



**Sérgio António Matos
Pinho**

**Caixa Automática e Manobras Especiais no
ATLASCAR**



**Sérgio António Matos
Pinho**

**Caixa Automática e Manobras Especiais no
ATLASCAR**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica de Vítor Manuel Ferreira dos Santos, Professor do Departamento de Engenharia Mecânica da Universidade de Aveiro

o júri / the jury

presidente / president

Prof. Doutor Jorge Duarte Fernandes Ferreira

Professor Auxiliar da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor Manuel Bernardo Cunha

Professor Auxiliar da Universidade de Aveiro

Prof. Doutor Vítor Manuel Ferreira dos Santos

Professor Associado da Universidade de Aveiro (orientador)

**agradecimentos /
acknowledgements**

A todas as pessoas que me acompanharam neste percurso - amigos, professores e familiares - em especial ao Professor Vítor Santos pelo acompanhamento, motivação e boa disposição mesmo nos momentos mais difíceis.

Por outro lado não devo deixar de parte os preciosos colegas Jorge Almeida, João Torrão e Ricardo Pascoal pela boa integração no LAR e por todas as ajudas prestadas durante este gratificante período.

Agradeço ainda aos meus colegas de laboratório pelo apoio, diversão e momentos de descontração, assim como pela sua paciência nos dias mais complicados. À família BEST por todos os momentos de aprendizagem, descontração e apoio.

E finalmente, não menos importante, agradeço à minha família e amigos mais próximos. A família porque sempre me apoio e acreditou no meu futuro, tornando possível esta jornada. Aos amigos que sabem o quão importante são na minha vida.

A todos, um muitíssimo obrigado.

Resumo

O AtlasCar é um protótipo de um veículo desenvolvido pelo Laboratório de Automação e Robótica (LAR) do Departamento de Engenharia Mecânica da Universidade de Aveiro, em que, o principal objetivo é o desenvolvimento e estudo de sistemas de condução autónoma, de condução assistida (ADAS) e de segurança.

Entre todos os desafios deste projeto, a inexistência de um sistema de comutação de velocidades automático fiável limitou, em muito, a condução autónoma. Este trabalho teve como objetivo a reconstrução de todo o sistema de comutação, necessitando da construção de um novo sistema mecânico (baseado no modelo anterior), da calibração/reajustes do sistema de comutação (unidade de controlo), da implementação do sistema no veículo e do desenvolvimento de *software* no sistema computacional (para realização de manobras especiais).

Para o efeito são utilizados protocolos de comunicação entre o *firmware* e o *software* do AtlasCar, *software* de calibração e um modo *ROS* para comunicação com todo o sistema.

Esta dissertação aborda o desenvolvimento de um novo comutador de mudanças que teve como base o modelo existente. Após concluído o comutador, implementou-se uma nova segurança no sistema eletrónico e tornou-se a unidade de controlo mais compacta e robusta, juntando as várias unidades de controlo num único bloco.

Por último, desenvolveu-se um *software* para a realização de manobras especiais como: avançar, recuar, estacionar, estacionar paralelamente, inverter marchar e parar.

Após realizados todos os trabalhos, concluiu-se que o comutador de mudanças é funcional e robusto, e as manobras são realizadas, mas não contém um grande grau de repetibilidade.

Abstract

The AtlasCar is a prototype of a vehicle developed by the Laboratory of Automation and Robotics (LAR) of the Department of mechanical engineering at the University of Aveiro, in which the main objective is to develop and study of autonomous driving systems, assisted driving (ADAS) and safety. Between all challenges this project, the lack of a automatic switch system of gearbox was a limitation for autonomous driving. This work had as main goal the reconstruction of all switching system, for that, is built a new mechanical system (based on the previous model), calibrated/adjusted switch system (control unit), implemented the system in a vehicle and developed a software in a computational system (for performing special maneuvers). For this purpose it used communication protocols between the firmware and the AtlasCar software, calibration software and a ROS mode to communicate with the entire system.

This thesis focus the development of a new switching system of gearbox that was based on the existing model. After the switching system it was implemented a new safety on the electronic system and became the control unit more compact and robust, joining both units in a single block.

Lastly, it was developed a software to performing special maneuvers like: forward, backward, parking, parallel parking, reverse and stop.

After all jobs performed, it was concluded that the switching system is functional and robust, and the maneuvers are performed, but do not contain a high repeatability.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	v
List of Acronyms	vii
1 Introdução	1
1.1 Projeto ATLAS	1
1.2 Objetivos	2
1.3 AtlasCar	2
1.3.1 Geração e gestão de energia	3
1.3.2 Sensores	4
1.3.3 Processamento	5
2 Infraestrutura existente	7
2.1 Sistema mecânico	7
2.2 Sistema eletrônico	9
2.3 Acionamento de pedais	12
3 Desenvolvimento da nova unidade	15
3.1 Sistema mecânico	15
3.1.1 Dimensionamento da placa	16
3.1.2 Transmissão	16
3.1.3 Sistema de guiamento	18
3.1.4 Sistema de (des)acoplamento	18
3.1.5 Veios	19
3.1.6 Modelo final	19
3.2 Sistema eletrônico	25
3.2.1 Unidade de Controlo	26
3.2.2 Segurança	26
3.3 Acionamento de pedais	29
4 <i>Software</i> para manobras	31
4.1 Definições	31
4.2 Arquitetura geral de <i>software</i>	31

4.2.1	Baixo Nível	32
4.2.2	Médio Nível	32
4.2.3	Alto Nível	33
4.3	Implementações	33
4.4	Estados	35
4.4.1	Parado	36
4.4.2	Arranque	36
4.4.3	Paragem	37
4.4.4	Movimento	37
4.5	Manobras	37
5	Experiências e Resultados	41
5.1	Experiências	41
5.2	Resultados	42
6	Conclusões	47
	Bibliografia	49
	Anexos	51

Lista de Figuras

1.1	AtlasCar	1
1.2	Redesign do alternador	3
1.3	Inversor	3
1.4	<i>Uninterruptible Power Supply</i> , UPS	4
1.5	Disposição dos equipamentos	4
1.6	Dispositivos: a) Laser 3D, b) Sensores infravermelhos, c) Câmara estéreo, d) IMU, e) GPS, f) Laser Planar, g) Câmaras Monoculares	5
2.1	Modelo existente: guia para movimento em yy (1) e em xx (2)	7
2.2	Modelo existente: 1) motores 2) polia 3) base superior 4) base inferior 5) correias	8
2.3	Motor	9
2.4	Potenciómetro multivolta 10k Ω	9
2.5	Comando de áudio adaptado para seleção de modo (manual ou automático) e de velocidade	9
2.6	Arduino <i>Shield Ethernet</i>	10
2.7	Conjunto de componentes do sistema eletrónico do comutador de mudanças	10
2.8	Simulador de um veículo - parâmetros/caraterísticas do veículo	11
2.9	Simulador de um veículo - interface para alteração direta de condições a que o veículo está sujeito	12
2.10	Sistema de acionamento de um pedal	12
2.11	Sistema de ajuste dos pedais colocado sob o veículo	13
2.12	Borboleta eletrónica	13
3.1	Espaço para movimento do manípulo: esquerda - base existente; direita - base nova	16
3.2	Corrente	17
3.3	Carreto	17
3.4	Sistema de esticadores das correntes	17
3.5	Sistema de patins duplos	18
3.6	Sistema de (des)acoplamento	19
3.7	Vista explodida da base de suporte das correntes	20
3.8	Vista explodida da base inferior	21
3.9	Vista explodida do modelo completo	22
3.10	Vista pormenorizada da base de suporte das correntes	23
3.11	Vista inferior do modelo, sem suporte	24
3.12	Vistas detalhadas do sistema de acoplamento das bases à corrente	24

3.13	Comutador de velocidades, montagem final	24
3.14	Comutador de velocidades, montagem final	25
3.15	Comutador de velocidades, montagem final	25
3.16	Unidade de controlo da caixa de velocidades	26
3.17	Fim de curso	27
3.18	Sistema de montagem	27
3.19	Sistema redutor para acionamento da embraiagem	29
4.1	<i>Packages</i> : alto nível, médio nível e baixo nível, respetivamente (situação existente)	32
4.2	Tópicos <i>ROS</i>	33
4.3	<i>Gamepad</i>	34
4.4	<i>Packages</i> utilizados para a implementação das várias manobras	34
4.5	Máquina de estados principais e as transições permitidas	35
4.6	Sub-estados e ações associadas	36
4.7	Sequência de ações	37
4.8	Sequência de ações	38
5.1	Tempos de comutação das várias velocidades	42
5.2	Velocidades atingidas durante a manobra de "estacionar paralelamente" . . .	43
5.3	Velocidades atingidas durante a manobra de "inverter marcha"	43
5.4	Deslocamentos realizados durante a manobra de "estacionar paralelamente" .	44
5.5	Deslocamentos realizados durante a manobra de "inverter marcha"	44
5.6	Trajectoria descrita pelo veículo durante a manobra de "estacionar paralelamente"	45
5.7	Trajectoria descrita pelo veículo durante a manobra de "inverter marcha" . . .	45
6.1	Esquema elétrico	51
6.2	Desenho de definição base maior	53
6.3	Desenho de definição base menor	54
6.4	Desenho de definição base inferior da coluna de ajuste	55
6.5	Desenho de definição base superior da coluna de ajuste	56
6.6	Desenho de definição base inferior da coluna fixa	57
6.7	Desenho de definição dos ligantes das colunas	58
6.8	Desenho de definição da garra	59
6.9	Desenho de definição do olhal da garra	60
6.10	Desenho de definição da base de suporte	61
6.11	Desenho de definição da lateral do suporte	62
6.12	Desenho de definição da lateral pequena do suporte	63
6.13	Desenho de definição do adaptador da lateral pequena do suporte	64
6.14	Desenho de definição do veio	65
6.15	Desenho de definição do veio	66
6.16	Desenho de definição do veio	67
6.17	Desenho de definição do veio	68

Lista de Tabelas

3.1	Legenda da figura 3.7	20
3.2	Legenda da figura 3.8	21
3.3	Legenda da figura 3.9	23
3.4	Pinos disponíveis no modelo existente	28
3.5	Pinos disponíveis no modelo final	28
4.1	Condições para a definição dos estados	36
5.1	Parâmetros utilizados para a realização a manobra "inverter marcha"	41
5.2	Parâmetros utilizados para a realização da manobra "estacionar paralelamente"	42

List of Acronyms

ADAS Sistema Avançado de Condução Assistida

DC Corrente contínua

UPS *Uninterruptible Power Supply*

PLC *Power Line Communication*

IMU Unidade de Medição Inercial

GPS Sistema de Posicionamento Global

ROS *Robot Operating System*

LCD *liquid Crystal Display*

RPM Rotações por minuto

PWM *Pulse-width modulation*

CW Sentido dos ponteiros do relógio

CCW Sentido contrário aos ponteiros do relógio

Capítulo 1

Introdução

1.1 Projeto ATLAS

O projeto ATLAS iniciou-se no ano letivo 2002/2003, no Laboratório de Automação e Robótica do Departamento de Engenharia Mecânica focando o desenvolvimento e estudo de sistemas ativos e de deteção avançada para futura implementação em casos reais (indústria automóvel ou similar).

Após vários desenvolvimentos por parte desta equipa, que ganhou destacados prémios em competições de condução autónoma [1], decidiu-se tornar este projeto um pouco mais ambicioso, aplicando-o ao ambiente real, criando-se o AtlasCar [2].



Figura 1.1: AtlasCar

O AtlasCar é um protótipo à escala real, de um Ford Escort Station Wagon de 1998 (fig. 1.1), usado para pesquisa e desenvolvimento de sistemas avançados de condução assistida (ADAS), de condução autónoma e de segurança. O ADAS é praticado em carros autónomos

que se focam num tema, cada vez mais, comum na literatura e em diversos laboratórios mundiais, sendo já estudado em outros projetos similares ao AtlasCar, como é exemplo, o Nissan's NSC-2015 [3], o Nissan da Universidade de Oxford [4], o Toyota Lexus LS [5], o carro da Google [6], o Mercedes-benz S500 Intelligent drive [7], o Volkswagen Passat da Universidade de Stanford [8] e o VisLab itália [9].

Segundo a definição de autonomia de George A. Bekey[10],

“Autonomy refers to systems capable of operating in the real-world environment without any form of external control for extended periods of time.”

Assim, estes veículos permitem realizar um determinado percurso de forma autónoma, ou seja, sem interferência do utilizador. Para que este objetivo seja concretizável, no caso dos veículos de combustão, é necessário que se movimentem autonomamente, isto é, necessitam de uma caixa de velocidades automática e controlo eletrónico dos pedais e volante.

1.2 Objetivos

O veículo utilizado (fig. 1.1) não possui caixa de velocidades automática, desta forma não é permitido, até ao momento, demonstrar todas as potencialidades do ADAS. A única manobra que é possível executar no AtlasCar é o seguimento de peões[11] com uma limitação da velocidade pois, não existe comutação de mudanças, utilizando sempre a primeira mudança.

Dado que no mercado não existem caixas automáticas para o modelo do veículo em questão, é necessário construir um sistema que comute o manípulo das velocidades no qual, se irá focar grande parte desta dissertação.

Assim, os principais objetivos desta dissertação são:

- desenvolvimento de um sistema electromecânico de comutação da caixa de velocidades do ATLASCAR e respetiva interligação ao software de gestão dessa mesma caia;
- ajuste de mecanismo de atuação dos pedais;
- implementação de *software* ao nível do sistema computacional do veículo para a execução de algumas manobras especiais (início de marcha, estacionamento, inversão de marcha, etc).

Após concluídos todos estes objetivos, o AtlasCar será capaz de comutar mudanças autonomamente e realizar várias manobras especiais, tornando assim, o AtlasCar um veículo muito mais versátil e melhor para aplicação e demonstração do ADAS.

1.3 AtlasCar

Numa visão muito geral, o AtlasCar, como já foi mostrado na figura 1.1, é uma carrinha a gasolina que, após modificada, mantém a conformidade legal do veículo e a capacidade de condução humana.

O hardware existente está subdividido em três secções:

- Geração e gestão de energia
- Sensores
- Processamento

1.3.1 Geração e gestão de energia

Para garantir a energia necessária para os equipamentos a bordo foi necessário acrescentar um gerador de energia. Para este fim, redesenhou-se o sistema de alternador e implementou-se um gerador DC acionado pelo motor (fig. 1.2).

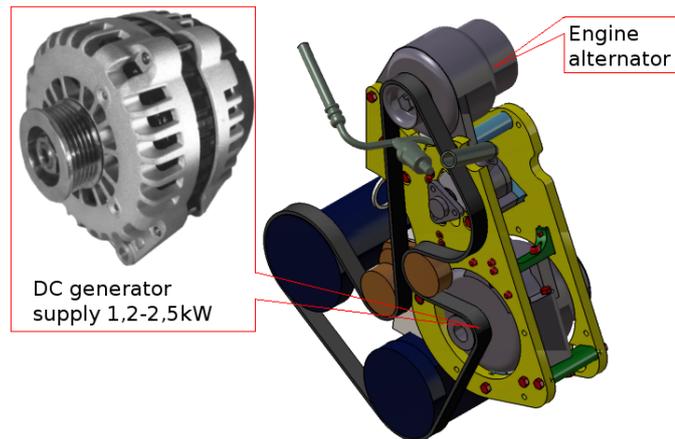


Figura 1.2: Redesign do alternador [2]

O gerador DC está ligado a um inversor (fig. 1.3) que fornece 220/230 V a partir de 12 V DC provenientes do alternador.



Figura 1.3: Inversor [2]

Por fim, temos a *Uninterruptible Power Supply* (UPS) (fig. 1.4) que é utilizada como estabilizador em situações em que não é gerada energia, como por exemplo, o carro parado; também é utilizado para efetuar a distribuição e gestão da energia para todos os equipamentos existentes. Como são necessárias tensões diversas, existem, como pode ser visualizada na figura 1.5, três quadros elétricos: o quadro C que contém um PLC; o quadro B que fornece 12V; e o quadro A responsável pelos 24V.



Supply 230V @ 50/60Hz

Figura 1.4: *Uninterruptible Power Supply*, UPS [2]

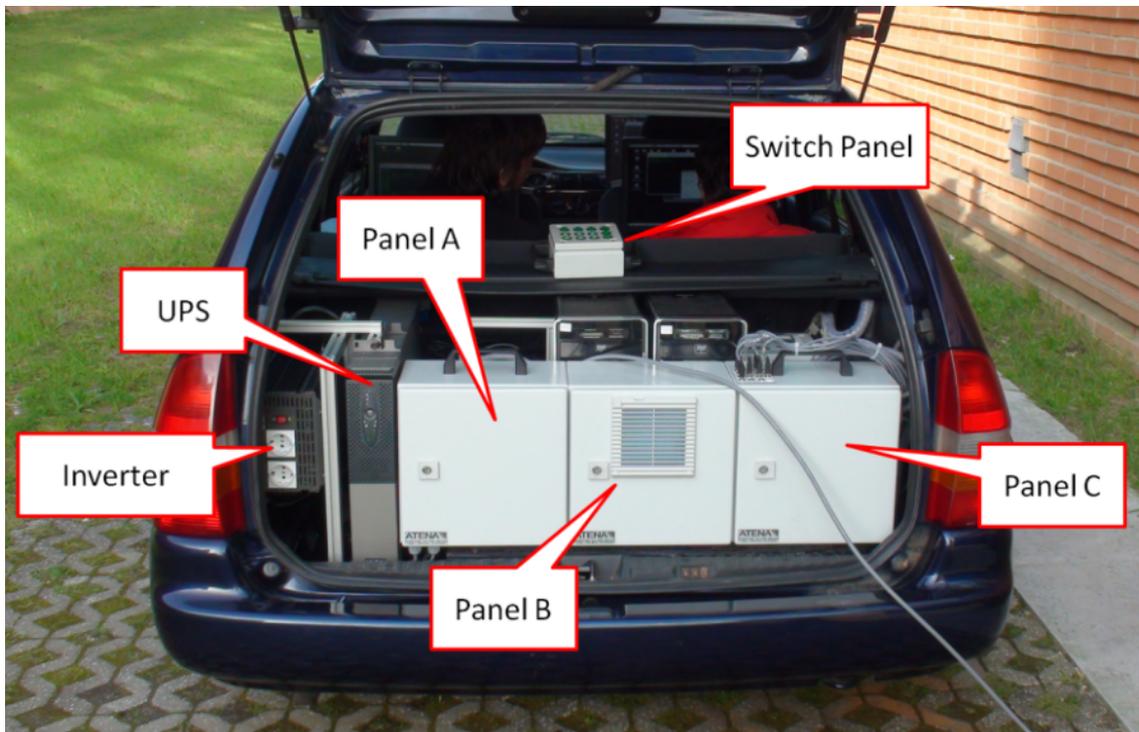


Figura 1.5: Disposição dos equipamentos [2]

1.3.2 Sensores

O Atlascar necessita de ter uma perceção do ambiente que o rodeia, para tal, foram instalados vários sensores e câmaras para permitir esta perceção. A lista de sensores e câmaras presente até à data é (fig. 1.6):

- Laser 3D
- Câmara estéreo
- 2 Câmaras monoculares numa unidade *pan-tilt*

- Lasers 2D
- Unidade de mediação inercial (IMU)
- Sistema de posicionamento global (GPS)
- Laser planar 270°
- Sensores infravermelhos

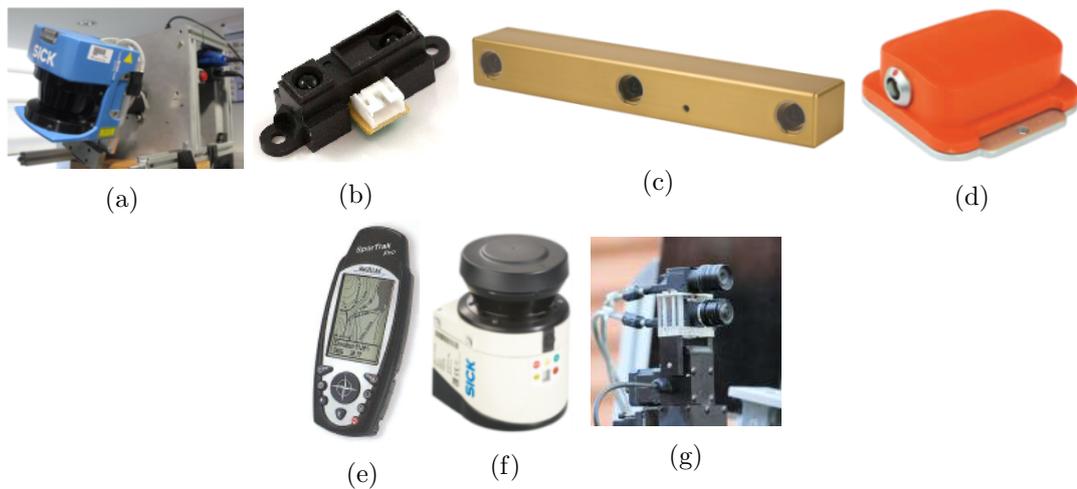


Figura 1.6: Dispositivos: a) Laser 3D, b) Sensores infravermelhos, c) Câmera estéreo, d) IMU, e) GPS, f) Laser Planar, g) Câmaras Monoculares

1.3.3 Processamento

Um dos grandes objetivos deste projeto é a aquisição e tratamento de dados em cenários reais, assim é necessário uma ou mais unidades de processamento. A unidade de processamento é constituída por um computador que está ligado aos sensores e permite receber os dados e ao mesmo tempo correr os algoritmos criados para os respetivos efeitos, desenvolvidos em ambiente ROS (*Robot Operating System*). O veículo também possui três ecrãs LCD que permitem ao utilizador monitorizar todo o sistema.

Capítulo 2

Infraestrutura existente

Neste capítulo apresenta-se uma visão do trabalho previamente desenvolvido, focando em particular os sistemas mecânicos e eletrônicos do comutador de mudanças.

2.1 Sistema mecânico

De forma a concretizar o objetivo principal desta dissertação é necessária a existência de um sistema mecânico que permita comutar as velocidades do AtlasCar. Para que esta comutação seja possível é necessário deslocar a alavanca para as posições necessárias, de forma a existir uma correta comutação sem danificar a caixa de velocidades do automóvel.

Até à data foi criado um modelo para solucionar esta questão, desenvolvido no ano letivo de 2010/2011, pelos alunos Miguel Mieiro e Nuno Silva na cadeira de Projeto em Automação e Robótica Industrial [12].

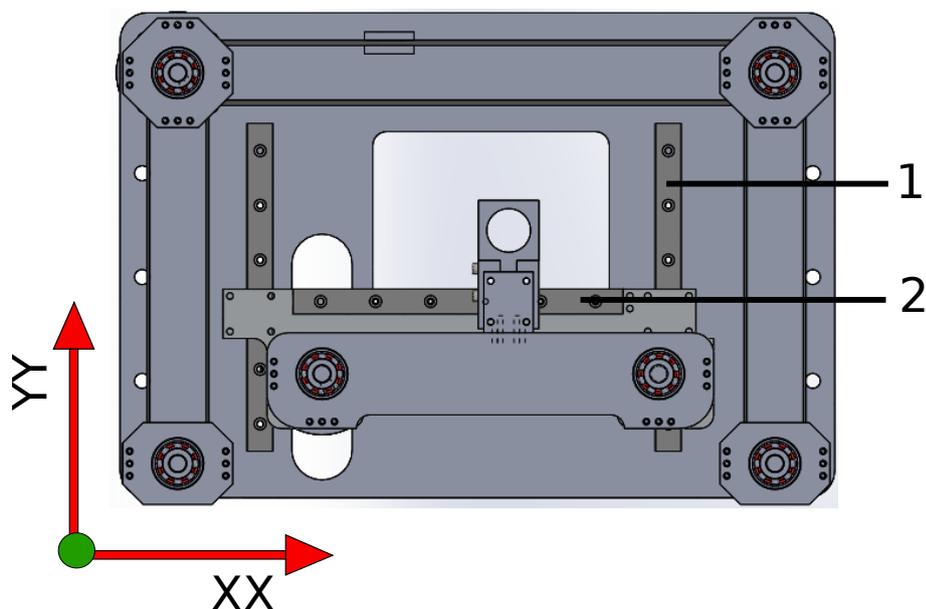


Figura 2.1: Modelo existente: guia para movimento em yy (1) e em xx (2)

O sistema utiliza dois motores, de características iguais, para realizar o deslocamento do

manipulo segundo os eixos xx e yy . Para a execução dos deslocamentos existem duas bases: na principal (a de maiores dimensões) estão fixas as guias que irão orientar a base de menores dimensões, sendo assim responsável pelo movimento em yy ; na secundária (a de menores dimensões) está acoplada a guia para o movimento em xx (fig. 2.1). Admitindo-se assim que a placa principal e secundária são responsáveis pelos movimentos em yy e xx , respectivamente.

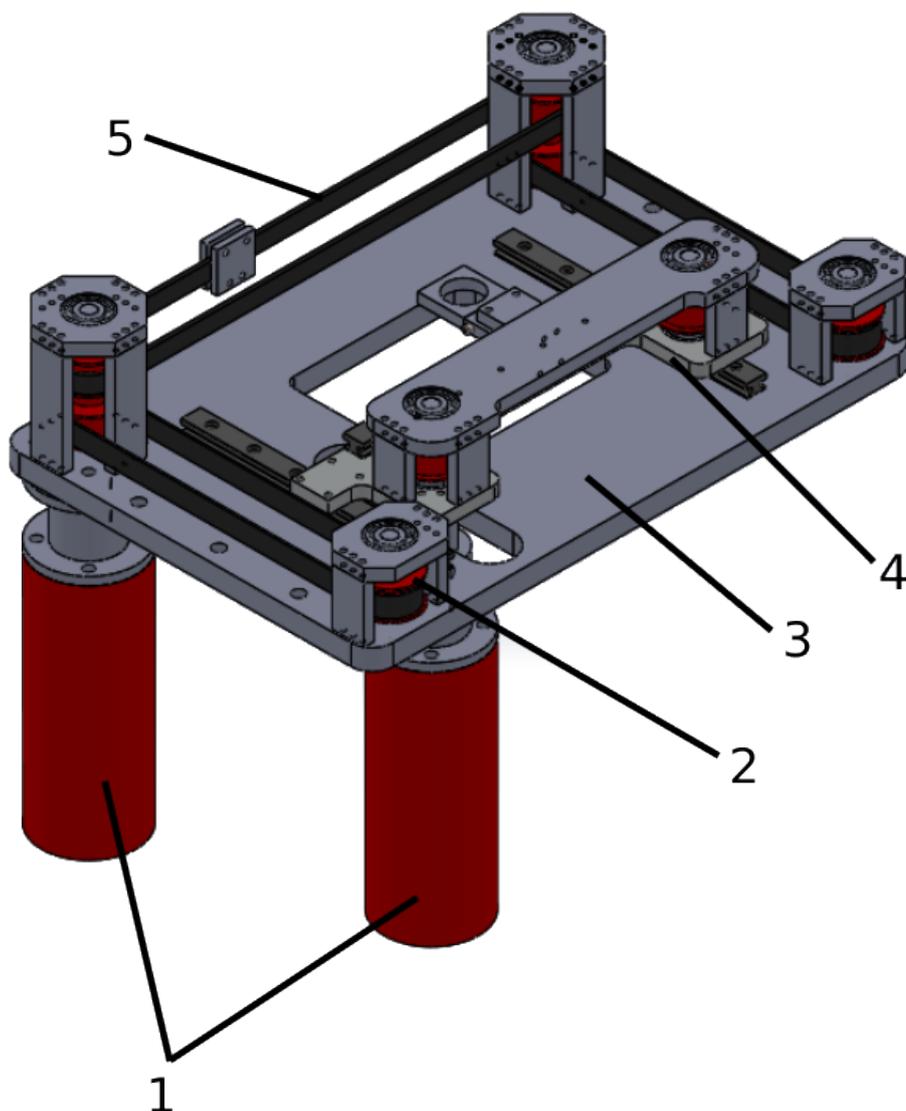


Figura 2.2: Modelo existente: 1) motores 2) polia 3) base inferior 4) base superior 5) correias [12]

Para a realização dos deslocamentos, referidos anteriormente, são utilizados dois motores da MFA (fig. 2.3), que possuem uma redução de 104:1 que debita 125 RPM à saída e 41.3 W de potência. A transmissão entre os motores e as bases é feita por meio de correias dentadas, como se pode verificar na figura 2.2.



Figura 2.3: Motor [13]

Como nesta aplicação é necessária a leitura da posição em que se encontra o manipulador, utilizam-se dois potenciômetros multivolta de $10k \Omega$ (fig. 2.4). Desta forma, é possível saber qual o deslocamento dos motores e saber as posições exatas das placas em cada instante, o que permite saber a localização exata do manipulador de forma a realizar uma correta comutação de velocidades.



Figura 2.4: Potenciômetro multivolta $10k \Omega$ [14]

Outro aspeto importante neste modelo é o sistema que agrega o manipulador ao comutador, que pode ser visto na figura 2.1, e que foi feito em aço de forma a suportar os esforços realizados nesta região. Devido ao facto de ser uma peça fixa, o sistema de comutação manual é realizado através de um comando que será descrito na secção seguinte.

2.2 Sistema eletrónico

A unidade eletrónica de comando foi desenvolvida por Pedro Pinheiro durante a sua dissertação em 2012 [15]. O sistema possui dois módulos, automático e manual, em que, para a comutação, utiliza um comando de áudio adaptado de um Renault Mégane (fig. 2.5).



Figura 2.5: Comando de áudio adaptado para seleção de modo (manual ou automático) e de velocidade [15]

Para que os dados do comando e dos potenciômetros possam ser lidos e processados, utiliza-se um Arduino *Shield Ethernet* (fig. 2.6) que, por sua vez, se encontra ligado a uma placa que contém duas pontes H para controlo dos motores (fig. 2.7).

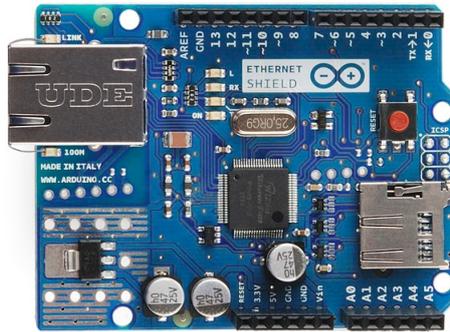


Figura 2.6: Arduino *Shield Ethernet* [15]

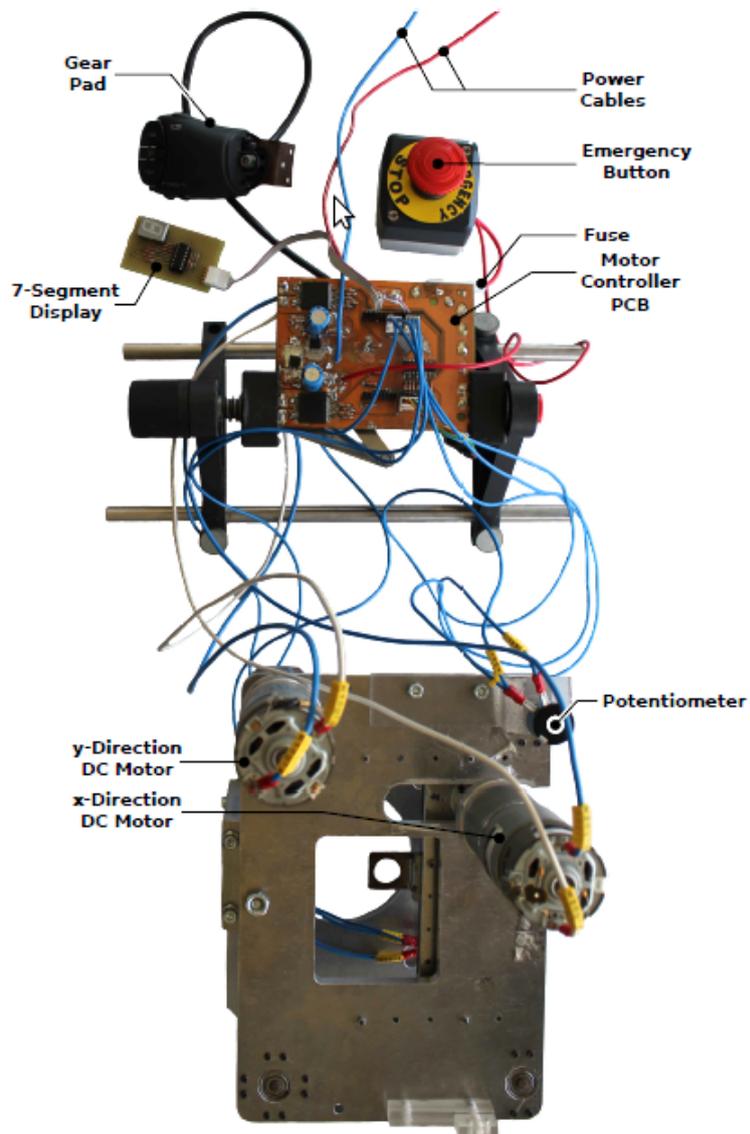


Figura 2.7: Conjunto de componentes do sistema eletrônico do comutador de mudanças [15]

Conforme apresentado na figura 2.7, este sistema contém um *display* de 7 segmentos e um botão de emergência que corta diretamente a alimentação dos motores.

O sistema manual, tal como referido, utiliza o comando apresentado na figura 2.5 para comutação das velocidades e possibilita ao utilizador comutar as velocidades de forma sequencial, ou seja, não permite "saltar" velocidades. Isto faz com que a marcha atrás só possa ser engrenada se a velocidade anterior for o ponto morto (posição neutra).

O sistema automático recebe a informação da velocidade pretendida, via *Ethernet*, executando essa mesma ordem. Neste caso, a comutação de velocidades pode não ser sequencial, permitindo engrenar a velocidade desejada assim que o pedido seja feito. Para o teste deste modo foi utilizado um simulador em Matlab (fig. 2.8 e 2.9), em que se pode variar alguns parâmetros de forma a alterar as características do veículo e situações a que é sujeito. Este simulador foi criado pois, na altura, não existia um sistema mecânico suficientemente robusto para realizar os testes.

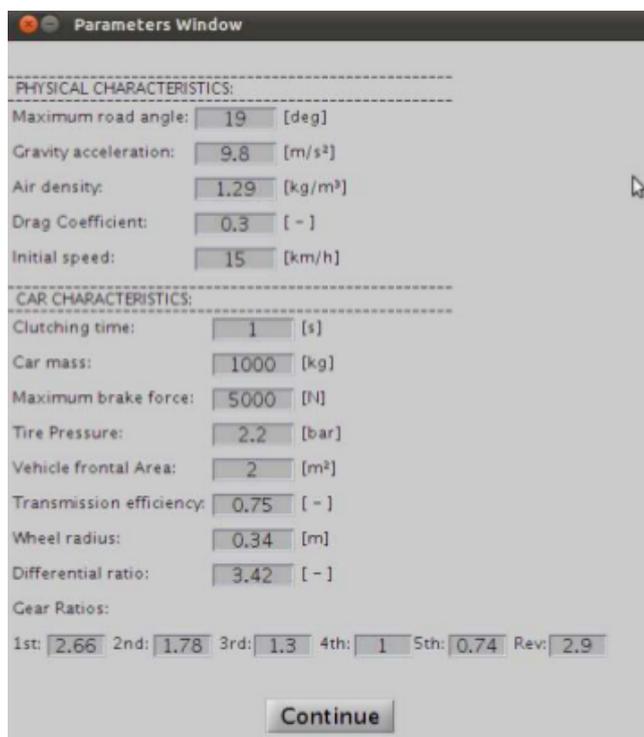


Figura 2.8: Simulador de um veículo - parâmetros/caraterísticas do veículo [15]

Na figura 2.9 encontra-se a interface que permite ao utilizador variar alguns parâmetros que influenciam diretamente a velocidade do veículo. Os parâmetros referidos são acelerador, travão e inclinação do piso, os quais podem ser alterados através de barras verticais (balão 2, 3 e 1 respetivamente) ou introduzindo um valor fixo (balão 5, 6 e 4, respetivamente). No caso de se introduzirem valores fixos estes apenas são atualizado no simulador após pressionado o botão *Override* (balão 7).

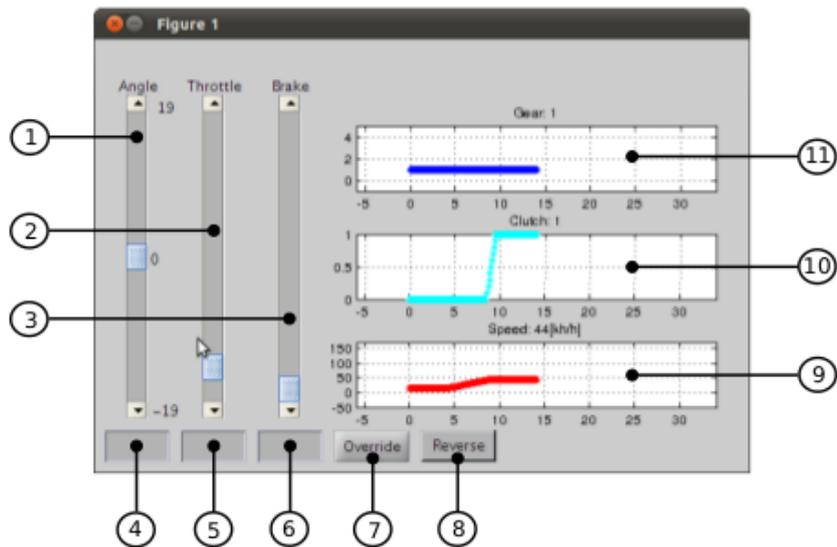


Figura 2.9: Simulador de um veículo - interface para alteração direta de condições a que o veículo está sujeito [15]

Além das possibilidades apresentadas anteriormente é possível inverter o sentido de marcha pressionando, para isso, o botão *Reverse* (balão 8) ou ainda visualizar a velocidade em que nos encontramos (balão 11), o nível de aceleração (balão 10) e a velocidade do veículo (balão 9).

2.3 Acionamento de pedais

Tal como nos outros sistemas, os pedais já sofreram intervenções de forma a permitir o acionamento automático[16]. O acionamento é feito por meio de atuadores de travão de mão de um Renault Vel Satis (fig. 2.10). A conexão entre os acionadores e os pedais é feita por meio de cabos de aços que tem uma zona intermédia que permite o seu ajuste. Este ajuste é feito na base inferior do veículo com auxílio de serra-cabos, como se pode visualizar na figura 2.11.



Figura 2.10: Sistema de acionamento de um pedal [16]



Figura 2.11: Sistema de ajuste dos pedais colocado sob o veículo

O sistema descrito anteriormente não se aplica ao acelerador, nesse caso foi alterada a borboleta mecânica (de origem) por uma variante eletrônica (fig. 2.12) que permite receber um sinal com os valores de abertura, sendo que não é necessário realizar esforços mecânicos para acelerar o veículo em modo automático.



(a)



(b)

Figura 2.12: Borboleta eletrônica

Capítulo 3

Desenvolvimento da nova unidade

Neste capítulo descreve-se o estudo, concepção e implementação do comutador, abordando-se principalmente a parte mecânica e eletrônica.

3.1 Sistema mecânico

Sobre o modelo existente verificaram-se algumas falhas a nível mecânico, o que impossibilita a sua implementação, pois o objetivo principal não é concretizado. Com o intuito de resolver estes problemas optou-se por desenvolver um novo sistema de raiz.

Após a análise de várias soluções como, por exemplo, um braço robótico ou um robô delta (com hidráulicos ou motores linear), optou-se pelo mesmo princípio de funcionamento. Esta escolha foi feita com base nos custos associados a cada sistema, no trabalho já desenvolvido e nos materiais que podem ser reutilizados, ou seja, optou-se pela utilização de um sistema com duas bases acionadas por dois motores independentes, responsáveis por cada um dos movimentos, xx e yy .

Para a seleção do sistema foi feita uma análise de erros de construção do modelo existente. Nesta análise verifica-se a existência de algumas falhas, tais como:

- sub-dimensionamento do espaço para o movimento do manipulador - Dimensionamento da placa
- escorregamento do sistema de transmissão - Transmissão
- demasiado atrito no sistema de guiamento das placas - Sistema de guiamento
- inexistência de um sistema de desacoplamento do manipulador do comutador - Sistema de (des)acoplamento
- flexão dos veios - Veios

Depois de identificadas as principais limitações e falhas, desenvolveram-se soluções para contrariar estas ocorrências e tornar, assim, o modelo viável e funcional.

3.1.1 Dimensionamento da placa

Nesta fase pensou-se numa forma de solucionar o problema do sub-dimensionamento do espaço para movimento do manipulador, optando-se por aumentar as suas dimensões como se pode visualizar na figura 3.1. A nova placa, além de permitir a correta comutação de velocidades sem ocorrer nenhuma obstrução, também permite desacoplar a base secundária da base principal sem necessidade de desmontar o motor que lhe está acoplado.

Este desacoplamento é possível pois, como é visível na figura 3.1, a placa existente contém um rasgo para permitir o deslocamento do motor, sendo que, na nova placa optou-se por cortar uma secção que permite realizar o movimento do manipulador e do motor.

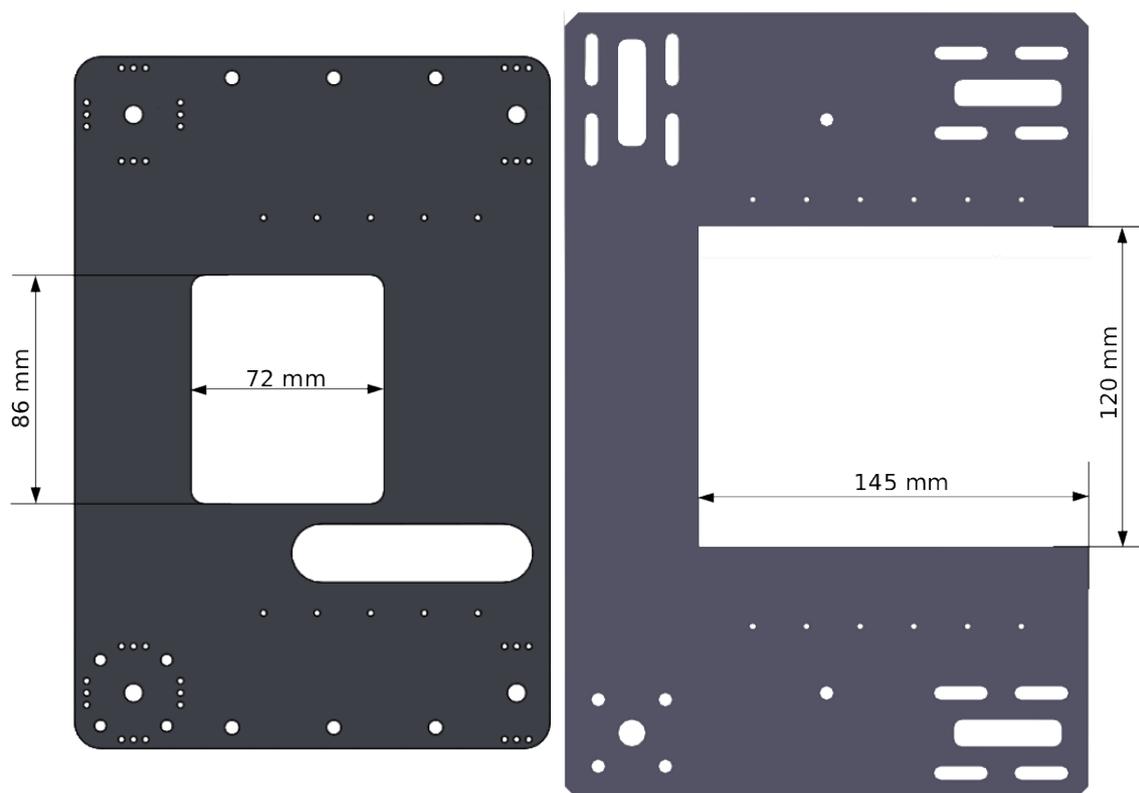


Figura 3.1: Espaço para movimento do manipulador: esquerda - base existente; direita - base nova

3.1.2 Transmissão

O sistema de transmissão de potência do modelo existente foi o sistema que se mostrou mais ineficiente. A ineficiência surge do facto de se utilizar correias dentadas sem utilização de esticadores, o que permitia o escorregamento. Ainda foram pensados alguns sistemas de esticadores de forma a solucionar o problema, porém, como não foram pensados de raiz, a aplicabilidade dos esticadores não se mostrou viável.

Para que no novo modelo não ocorra escorregamento optou-se por utilizar um sistema de transmissão suficientemente robusto, utilizando-se para o efeito um sistema de correntes e carretos, após realizada um estudo de mercado:

- corrente simples 06-B1 3/8" link-belt [17], figura 3.2
- carretos Z 9 3/8"06-B1 simples [18], figura 3.3



Figura 3.2: Corrente



Figura 3.3: Carreto

Além destes novos componentes, também se usou num sistema de esticadores (fig. 3.4) para que possa existir um ajuste da corrente, evitando assim folga ou aperto excessivo.



Figura 3.4: Sistema de esticadores das correntes

Os motores utilizados para realizar os movimentos segundo os eixos xx e yy foram os utilizados no modelo existente. Esta escolha foi feita com base nas equações (eq.3.1 a 3.3) utilizadas no dimensionamento dos motores do modelo existente [12], onde o diâmetro primitivo dos carretos é 27.85 mm (eq. 3.2) e a velocidade média é 152 mm/s , valor utilizado no modelo existente, em que se considerou necessários $0,25 \text{ s}$ para realizar um deslocamento de 30 mm .

$$RPM = \frac{velocidade}{\pi \times diametro} = \frac{152 \left[\frac{mm}{s}\right]}{3.141 \times 27.85 [mm]} \times 60 = 104.26 \text{ rpm} \quad (3.1)$$

$$Momento = forca \times raio = 178.46 [N] \times \frac{0.02785}{2} [m] = 2.49 \text{ Nm} \quad (3.2)$$

$$Potencia = momento \times \frac{rpm}{60} \times 2\pi = 2.49 \times \frac{104.26}{60} \times 2 \times 3.141 = 27.18 \text{ W} \quad (3.3)$$

Nas equações pretende-se calcular a potência necessária (eq.3.3) para o movimento do manipulador, para tal, é necessário calcular as RPM necessárias (eq.3.1) e o momento criado (eq.3.2), em que a força é 178.56 N, valor que surge da divisão do momento da alavanca (23,2 Nm) pelo ponto de aplicação (130 mm). A diferença entre o cálculo do modelo existente e o do novo modelo é o diâmetro dos carretos. Dos cálculos concluiu-se que a potência necessária é de 27.18 W que é inferior à potência dos motores (41.3 W), bem como, as rotações necessárias (104.26 rpm) são inferiores às debitadas pelo motor (125 rpm).

3.1.3 Sistema de guiamento

Além do sistema de transmissão, também se verificou uma falha grave no sistema de guias e patins para guiamento das bases [19]. A falha consistia no facto do sistema permitir rotação da base sempre que a força de repulsão do manipulador não fosse aplicada no centro das bases. Por sua vez, este efeito criava demasiado atrito entre os patins e as guias necessitando, assim, de um elevado acréscimo de força para realizar o escorregamento. O acréscimo de força, por sua vez, era tão elevado que os motores ficavam imobilizados.

A solução encontrada para a resolução deste problema foi colocar dois patins alinhados, como se pode visualizar na figura 3.5. Como na solução proposta o comprimento dos patins é aumentado para o dobro, a distância entre pontos de contacto é maior, o que reduz o ângulo de rotação, ou seja, diminui-se assim as forças de atrito existentes, o que faz com que os motores não sejam sobrecarregados.



Figura 3.5: Sistema de patins duplos

3.1.4 Sistema de (des)acoplamento

Sendo que se pretende que a caixa possibilite um modo automático e manual, em que o modo manual deva ser a utilização normal da caixa, deverá existir um desacoplamento entre o manipulador e o comutador de velocidades.

Como no modelo existente este acoplador era fixo, o modo manual foi realizado por *software* onde apenas era permitida a comutação sequencial por meio de um comando. De forma a alterar esta situação, desenvolveu-se um novo acoplador que já permite ao utilizador desacoplar o manipulador e utilizar a caixa manualmente, sem qualquer necessidade de *software* ou *hardware*.

O sistema criado, representado na figura 3.6, é constituído por duas peças unidas por pernos, que permitem abrir o sistema e retirar a manipulador para uso manual.

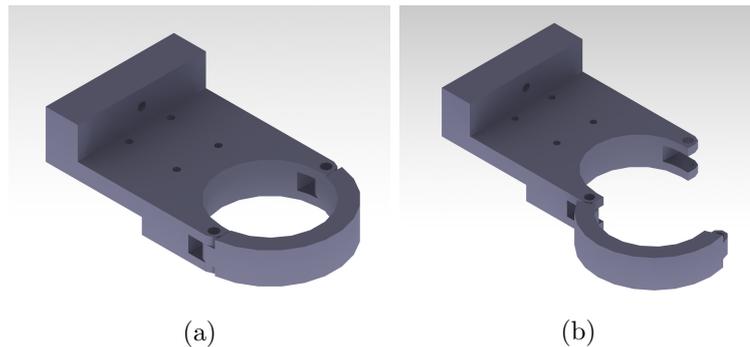


Figura 3.6: Sistema de (des)acoplamento

3.1.5 Veios

Por último, verificou-se que os veios fletiam um pouco quando aplicadas grandes forças; de forma a solucionar este problema optou-se por aplicar veios de $8mm$ em substituição dos antigos veios de $5mm$, já que os novos carretos têm um furo interno de $8mm$. Além deste aumento de diâmetro também se procurou um aço com maior resistência. Nesta fase não se realizaram quaisquer tipo de cálculos, pois apenas se procurou uma solução melhor, ou seja, como os veios existentes satisfaziam, no limite, os requisitos, apenas se pensou em aumentar o tamanho (por necessidade) e a dureza, criando, assim, o sobre-dimensionamento dos mesmos. Contudo, no futuro pode ser feito um redimensionamento do comutador de forma a reduzir todo o tamanho da caixa e tornar o sistema mais compacto.

3.1.6 Modelo final

Após desenvolvidas todas as soluções apresentadas, procedeu-se ao desenho das restantes peças. Nas figuras seguintes, 3.7 a 3.9, é possível ver algumas vistas explodidas do modelo, sendo que apenas são representados alguns elementos, descritos nas tabelas 3.1 e 3.2. Esta omissão deve-se à sua semelhança, como é exemplo as bases de suporte das correntes, que se diferenciam no tamanho, no número de carretos e na (in)existência de ajuste.

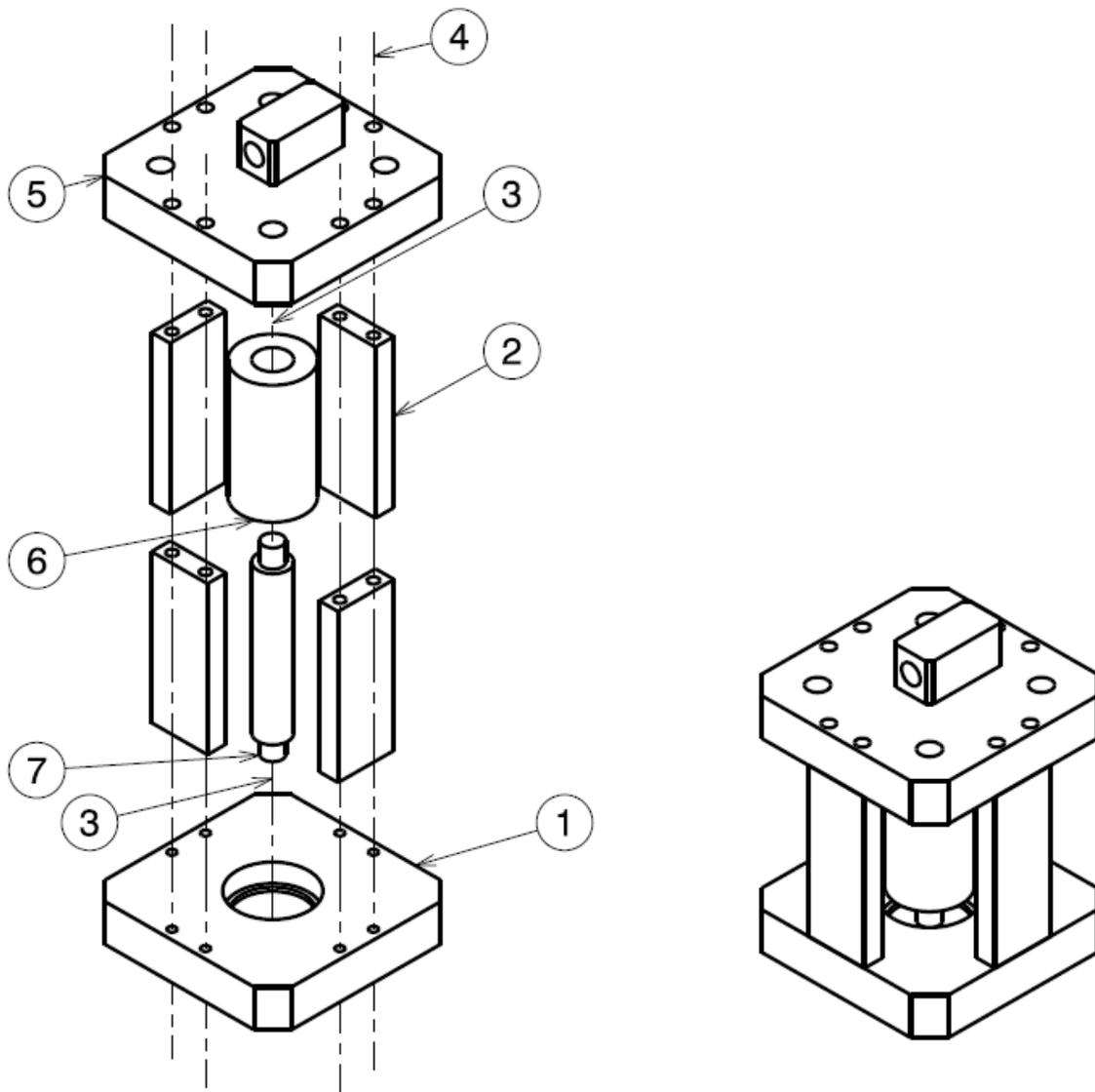


Figura 3.7: Vista explodida da base de suporte das correntes (legenda - tabela 3.1)

Tabela 3.1: Legenda da figura 3.7

Identificação	Número de peças	Descrição
1	1	Base
2	4	Elos de ligação
3	2	Rolamentos esféricos (não representados)
4	16	Parafusos sextavados M2x6
5	1	Base de ajuste
6	1	Carreto (esquemático)
7	7	Veio

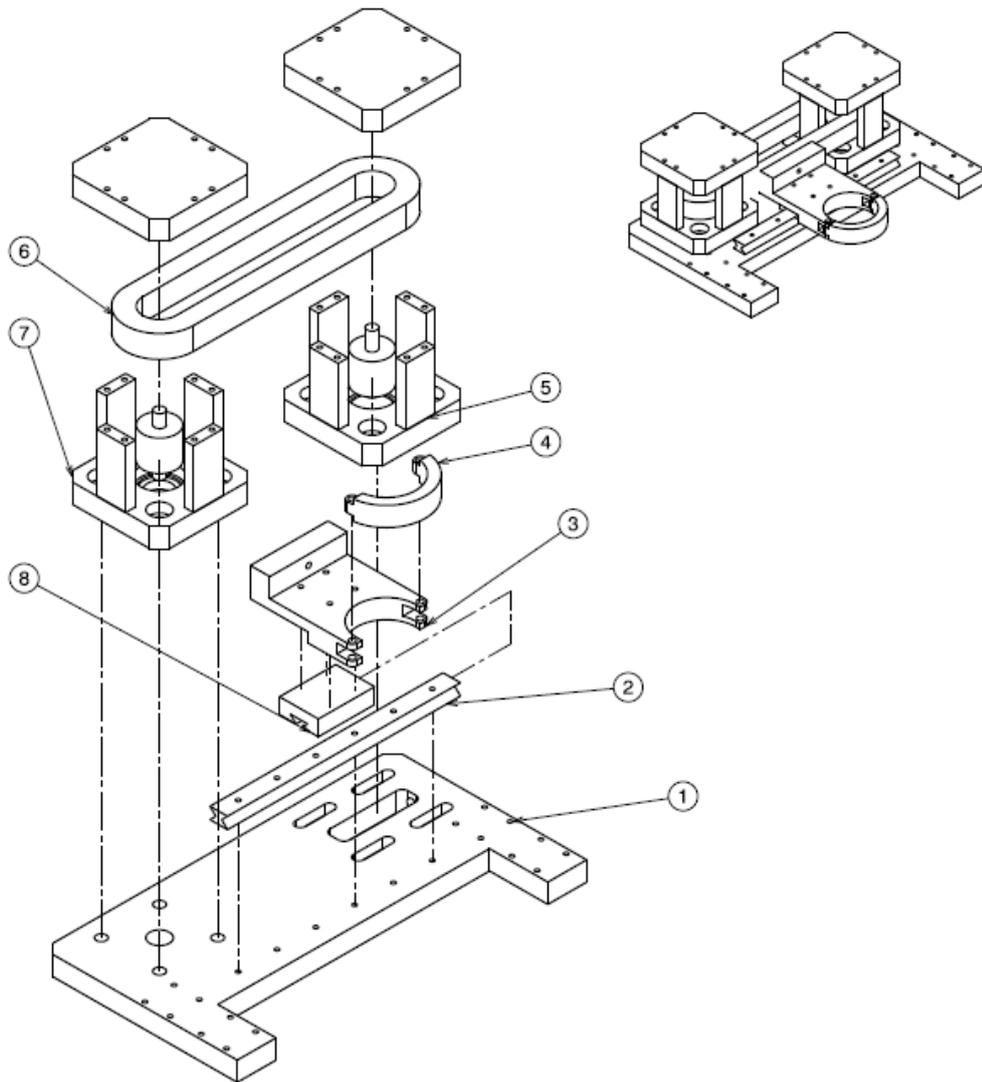


Figura 3.8: Vista explodida da base inferior (legenda - tabela 3.2)

Tabela 3.2: Legenda da figura 3.8

Indentificação	Número de peças	Descrição
1	1	Base principal
2	1	Guia
3	1	Sistema de acoplamento, base fixa
4	1	Sistema de acoplamento, base móvel
5	1	Base de suporte das correntes com ajuste
6	1	Corrente
7	1	Base de suporte das correntes fixa
8	1	Patins

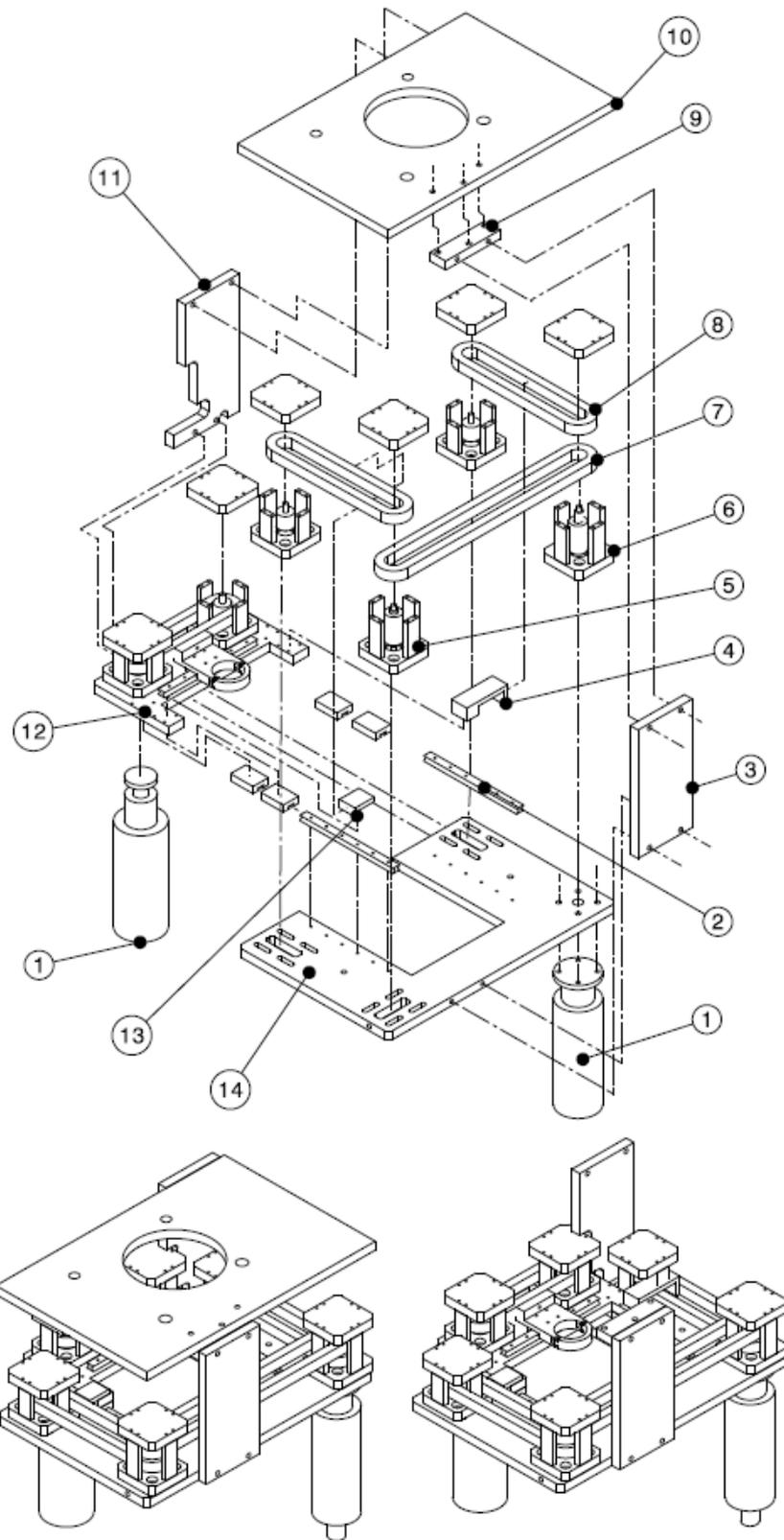


Figura 3.9: Vista explodida do modelo completo (legenda - tabela 3.3)

Tabela 3.3: Legenda da figura 3.9

Identificação	Número de peças	Descrição
1	2	Motores
2	2	Guia
3	1	Suporte lateral pequeno grande
4	1	Acoplador base-corrente
5	3	Base de suporte das correntes com ajuste
6	1	Base de suporte das correntes fixa
7	1	Corrente grande
8	2	Corrente pequena
9	1	Apoio suporte lateral pequeno
10	1	Base de suporte
11	1	Suporte lateral grande
12	1	Estrutura da base inferior
13	1	Acoplador base-corrente pequeno
14	1	Base principal

Após realizados todos os desenhos das peças necessárias para a construção do novo modelo, procedeu-se ao seu fabrico (desenhos no anexo). No final deste processo, montaram-se todas as peças e componentes obtendo-se assim o modelo final que pode ser visualizado da figura 3.10 a 3.15.

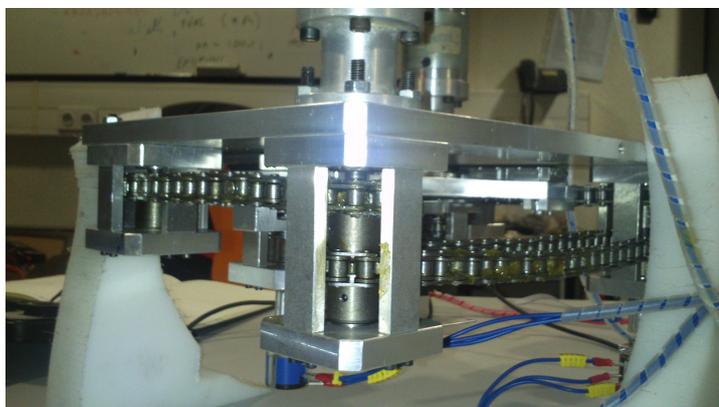


Figura 3.10: Vista pormenorizada da base de suporte das correntes

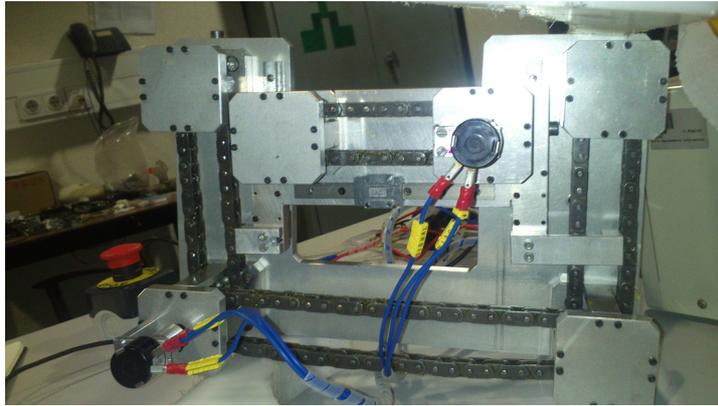
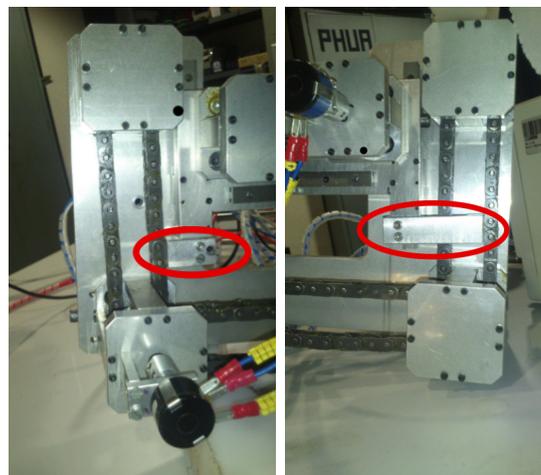


Figura 3.11: Vista inferior do modelo, sem suporte



(a)

(b)

Figura 3.12: Vistas detalhadas do sistema de acoplamento das bases à corrente

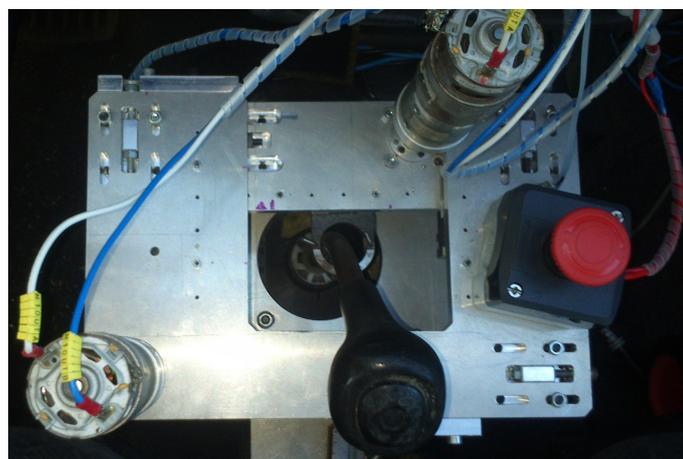


Figura 3.13: Comutador de velocidades, montagem final - Vista de cima com sistema acoplado

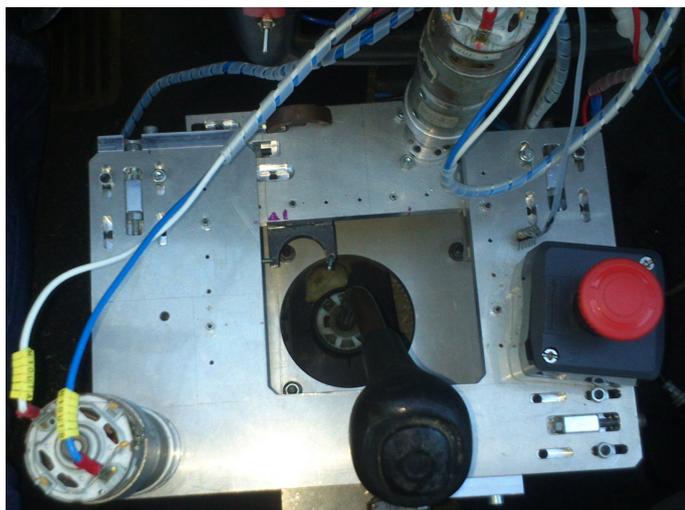


Figura 3.14: Comutador de velocidades, montagem final- Vista de cima com sistema de-sacoplado



Figura 3.15: Comutador de velocidades, montagem final

3.2 Sistema eletrônico

Da mesma forma que se avaliaram as falhas do sistema mecânico, também se procurou avaliar as falhas no sistema eletrônico. Após vários testes em bancada e simulação de situações

críticas (e.g. bloqueios), conclui-se que o sistema era suficientemente robusto para a aplicação em causa. Assim, apenas foram alterados alguns pormenores de forma a tornar o sistema mais seguro, quando implementado no veículo.

As alterações realizadas no sistema eletrónico, consistiram em:

- alterar o unidade de controlo - Unidade de Controlo
- implementar uma segurança a nível de *hardware* para proteção da caixa do AtlasCar - Segurança

3.2.1 Unidade de Controlo

No modelo existente é utilizado um sistema Arduino, para processamento, conjuntamente com um *Ethernet Shield*. O novo modelo utiliza um arduino *Ethernet* (fig. 3.16) que junta num único bloco a unidade de processamento e de comunicação. Desta forma, reduz-se a possível ocorrência de desconexão entre as duas unidades de controlo utilizadas anteriormente, como se reduz o espaço necessário, tornando assim o sistema eletrónico mais compacto.

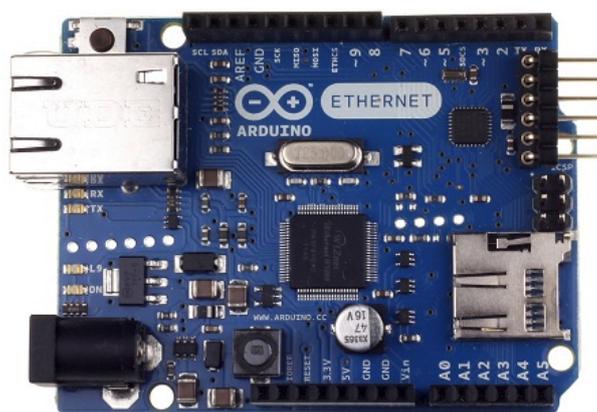


Figura 3.16: Unidade de controlo da caixa de velocidades [20]

3.2.2 Segurança

O sistema previamente existente apenas continha um dispositivo de segurança que consiste num botão de emergência que corta a alimentação dos motores. Esta segurança apenas efetua o corte de energia quando o utilizador pressionar o botão de emergência, contudo, é necessário uma segurança que permita proteger a caixa de velocidades do veículo, isto é, uma segurança que impossibilite o movimento do comutador sempre que a embraiagem não se encontra pressionada.

Para o efeito, foi desenvolvido um sistema de segurança que consiste na implementação de um fim de curso (fig. 3.17a), que quando pressionado permite que a caixa comute de mudança. De forma a que utilize uma segurança que contemple falhas nas ligações, sempre que o fim de curso não se encontra pressionado o valor recebido no Arduino é 0, ou seja, não é permitida comutação de velocidades; caso este se encontra pressionado, o valor recebido é 1, e é permitido comutar velocidade. Como medida de prevenção, se durante uma comutação o fim de curso deixar de estar pressionado, o comutador irá parar o movimento e esperar que este volte a ser pressionado (ativo) para continuar a comutação.



Figura 3.17: Fim de curso

Como o acionamento da embraiagem é feito diretamente no cabo, o pedal movimenta-se, mas apenas sob a força criada por uma mola, o que faz com que o deslocamento do pedal seja menor quando acionado automaticamente ou manualmente. Por esta razão, foi necessário aplicar uma mola na ponta do fim de curso (fig. 3.17b), para quando se pressionar o pedal manualmente, este não fique bloqueado numa posição demasiado recuada. Contudo, quando acionada automaticamente, a mola, ao tocar na base inferior do veículo, ativa o fim de curso e permite o movimento do comutador, pois a embraiagem já se encontra totalmente pressionada.



Figura 3.18: Sistema de montagem

Para a implementação do fim de curso é necessário um pino na placa do Arduino, e sendo que as entradas existentes se encontram todas utilizadas (tabela 3.4), foi necessário retirar o display de 7 segmentos, libertando-se assim os pinos A3, A4 e A5. Dos três pinos livres optou-se por utilizar o pino A5 para receber o sinal do fim de curso. Sendo que as entradas não foram alteradas na transição de unidade de controlo, a tabela 3.5 contém apenas os *Analogue Input*, que ilustra as alterações feitas para a implementação do fim de curso.

Tabela 3.4: Pinos disponíveis no modelo existente [15]

	Pins	Arduino UNO	Ethernet "Shield"	Used For	I/O
Digital IO	0	RX	RX	—	—
	1	TX	RX	—	—
	2	FREE	FREE	Encoder Bit A0	Output
	4	FREE	SDCS	—	—
	7	FREE	FREE	Encoder ENABLE	Output
	8	FREE	FREE	MAN/AUTO Select	Input
	12	FREE	SPI	—	—
PWM	13	FREE	SPI	—	—
	3	FREE	FREE	Encoder Bit A1	Output
	5	FREE	FREE	PWM Motor 2	Output
	6	FREE	FREE	PWM Motor 1	Output
	9	FREE	FREE	MAN/AUTO Gear Up	Input
	10	FREE	ETHCS	—	—
Analogue Input	11	FREE	ETHCS	—	—
	A0	FREE	FREE	Potentiometer 1	Input
	A1	FREE	FREE	Potentiometer 2	Input
	A2	FREE	FREE	MAN/AUTO Gear Down	Input
	A3	FREE	FREE	7-Seg Bit A	Output
	A4	FREE	FREE	7-Seg Bit B	Output
	A5	FREE	FREE	7-Seg Bit C	Output

Tabela 3.5: Pinos disponíveis no modelo final

	Pins	Arduino Ethernet	Used For	I/O
Analogue Input	A0	FREE	Potentiometer 1	Input
	A1	FREE	Potentiometer 2	Input
	A2	FREE	MAN/AUTO Gear Down	Input
	A3	FREE	—	—
	A4	FREE	—	—
	A5	FREE	Mechanical Limit	Input

3.3 Acionamento de pedais

Em relação aos pedais, apenas se detetaram algumas falhas durante os testes, nomeadamente as seguintes:

- más conexões nas ligações dos sistemas de acionamento dos pedais
- atraso na resposta do acelerador
- longo período de tempo para pressionar a embraiagem
- longa distância percorrida para imobilização do veículo

Após determinadas as ocorrências, apontadas anteriormente, foram realizadas as devidas intervenções. Sendo que apenas foi necessário ajustar alguns cabos e fios de forma a solucionar o problema do tempo de travagem e da má conexão.

Em relação ao sistema de embraiagem foi pensado num sistema de redução, apresentado na figura 3.19, para aumentar a velocidade de acionamento. Contudo deixou-se esta ideia de parte, pois iria fazer com que o alívio do pedal fosse demasiado rápido para o arranque. Decidiu-se assim deixar a embraiagem no mesmo estado, propondo-se, para trabalho futuro, a introdução de um sistema de acionamento variável.

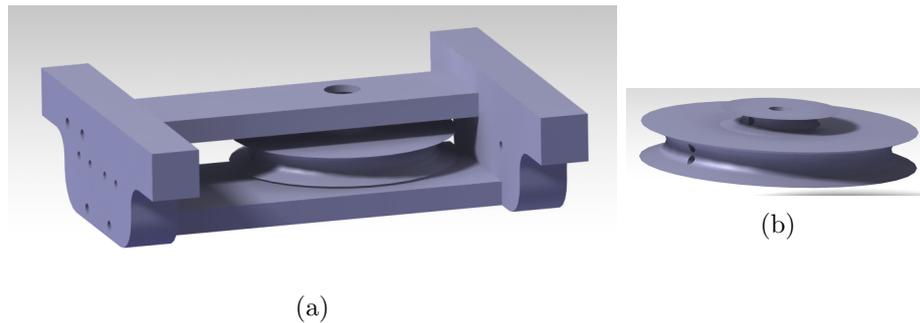


Figura 3.19: Sistema redutor para acionamento da embraiagem

Por último, tem-se o atraso na resposta do acelerador que, após algumas verificações e análises, se pensa ser o aquecimento excessivo da borboleta ou de algum dos componentes de controlo. Como até à data não se conseguiu descobrir exatamente o problema, é proposto, mais uma vez como trabalho futuro, a resolução deste atraso de resposta.

Capítulo 4

Software para manobras

Neste capítulo explica-se a estrutura do *software* desenvolvido para o AtlasCar e o modelo utilizado para a implementação das várias manobras.

4.1 Definições

Para facilitar a compreensão do capítulo, apresenta-se a definição de alguns conceitos utilizados:

- *Package*: define um grupo de executáveis responsáveis por algum processamento e/ou comunicação;
- Módulo: conjunto de *packages* da mesma categoria, por exemplo, *base* (permitem a comunicação com dispositivos de controlo), *sensors* (recebem informações de dispositivos, sensores, *encoders*, etc), entre outros;
- Estado: conjunto de variáveis que representam a situação de movimento do AtlasCar e incluem, especialmente, velocidades desejadas e correntes;
- Manobra: sequência ordenada de movimentos realizados pelo utilizador, por exemplo: avançar, recuar, estacionar, etc.

4.2 Arquitetura geral de *software*

O *software* no AtlasCar encontra-se dividido em vários módulos; contudo, nesta secção apenas se aborda o módulo *base*, onde está implementado o software desenvolvido para a realização das manobras. Para a realização das manobras são utilizados todos os *packages* do módulo *base* e o *package odometry* do módulo *sensors*.

Para facilitar a implementação, alterações e análise, o *software*, no AtlasCar, está dividido em três níveis:

- baixo nível - interação com atuadores e sensores
- médio nível - gestor de funções e comunicação entre módulos
- alto nível - unidade de interface

A figura 4.1 ilustra a estrutura para o caso do módulo *base*.

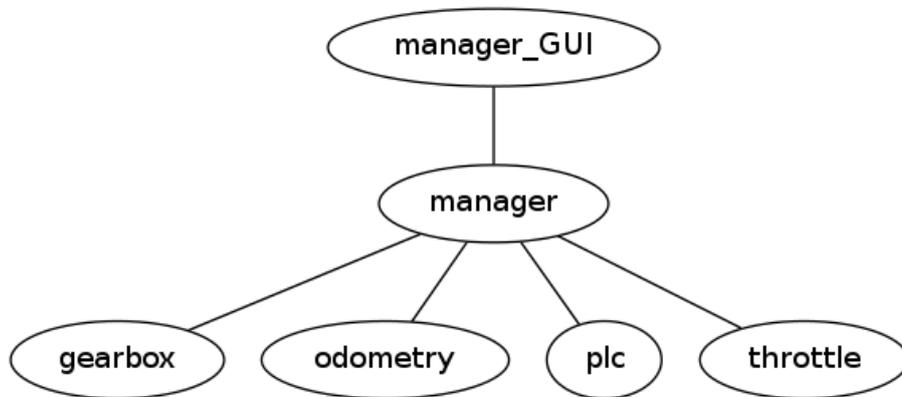


Figura 4.1: *Packages*: alto nível, médio nível e baixo nível, respetivamente (situação existente)

A interligação entre os vários níveis é realizada em ambiente ROS (*Robot Operating System*). Para formalizar uma exposição mais clara efetua-se, de seguida, uma abordagem detalhada de cada nível.

4.2.1 Baixo Nível

A classe de baixo nível do *software* é responsável pela comunicação com o *hardware*, isto é, recebe e envia dados/instruções para os dispositivos. Para cada dispositivo existe um *package*, que se encontram enumerados a seguir:

- *gearbox* [15]
- *odometry* [21]
- PLC [22]
- *throttle*

O *gearbox* é responsável pela comunicação com o comutador de velocidade, que por sua vez é responsável por ordenar uma comutação de velocidade, e pedir informações sobre a mudança atual. A *odometry* recebe informação do *encoder* que permite saber a velocidade e deslocamento do veículo. O *package PLC* comunica com o sistema físico (plc) controlando, assim, o travão, a embraiagem, a direção e as luzes. Por último, temos o *throttle* que está encarregue de controlar o acelerador, ou seja, acelera ou desacelera o veículo.

4.2.2 Médio Nível

Neste nível existe apenas um *package*, denominado *manager*, que surge como meio de ligação entre o alto e o baixo nível, tendo como função agrupar a informação vinda de todos os *packages* de baixo nível, processar a informação e enviar para o alto nível, bem como, receber informação do alto nível, processar e enviar para o vários *packages* de baixo nível. Para uma melhor perceção de toda a comunicação apresenta-se na figura 4.2 os tópicos ROS envolvidos no *software* a desenvolver.

O processamento realizado por este nível é o agrupamento/gestão de mensagens provenientes dos vários níveis, permitindo assim receber várias informações do baixo nível, agrupando tudo numa mensagem para enviar para o alto nível e/ou receber uma mensagem do alto nível separando a informação, da mesma, em várias mensagens para enviar para o baixo nível.

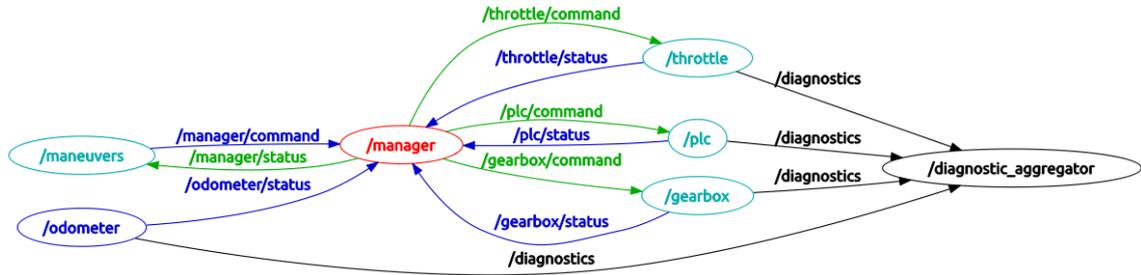


Figura 4.2: Tópicos *ROS*

4.2.3 Alto Nível

Por último, temos o alto nível que é responsável por interpretar os pedidos feitos pelo utilizador, como por exemplo, receber informações enviadas por um *gamepad* e enviar essa informação ao médio nível para que posteriormente seja enviado ao baixo nível. Desta forma, pode-se definir este nível como a interface e meio de comunicação com o utilizador.

Neste nível existe um *package* chamado *manage_GUI*, responsável por receber informações de um *gamepad* que permitem controlar vários parâmetros do AtlasCar, tais como, ativar/desativar o modo automático dos vários pedais e volante, acionar as luzes, acelerar, travar, pressionar a embraiagem, mudar de direção, etc. Além da comunicação com o utilizador este *package* também comunica com o médio nível, do qual, recebe informações, como velocidade, rpm, estado do acelerador, do travão, da embraiagem, das luzes, entre outros.

4.3 Implementações

Para a implementação do *software*, responsável pela realização das várias manobras, foi necessário acrescentar e alterar alguns *packages* existentes. Neste caso em específico, como se pretende ter uma utilização diferente do *gamepad*, ou seja, utilizar os botões para ações diferentes, foi criado um novo *package* de alto nível, denominado *maneuvers* (fig. 4.4). Desta forma, o *gamepad* permite ao utilizador indicar as seguintes manobras ao veículo (fig. 4.3):

- avançar
- recuar
- parar
- estacionar
- estacionar paralelamente
- inverter marcha



Figura 4.3: *Gamepad*

Ao pressionar o botão de avançar o utilizador pede ao *software* que o veículo se desloque com uma velocidade positiva (velocidade fixa de 2 m/s); de forma inversa, no caso de recuar a velocidade pedida é negativa (velocidade fixa de -2 m/s); o botão de parar pede uma velocidade de 0 m/s que irá imobilizar ou manter imobilizado o veículo, tendo em qualquer um dos casos uma rotação do volante de 0 rad . O pressionar do botão estacionar apenas faz com que o veículo coloca a mudança neutra, após mobilizado o veículo. Por último, temos o botão estacionar paralelamente e o inverter marcha que, ao contrário dos outros, solicitam ao sistema a realização de uma sequência de pedidos, em que as variáveis utilizadas são a velocidade, a direção do veículo e o deslocamento.

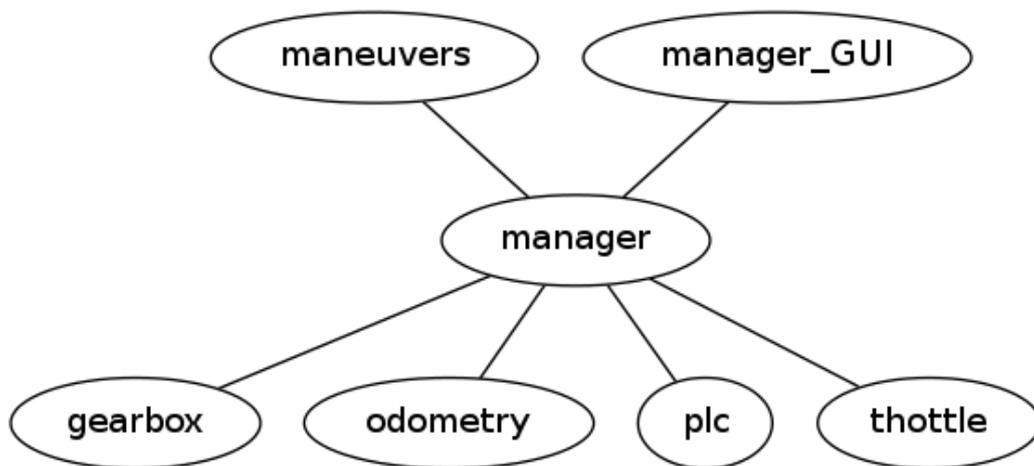


Figura 4.4: *Packages* utilizados para a implementação das várias manobras

Além do novo *package* de alto nível foram feitas algumas alterações nos *package manager* de médio nível e *gearbox* de baixo nível. No *gearbox* implementou-se o envio de uma mensagem com o intuito do hardware informar o *software* sobre o estado do comutador (mudança engrenada). Relativamente ao *package manager* foram feitas várias alterações, tais como, implementação de uma máquina de estados e implementação de parâmetros na mensagem de *status* e *command*.

Para isto, foram implementadas duas estruturas de informações e de procedimentos:

- estados
- manobras

Os estados estão implementados no *package manager* e são responsáveis pelos movimentos base do veículo, como por exemplo, arranque, paragem, andamento, etc. As manobras estão no *package maneuvers* e estão encarregues de executar uma sequência de pedidos para realizar manobras como estacionar, avançar, parar, etc.

4.4 Estados

Como já foi referido, os estados descrevem os contextos de movimento do veículo e são determinantes para realizar as manobras. Para a sequenciação dos estados foi criada uma máquina de estados (fig. 4.5). De forma a simplificar a análise e implementação do código foram criados apenas quatro estados principais:

- parado
- arranque
- paragem
- movimento

Os quatro estados apresentados são o mínimo necessário e suficiente para descrever e realizar as manobras. Contudo, alguns destes estados têm uma execução mais complexa, para os quais, foram criados sub-estados internos que têm associadas ações sequenciais (fig. 4.6).

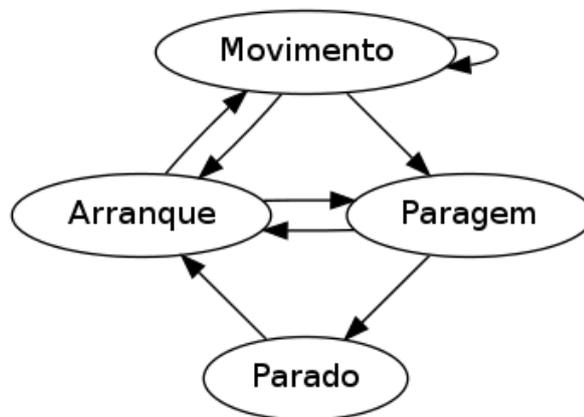


Figura 4.5: Máquina de estados principais e as transições permitidas

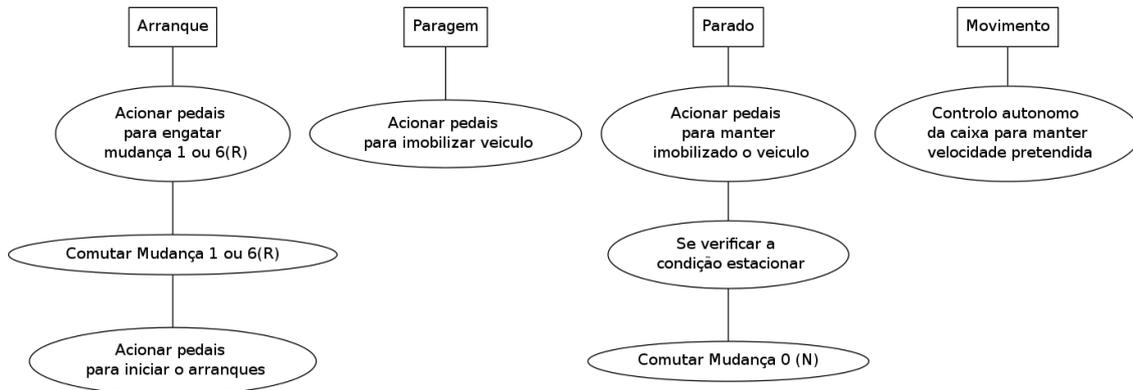


Figura 4.6: Sub-estados e ações associadas

Tabela 4.1: Condições para a definição dos estados (limite = velocidade de transição de arranque para andamento)

Estado	Velocidade atual [<i>Km/h</i>]	Velocidade pretendida [<i>Km/h</i>]
Parado	0	0
Arranque	< limite	> limite
Paragem	> 0	0
Movimento	> limite	> limite

4.4.1 Parado

O estado "parado", como o nome indica, corresponde ao veículo imobilizado. Nesta situação o *manager* pede ao sistema que acione o travão e a embraiagem. As condições para se entrar neste estado é quando são quando é pedida uma velocidade de 0 *Km/h* e o veículo está imobilizado, ou seja, com 0 *Km/h* (tabela 4.1). Este estado, tem uma variável extra que comuta a caixa de velocidade para a posição neutra no caso do utilizador pretender estacionar o carro (pressionar a tecla estacionar do *gamepad*), se não se pretender estacionar o veículo, não é realizada qualquer comutação, permanecendo a mudança onde está (1^a ou R).

4.4.2 Arranque

O "arranque", como se pode visualizar na figura 4.6, tem um conjunto de ações mais complexas que necessita de sub-estados para realizar os vários estágios relativos à manobra. Entra-se neste estado quando há um pedido de velocidade superior à velocidade limite e velocidade atual for menor que a velocidade limite (tabela 4.1).

O primeiro estágio aciona o travão, a embraiagem, bem como a comutação de caixa para a velocidade 1 - no caso de se pretender uma velocidade positiva - ou, para velocidade 6 (R) - no caso de se pretender uma velocidade negativa. Após concluída a comutação, dá-se início ao segundo estágio, que acelera o veículo e deixa de pressionar o travão e embraiagem, dando início ao movimento (fig. 4.6).

4.4.3 Paragem

A "paragem" surge da necessidade de imobilizar o veículo, ou seja, este estado ocorre quando se tem uma velocidade superior a 0 Km/h e se pretende uma velocidade igual a 0 Km/h (tabela 4.1). Para a imobilização do veículo são pressionados os pedais da embraiagem e do travão e aliviado o acelerador.

4.4.4 Movimento

Por último, temos o estado "moviementto" onde é feita a gestão de velocidades após o arranque. Assim, este estado corresponde à necessidade de ter e querer uma velocidade superior à velocidade limite (tabela 4.1).

Atualmente, as ações realizadas neste estado apenas mantém o acelerador numa posição fixa, de forma a manter uma velocidade baixa e constante, não necessitando de mudanças superiores à primeira. Contudo, é deixado para trabalho futuro a implementação de um sistema de controlo autónomo da caixa, já anteriormente desenvolvido, em *Matlab*, por Pedro Pinheiro [15].

4.5 Manobras

A manobra é uma sequência de estágios caracterizados por velocidade, direção e condição de fim de estágio. Como já foi referido, as manobras foram implementadas no *package maneuvers*, sendo que até a data foram implementadas seis manobras: avançar, recuar, parar, estacionar paralelamente, inverter marcha e parquear. Estas são pedidas pelo utilizador por meio do *gamepad* (fig. 4.3), como abordado.

Para a implementação das manobras foi realizada uma máquina de estados em que as condições de entrada são as manobras pretendidas, desta forma, sempre que se carrega no *gamepad* altera-se o estado, iniciando-se a manobra pedida, como segurança foi implementado um estado de transição, que permite parar as manobras e impossibilitar a alteração da manobra enquanto esta não estiver terminada (fig. 4.7).

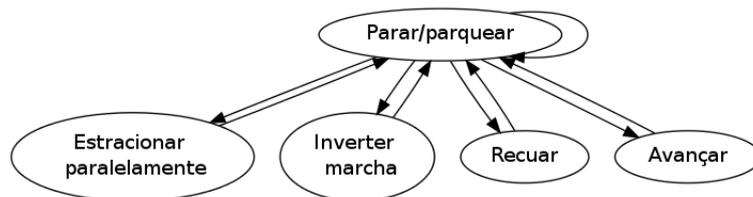


Figura 4.7: Sequência de ações

Nas manobras "avançar", "recuar" e "parar", como apenas é imposta uma velocidade e uma rotação do volante, não sendo necessária nenhuma condição para final de manobra, apenas existe um estágio, criado dentro de cada um dos estados. Assim, em ambas as manobras é enviada para o *manager* um rotação do volante que varia entre -0.5 e 0.5 *rad*, bem como uma velocidade, em que para a manobra "avançar" é enviada uma velocidade positiva, no "recuar" uma velocidade negativa e, por último, no "parar" uma velocidade de 0 *m/s*.

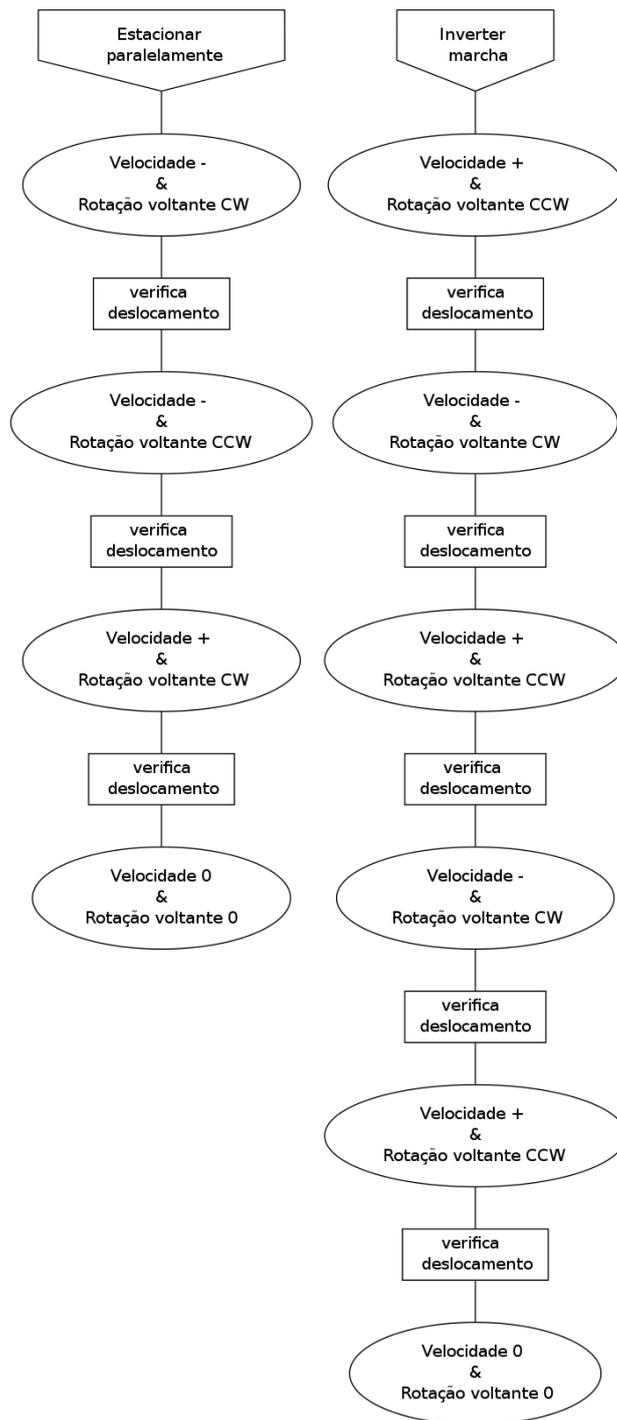


Figura 4.8: Sequência de ações

No caso da manobra "estacionar paralelamente" e "inverter marcha", como são manobras mais complexas, é necessário um maior número de ações, o que implica um maior número de estágios. Para o efeito, foram criadas funções com os vários estágios para cada uma das manobras (fig. 4.8). Nestas manobras, cada estágio, além de impor uma velocidade e direção,

contêm uma condição de fim de estágio que, neste caso, é o deslocamento. Entre cada estágio é pedida uma velocidade de 0 m/s como segurança e/ou para permitir a rotação do volante para a posição pretendida.

A manobra "estacionar paralelamente", como se pode ver na figura 4.8, está dividida em quatro estágios. Inicia-se a manobra impondo uma velocidade negativa e uma rotação do volante no sentido dos ponteiros do relógio (CW). Após verificado o deslocamento, inicia-se o segundo estágio, onde é rodado o volante no sentido contrário aos ponteiros do relógio (CCW), e impõe-se na mesma uma velocidade negativa. De forma a ajustar a posição do veículo após verificar o deslocamento, entra-se no terceiro estágio, onde é enviada uma velocidade positiva e uma rotação no sentido dos ponteiros do relógio (CW). Quando concluído o terceiro deslocamento dá-se a entrada no último estágio que impõe uma velocidade e rotação igual a 0 m/s e 0 rad , respetivamente.

A manobra de "inverter marcha" (fig. 4.8), da mesma forma que na manobra "estacionar paralelamente", define-se uma velocidade e rotação, sendo que a transição também é condicionada pelo deslocamento. Com isto, quando a manobra é iniciada, define-se uma velocidade positiva e rotação contrária aos ponteiros do relógio (CCW). De seguida e após verificado o deslocamento necessário, altera-se a velocidade para um valor negativo e a rotação do volante para o sentido oposto (sentido dos ponteiros do relógio (CW)). Após verificado o segundo deslocamento, entra-se no terceiro estágio onde a velocidade é positiva e a rotação do volante é a contrária ao sentido dos ponteiros do relógio (CCW). Quando é verificado o novo deslocamento é iniciado o quarto estágio, onde é imposta uma velocidade negativa e uma rotação do volante no sentido dos ponteiros do relógio (CW). Da mesma forma é verificado um novo deslocamento, que em seguida dá início ao quinto estágio onde é indicada uma velocidade positiva e rotação contrária aos ponteiros do relógio (CCW). Por último, temos o sexto estágio, que apenas ocorre após se verificar o quinto estágio, colocando a velocidade e a rotação do volante a 0 m/s e 0 rad , respetivamente.

Para finalizar, tem-se a manobra "parquear" que apenas indica ao *package manager* a intenção de se desengatar o veículo, por meio de uma variável. Nesta situação o *package manager* inicia a comutação da caixa para a velocidade 0 (N) .

Capítulo 5

Experiências e Resultados

Neste capítulo aborda-se as experiências realizadas para testar o sistema desenvolvido e os respectivos resultados.

5.1 Experiências

Para provar a fiabilidade e robustez da solução desenvolvida ao longo desta dissertação, bem como, dos vários sistemas existentes no AtlasCar, necessários para a realização de manobras, foram realizados alguns testes.

Além de se verificar a execução das manobras "avançar", "recuar", "parar", "estacionar paralelamente", "inverter marcha" e "estacionar", para as manobras de "estacionar paralelamente" e "inverter marcha" realizou-se um série de cinco repetições, para cada manobra, com as mesmas condições (tabelas 5.1 e 5.2), durante as quais foram retirados vários valores, tais como, velocidade, deslocamento, rotação do volante, entre outros, de forma a mostrar a repetibilidade das manobras.

Tabela 5.1: Parâmetros utilizados para a realização a manobra "inverter marcha"

Número da ação	Velocidade [m/s]	Rotação do volante [rad]	Condição de execução [pulsos]
1	2	0.45	10
2	-2	-0.45	-2
3	2	0.45	2
4	-2	-0.45	-2
5	2	0.45	5
6	0	0	—

De forma a testar o comutador, foram realizados alguns testes, onde se executaram séries de comutação, como o motor a funcionar e a embraiagem pressionada, que permitiram retirar vários valores como tempos e/ou falha nas comutações.

Os resultados e representações gráficas estão expostos na secção seguinte.

Tabela 5.2: Parâmetros utilizados para a realização da manobra "estacionar paralelamente"

Número da ação	Velocidade [m/s]	Rotação do volante [rad]	Condição de execução [pulsos]
1	-2	-0.4	-10
2	-2	0.4	-2
3	2	-0.4	1
4	0	0	—

5.2 Resultados

Após realizadas as manobras e respectivos testes, concluiu-se que a execução das manobras é satisfatória mas pouco repetitiva. Da mesma forma, após realizados os testes de comutação é possível mostrar que é um sistema suficientemente resistente e preciso.

Para provar as afirmações anteriores apresentam-se alguns gráficos e tabelas. Relativamente ao comutador, podem ser visualizados os tempos de comutação das várias velocidades no gráfico de colunas da figura 5.1, em que, para cada comutação foram executados 5 ensaios. É também de salientar que durante os ensaios não ocorreu nenhuma falha na comutação das mudanças, ou seja, o comutador permitiu comutar 60 vezes, em blocos de 10 comutações.

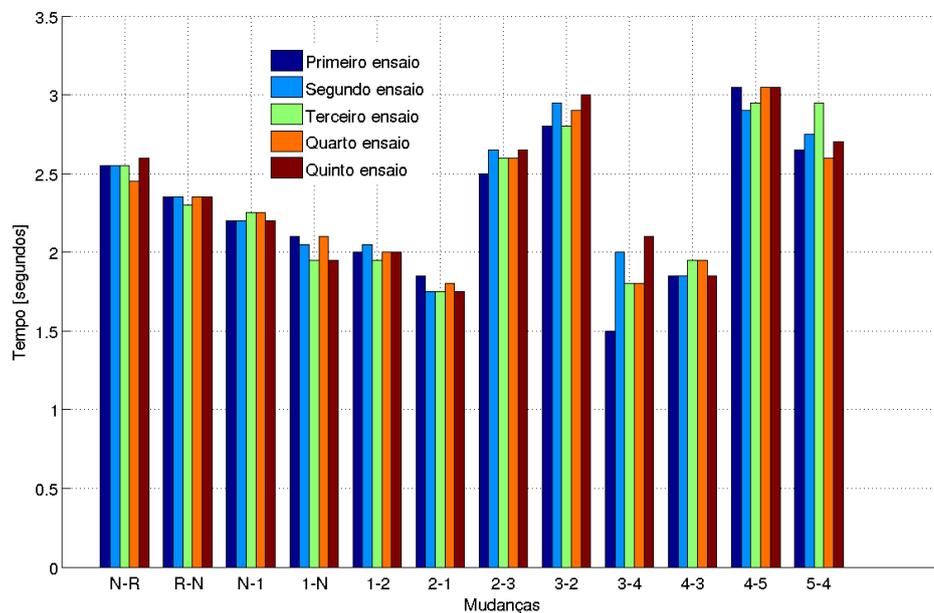


Figura 5.1: Tempos de comutação das várias velocidades

Nas manobras, como referido, foram retirados vários valores no decorrer dos testes relativos às manobras. Porém, decidiu-se apenas apresentar alguns (velocidade e deslocamento), pois foram os que se mostraram mais relevantes para a repetibilidade das manobras. Assim as

representações gráficas das velocidades das manobras "estacionar paralelamente" e "inverter marcha" podem ser visualizadas nas figuras 5.2 e 5.3, respectivamente.

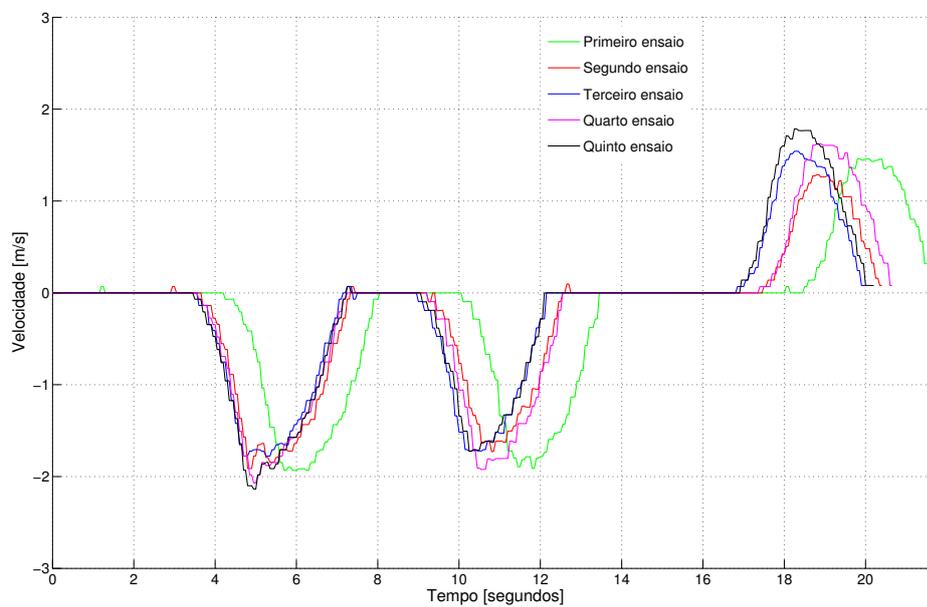


Figura 5.2: Velocidades atingidas durante a manobra de "estacionar paralelamente"

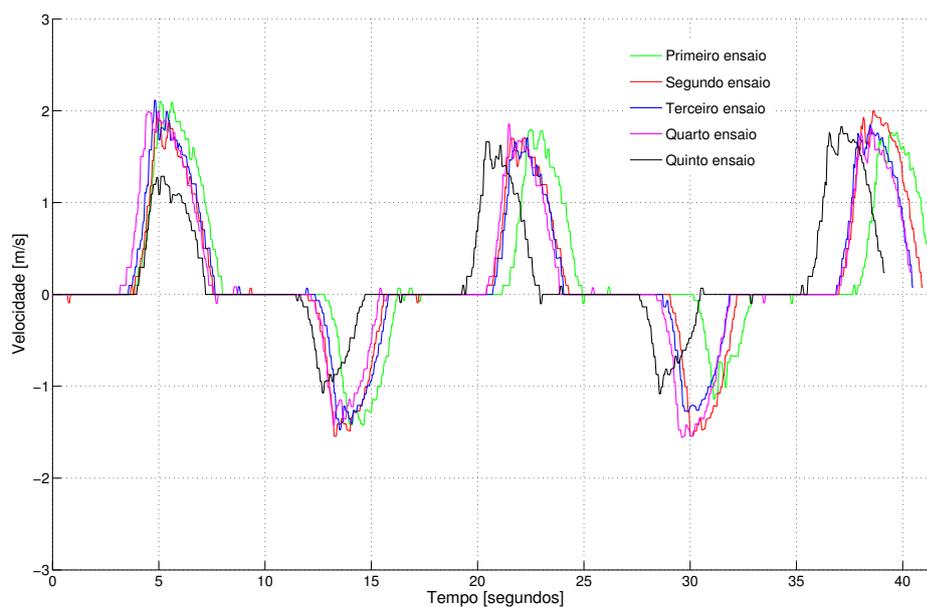


Figura 5.3: Velocidades atingidas durante a manobra de "inverter marcha"

O deslocamento, sendo um dos grandes fatores na boa repetibilidade das manobras, dado que permite definir o espaço percorrido pelo veículo, também é apresentado. Nas figuras 5.4 e 5.5, é possível visualizar os deslocamentos realizados pelo AtlasCar durante as manobras de "estacionar paralelamente" e "inverter marcha", respetivamente.

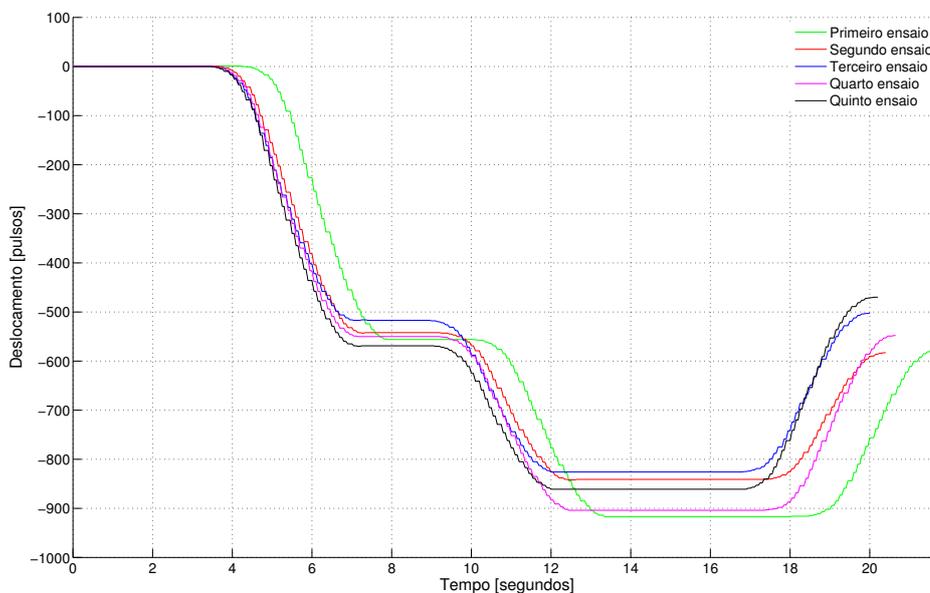


Figura 5.4: Deslocamentos realizados durante a manobra de "estacionar paralelamente"

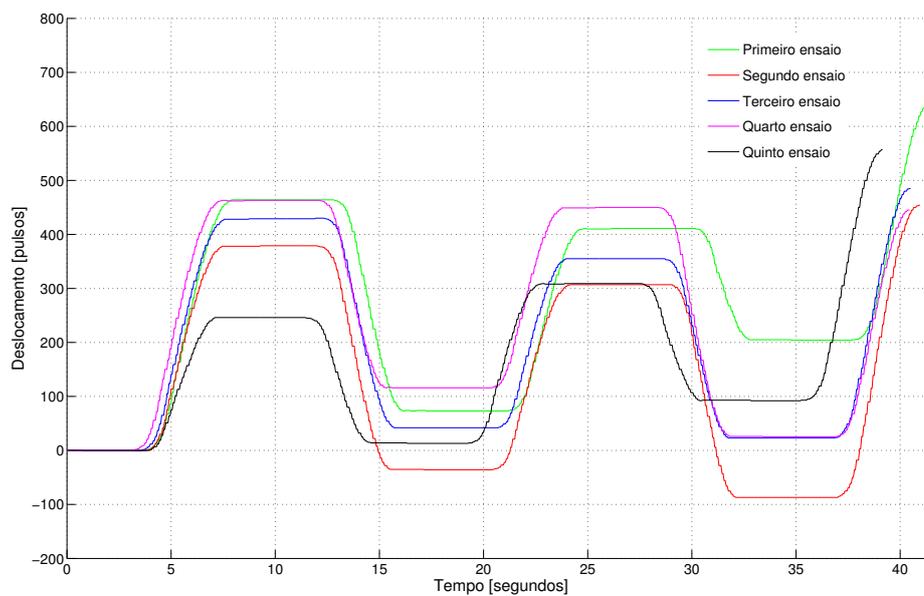


Figura 5.5: Deslocamentos realizados durante a manobra de "inverter marcha"

Para auxiliar a visualização da influência destes parâmetros na execução das trajetórias são apresentados dois gráficos com a representação das trajetórias, em xx e yy , realizadas durante a manobra "estacionar paralelamente" e "inverter marcha", nas figuras 5.6 e 5.7 respectivamente.

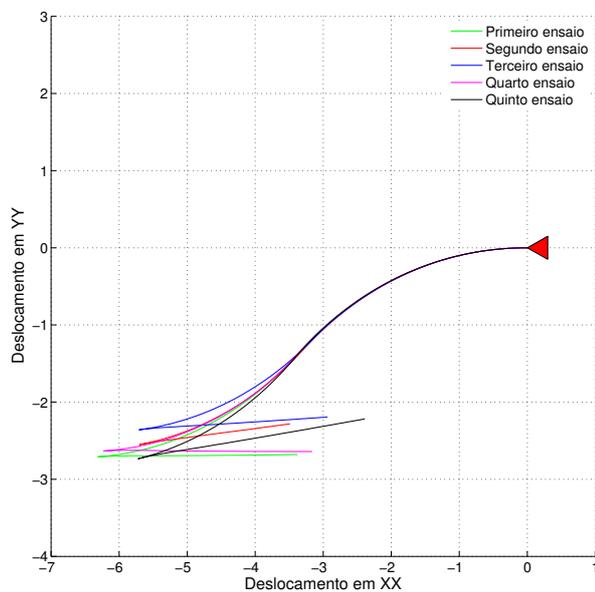


Figura 5.6: Trajetória descrita pelo veículo durante a manobra de "estacionar paralelamente"

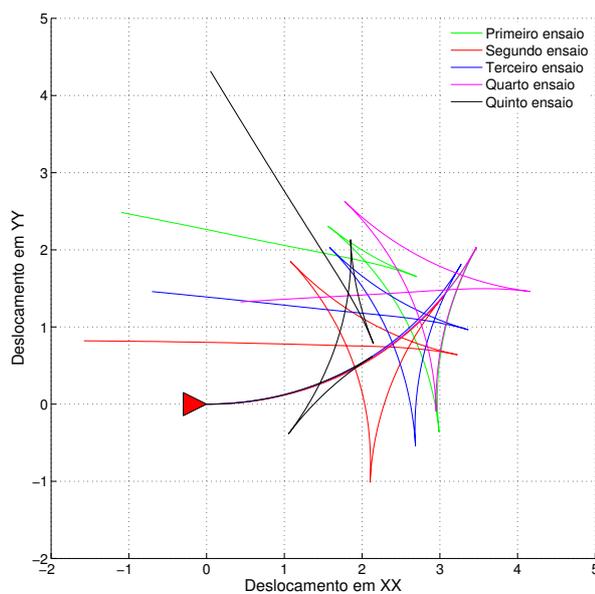


Figura 5.7: Trajetória descrita pelo veículo durante a manobra de "inverter marcha"

Da análise dos gráficos, conclui-se que a velocidade atingida não é constante ao longo das várias manobras, fenómeno que se deve a um problema técnico do AtlasCar que impossibilita a imposição de uma aceleração sempre igual. Como o método de travagem existente só tem uma velocidade de atuação, faz com que o tempo e distância de travagem variem em função das velocidades alcançadas, como se pode visualizar nos gráficos das figuras 5.4 e 5.5.

As diferenças nas trajetórias realizadas pelo veículo (fig. 5.6 e 5.7) devem-se às diferentes distâncias de travagem pois o *software* desenvolvido em malha aberta apenas contempla as distâncias percorridas antes do início da travagem. Assim, como não se controlam as distâncias de travagem, as trajetórias realizadas não são idênticas, verificando-se com mais intensidade neste tipo de manobras pois são trajetórias circulares.

De forma a solucionar esta falta de repetibilidade, poderiam ser pensadas e/ou desenvolvidas algumas soluções, tais como, controlo preciso do acelerador de forma a ter sempre uma aceleração constante, um sistema de travagem instantâneo com o intuito de evitar distâncias de travagem diferentes, e ainda a implementação de software de planeamento de trajetória, desenvolvido no passado por Joel Pereira [23].

Capítulo 6

Conclusões

Ao longo desta dissertação atingiram-se vários objetivos, durante os quais se retiraram várias conclusões, bem como, foram definidos alguns trabalhos a realizar no futuro. De forma a facilitar a interpretação das várias conclusões e proposta desses mesmos trabalhos, optou-se por dividir em quatro secções: sistema mecânico, sistema eletrónico, controlo de pedais e *software*.

No sistema mecânico, aquando do teste em bancada, conclui-se que é bastante robusto e satisfaz os requisitos. Contudo, após aplicado no AtlasCar, embora funcional, verificou-se que o sistema poderia ser um pouco mais pequeno de forma a permitir maior mobilidade aos passageiros e maior robustez ao sistema de transmissão. Assim, propõem-se que seja desenvolvido um sistema mais compacto e dimensionado, evitando-se o sobre-dimensionamento desnecessário, bem como, o desenvolvimento de um sistema de acoplamento dos veios aos carretos mais robusto.

Em relação ao sistema eletrónico, como no sistema mecânico, durante o teste em bancada este mostrou-se eficaz e robusto, cumprindo assim todos os requisitos necessários para uma boa comutação. Após aplicado no veículo, este mostrou algumas fragilidades que não comprometeram o seu funcionamento, mas que, devem ser revistas posteriormente de forma a tornar-se o sistema mais preciso. Assim, sugere-se a implementação de um novo sistema de calibração que, em vez de adquirir apenas algumas posições do manípulo, contemple todas as posições possíveis para as várias mudanças, e ainda uma interface de calibração. Além destas alterações, também se propõe a correção de um *bug* no modo automático do *software* de forma a impossibilitar a passagem direta da velocidade 5 para a 6(R), pois é mecanicamente impossível.

A secção relativa aos atuadores dos pedais é a parte onde o trabalho futuro revela ser mais importante, pois estes condicionam o funcionamento do *software* desenvolvido. Além dos presentes atuadores serem suficientes para a realização e conclusão desta dissertação, contêm algumas limitações, tais como, velocidades constantes de acionamento dos atuadores e conexões frágeis. Assim, para esta parte, sugere-se o desenvolvimento de um novo sistema de acionamento para ambos os pedais, embraiagem e travão, que permitam o acionamento com velocidade variável com o intuito de permitir uma travagem ou acionamento da embraiagem, mais rápida ou lenta, em função da necessidade. Com estes novos sistemas, também se resolveriam as fragilidades existentes nas conexões dos atuais atuadores.

Relativamente ao acelerador, que é controlado eletronicamente, verificou-se uma irregularidade na aceleração que impossibilita uma velocidade de arranque constante. Assim, sugere-se,

para trabalho, a resolução deste fenómeno, pois como já foi explicado anteriormente, influencia a boa execução das manobras.

O *software* desenvolvido, como foi referido, contém algumas limitações devido ao facto de ser realizado em malha aberta e não se ter controlo sobre os tempos de travagem e aceleração, não permitindo uma boa repetibilidade das manobras. Contudo, a forma simplificada como são descritas as ações para a realização das manobras, mostrou-se robusta e de fácil adaptação às várias situações. Assim, as grandes alterações a realizar no *software* incidem na sequência de ordens enviadas para realização das manobras; com isto, propõe-se a alteração das condições de movimento para variáveis provenientes do *software* de planeamento de trajetória, desenvolvido por Joel Pereira [23], que permitirá descrever melhor as manobras.

Ainda no *software* é deixado como trabalho a implementação do algoritmo, desenvolvido por Pedro Pinheiro [15], no estado "andamento", para permitir comutação de mudanças a velocidades mais elevadas.

Analisando o AtlasCar como um todo, conclui-se que este é capaz de realizar as manobras tendo algumas limitações na repetibilidade das mesmas quando mais complexas, o que, não deixa de tornar o AtlasCar um veículo autónomo, pois permite avançar ou recuar sem necessidade de um condutor para comutação de velocidades. Após solucionados os problemas apresentados anteriormente, prevê-se que possa ser um veículo autónomo e preciso na execução das várias manobras.

Para finalizar, após todos os esforços realizados durante um ano é possível executar a mudança de velocidade que, por sua vez, possibilita o avançar, o parar e o recuar. Estas novas funcionalidades trazem novos desafios como o desenvolvimento de sistemas de perceção na retaguarda do AtlasCar, tais sistemas, ainda não foram abordados devido à impossibilidade de recuar o veículo autonomamente.

Além de todas os avanços realizados, não é deixado de parte o grande fato de, além de implementados todos estes sistemas, o AtlasCar continuar homologado, ou seja, autoriza a circulação em via pública. Isto é conseguido em virtude do sistema de comutação implementado ter um sistema de desacoplamento do manípulo, concedendo assim uma condução totalmente manual.

Bibliografia

- [1] M. O. e Miguel Neta, “Desenvolvimento de um robot móvel para participação no festival robótica 2004, classe condução autónoma,” tech. rep., Universidade de Aveiro, 2004.
- [2] LARlabs, “Projecto atlas - universidade de aveiro.” <http://atlas.web.ua.pt/>, Março 2014.
- [3] B. N. Technology, “Nissan car drives and parks itself at ceatec.” <http://www.bbc.com/news/technology-19829906>, 2012.
- [4] M. O. S. . Tech, “The car that drives itself... using an ipad! oxford university unveils robot car.” <http://www.dailymail.co.uk/sciencetech/article-2278725/The-car-drives-using-iPad-Oxford-University-unveils-robot-car.html#ixzz2KtzUvWhy>, 2013.
- [5] B. N. Technology, “Toyota sneak previews self-drive car ahead of tech show.” <http://www.bbc.com/news/technology-20910769>, 2013.
- [6] T. Atlantic, “Google’s self-driving cars: 300,000 miles logged, not a single accident under computer control.” <http://www.theatlantic.com/technology/archive/2012/08/googles-self-driving-cars-300-000-miles-logged-not-a-single-accident-under-computer-control/260926/>, 2012.
- [7] Mercedes-Benz, “Autonomous long-distance drive.” <http://www5.mercedes-benz.com/en/innovation/autonomous-long-distance-drive-research-vehicle-s-500-intelligent-drive/>, 2014.
- [8] M. T. Review, “Stanford’s new driverless car.” <http://www.technologyreview.com/news/408059/star-new-driverless-car/>, 2007.
- [9] Vislab, “Automotive.” <http://vislab.it/automotive/>, 2014.
- [10] G. A. Bekey, *Autonomous Robots*. MIT PRESS LTD, 2005.
- [11] P. S. Stein, “Navigation in dynamic environments taking advantage of moving agents,” Master’s thesis, Universidade de Aveiro, 2013.
- [12] M. M. e Nuno Silva, “Atlascar - automatização da caixa de velocidades,” tech. rep., Universidade de Aveiro, 2011.
- [13] F. L. W. D. K. C. OBT, *975D Series 42mm (45mm motor) Planetary (Epicyclic) Metal Gearbox*.
- [14] V. Spectrol, *7/8” (22.2 mm) Multi Turn Wirewound Potentiometer - 533: 3 Turns/534: 10 Turns/535: 5 Turns*, 2014.

- [15] P. F. P. Pinheiro, “Sistema automático para gestão da caixa de velocidades do atlascar,” Master’s thesis, Universidade de Aveiro, 2012.
- [16] J. L. de Jesus Bettencourt Viana, “E-stop e consola de controlo e monitorização para o atlascar,” Master’s thesis, Universidade de Aveiro, 2013.
- [17] Eurocorreias, “Corrent de rolos c/ pino saido.” <http://www.eurocorreias.pt/catalogos/correntes.pdf>, Maio 2014.
- [18] Dunbelt, “Carretos e cremalheiras.” http://www.dunbelt.com/catalogos/cat_carretos.pdf, Dezembro 2008.
- [19] Igus, “Drylin® t - miniatura.” http://www.igus.pt/wpck/2201/DryLin_T_Miniaturfuehrung, 2014.
- [20] Arduino, “Arduino ethernet.” <http://arduino.cc/en/Main/ArduinoBoardEthernet>, 2014.
- [21] A. J. L. Oliveira, “Sistema de monitorização da condução de um automóvel,” Master’s thesis, Universidade de Aveiro, 2012.
- [22] T. N. da Rocha, “Piloto automático para controlo e manobras de navegação do atlascar,” Master’s thesis, Universidade de Aveiro, 2011.
- [23] J. F. Pereira, “Estacionamento autónomo usando perceção 3d autonomous parking using 3d perception,” tech. rep., Universidade de Aveiro, 2012.

Anexos

Intervenção no Acelerador

O sistema de alimentação existente no controlador do acelerador era proveniente da bateria do AtlasCar que fornecia uma tensão de aproximadamente $14V$, que alimentava a unidade de controlo *ETHERNET* e a placa de controlo do servo motor da borboleta do acelerador. De forma a alimentar o sistema com uma tensão regulável e adequada, implementou-se um regulador de tensão LM317, em conjunto com duas resistências de forma a ter um *output* de aproximadamente $8V$. Para atenuar os picos provenientes da bateria foi colocada em série uma bobine (fig. 6.1).

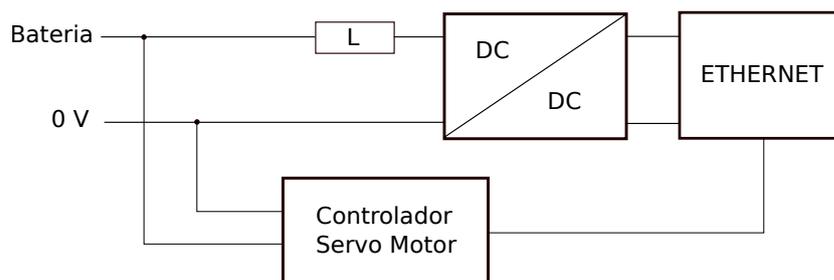


Figura 6.1: Esquema elétrico

Estas alterações foram realizadas para a alimentação da unidade de controlo *ETHERNET*, continuando-se a alimentar a placa de controlo do servo motor com os $14V$ proveniente da bateria do AtlasCar.

Desenhos de peças fabricadas

- Lista
 - Base maior;
 - Base menor;
 - Base inferior da coluna de ajuste;
 - Base superior da coluna de ajuste;
 - Base inferior da coluna fixa;
 - Ligantes das colunas;
 - Garra;

Olhal da garra;
Base de suporte;
Lateral do suporte;
Lateral pequena do suporte;
Adaptador da lateral pequena do suporte;
Veios;

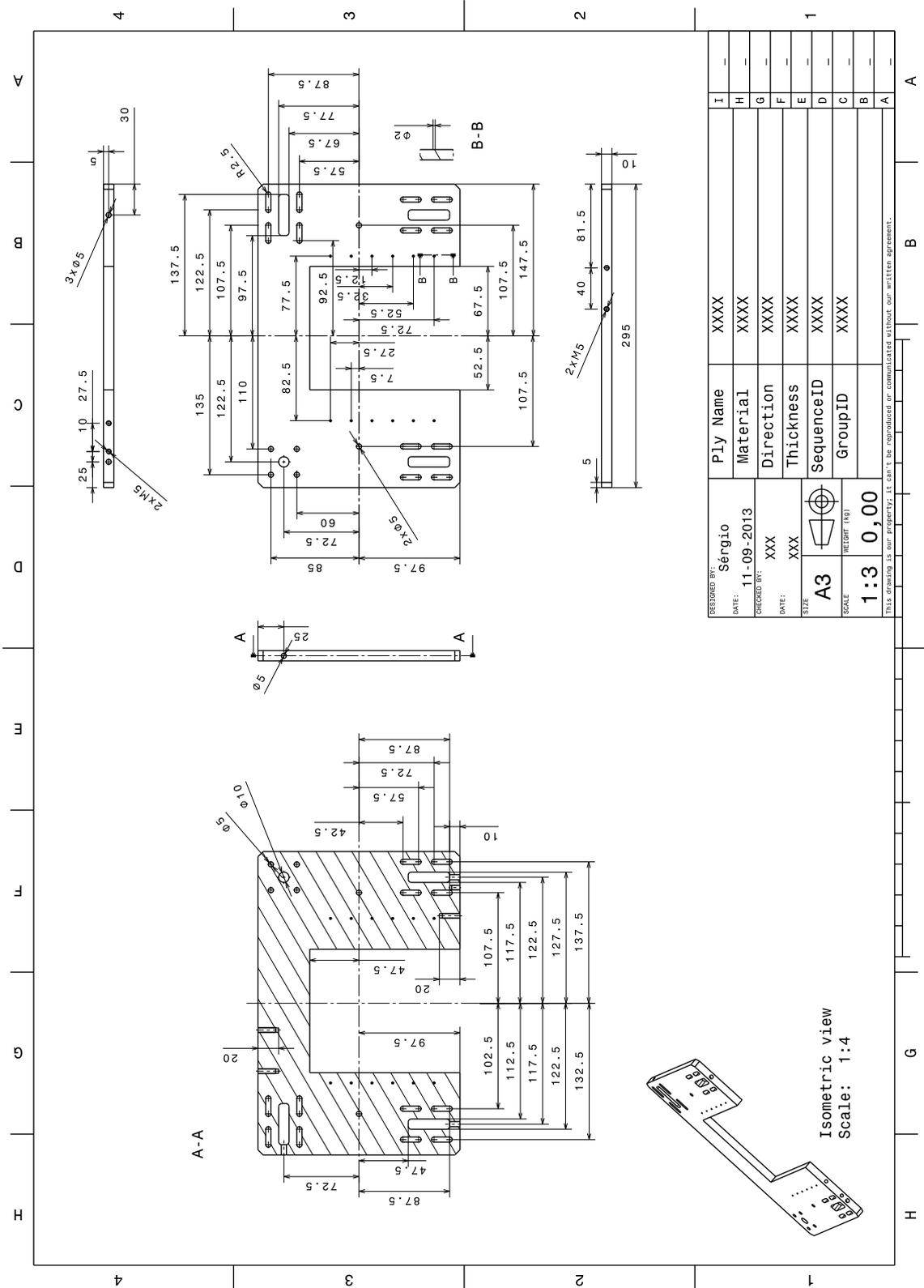


Figura 6.2: Desenho de definição base maior

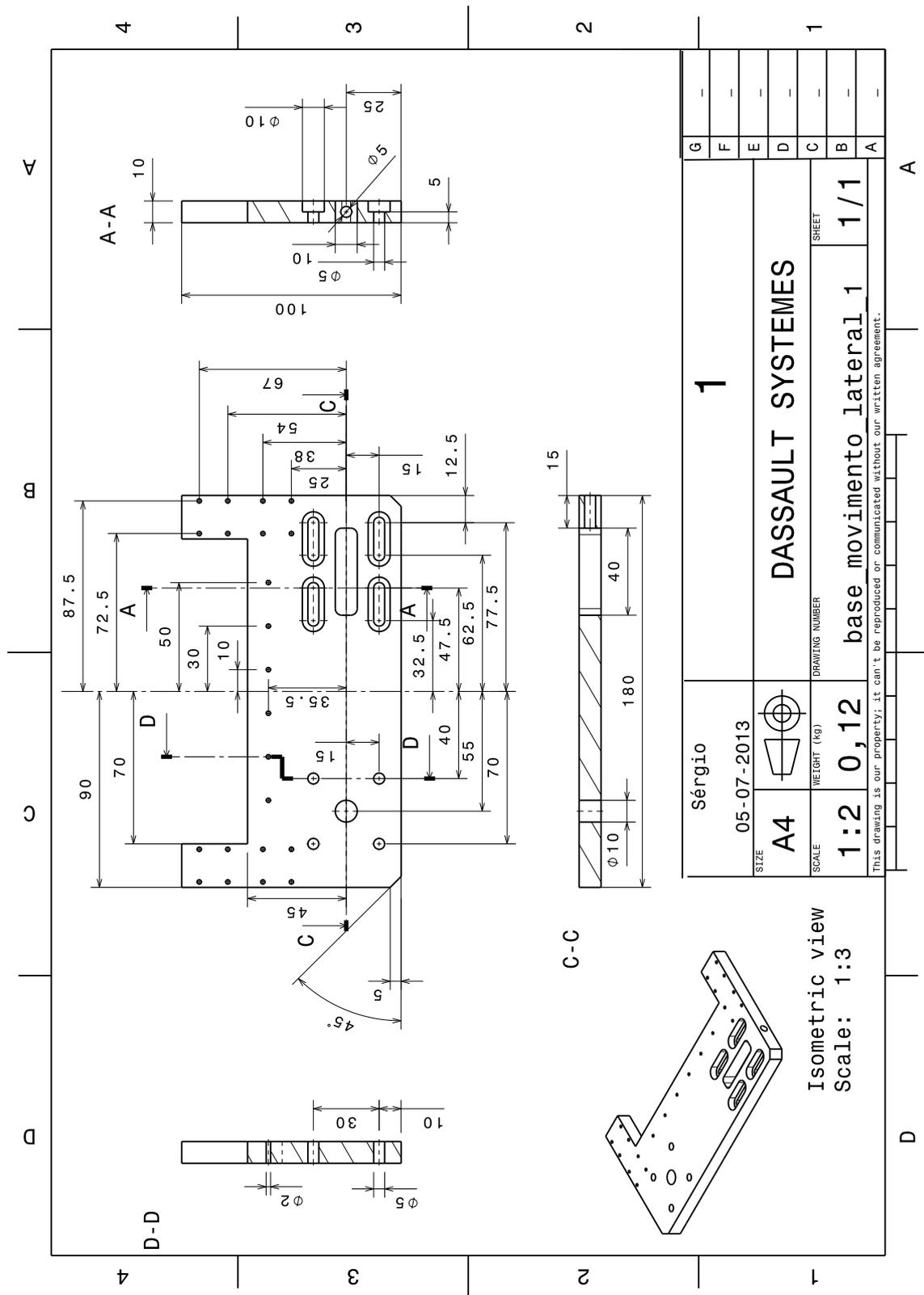


Figura 6.3: Desenho de definição base menor

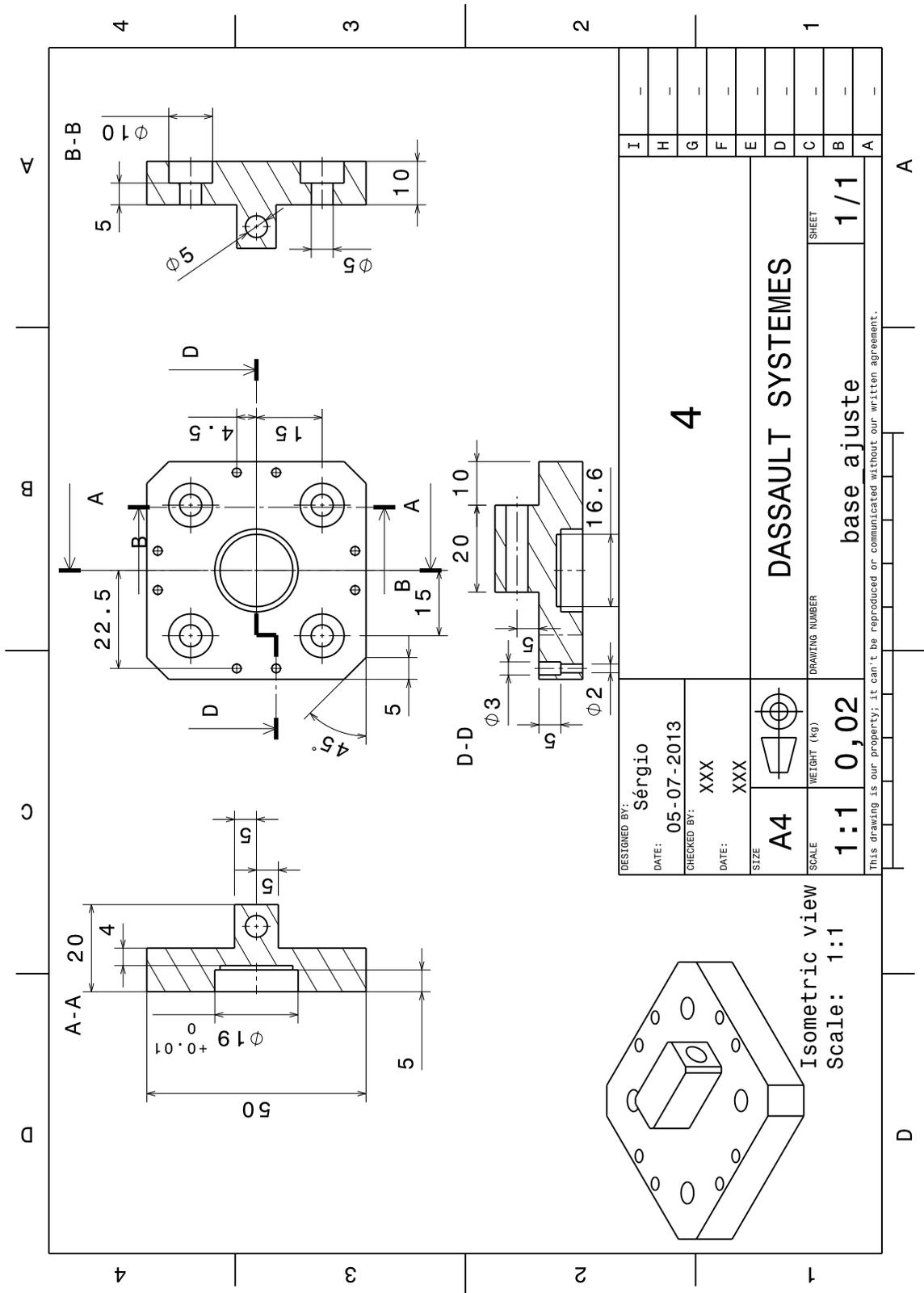


Figura 6.4: Desenho de definição base inferior da coluna de ajuste

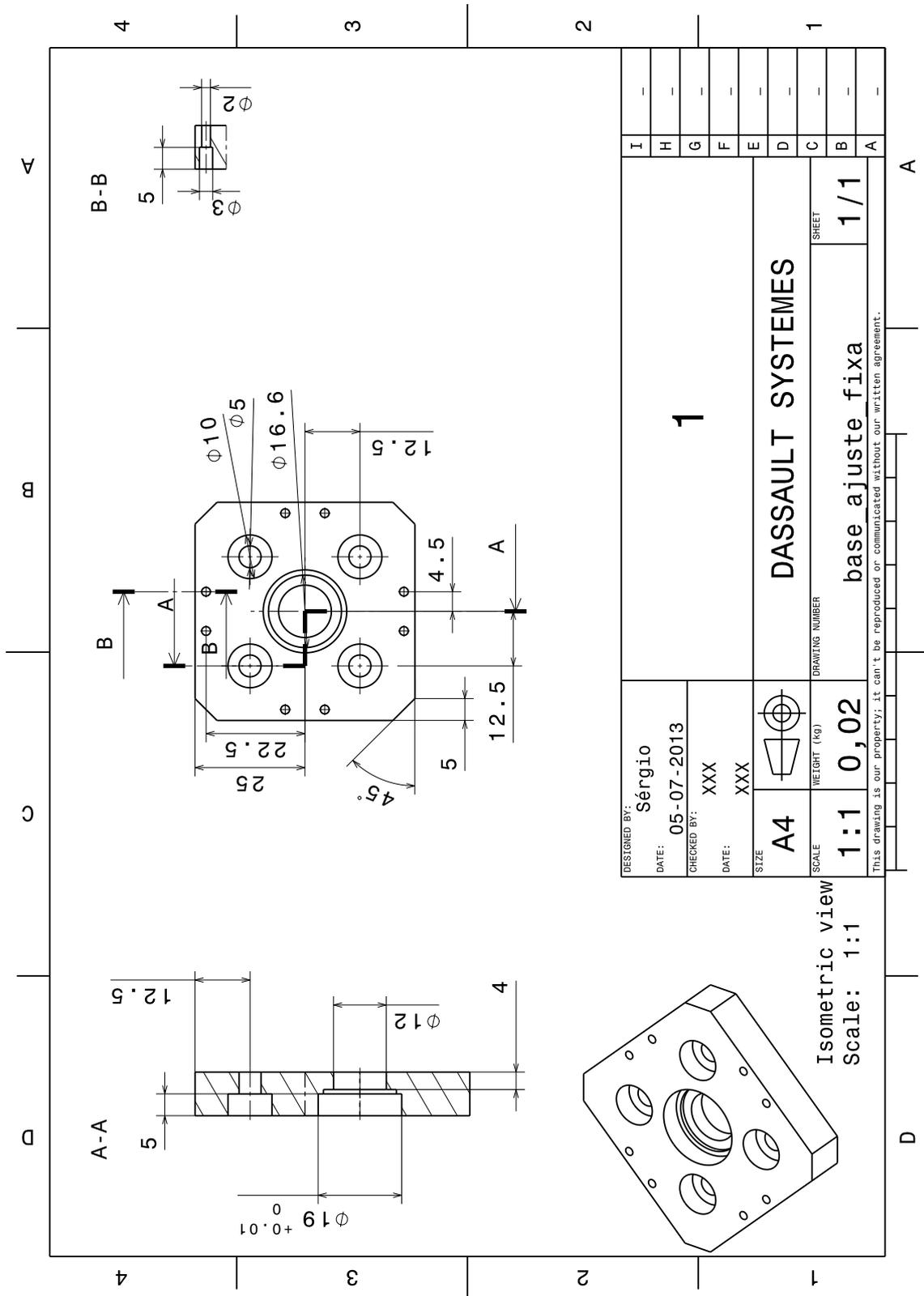


Figura 6.5: Desenho de definição base superior da coluna de ajuste

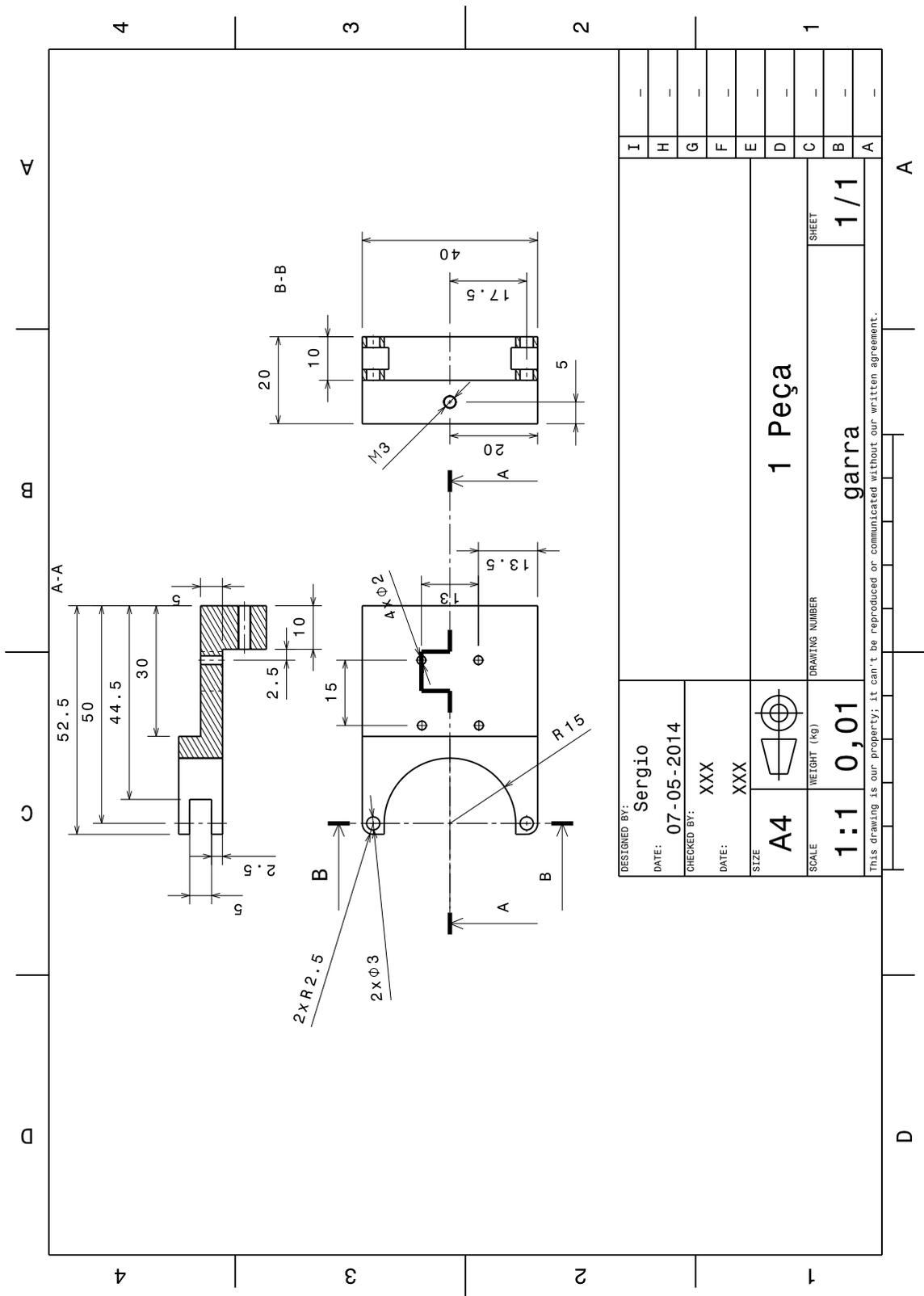


Figura 6.8: Desenho de definição da garra

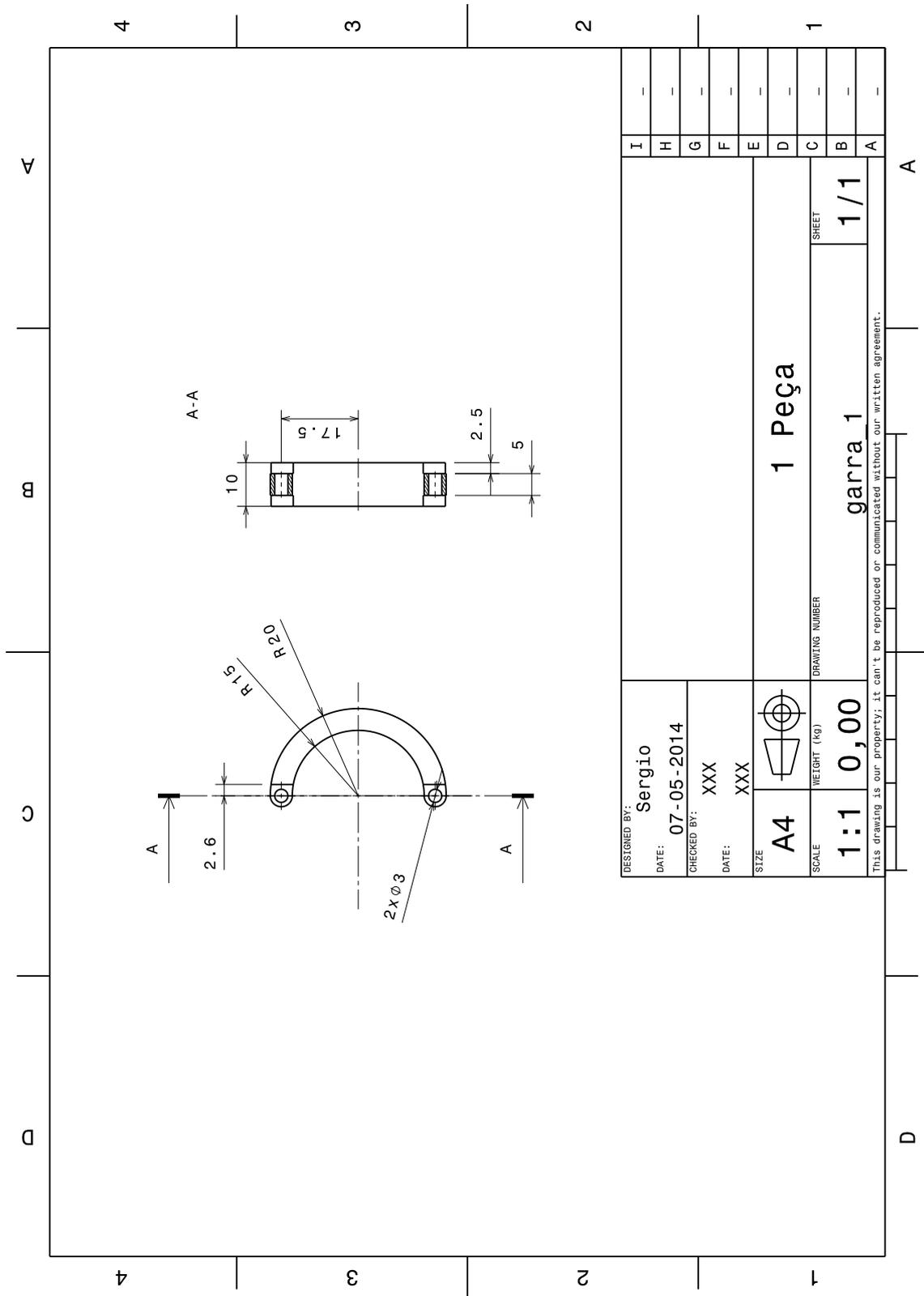


Figura 6.9: Desenho de definição do olhal da garra

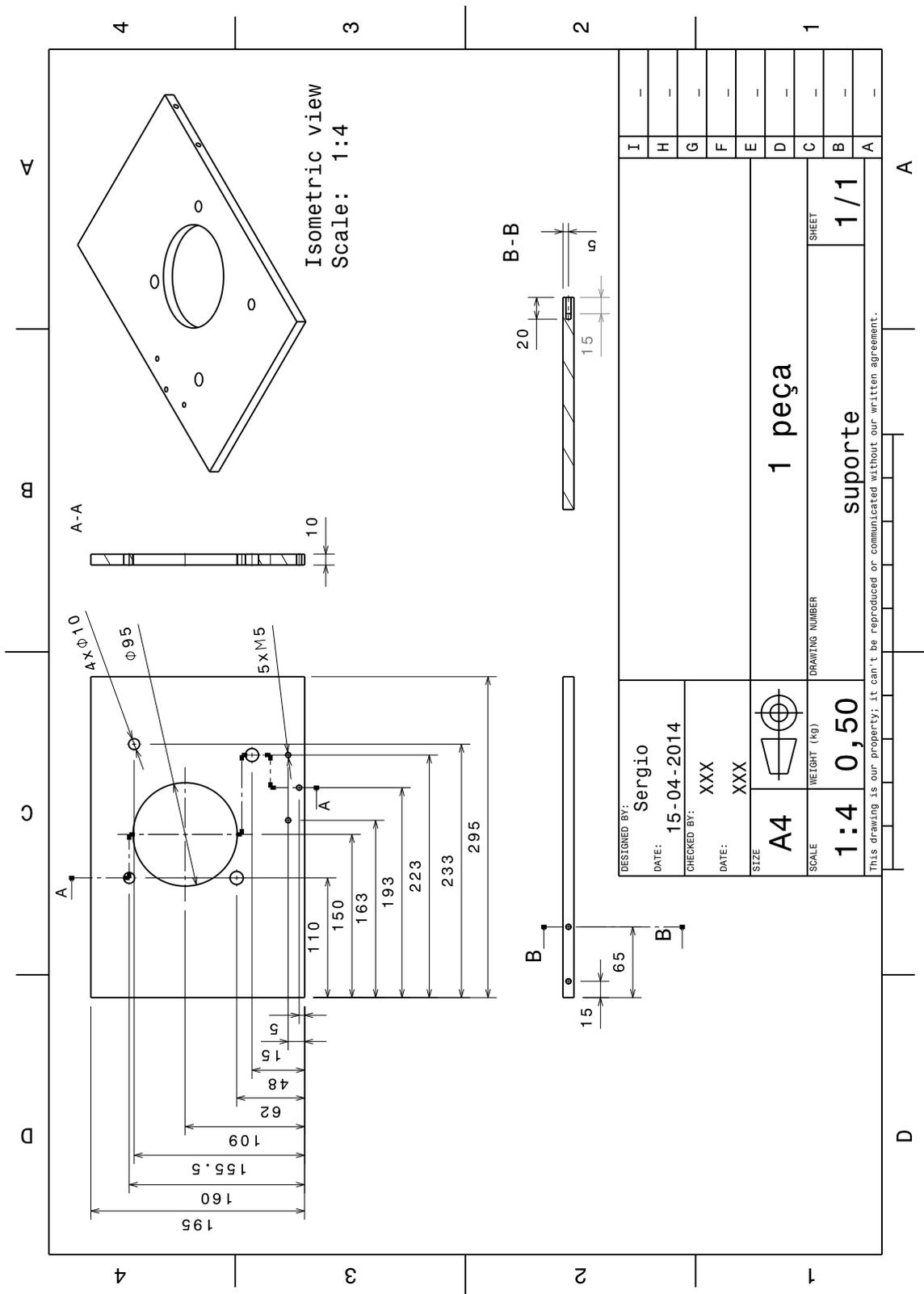


Figura 6.10: Desenho de definição da base de suporte

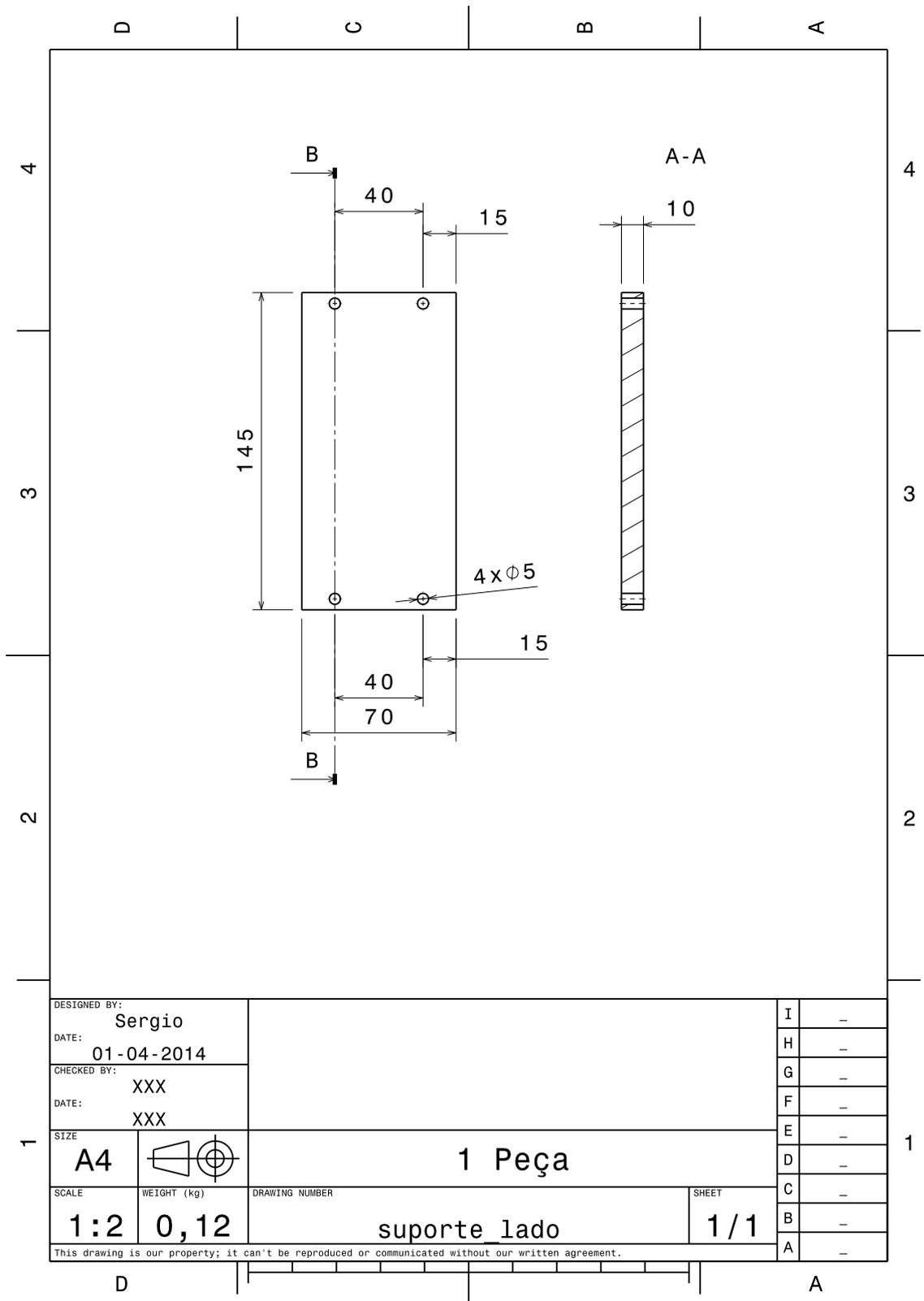


Figura 6.11: Desenho de definição da lateral do suporte

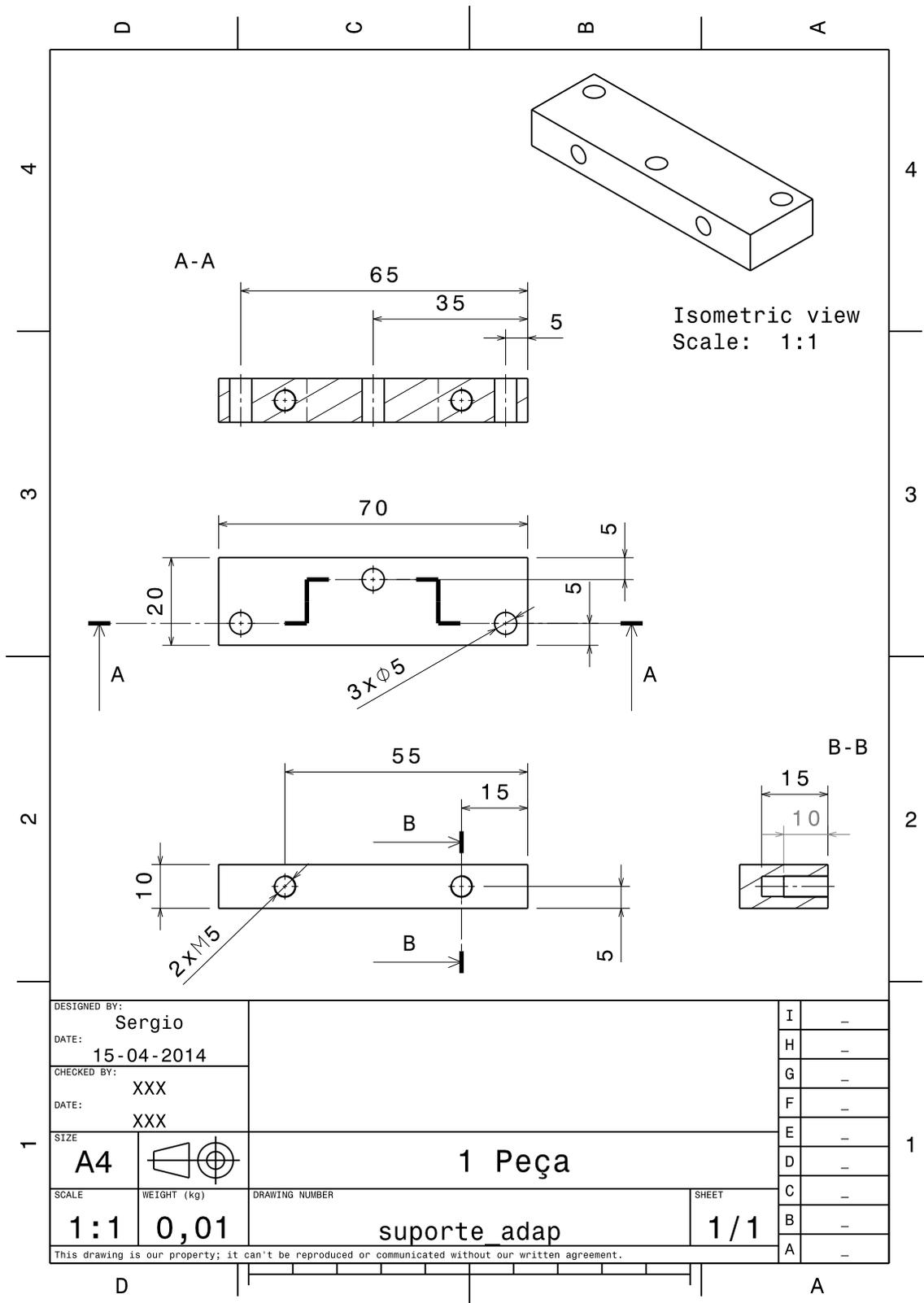


Figura 6.13: Desenho de definição do adaptador da lateral pequena do suporte

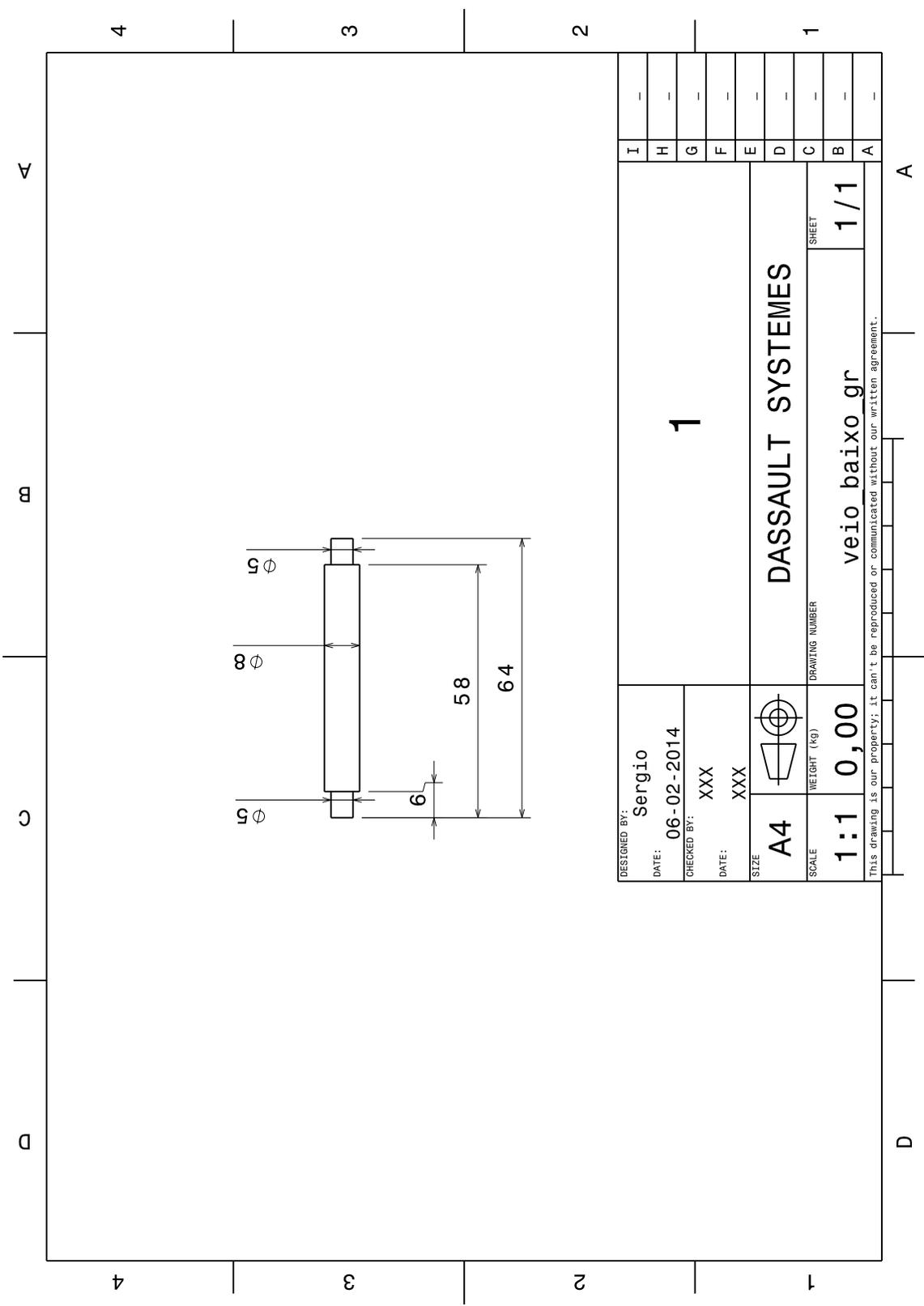


Figura 6.14: Desenho de definição do veio

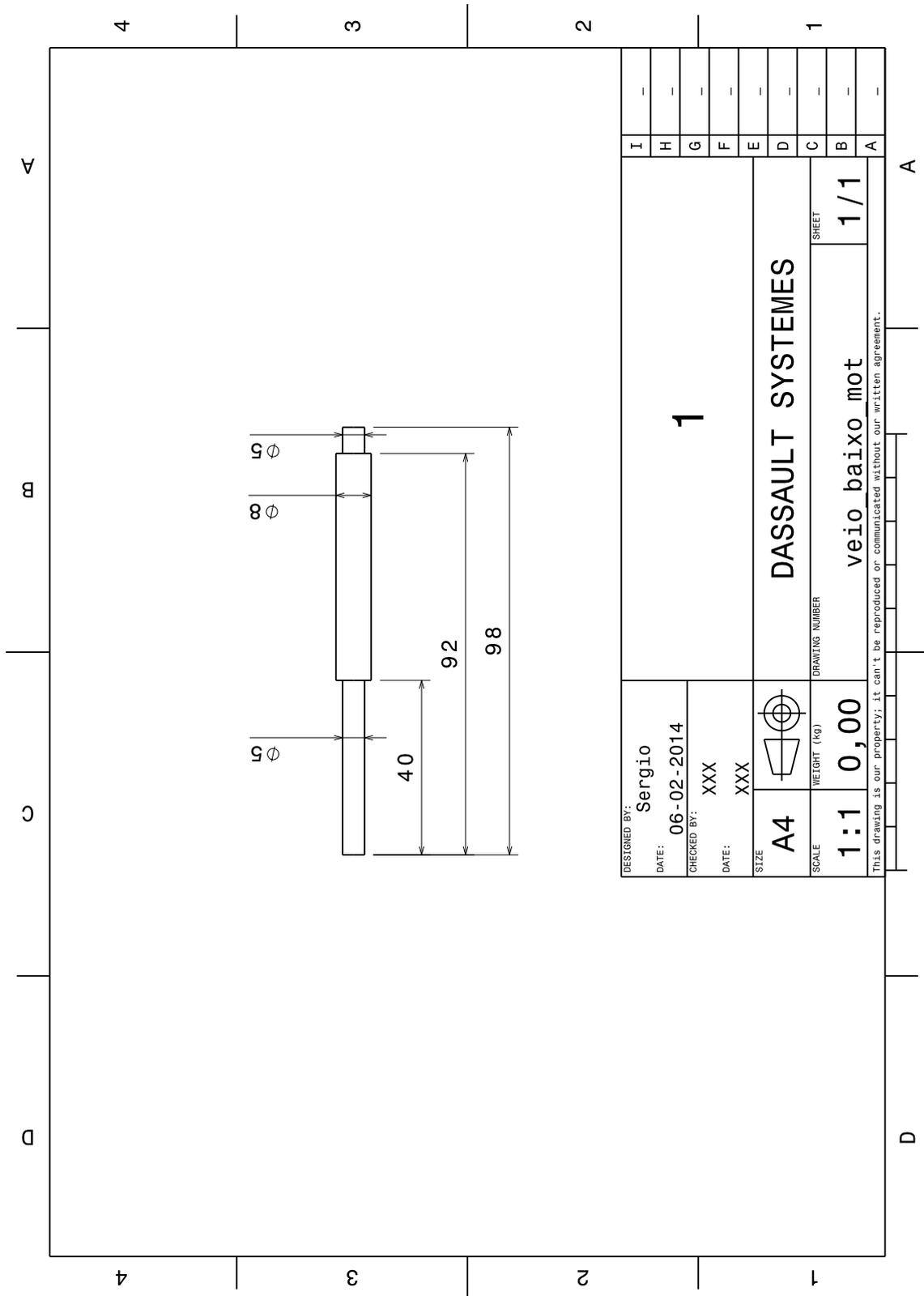


Figura 6.15: Desenho de definição do veio

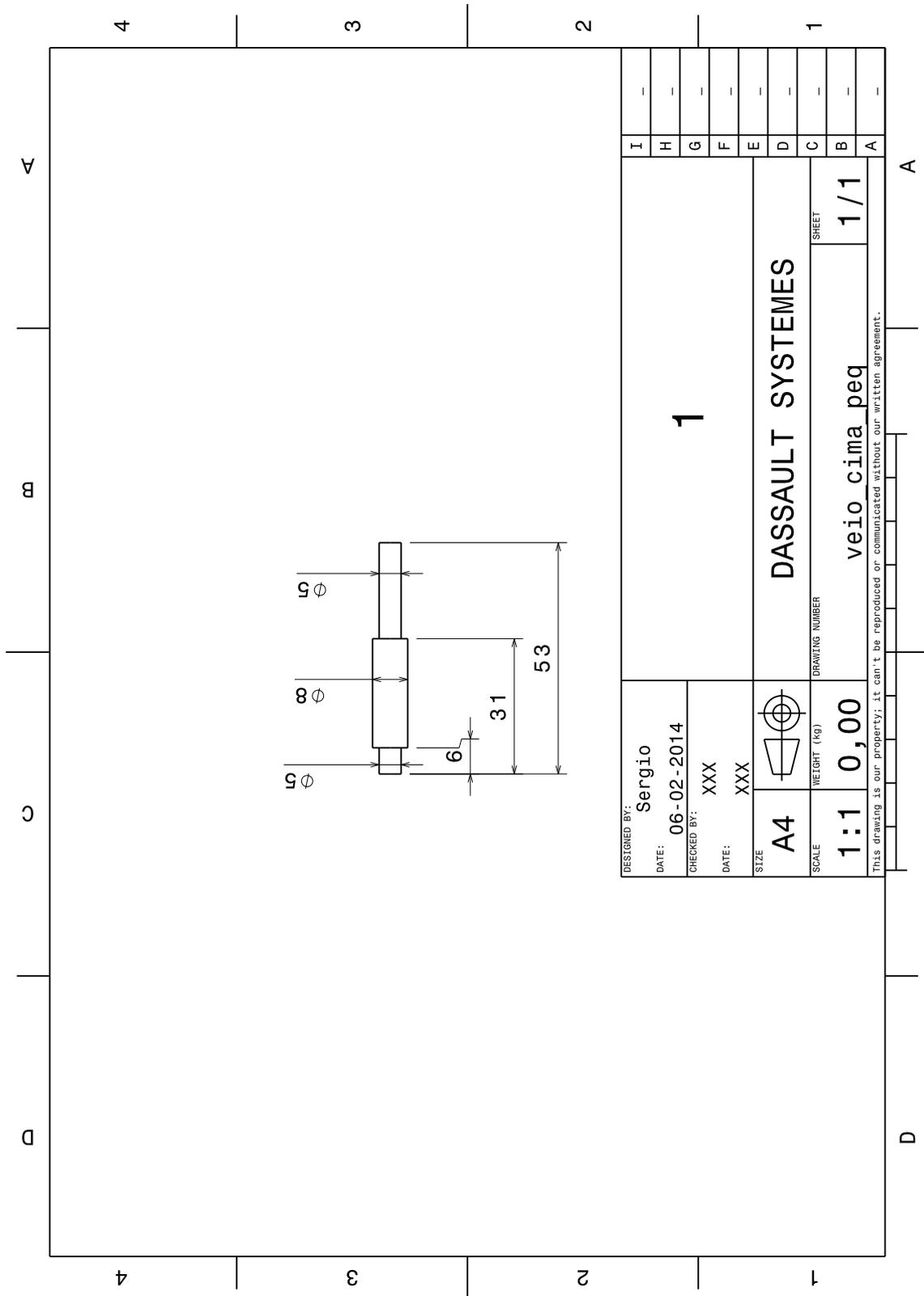


Figura 6.17: Desenho de definição do veio