

Mauro dos Santos Martins

Desenvolvimento de sistema de condução autónoma de um veículo eléctrico

Dissertação de Mestrado

5 De Setembro de 2011



UNIVERSIDADE DE COIMBRA



Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica e de Computadores
Dissertação de Mestrado em Engenharia Electrotécnica e de Computadores
- Especialização em Automação -

Desenvolvimento de sistema de condução autónoma de um veículo eléctrico

Mauro dos Santos Martins

Orientador:

Professor Doutor Urbano José Carreira Nunes

Co-orientador:

Professor Doutor Sérgio Manuel Ângelo da Cruz
Mestre Marco José da Silva

Júri:

Presidente: Professor Doutor Rui Pedro Duarte Cortesão

Vogais:

Professor Doutor Rui Paulo Pinto da Rocha
Professor Doutor Sérgio Manuel Ângelo da Cruz
Professor Doutor Urbano José Carreira Nunes

Coimbra, 2011

Dedicatória:

Quero dedicar esta dissertação aos meus pais António Martins e Maria Helena, à minha irmã Neuza, aos meus avós Maria Augusta e Mário Santos, e à minha namorada Eliana Almeida.

”In theory, there is no difference between theory and practice. But, in practice, there is.”

Jan L. A. van de Snepscheut/Yogi Berra.

Agradecimentos:

Quero prestar o meu agradecimento ao meu orientador Professor Doutor Urbano Nunes pela sua orientação ao longo desta dissertação.

Agradeço ao meu co-orientador Mestre Marco José da Silva pelo apoio, orientação, coordenação e amizade.

Gostaria de agradecer aos colegas Luís Garrote, Hugo Faria, Fernando Moita e Cristiano Premebida pela ajuda e acompanhamento durante o desenvolvimento da Dissertação.

Agradeço ao Departamento de Engenharia Electrotécnica e de Computadores e sobretudo ao Instituto de Sistemas e Robótica, que disponibilizaram as instalações e todo o equipamento necessário para a realização deste projecto.

Quero em especial agradecer à minha Mãe, Maria Helena e ao meu Pai, António Martins pela compreensão apoio e paciência que demonstraram para comigo ao longo dos anos.

Conteúdo

1	Introdução	1
1.1	Introdução	1
1.2	Enquadramento e objectivos do trabalho	2
1.3	Trabalho realizado	2
2	Estado da Arte	4
2.1	Estado da arte	4
2.2	Descrição da plataforma autónoma	5
2.2.1	Módulo de controlo de tracção	7
2.2.2	Módulo de gestão de energia	7
2.2.3	Rede CAN	8
2.2.4	Módulo de aquisição e sincronização	8
2.2.5	Módulo de processamento	8
3	Modelação matemática do sistema de controlo da direcção	9
3.1	Geometria da plataforma (Ackermann)	9
3.2	Controlador de seguimento de caminho Kanayama.	10
3.3	Motores DC	12
3.4	Controlador Proporcional Integral e Derivativo	12
3.5	Drives para motores DC	13
3.5.1	Drives Lineares	13
3.5.2	Drives Comutados	13
3.6	Sensores	16
3.6.1	Sensores Existentes na Plataforma	17
3.6.2	Sensores Existentes no Sistema de controlo da direcção	18
4	Sistema de Controlo da Direcção	20
4.1	Hardware	20
4.1.1	Submódulo de controlo	21
4.1.2	Submódulo de potência (Ponte-H)	23
4.2	Firmware	27
4.3	Protocolo de comunicação	28
4.3.1	Robot Operating System – > Microcontrolador	28
4.3.2	Microcontrolador – > Robot Operating System	29
4.4	Relação pulsos-ângulo	30
4.5	Projecto controlador PID	31
4.6	Utilitário de criação de perfis	32

5	Sistema de simulação	34
5.1	Simulador	34
5.1.1	Integração com o Google Earth	35
5.2	Modelo simulação para controlador lateral	36
5.2.1	Cálculo dos ganhos do controlador PID	37
5.2.2	Modelo de Simulação	38
6	Robotic Operating System (ROS)	40
6.1	Sobre o Robot Operating System	40
6.2	Nó de comunicação com plataforma commIO	42
6.3	Nó de comunicação com GPS diferencial TopCon	42
6.3.1	Conversão LLH para ECEF	43
6.3.2	Conversão de ECEF para ENU	43
6.4	Nó de Log do sistema loggerISR	43
7	Contribuições no âmbito do projecto ISRobotCar	44
7.1	Sincronização de aquisição sensorial	44
7.1.1	Sincronização CAN e GPS	45
7.1.2	Sincronização entre módulos CAN	45
8	Validação dos resultados	47
8.1	Sistema de Controlo da direcção	47
8.2	Navegação Autónoma	49
9	Conclusões e Trabalho Futuro	51
9.1	Trabalho Futuro	51

Lista de Figuras

1.1	Yamaha, Floriade	1
1.2	ParkShuttle3	1
1.3	Robosoft, La Rochelle	2
1.4	Move, Tocha	2
2.1	Plataforma de testes ISRobotCar	5
2.2	Estrutura geral dos sistemas presentes na plataforma	6
2.3	Estrutura detalhada dos sistemas presentes na plataforma	6
2.4	Níveis e módulos do sistema de controlo de movimento	7
3.1	Geometria de Ackermann	9
3.2	Estrutura de navegação com controlador Kanayama	11
3.3	Controlador Proporcional Integral e Derivativo	12
3.4	Amplificador em Push-Pull	13
3.5	Estrutura de uma Ponte-H	14
3.6	Fluxo de corrente em função dos pares activados	14
3.7	Amplificador comutado(Ponte-H)	15
3.8	Diferentes Duty-Cycle	16
3.9	Perdas nos comutadores	16
3.10	Sinal em quadratura	17
3.11	Crescimento das elipses de incerteza	18
3.12	Multipath em sistemas GPS	18
4.1	Estrutura do sistema de direcção	20
4.2	Diagrama do módulo de controlo da direcção	21
4.3	Estrutura interna do controlador LM629	21
4.4	Perfis de velocidade trapezoidais	22
4.5	Representação 3D do Submódulo de controlo	22
4.6	Diagrama do submódulo de potência	23
4.7	Esquema do mosfet escolhido com diodo de roda livre e protecção de Gate	24
4.8	Utilização típica do controlador de meia ponte	24
4.9	Representação 3D Ponte-H desenvolvida	25
4.10	Formas de onda sob carga leve	25
4.11	Formas de onda sob carga elevada	25
4.12	Firmware para sistema embebido	27
4.13	Relação potenciómetro linear/encoder óptico	28
4.14	Comunicação Computador/Microcontrolador	28
4.15	Comunicação Computador/Microcontrolador	29
4.16	Exemplo de comando em Hexadecimal	29
4.17	Exemplo de comando em Hexadecimal	29

4.18	Mensagem dados sensoriais	29
4.19	Relação entre pulsos de encoder e angulo de direcção	30
4.20	Malha de controlo equivalente	31
4.21	Utilitário para cálculo de perfis trapezoidais	32
4.22	Movimento de 32000 pulsos em 4 segundos	33
4.23	Movimento de 32000 pulsos em 2 segundos	33
5.1	GUI do simulador	34
5.2	Diagrama de controlo em Simulink	35
5.3	Trajecto definido no software Google Earth	35
5.4	Percurso importado do Google Earth	36
5.5	Simulação de navegação com controlador Kanayama	36
5.6	Trajectória real e simulação no Google Earth	36
5.7	Resposta do sistema e comando aplicado	37
5.8	Aproximação do sistema a um modelo de primeira ordem	37
5.9	Diagrama simulink para modelo do controlador lateral	38
5.10	Diagrama temporal dos percursos a efectuar	38
5.11	Sobreposição dos resultados reais e de simulação	39
6.1	Diagrama produzido pelo rxgraph	40
6.2	Exemplo de malha de controlo usando ROS	41
6.3	Integração do ROS com a plataforma	41
6.4	Vários sistemas de coordenadas (LLH, ECEF, ENU)	42
7.1	Estrutura de um frame CAN	44
7.2	Estructura de mensagem de trigger	44
7.3	Estructura de mensagem de trigger alterada	45
7.4	Diagrama temporal de aquisição	46
8.1	Posicionamento nos 32000 pulsos de encoder (22°)	47
8.2	Resposta de posicionamento durante navegação autónoma (ref.-azul/real-vermelho)	48
8.3	Trajectoria desejada, simulada e real	49
8.4	Sobreposição de resultados no Google Earth	49
8.5	SnapShots da execução do percurso	50

Nomenclatura:

ABS - Anti-lock Braking System
ADAS - Advanced driver assistance systems
ADC - Analog-to-digital converter
AGV - Automated guided vehicle
ASCII - American Standard Code for Information Interchange
BJT - Bipolar junction transistor
CAN - Controller area network
DAC - Digital-to-analog converter
DGPS - Differential Global Positioning System
ECEF - Earth-Centered, Earth-Fixed
ECU - Electronic Control Unit
ENU - East North Up
EPS - Electronic Power Steering
FET - Field Effect Transistor
GPS - Global Positioning System
GUI - Graphical User Interface
ID - Identificador
IGBT - Insulated Gate Bipolar Transistor
IMU - Inertial Measuring Unit
LLH - Latitude Longitude Height
PID - Proporcional Integral e Derivativo
PRT - Personal Rapid Transportation
PWM - Pulse With Modulation
ROS - Robotic Operating System
SOC - State of Charge

Resumo

Na actualidade, o desenvolvimento de frotas de veículos autónomos quer para transporte de passageiros quer para transporte de bens começa a tornar-se uma realidade. No enquadramento de tornar um veículo eléctrico capaz de navegar sozinho através de um percurso previamente definido, foi desenvolvido durante esta dissertação um sistema de controlo da direcção para um veículo autónomo.

Este sistema só por si não permite a navegação autónoma, sendo necessária a integração noutros sistemas tanto de hardware como de software. Neste contexto foram desenvolvidos módulos de potência e de controlo baseados em microcontroladores PIC.

Foram também desenvolvidos módulos de software utilizando o Framework ROS (Robot Operating System) que possibilitam a comunicação entre o software de controlo e o hardware.

Esta dissertação foi realizada no âmbito do projecto EVSIM09-Modelos para simulação de tráfego de veículos eléctricos com comunicação e capacidade de decisão dinâmica (Projecto PTDC/SEN-TRA/099413/2008, 2010-12), financiado pela Fundação para a Ciência e Tecnologia (FCT).

Palavras-chave: Controlo de direcção, Controlo Proporcional Integral e Derivativo (PID), Veículo Autónomo, Robot Operating System (ROS)

Abstract

Nowadays, the development of a fleet of autonomous vehicles both for passengers and goods transportation is becoming a reality. Within the framework of making an electric vehicle to navigate autonomously on a predetermined route, it was developed during this dissertation a lateral control system. This system by itself doesn't allow autonomous navigation, requiring the integration with other systems both hardware and software. In this context power and PIC microcontrollers based control modules were developed.

Were also developed software modules using ROS Framework (Robot Operating System) that enables communication between the control software and hardware.

This thesis was performed in the framework of the project EVSIM09 - Models for traffic simulation of electric vehicles with communications and dynamic decision capacity (Project PTDC/SEN-TRA/099413/2008, 2010-2012), funded by Fundação para a Ciência e Tecnologia (FCT).

Keywords: Steering Control, Proportional Integral and Derivative control, Autonomous Vehicle, Robot Operating System

Capítulo 1

Introdução

1.1 Introdução

Milhões de automóveis são utilizados diariamente como meio de transporte para pessoas e mercadorias. O nível de popularidade deste veículo provém da liberdade e flexibilidade que proporciona. O seu uso abrange um espectro amplo de utilizações como por exemplo, deslocação para o emprego, compras, lazer, entre outras. À medida que a população mundial aumenta, o fluxo de pessoas a transportar é maior, bem como o tráfego e a poluição produzidos. Um dos principais problemas associado ao fluxo crescente de deslocações, é o maior número de acidentes rodoviários. Todos os anos, só nas estradas Portuguesas morrem ou ficam com lesões permanentes centenas de pessoas [1]. O erro humano é apontado como a principal causa de acidente em cerca de 90% dos casos. Factores como o cansaço e o consumo de álcool ou drogas influenciam na percepção do ambiente, na avaliação incorrecta de uma situação, na decisão correcta a tomar, entre outras. Numa tentativa de tornar mais seguro o acto de conduzir, nos últimos anos tem-se assistido à instrumentação dos automóveis e à criação de sistemas de assistência ao condutor designados por A.D.A.S (Advanced Driver Assistance Systems). São exemplo destes sistemas o ABS (Anti-lock Braking System), o GPS (Global Positioning System) com aviso automático de excesso de velocidade, sistemas de detecção de saída de faixa, Cruise Control adaptativo, detecção de sinais de trânsito e estacionamento automático. Apesar da existência destes sistemas, o condutor continua a ter o factor de decisão [2]. A criação de uma rede de transportes autónoma (sem a intervenção de um condutor) e preferencialmente eléctrica poderá vir a ser a solução para a deslocação de um número cada vez maior de pessoas de forma mais eficaz, económica e “limpa”, sobretudo nas grandes cidades.



Figura 1.1: Yamaha, Floriade



Figura 1.2: ParkShuttle3



Figura 1.3: Robosoft, La Rochelle



Figura 1.4: Move, Tocha

Devido à complexidade do sistema rodoviário tradicional, é um desafio científico enorme a integração de veículos completamente autónomos. É necessário o estudo de novos algoritmos para navegação e fusão sensorial bem como torná-los robustos para conseguir esse objectivo.

Apesar dos veículos autónomos ainda não fazerem parte do sistema rodoviário convencional, encontram-se já alguns nichos onde, devido às suas características especiais têm operado com sucesso. Aeroportos como o de “Heathrow” possuem veículos eléctricos e autónomos conhecidos por PRT (Personal Rapid Transporter) destinados a levar os seus passageiros desde o terminal até ao parque de estacionamento. Cidades como Antibes, La Rochelle, Lausanne, entre outras, possuem a sua própria frota de veículos autónomos para servirem zonas turísticas. Em Portugal está em funcionamento uma frota de veículos autónomos capazes de transportar utentes entre os pavilhões do Hospital Rovisco Pais. O trabalho realizado durante esta dissertação pretende ser um contributo para o desenvolvimento de veículos autónomos.

1.2 Enquadramento e objectivos do trabalho

Nos últimos anos, o ISR-Coimbra tem vindo a desenvolver sistemas dedicados à navegação autónoma e vários investigadores têm contribuído para este objectivo. Encontram-se em estudo e desenvolvimento metodologias orientadas a este propósito com realce nos sistemas para localização, navegação, detecção de peões, estimação do estado de carga (SOC), entre outros [7] [8] [9].

O trabalho exposto nesta dissertação surge no seguimento da linha de investigação em desenvolvimento no ISR, e tem como objectivo principal o desenvolvimento de um sistema de controlo da direcção capaz de controlar em posição, aceleração e velocidade angular.

1.3 Trabalho realizado

O desenvolvimento de um sistema de controlo da direcção para um veículo autónomo exigiu as seguintes fases:

- Estudo do algoritmo de navegação “Kanayama”.
Este algoritmo pretende eliminar o erro de pose num veículo, calculando a velocidade linear e angular necessárias para o efeito. Em conjunto com outros módulos permite a navegação autónoma de um veículo.
- Projecto de hardware necessário ao módulo de controlo da direcção

- Submódulo de controlo.
Projecto de um controlador lógico (microcontrolador) responsável por receber comandos de posicionamento e executá-los.
- Submódulo de potência (Ponte-H).
Projecto de módulo de potência que integra o controlador de direcção.
- Firmware
Aplicação criada para execução no microcontrolador embebido de forma a cumprir os requisitos do sistema.
- Simulação
 - Simulador da plataforma ISRobotCar
Actualização de um simulador em ambiente Matlab para estudo dos algoritmos de navegação.
 - Modelo de simulação para o controlador de direcção.
Desenvolvimento de um modelo representativo do sistema de direcção para fins de simulação.
- Robotic Operating System "ROS"
Estudo do funcionamento deste framework e criação de diversos "módulos" de software.
 - Módulo de comunicação com hardware
Este módulo permite o interface entre o ROS e o Hardware.
 - Módulo de comunicação com GPS
Este módulo permite o interface entre o ROS e o GPS diferencial.
 - Módulo de LOG do sistema
Este módulo permite o registo de todos os dados que circulam no ROS.
- Contribuições no âmbito ISRobotCar
 - Sincronização de aquisição sensorial
De forma a sincronizar a aquisição de sensores para o processo de fusão sensorial foi necessário implementar uma nova estratégia para a rede CAN.

Capítulo 2

Estado da Arte

2.1 Estado da arte

Para muitos condutores, rodar o volante de um veículo que não possua direcção assistida exige mais binário do que aquele que o condutor considera confortável. Este esforço é especialmente notório em manobras a baixa velocidade. A força necessária para rodar a direcção depende sobretudo do peso e comprimento do veículo.

Foi em 1902 que Frederick W. patenteou a sua invenção que fazia uso de potência hidráulica para mover a direcção de um veículo terrestre. Francis W. Davis, um engenheiro a trabalhar na divisão de camiões do fabricante Pierce Arrow começou a explorar formas de tornar a direcção mais fácil e em 1926 demonstrou o primeiro sistema de direcção assistida[3]. Com este mecanismo o condutor do veículo apenas tem de exercer uma fracção do binário necessário para mover a direcção, visto que o restante é fornecido pelo mecanismo de direcção assistida. Mais tarde, a trabalhar para a General Motors, Davis melhorou o seu sistema de direcção hidráulico verificando no entanto que seria demasiado dispendioso para produção [4]. Foi apenas em 1951 que surgiu o primeiro veículo de passageiros (Chrysler Imperial) com sistema de direcção assistida conhecido como “Hydraguide” [5]. O sistema usava algumas das patentes feitas por Davis. Em 1952 a General Motors equipou o seu Cadillac com um sistema de direcção assistida baseado no trabalho que Davis desenvolveu para a empresa quase 20 anos antes [6].

Até ao presente a maioria dos sistemas de direcção assistida baseiam-se num circuito hidráulico para auxiliar o condutor, no entanto este sistema apresenta várias desvantagens. A pressão hidráulica necessária provém de uma bomba que por sua vez é accionada pelo motor do veículo através de um sistema de correias. Este facto faz com que o sistema de direcção consuma energia mesmo quando nenhum esforço seja requerido na direcção. Em alguns casos, a potência consumida por este sistema situa-se entre os 3 e os 5 CV traduzindo-se num maior consumo de combustível e aumentando a produção de CO_2 . Para contornar o facto da bomba hidráulica estar constantemente em funcionamento foram também desenvolvidos sistemas Electro-hidráulicos em que a bomba é accionada por um motor eléctrico diminuindo assim o consumo. Apesar da melhoria, estes sistemas ainda dependem de um circuito hidráulico com as suas perdas energéticas e possíveis fugas/rupturas que levantam questões ambientais devido ao fluido que circula no sistema.

Em 1993 a Honda introduziu no mercado o primeiro sistema de direcção assistida totalmente eléctrico (Electronic Power Steer ou EPS) no modelo Acura NSX. Um motor DC instalado concentricamente em torno do pinhão, providencia a força necessária, e um sensor de binário montado na coluna de direcção monitoriza as intenções do condutor. Esta solução torna-se preferível em relação às anteriores devido ao baixo consumo, peso e

dimensões reduzidas, inexistência de fluidos prejudiciais e uma maior capacidade de controlo. A ECU (Electronic Control Unit) responsável pela EPS pode dosear a quantidade de força a aplicar ao motor em função da velocidade do veículo ou preferências do utilizador.

O bom funcionamento do sistema de direcção é fulcral para o correcto e seguro uso de um veículo. Uma falha neste sistema pode ter consequências bastante graves, por este motivo todas as aplicações até agora descritas mantêm uma ligação mecânica ao sistema de direcção (coluna de direcção). Existem no entanto estudos para a remoção deste elemento mecânico tornando o sistema completamente eléctrico. Esta solução enquadra-se nos sistemas “X-by-Wire” sendo no caso particular conhecida como “Steer-by-wire”. O desacoplamento mecânico do sistema permite uma redução de peso, diminuição do esforço de condução, e eliminação de vibração do volante quando se passa num pavimento em mau estado. Esta tecnologia ainda não se encontra massificada por levantar problemas de segurança e estima-se que demore ainda até que o seja.

Outro campo de investigação são os ADAS. Um destes sistemas que começa a aparecer comercialmente é o estacionamento (paralelo) autónomo. Marcas como a Lexus, a Mercedes e a Ford já possuem modelos com esta funcionalidade. Durante o estacionamento autónomo a direcção é completamente controlada pelo computador de bordo. Para este fim é necessário desenvolver toda uma infra-estrutura de controlo e accionamento do sistema de direcção.

Funcionalidades como esta levam-nos em direcção a sistema ADAS cada vez mais robustos e aproximamos-nos de veículos capazes de navegar autonomamente sem a necessidade de intervenção humana.

2.2 Descrição da plataforma autónoma

A plataforma utilizada no decorrer desta dissertação foi o ISRobotCar, baseada no AGV (Autonomous Guided Vehicle) da Yamaha. Originalmente a plataforma foi concebida para aplicações de AGV filo-guiado. Esta tecnologia baseia-se na detecção do campo electromagnético de um condutor embebido no solo que descreve o percurso a percorrer. Este tipo de AGV é bastante utilizado em aplicações onde o trajecto a percorrer seja constante, por exemplo em ambiente fabril, turismo, aeroportos, entre outros.

Devido à falta de documentação sobre o sistema e à impossibilidade de alterar o seu firmware, surgiu a necessidade de reinstrumentar a plataforma. Apenas os elementos estruturais como a carroçaria, o motor de tracção e o de direcção se mantiveram.

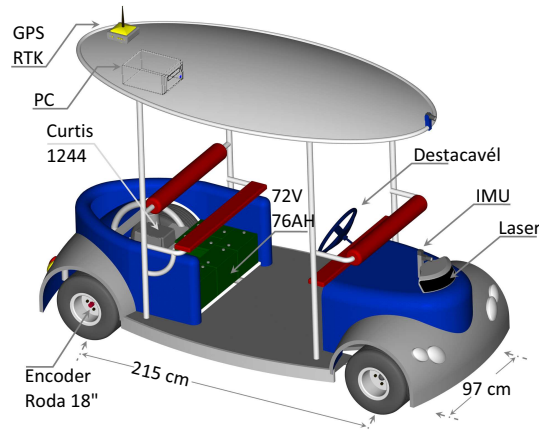


Figura 2.1: Plataforma de testes ISRobotCar

A plataforma está organizada em diferentes módulos mediante a função que desempenham.

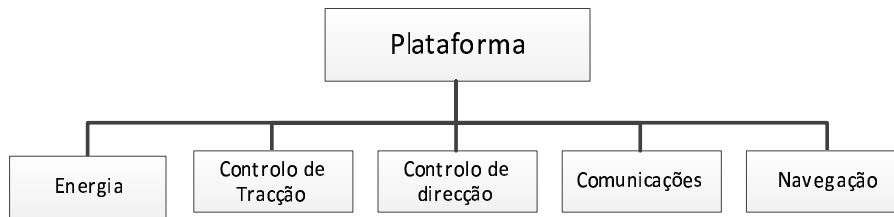


Figura 2.2: Estrutura geral dos sistemas presentes na plataforma

A figura anterior pretende proporcionar uma visão da organização lógica da plataforma, enquanto que a seguinte pretende fornecer uma representação mais detalhada do funcionamento da mesma.

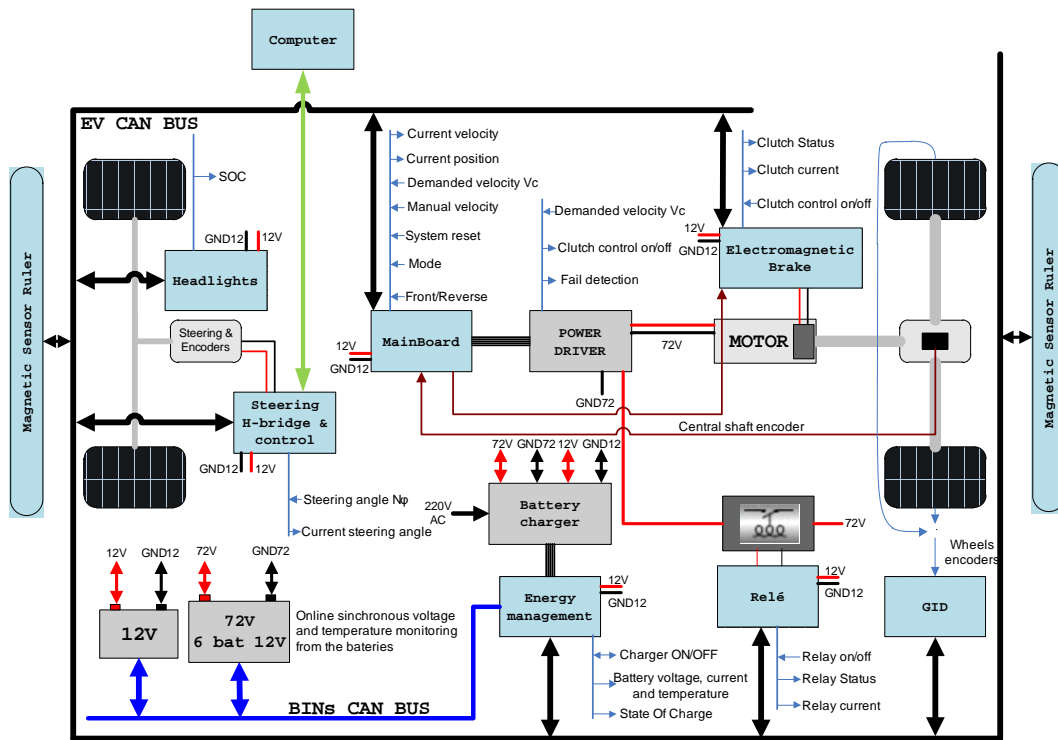


Figura 2.3: Estrutura detalhada dos sistemas presentes na plataforma

A figura seguinte mostra a integração dos sistemas de controlo necessários à navegação autónoma.

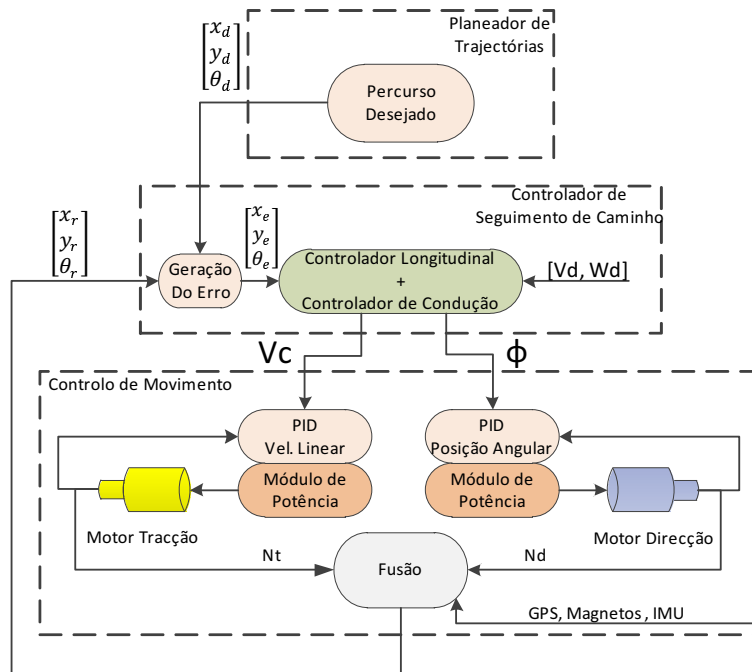


Figura 2.4: Níveis e módulos do sistema de controlo de movimento

O módulo de controlo da direcção foi o principal objecto de trabalho durante esta dissertação e será descrito nos próximos capítulos.

2.2.1 Módulo de controlo de tracção

O controlo longitudinal da plataforma é feito através de um Power Driver do fabricante “CURTIS” que por sua vez controla um motor DC de excitação separada de 72V e 3.5kW. Uma característica importante deste Power Driver é a capacidade de fazer travagem regenerativa, aproveitando alguma da energia que de outra forma seria perdida.

Em conjunto com o motor de tracção e o Power Driver existe o módulo “Mainboard”. A mainboard monitoriza um interruptor rotativo (chave) seleccionando o modo de funcionamento da plataforma. Caso a chave esteja na posição “manual”, o módulo reencaminha o sinal do acelerador (potenciómetro 0 a 5V) para a entrada de comando do Power Driver. Caso a chave esteja em “auto”, o módulo recebe comandos de velocidade bem como feedback das rodas através da rede CAN. Neste modo, o módulo implementa um controlador PID de velocidade e através de um DAC (Digital to Analog Converter) aplica ao Power Driver o comando necessário para manter a velocidade linear no valor desejado.

2.2.2 Módulo de gestão de energia

O módulo de energia é dividido em duas secções: Alimentação e gestão. O pack de baterias é constituído por sete baterias de gel de chumbo. Seis destas encontram-se ligadas em série perfazendo 72V com uma capacidade de 78AH, e são responsáveis por fornecer alimentação ao módulo de controlo longitudinal. A sétima bateria serve de alimentação a todos os módulos de controlo bem como ao motor da direcção.

Para a estimação do estado da carga (State of Charge) existe um módulo BMS (Battery Managing System) constituído por um sistema principal responsável por medir a corrente total fornecida pelas baterias bem como agregar vários dispositivos de monitorização (tensão e temperatura) BIN (Battery Intelligent Node) montados em cada bateria. Esta informação é disponibilizada na rede CAN para que seja utilizada pelo módulo “sinalética” onde é possível aos utilizadores consultar os parâmetros do sistema.

2.2.3 Rede CAN

De forma a trocar informações/comandos entre os diferentes ECU (Electronic Control Unit), a plataforma conta com uma rede CAN (Controller Area Network). Existe um nó responsável pela gestão da rede, definindo intervalos temporais (time-slots) em que cada um dos restantes nós pode comunicar.

O funcionamento da rede CAN do veículo será descrita com mais detalhe no capítulo sete.

2.2.4 Módulo de aquisição e sincronização

Montados na plataforma encontram-se diversos sensores como encoders, GPS diferencial, régua para detecção de magnetos e IMU (Inertial Measuring Unit) contendo bússola digital, acelerómetros e giroscópios. De forma a centralizar o processo de aquisição sensorial, existe o módulo de aquisição e sincronização. Este módulo é responsável por adquirir as informações dos sensores e enviá-las através da rede CAN. Os dados provenientes destes sensores servem de entradas ao processo de fusão sensorial para a localização do veículo.

Durante esta dissertação este módulo passou também a ser responsável pela gestão e sincronização da rede CAN como será detalhado no capítulo sete.

2.2.5 Módulo de processamento

De forma a executar os algoritmos de navegação e fusão sensorial que possibilitam a navegação autónoma da plataforma, existe o módulo de processamento. Este módulo é constituído por um computador com o sistema operativo Ubuntu e o framework vocacionado para a robótica “ROS” (Robot Operating System). Algumas considerações sobre este sistema serão feitas no capítulo seis.

Capítulo 3

Modelação matemática do sistema de controlo da direcção

As secções seguintes servem de introdução teórica necessária para o controlo e navegação de um veículo autónomo, assim como introdução às considerações necessárias para o projecto de um sistema de controlo da direcção.

3.1 Geometria da plataforma (Ackermann)

A plataforma ISRobotCar apresenta uma geometria equivalente à de um carro tradicional. Este tipo de geometria é conhecida como geometria de Ackermann.

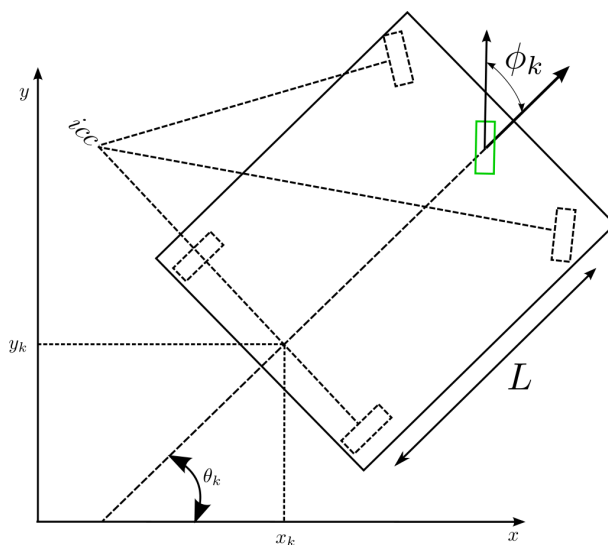


Figura 3.1: Geometria de Ackermann

As duas rodas dianteiras têm ângulos diferentes (canvas), no entanto podem ser modeladas como sendo apenas uma central. Para esta geometria são aplicáveis as seguintes relações:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ \tan \frac{\phi}{L} \\ 0 \end{bmatrix} \cdot v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \cdot v_2 \quad (3.1)$$

Onde:

$[\dot{x}, \dot{y}, \dot{\theta}]$ - Deslocação do veículo em cada eixo

L - Distância entre eixos (traseiro e frontal)

ϕ - Ângulo da direcção

v_1 - Velocidade linear do veículo

v_2 - Velocidade do steering

A variável que controla a direcção do veículo é o ϕ e representa o ângulo da direcção. Esta grandeza pode ser calculada em função da velocidade linear e angular impostas ao veículo através de:

Velocidade angular do veículo:

$$\omega = \dot{\theta} = \frac{\tan \phi}{L} v_1 \quad (3.2)$$

Ângulo de direcção:

$$\phi = \tan^{-1} \frac{\omega L}{v_1} \quad (3.3)$$

A velocidade a impor ao sistema de tracção é igual à velocidade linear do veículo.

3.2 Controlador de seguimento de caminho Kanayama.

Para que o veículo consiga seguir a trajectória desejada é necessário um algoritmo que o consiga fazer convergir da pose actual para a pose desejada. São vários os controladores para este efeito como o Kanayama[10], Sliding Mode[12], Fuzzy Logic[11], entre outros. Neste trabalho apenas será abordado o controlador Kanayama, este controlador actua sobre o erro de pose em relação à pose desejada. O erro é convertido do referencial mundo para o referencial do veículo através da transformação T_e . A cada iteração é calculada uma nova velocidade linear e angular, necessária para que o veículo convirja para a pose desejada. Através da cinemática inversa, é possível calcular qual a velocidade e ângulo a impor aos actuadores.

As equações que governam este controlador são as seguintes:

Erro de pose:

$$E_p = T_e \cdot (P_d - P_r) \quad (3.4)$$

CAPÍTULO 3. MODELAÇÃO MATEMÁTICA DO SISTEMA DE CONTROLO DA DIRECÇÃO

Em termos matriciais:

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos \theta_r & \sin \theta_r & 0 \\ -\sin \theta_r & \cos \theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_d - x_r \\ y_d - y_r \\ \theta_d - \theta_r \end{bmatrix} \quad (3.5)$$

Por fim as equações que fornecem as velocidades lineares e angulares são:

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} k_x \cdot x_e + v_r \cdot \cos \theta_e \\ \omega_r + v_r \cdot (k_y \cdot y_e + k_\theta \cdot \cos \theta_e) \end{bmatrix} \quad (3.6)$$

Onde:

E_p - Erro de Pose

P_d, P_r - Pose desejada e actual

T_e - Transformação para o referencial do veículo

$[x_e, y_e, \theta_e]$ - Erro longitudinal, lateral e de orientação

$[v_r, \omega_r]$ - Velocidade linear e angular de referência

$[v_c, \omega_c]$ - Velocidade linear e angular a impor

k_x, k_y, k_θ - Ganhos do controlador

A função do algoritmo apresentado é fazer convergir a pose actual do veículo para a pose desejada. Para que seja possível fazer o seguimento de trajectórias são necessários módulos adicionais. O bloco indicado como “Planeador de trajectórias” é responsável por decidir qual a próxima pose a atingir até completar o percurso desejado.

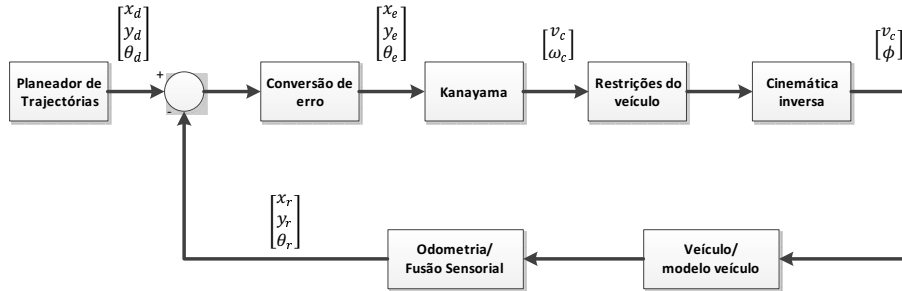


Figura 3.2: Estrutura de navegação com controlador Kanayama

O processo de navegação começa com a diferença entre a pose actual e a desejada. Esse valor é depois transformado para o referencial do robô através de T_e e fornecida ao navegador. A tarefa do navegador é gerar as velocidades angulares e lineares para o veículo, em função do erro em posição e dos ganhos do controlador. Através da cinemática inversa são calculados os comandos adequados para os actuadores, que vão fazer o veículo deslocar-se para a pose desejada. Na iteração seguinte o processo repete-se até ter terminado o percurso.

O cálculo das constantes do controlador “ k_x, k_y, k_θ ” segue o procedimento descrito em [10].

3.3 Motores DC

O motor DC de ímanes permanentes é um tipo de máquina eléctrica rotativa, onde se converte energia eléctrica em energia mecânica. Neste tipo de máquinas, existem ímanes permanentes para criar o campo magnético do estator. Ao aplicar uma tensão aos terminais do rotor será gerada uma corrente e consequentemente um campo magnético. A interacção dos dois campos magnéticos produz binário pelo princípio da lei de “Lorentz” [13].

A potência mecânica do motor é proporcional à corrente que o atravessa como se pode verificar na seguinte equação:

$$P = I \cdot V_{cemf} \quad (3.7)$$

Por sua vez, a corrente é proporcional à tensão aplicada:

$$I = \frac{V_{aplicada} - V_{cemf}}{R_{armadura}} \quad (3.8)$$

Onde:

P - Potência mecânica

I - Corrente no motor

V_{cemf} - Tensão contra electromotriz

$V_{aplicada}$ - Tensão aplicada

$R_{armadura}$ - Resistência de armadura

3.4 Controlador Proporcional Integral e Derivativo

O controlador Proporcional, Integral e Derivativo (PID) é um controlador de uso genérico muito utilizado em aplicações industriais.

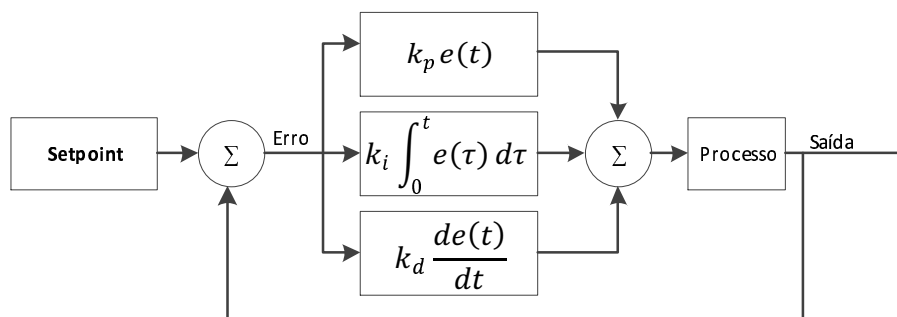


Figura 3.3: Controlador Proporcional Integral e Derivativo

Este tipo de controlador tem inúmeras aplicações, desde controlo de temperatura, velocidade, posição, entre outras. Foi estudada a aplicação deste filtro num sistema de controlo da direcção, tendo-se chegado à conclusão que seria uma solução prática e eficaz.

Como é possível observar pela figura acima, o PID actua sobre o erro entre a saída do processo e o setpoint definido. Em função deste valor e dos ganhos k_p , k_i e k_d o controlador produz o comando necessário para manter ou levar o processo até ao setpoint

definido. Escolhendo cuidadosamente os valores dos ganhos pode-se (dentro das limitações do sistema) obter o tipo de resposta desejado.

O facto de o sistema ser implementado num microcontrolador, implica a discretização do PID. No caso do controlador de movimento LM629 (ver capítulo 4) a função implementada é a seguinte:

$$u(n) = k_p \cdot e(n) + k_i \cdot \sum_{n=0}^N e(n) + k_d \cdot [e(n') - e(n' - 1)] \quad (3.9)$$

Onde:

$u(n)$ - sinal de comando no instante n

$e(n)$ - erro no instante n

n' - instante de amostragem derivativo

k_p, k_i, k_d - ganhos do controlador

A componente derivativa pode ser calculada apenas em instantes desejados múltiplos do período de amostragem ($n' = \alpha \cdot n$)

3.5 Drives para motores DC

De forma a aplicar tensão e polaridade adequada ao motor, é necessário amplificar os sinais de controlo. Nas secções seguintes serão abordados dois tipos de amplificador possíveis para esta aplicação.

3.5.1 Drives Lineares

Este tipo de amplificador com uma arquitectura em “push-pull” permite a amplificação da tensão de entrada bem como uma maior corrente de saída.

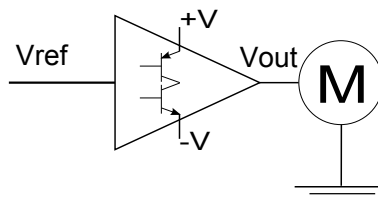


Figura 3.4: Amplificador em Push-Pull

No entanto, como os transístores de potência estão a funcionar na região linear, têm de dissipar uma potência elevada não sendo por isso adequados para aplicações que exijam correntes altas na carga.

3.5.2 Drives Comutados

Esta configuração faz uso de quatro transístores numa configuração em “H” (motor no ramo do meio). Apenas um par de transístores conduz em cada instante.

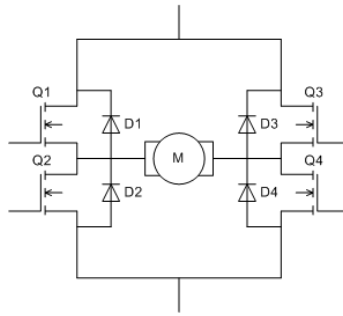


Figura 3.5: Estrutura de uma Ponte-H

Accionando o transístor Q1 e Q4 fará com que o motor rode num sentido, rodando para o outro caso seja activado Q2 e Q3. Ao ligar o transístor Q2 e Q4 ou Q1 e Q3 faz com que o motor esteja electricamente travado. Combinações como Q1 e Q2 ou Q3 e Q4 são proibidas pois criam um curto-circuito no ramo esquerdo ou direito que pode danificar os transístores, esta condição é conhecida como “shoot-through”.

As figuras seguintes pretendem demonstrar o fluxo da corrente quando se altera a sequência de transístores em condução:

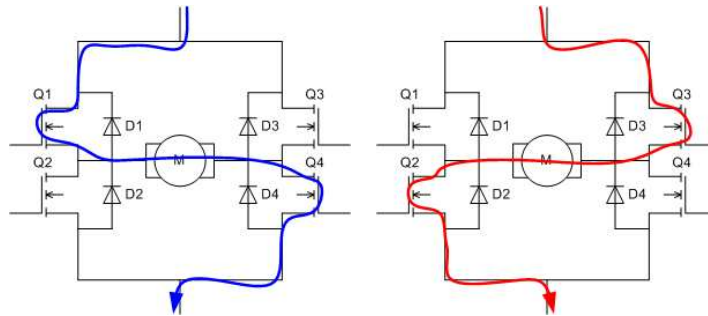


Figura 3.6: Fluxo de corrente em função dos pares activados

Devido às propriedades físicas das cargas indutivas (como um motor), quando a corrente está a fluir numa direcção a indutância tende a mantê-la mesmo que as condições se alterem. Este facto verifica-se quando a ponte é desactivada ou é trocada a polaridade a aplicar (por exemplo, quando o comando de direcção se altera). Enquanto o campo electromagnético em redor dos enrolamentos do motor não se extinguir, uma corrente fluirá na tentativa de o manter. É necessário fornecer um caminho seguro para esta corrente fluir enquanto vai decaindo até zero ou muda para a nova direcção. Não providenciar este caminho poderá resultar em danos nos interruptores da ponte, devido à tensão elevada gerada aos terminais da indutância pela derivada da corrente ($v_L = -L \frac{di}{dt}$). Uma forma comum de fornecer este caminho é a utilização de diodos de roda livre (freewheeling diodes) em paralelo com os mosfets. Estes começam a conduzir após o mosfet ter sido desactivado até que o campo electromagnético colapse e a corrente se anule. Outra técnica possível de utilizar para a dissipação da Fcem (Força Contra Electromotriz) é a activação de ambos os mosfets inferiores ou superiores. Esta situação é desejada uma vez que além de permitir a dissipação da corrente contra electromotriz faz com que o motor pare rapidamente porque é criado um curto-circuito aos terminais do motor travando-o electricamente.

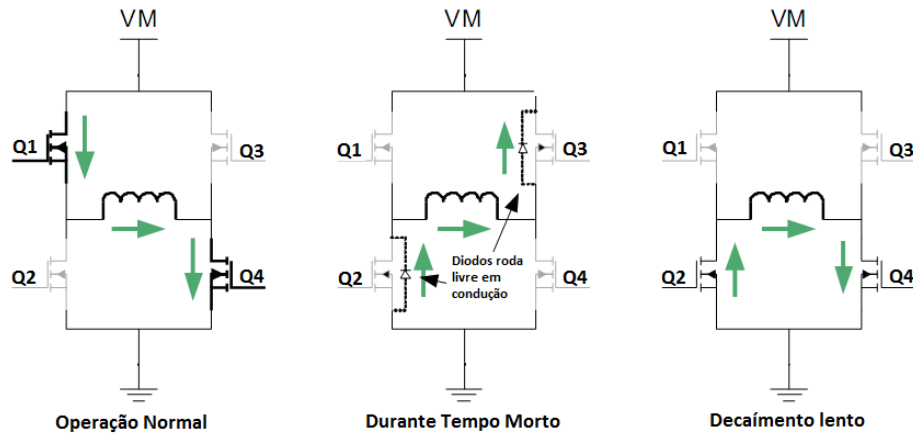


Figura 3.7: Amplificador comutado(Ponte-H)

Apesar da activação de ambos os mosfets criar um percurso para que a corrente flua, os diodos de roda livre têm ainda uma função importante nos instantes entre a desactivação de mosfets e activação dos complementares por exemplo quando é invertido o sentido de rotação.

Pulse Width Modulation

Como já foi descrito, a ponte-H permite controlar a polaridade da tensão aplicada ao motor, no entanto é também necessário controlar a tensão média. Para esse efeito usa-se PWM (Pulse With Modulation). Esta técnica consiste em variar o tempo que uma onda quadrada está “ligada” (Duty-Cycle) mantendo constante o período da mesma. A saída será uma tensão média proporcional ao Duty-Cycle:

$$V_{out} = V_{in} \cdot DutyCycle \quad (3.10)$$

Em que o Duty-Cycle é percentagem de tempo a on:

$$DutyCycle = \frac{t_{on}}{(t_{on} + t_{off})} \quad (3.11)$$

Onde:

V_{in} - Tensão de entrada

V_{out} - Tensão de saída

A figura 3.8 pretende mostrar sinais de PWM com diferentes Duty-Cycles:

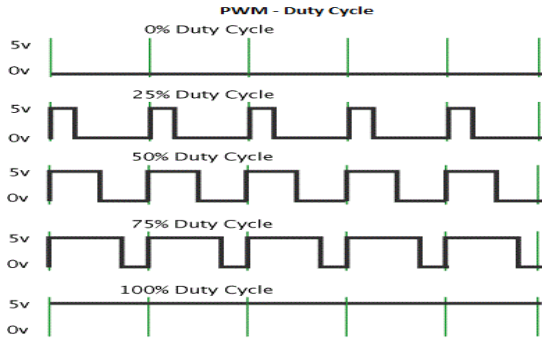


Figura 3.8: Diferentes Duty-Cycle

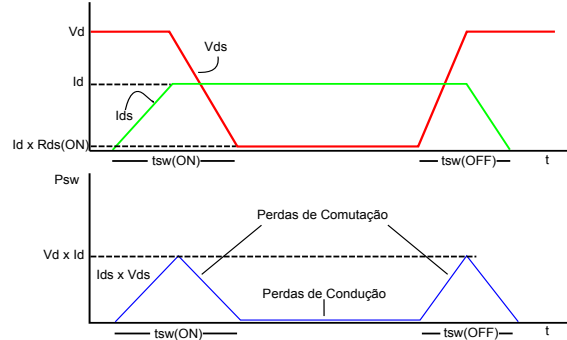


Figura 3.9: Perdas nos comutadores

Perdas nos comutadores

Como considerações finais serão abordadas as perdas nos mosfets de potência. Os mosfets presentes na ponte funcionam na região de saturação. Nesta região a resistência “Drain-Source” dos mosfets é na casa dos mili-Ohms, facto que minimiza as perdas por efeito de Joule.

$$P_{cond} = R_{dson} \cdot I_{ds}^2 \cdot D_{cycle} \quad (3.12)$$

No entanto, os mosfets utilizados não são ideais e não conseguem entrar na saturação instantaneamente. Assim sendo, as perdas mais significativas ocorrem quando o mosfet entra e sai de condução. Durante este período, a potencia dissipada é grande quando comparada com a de saturação.

$$P_{sw} = \frac{1}{2} \cdot V_{in} \cdot I_{ds} \cdot (t_{swON} + t_{swOFF}) \cdot F_{sw} \quad (3.13)$$

A potência total dissipada pelo mosfet pode ser calculada pela soma das duas anteriores:

$$P_{total} = P_{cond} + P_{sw} \quad (3.14)$$

Onde:

- P_{cond} - Potência dissipada na zona de saturação
- R_{dson} - Resistência de canal na zona de saturação
- I_{ds} - Corrente que atravessa o mosfet
- D_{cycle} - Tempo em que a onda se encontra a ON
- P_{sw} - Potência dissipada durante a comutação
- V_{in} - Tensão de entrada
- $T_{swON} e T_{swOFF}$ - Tempo de comutação
- F_{sw} - Frequência de comutação

3.6 Sensores

Esta secção está dividida em duas partes e pretende apresentar os sensores existentes na plataforma ISRobotCar. A primeira abordará de forma genérica quais os sensores da

plataforma enquanto que a segunda abordará os sensores usados no sistema de controlo da direcção.

3.6.1 Sensores Existentes na Plataforma

A plataforma ISRobotCar conta com inúmeros sensores para permitir a navegação autónoma:

- **Encoders**

Um encoder é um dispositivo electromecânico que produz pulsos eléctricos a partir do movimento rotacional do seu eixo, é por isso um transdutor de posição/velocidade angular.

O sinal produzido pelo encoder é constituído por duas ondas quadradas desfasadas de 90° (canal A e B). Contando o número de transições consegue-se saber a posição relativa do eixo, e verificando qual das ondas está em avanço em relação à outra determina-se o sentido de rotação.

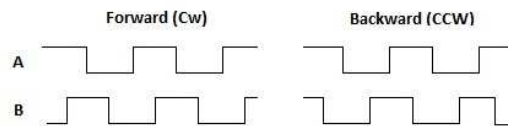


Figura 3.10: Sinal em quadratura

Existem quatro encoders montados na plataforma, dois nas rodas traseiras, um no eixo de tracção e um no actuador da direcção. Excepto o encoder do eixo de tracção que possui 312ppr (pulsos por revolução), os restantes possuem uma resolução de 1000ppr sendo aumentada para 4000 se forem usadas todas as transições do sinal em quadratura.

- **Inertial Measurement Unit (IMU)**

Um sistema inercial é regra geral constituído por acelerómetros e giroscópios. Através destes é possível saber a aceleração instantânea segundo cada um dos eixos bem como a velocidade angular. Através de técnicas de filtragem e de integração estes sensores ajudam no processo de fusão sensorial.

Montada na plataforma existe uma IMU de 9 graus, significando que possui acelerómetros e giroscópios nos três eixos (XYZ) bem como magnetómetro que permite a medição do campo magnético da terra e desta forma obter uma referência de orientação absoluta.

- **Régua de detectores magnéticos**

Como é de conhecimento geral, apenas a utilização de odometria não é viável para navegação autónoma. Sem forma de a corrigir, o erro de pose crescerá indeterminadamente ao longo do tempo. A incerteza da posição real da plataforma cresce como é possível verificar pelas elipses de incerteza:

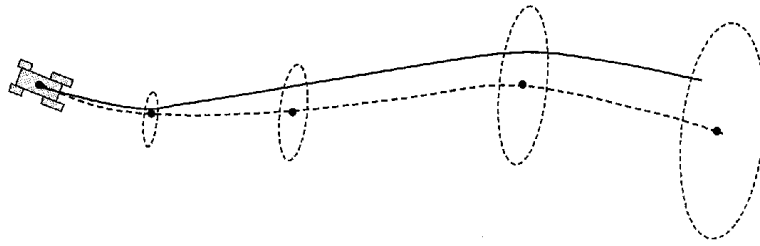


Figura 3.11: Crescimento das elipses de incerteza

Para corrigir a sua posição, a plataforma possui uma régua constituída por sensores de HALL, capazes de detectar magnetos embebidos no solo e cuja posição absoluta é conhecida com elevada precisão. Estas medidas fornecem ao processo de fusão sensorial uma contribuição com variância bastante baixa.

• GPS Diferencial

O Sistema de Posicionamento Global Diferencial (DGPS) é um melhoramento do Sistema de Posicionamento Global. Um sistema DGPS consegue precisões de localização na ordem de alguns centímetros contra alguns metros de um sistema convencional. O DGPS utiliza uma rede de estações terrestres de referência, que transmitem a diferença entre as posições indicadas pelos sistemas de satélite e as suas posições reais. Conhecida esta informação, as estações móveis conseguem fazer uma correcção no seu posicionamento.

Embora o sistema DGPS apresente vantagens em relação ao GPS, continua a sofrer de perturbações como o número de satélites “visíveis” e multipath. Por este facto é necessário prestar atenção a indicadores da qualidade do sinal como o HDOP (Horizontal Dilution of Precision) e o VDOP (Vertical Dilution of Precision).

Para servir de interface com diversos equipamentos nomeadamente o GPS, foi definido pela “National Marine Electronics Association” o standard NMEA. Este standard define uma variedade de strings que contêm informação diferenciada. Existem vários tipos de string como a AAM, ALM, GGA, entre outras.

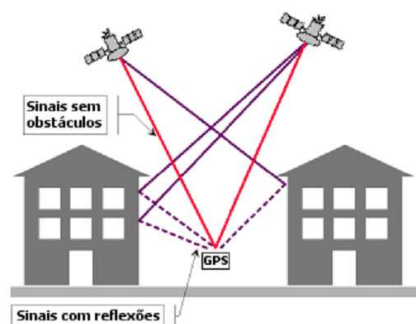


Figura 3.12: Multipath em sistemas GPS

3.6.2 Sensores Existentes no Sistema de controlo da direcção

Os sensores mais relevantes ao funcionamento do sistema de controlo da direcção são um encoder óptico rotativo e um potenciómetro linear.

CAPÍTULO 3. MODELAÇÃO MATEMÁTICA DO SISTEMA DE CONTROLO DA DIRECÇÃO

Embora ambos os dispositivos forneçam informação sobre a mesma grandeza, têm utilizações diferentes e complementares.

O encoder óptico possui uma grande resolução, no entanto a informação que fornece é relativa ao ângulo em que o sistema se encontra no momento em que é ligado. Existe a necessidade de conhecer o ângulo absoluto do sistema de direcção, para tal existe um potenciómetro linear montado na carroçaria e solidário com o sistema de direcção. Este sensor, ainda que não muito preciso permite dar uma estimativa do ângulo da direcção quando o sistema arranca. O valor do potenciómetro é lido e servirá para criar um offset entre o valor do ângulo absoluto e o relativo dado pelo encoder óptico.

Capítulo 4

Sistema de Controlo da Direcção

4.1 Hardware

Montado na coluna da direcção encontra-se o motor que actua o sistema, como demonstra a figura seguinte:

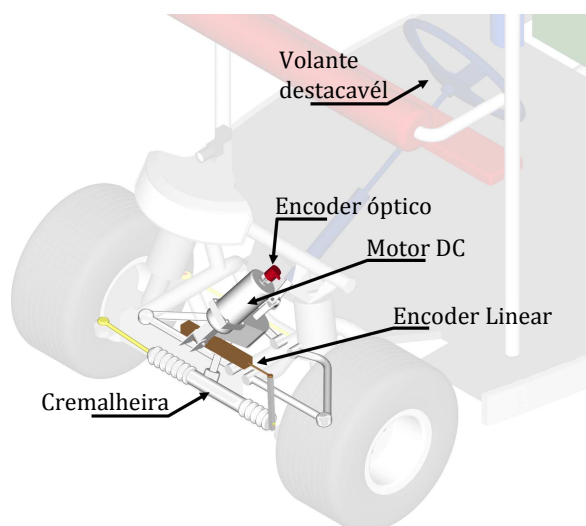


Figura 4.1: Estrutura do sistema de direcção

Foi já abordada a problemática de como controlar a posição de um motor DC, de como controlá-lo electricamente e que sensores se encontram disponíveis. O sistema de controlo desenvolvido foi dividido em dois submódulos principais.

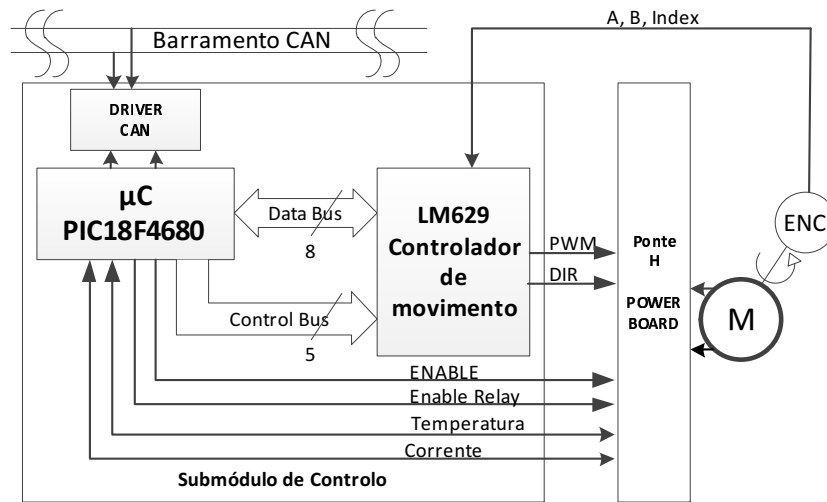


Figura 4.2: Diagrama do módulo de controlo da direcção

Destes, o primeiro é o controlador lógico, é responsável por receber novos comandos de posição angular criados pelo módulo de navegação e colocá-los em execução. O segundo, é o submódulo de potência (ponte-H), responsável pela amplificação dos sinais eléctricos necessários para controlar o motor.

4.1.1 Submódulo de controlo

A tarefa primordial do submódulo de controlo é receber comandos de posição angular vindos do módulo de navegação e executá-los. Para cumprir o objectivo, o submódulo de controlo conta com dois blocos principais. O primeiro destes blocos é um microcontrolador e o segundo, o circuito integrado “LM629”.

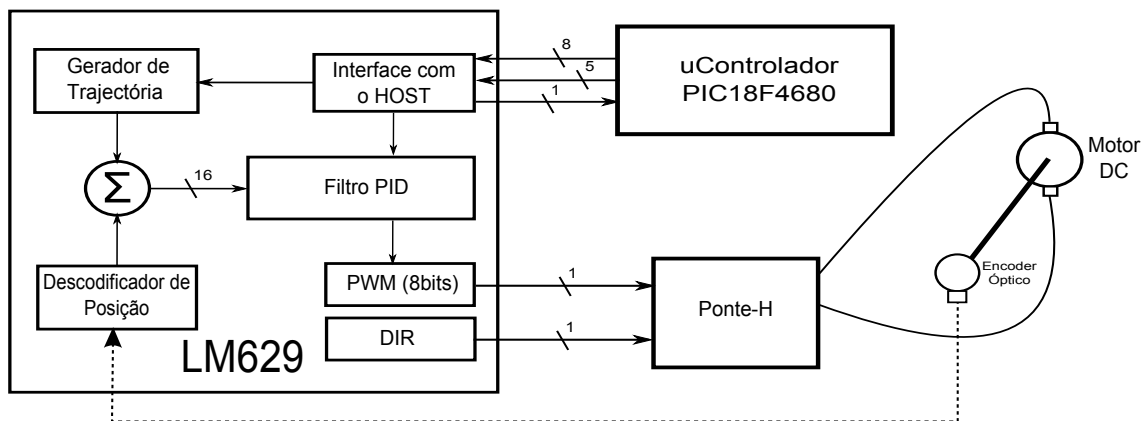


Figura 4.3: Estrutura interna do controlador LM629

O LM629 é um circuito integrado com funções de controlador de movimento que integra um controlador PID (“PID Filter”), gerador de perfis de velocidade trapezoidais (“Trajectory Generator”) e descodificador dos sinais provenientes do encoder óptico (“Position Decoder”). A sua saída são os sinais de comando, “DIR” e “PWM” que representam o sentido de rotação e a amplitude do sinal.

De forma a não forçar os componentes mecânicos com acelerações/desacelerações bruscas, o controlador permite a criação de perfis de velocidade trapezoidais. Este tipo de perfil

é constituído por uma fase de aceleração, desaceleração e uma fase de velocidade constante. O posicionamento obtido utilizando estes perfis permite uma resposta “criticamente amortecida”.

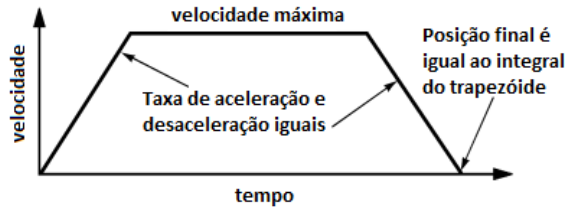


Figura 4.4: Perfis de velocidade trapezoidais

O LM629 não permite uma utilização “stand alone” dependendo do microcontrolador para o comandar. A comunicação entre os dois é feita através de dois barramentos paralelos, um de 8 bits para dados e outro de 5 bits para o controlo. Foi desenvolvida uma biblioteca para este integrado que permite a comunicação, a inicialização e a configuração de todos os parâmetros relativos ao movimento, como por exemplo, posição, aceleração e velocidade máxima. Os parâmetros relacionados com o controlador PID podem também ser configurados pela aplicação desenvolvida.

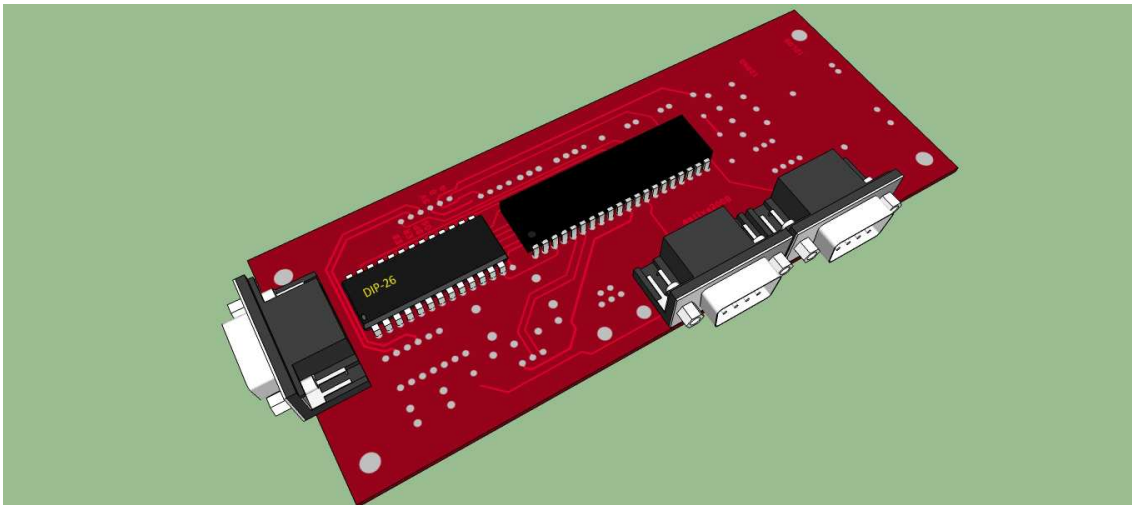


Figura 4.5: Representação 3D do Submódulo de controlo

O microcontrolador presente na placa de controlo serve vários propósitos tais como:

- Inicializar o LM629 e todos os parâmetros associados ao posicionamento.
- Leitura do potenciómetro linear para estimar a posição angular absoluta da direcção durante o “arranque” do sistema. Este valor serve como offset que será somado aos comandos de posicionamento do LM629.
- Monitorização da posição da chave da plataforma, activando a ponte-H se a chave estiver na posição “Auto” e desactivando-a nas restantes situações.
- “Gateway” entre o veículo ISRobotCar (barramento CAN) e o computador. Esta ligação tem como objectivo enviar informação para as camadas de controlo superiores (como o navegador) e receber comandos de posição para a direcção e velocidade

linear para o sistema de tracção. Ao receber um novo comando de posição, o microcontrolador serve de intermediário e informa o LM629 de novos parâmetros. Irá também enviar através da rede CAN comandos para a velocidade de tracção.

- Monitorização global do sistema de controlo da direcção, temperatura, corrente e posicionamento absoluto.

4.1.2 Submódulo de potência (Ponte-H)

Sob uma carga elevada na direcção, o motor pode atingir máximos de corrente de cerca de 50 Amperes. Estes valores de corrente são elevados e exigem algumas considerações no projecto da Ponte-H.

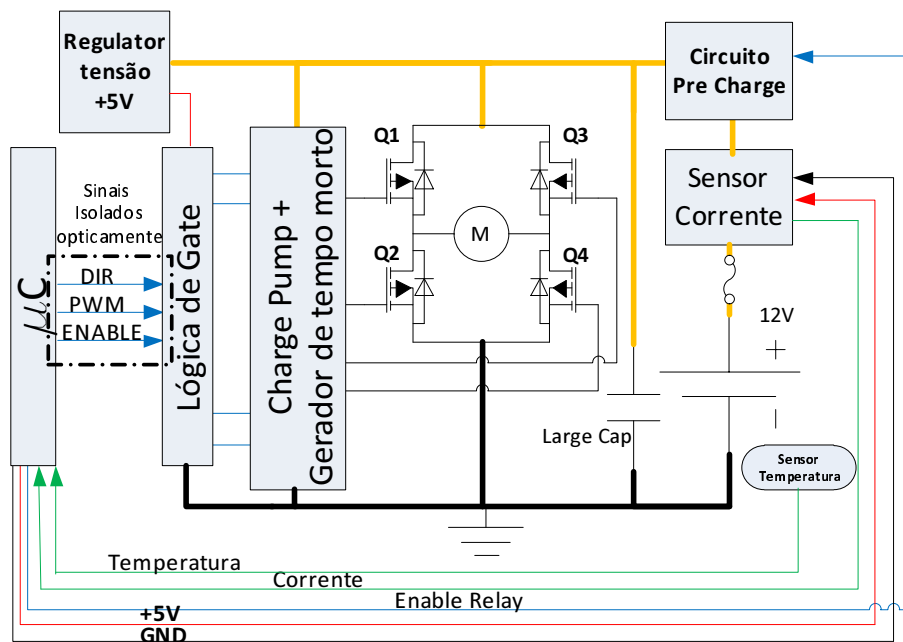


Figura 4.6: Diagrama do submódulo de potência

A escolha do tipo de comutadores para esta aplicação recaiu sobre factores como, tensão, corrente máxima, velocidade de comutação e robustez em ambientes automotivos. Após alguma pesquisa verificou-se que FETs (Field Effect Transistor) do tipo MOS (Metal Oxide Semiconductor) eram os indicados para a aplicação, em detrimento dos BJTs (Bipolar Junction Transistor) ou IGBTs (Insulated Gate Bipolar Transistor). Os baixos ganhos (h_{fe}) dos BJTs de potência, implicariam correntes de base (controlo) elevadas o que levaria a maiores perdas e a necessidade de buffers entre a lógica de controlo e o dispositivo. Por outro lado, os IGBTs são largamente utilizados em aplicações de potência, no entanto têm tipicamente tensões V_{ce} na ordem dos 2V quando completamente saturados. Numa configuração em ponte-H este problema é agravado pelo facto de dois destes dispositivos se encontrarem em série criando uma queda de potencial de aproximadamente 4V. Numa aplicação de 12V esta queda de tensão é significativa reduzindo para 8V a tensão máxima aplicada ao motor. Os mosfets de potência são preferíveis pois permitem correntes de dreno elevadas com perdas mínimas (R_{dsON} baixa), e correntes de gate (controlo) praticamente nulas.

Por reunir as especificações de corrente e tensão necessárias, por ter uma resistência de canal (R_{dsON}) baixa, e por possuir internamente protecções de gate que o tornam mais

robusto em ambientes difíceis, o dispositivo escolhido foi o mosfet canal-N “2SK3061”. É necessário fazer referência ao facto deste dispositivo já conter um “díodo de roda livre” necessário para lidar com a F_{cem} provocada pelos enrolamentos do motor.

Optou-se por utilizar mosfets canal-N na parte superior da ponte, (mosfets Q1 e Q3 da imagem 4.6) visto que estes dispositivos apresentam uma menor resistência de canal (R_{dsON}) para características tensão e corrente semelhantes num canal-P.

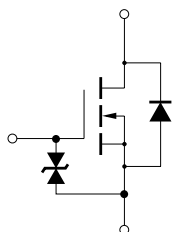


Figura 4.7: Esquema do mosfet escolhido com diodo de roda livre e protecção de Gate

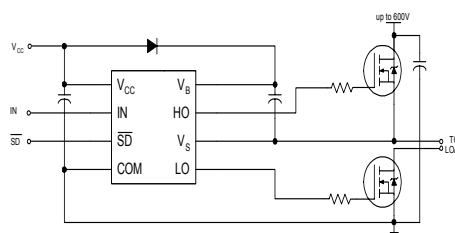


Figura 4.8: Utilização típica do controlador de meia ponte

Na figura 4.7 apresenta-se o esquema interno do mosfet escolhido.

A utilização de mosfets canal-N na parte superior da ponte exige a criação de uma tensão V_{gs} superior à de alimentação para o activar. Para este efeito foi usado um controlador de meia ponte com “charge-pump” incorporado. O circuito “charge-pump” cria uma tensão desacoplada (não referenciada aos 0V) que é aplicada entre o source e a gate do mosfet superior quando é necessário activa-lo.

Mesmo utilizando dispositivos iguais na parte superior e inferior da ponte, estes vão possuir tempos de ligar e desligar diferentes, sendo tipicamente maior o tempo para entrar ao corte do que a entrar na condução. Tomando o ramo esquerdo da ponte como exemplo e considerando que o transístor Q1 está em condução, é mudado o sentido do movimento. Para tal, é necessário desligar Q1 e activar Q2, como Q1 possui um tempo de desligar superior ao tempo de ligar de Q2 haverá por um curto espaço de tempo em que se verifica a condição de “shoot-through” podendo destruir os mosfets de potencia. De forma a evitar esta situação, o IR2184 gera um “tempo morto” em que permite aos dispositivos desligar antes de ligar o dispositivo complementar.

O IR2184 apenas possui uma entrada de controlo para ambos os mosfets. De forma a ser compatível com os sinais sentido de rotação (S) e PWM (P) gerados pelo LM629 é necessário “misturá-los”. A função lógica implementada para o efeito activa os transístores correspondentes e deixa o motor “travado” se nenhum sinal for aplicado (ambos os transístores da parte inferior ficam ligados).

S	P	Ramo Esquerdo	Ramo Direito
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

Devido aos condensadores de elevada capacidade montados na placa de circuito impresso, foi incluído um circuito de pré carga dos mesmos. O circuito de pré carga é uma malha RC (Resistência ligada em série com condensador). O facto de incluir uma resistência, faz com que a constante de tempo do sistema aumente (demora mais a “encher”

CAPÍTULO 4. SISTEMA DE CONTROLO DA DIRECÇÃO

o condensador) mas permite limitar a corrente durante o transitório que poderia danificar a placa de circuito impresso (PCI).

Constante de tempo:

$$\tau = r \cdot c[s] \quad (4.1)$$

Ao fim de 5τ o fenómeno transitório termina e é activado um relé em paralelo com a resistência fornecendo um caminho directo à alimentação.

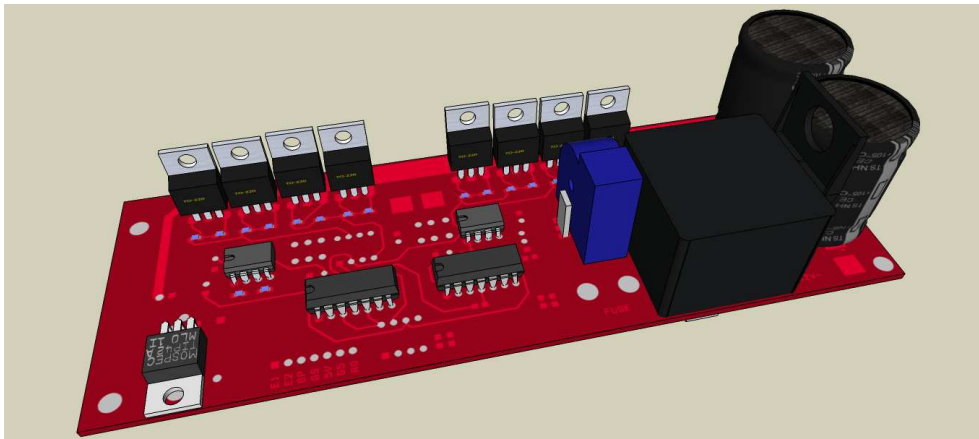


Figura 4.9: Representação 3D Ponte-H desenvolvida

Por se tratar de uma aplicação onde a segurança também é um factor importante, foram usados dois mosfets em paralelo em cada braço da ponte. Esta configuração ajuda na condução de correntes elevadas e cria redundância no caso de falha de um dos comutadores.

Para protecção da ponte, existe um sensor de temperatura e um sensor de corrente. Ao ultrapassar os limites de temperatura e/ou corrente definidos no firmware do microcontrolador a ponte é desactivada através dos pinos de controlo (SD) do IR2184 que por sua vez desactiva os transístores de potencia.

Todos os sinais de comando provenientes da placa de controlo estão isolados electricamente através de opto acopladores. Em último recurso um fusível à entrada da ponte-H protege-a de correntes suficientemente altas que a possam danificar.

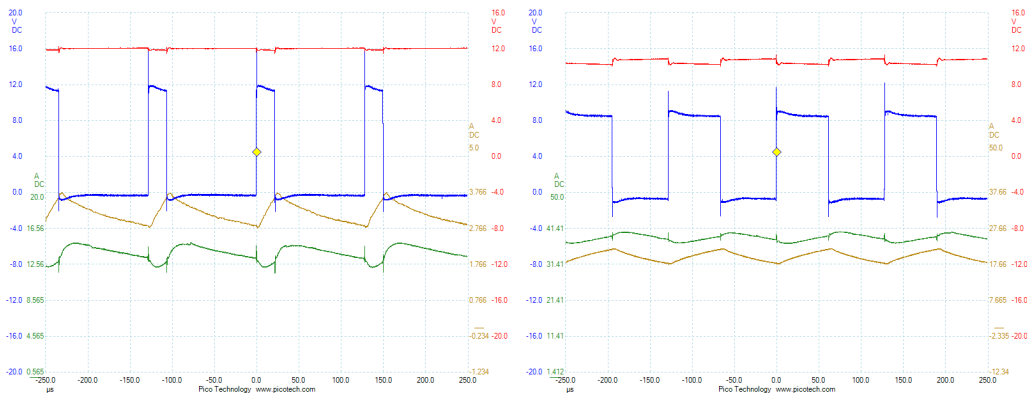


Figura 4.10: Formas de onda sob carga leve
Figura 4.11: Formas de onda sob carga elevada

CAPÍTULO 4. SISTEMA DE CONTROLO DA DIRECÇÃO

As figuras anteriores pretendem mostrar as formas de onda quando a ponte-H está em funcionamento. Os traços vermelho e amarelo representam tensão e corrente de entrada (fornecidos pela bateria), a azul o sinal de PWM gerado pelo LM629 e a verde a corrente no motor. Como se pode verificar, para manter o ângulo da direcção quando é aplicada uma força externa (perturbação), o motor necessita de exercer mais binário. Para este efeito o LM629 aumenta o Duty-Cycle do PWM e a corrente no motor sobe. Na figura é possível verificar picos de corrente de aproximadamente 50 Amperes.

Nas condições da figura 4.11 em que a corrente média é cerca de 50[A] e o Duty-Cycle 60% pode-se calcular a potência dissipada no comutador. Através das equações 3.11 a 3.13 chega-se ao valor de 13W, no entanto como cada ramo da ponte possui dois mosfets em paralelo a potência dissipada em cada é de aproximadamente 6.5[W].

4.2 Firmware

Uma vez apresentada a estrutura de comando e controlo do sistema de direcção, é agora detalhado o firmware desenvolvido.

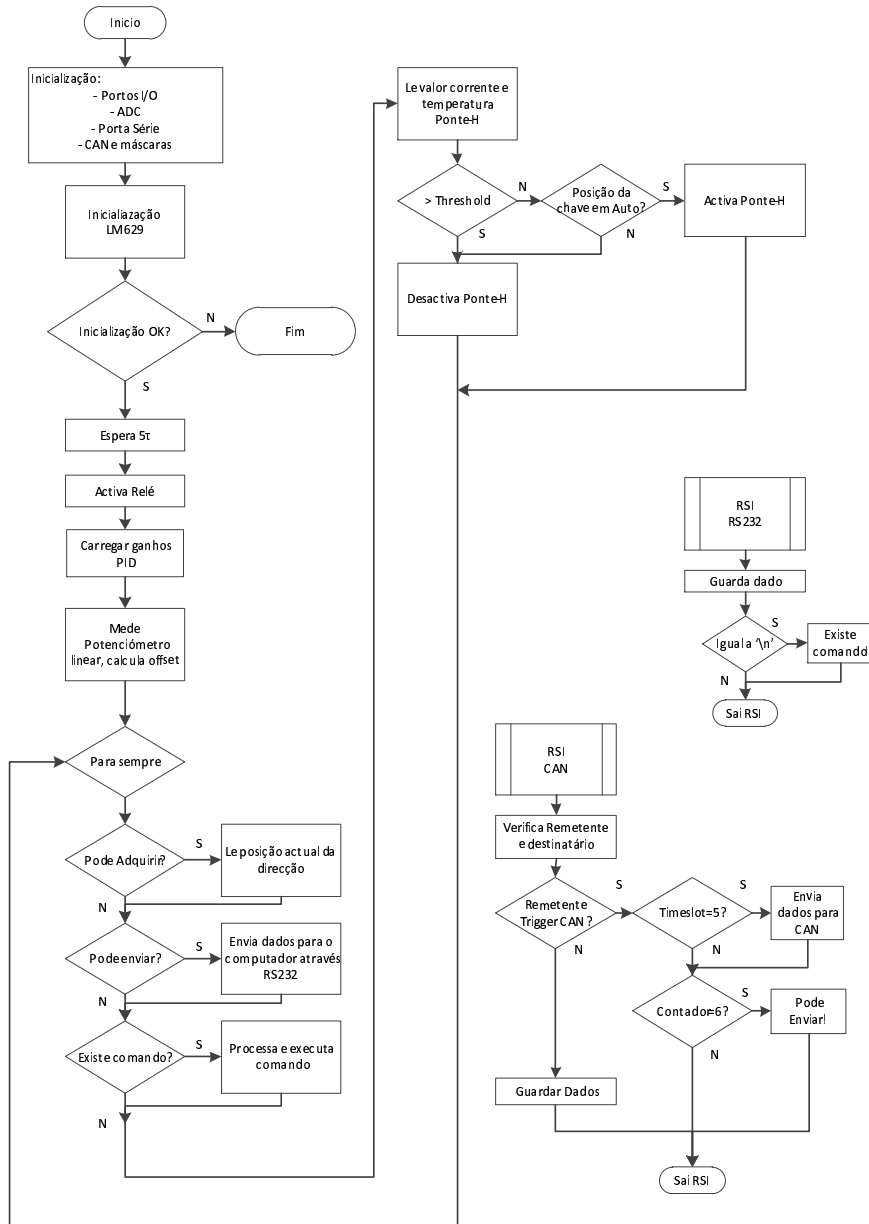


Figura 4.12: Firmware para sistema embebido

O firmware começa por inicializar diversos periféricos do microcontrolador que servem para conversão analógica digital, comunicações, controlo do LM629, entre outros. De seguida o firmware tenta inicializar o LM629; caso seja bem sucedido continua a execução, caso não consiga, termina a aplicação por não haver resposta do controlador de movimento.

Em caso de sucesso, o microcontrolador aguarda até que o circuito de “pre-charge” carregue o banco de condensadores e ao fim de 5τ é activada a relé ligando a ponte directamente à fonte.

Seguidamente são configurados os parâmetros relativos ao controlador PID.

O potenciómetro linear é lido aplicando-lhe uma tensão de 5V e identificando a posição do ponto central através de um ADC de 10 bits. Este valor permite determinar o ângulo absoluto do sistema. A relação entre os valores provenientes do potenciómetro e os equivalentes em pulsos de encoder é linear como se pode observar na figura seguinte:

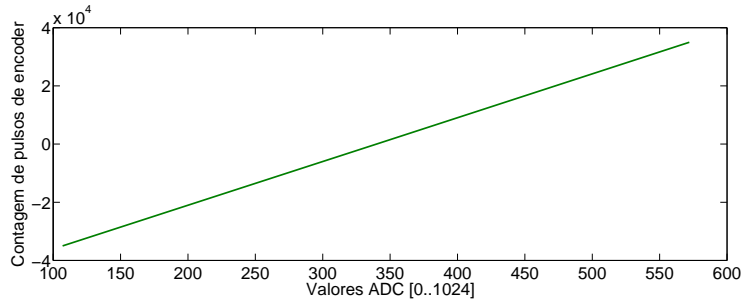


Figura 4.13: Relação potenciómetro linear/encoder óptico

Através destes dados foi possível deduzir a seguinte equação que relaciona as duas leituras:

$$P_{enc} = 150.54 \cdot Pot_{linear} - 51108 \quad (4.2)$$

Onde:

P_{enc} - Número de pulsos de encoder óptico

Pot_{linear} - Valor lido pelo ADC

O valor calculado serve como offset e é somado aos comandos de posição que forem enviados.

Nesta fase o programa entra num loop infinito dependendo de comandos/dados provenientes da porta série ou da rede CAN para efectuar as acções necessárias. Estes canais de comunicação são acedidos através de rotinas de serviço a interrupção despoletadas cada vez que existem dados novos. Aquando da existência de um novo comando a ser executado, uma flag no programa é activada para ser processado. O programa também é responsável por adquirir a posição da direcção e por concentrar a informação proveniente de outros ECU na rede CAN e envia-la para o computador na temporização correcta.

Por fim, o programa ainda é responsável por monitorizar a posição da chave activando a ponte-H caso esta esteja na posição “Auto” bem como a temperatura e a corrente máxima desligando-a caso estes valores saiam dos parametros definidos.

4.3 Protocolo de comunicação

Para que a aplicação a executar no computador (ROS) e o microcontrolador possam comunicar, foi criado um protocolo baseado em ASCII com a seguinte estrutura:

4.3.1 Robot Operating System – > Microcontrolador

P	Valor entre -35000 a 35000 (ASCII)	T	Endereço CAN (2 bytes)	Valor – 0 a 200 (ASCII)	“\n”
---	------------------------------------	---	------------------------	-------------------------	------

Figura 4.14: Comunicação Computador/Microcontrolador

O primeiro campo do protocolo é o carácter “P” e indica um comando para posição angular da direcção. Seguidamente é enviado em ASCII a posição angular da direcção (em pulsos de encoder). O carácter “T” é enviado indicando comando para a tracção, seguindo-se o endereço CAN para onde deverá ser enviado e um valor de 0 a 200. O valor de 0 a 200 representa em percentagem a velocidade da tracção, sendo de 1 a 100 velocidades positivas e de 101 a 200 velocidades negativas. Para finalizar o comando é enviado o carácter “\n”.

Exemplo de um comando que posicione a direcção nos 32000 pulsos de encoder e imponha uma velocidade linear de 20% no sistema de controlo longitudinal:

P	3	2	0	0	0	T	0x00	0xA8	2	0	“\n”
---	---	---	---	---	---	---	------	------	---	---	------

Figura 4.15: Comunicação Computador/Microcontrolador

Ou em hexadecimal:

0x50	0x33	0x32	0x30	0x30	0x30	0x54	0x00	0xA8	0x32	0x30	0x0A
------	------	------	------	------	------	------	------	------	------	------	------

Figura 4.16: Exemplo de comando em Hexadecimal

4.3.2 Microcontrolador – > Robot Operating System

De forma semelhante ao protocolo detalhado anteriormente foi desenvolvido um protocolo para a comunicação no sentido inverso também baseado em ASCII.

A estrutura genérica de uma mensagem é a seguinte:

Identificador de mensagem	Restante Mensagem	“\n”
---------------------------	-------------------	------

Figura 4.17: Exemplo de comando em Hexadecimal

O primeiro byte enviado pelo microcontrolador para o computador identifica o tipo de mensagem. Até à data, o protocolo ainda só possui definidas mensagens do tipo “0” que contêm informações de diversos sensores presentes na plataforma. O carácter que termina a transmissão é o “\n”.

No entanto outros tipos de mensagem estão previstos e podem ser definidos como por exemplo: monitorização de tensões, correntes e temperatura das baterias, mensagens de erro de hardware, entre outras. Para o caso das mensagens do tipo 0 a mensagem é definida da seguinte forma:

PD	VL	EE	ED	EC	MS	MP	MC	CX	CY	GZ	AX	AY	AZ
----	----	----	----	----	----	----	----	----	----	----	----	----	----

Figura 4.18: Mensagem dados sensoriais

Onde:

PD - Ângulo direcção em pulsos de encoder.	MC - Contador de Magneto
VL - Velocidade linear em mm/s	CX - Bússola eixo X
EE - EncoderEsquerdo	CY - Bússola eixo Y
ED - Encoder Direito	GZ - Giroscópio eixo Z
EC - Encoder Central	AX - Acelerómetro eixo X
MS - Polaridade Magneto	AY - Acelerómetro eixo Y
MP - Posição Magneto	AZ - Acelerómetro eixo Z

4.4 Relação pulsos-ângulo

O controlador de posição LM629 recebe feedback de velocidade e posição através de um encoder em quadratura com 4000ppr. De forma a integrar este sistema com o módulo de navegação, foi necessário estabelecer a relação entre o ângulo real da direcção (em graus/radianos) e a contagem de pulsos de encoder. Para o efeito foram feitas algumas experiências práticas que consistem em descrever trajectórias circulares com diferentes números de pulsos (ângulo) e registar o raio de curvatura efectuado pela plataforma.

Através da relação seguinte foi calculado o ângulo da direcção correspondente em graus/radianos:

$$\phi = \tan^{-1} \frac{L}{R} \quad (4.3)$$

Onde:

- ϕ - Ângulo da direcção
- L - Distancia entre eixos
- R - Raio de curvatura

Os resultados dos testes são os seguintes:

Número de pulsos	Diâmetro de curvatura [m]	Ângulo direcção [°]
0	∞	0
5000	80,4	3,0614098
10000	39,4	6,228441969
20000	17,94	13,48026916
35000	9,02	25,48799689
40000	7,62	29,43623109
50000	5,68	37,12726045

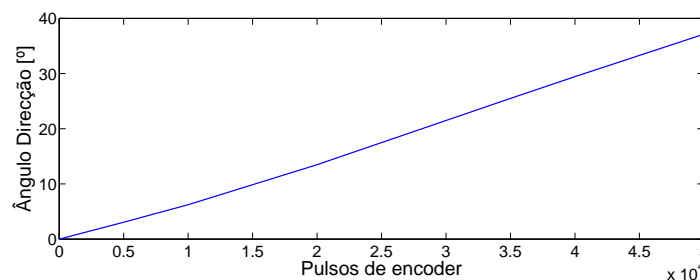


Figura 4.19: Relação entre pulsos de encoder e angulo de direcção

Com os testes descritos chegou-se ao valor de $6,86E^{-4} \pm 4,959E^{-5}$ graus/pulso.

4.5 Projecto controlador PID

Como será detalhado no capítulo seguinte, o sistema real foi aproximado a um sistema de primeira ordem. O sistema de controlo da direcção faz uso de um controlador PID do qual é necessário calcular os ganhos K_p , K_i e K_d .

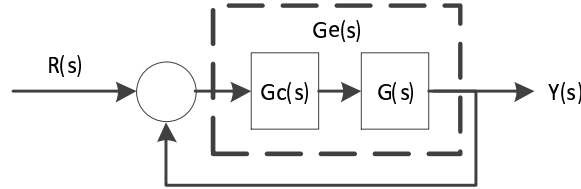


Figura 4.20: Malha de controlo equivalente

Nos capítulos anteriores foi abordado a estrutura do controlador PID para o qual se podem deduzir as seguintes igualdades:

Função transferência do sistema:

$$G(s) = \frac{b_0}{a_0s + a_1} \quad (4.4)$$

Função transferência do controlador:

$$G_c(s) = \frac{K_d s^2 + K_p s + K_i}{s} \quad (4.5)$$

Função transferência equivalente:

$$G_e(s) = G_c(s) \cdot G(s) \quad (4.6)$$

Função transferência do sistema em malha fechada:

$$\frac{Y(s)}{R(s)} = \frac{G_e(s)}{1 + G_e(s)H(s)} = \frac{b_0(K_d s^2 + K_p s + K_i)}{a_0 s^2 + a_1 s + b_0(K_d s^2 + K_p s + K_i)} \quad (4.7)$$

Um sistema de segunda ordem tem a seguinte equação característica:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (4.8)$$

Igualando os termos com grau equivalente da equação característica do sistema de segunda ordem e do sistema em malha fechada:

$$\begin{aligned} K_p &= \frac{2\zeta\omega_n - a_1}{b_0} \\ K_i &= \frac{\omega_n^2}{b_0} \\ K_d &= \frac{1 - a_0}{b_0} \end{aligned} \quad (4.9)$$

As equações 4.9 permitem o cálculo dos ganhos PID em função de ζ e ω_n controlando assim o tipo de resposta desejada.

4.6 Utilitário de criação de perfis

Durante a fase de desenvolvimento do sistema de controlo lateral, foi necessário fazer testes com diferentes parâmetros de movimento. Para isso, foi desenvolvido um GUI na ferramenta Matlab que simplificasse o processo de gerar os parâmetros necessários ao perfil desejado.

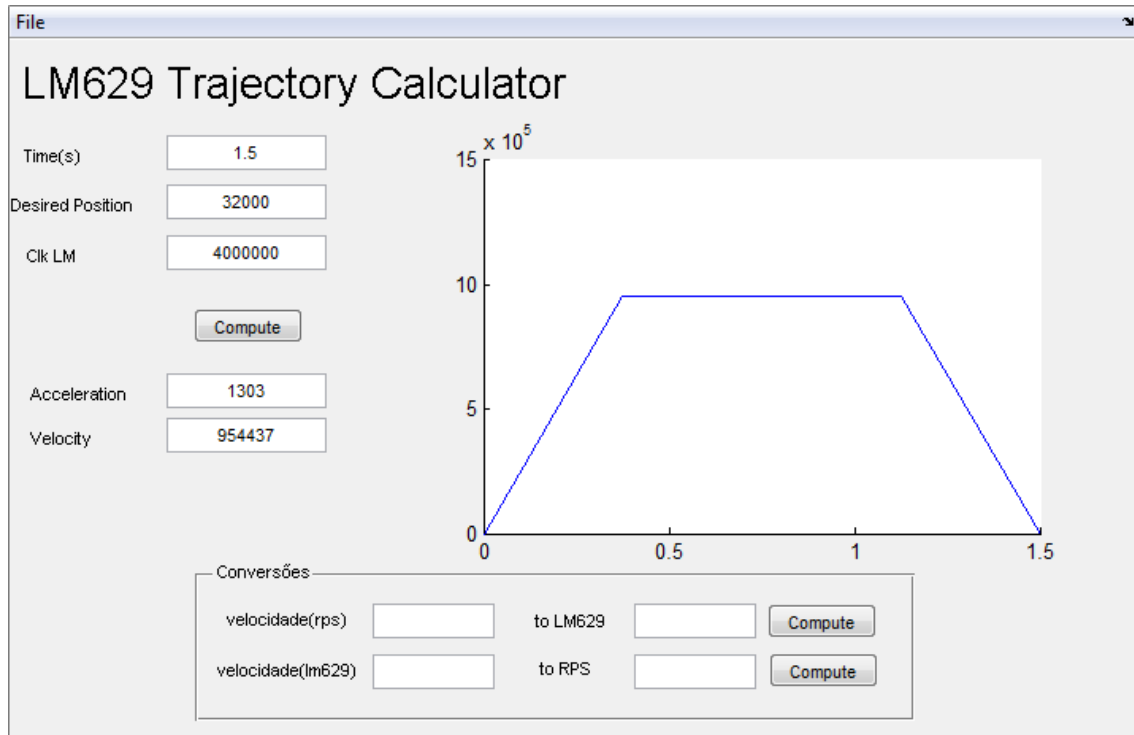


Figura 4.21: Utilitário para cálculo de perfis trapezoidais

Dado o tempo desejado para o movimento, assim como o número de pulsos a percorrer, o software calcula o valor de velocidade e aceleração a configurar no LM629. Permite também de forma gráfica inspeccionar o perfil gerado, mantendo o valor limite de velocidade (Y máximo do gráfico) fixo, para possibilitar uma comparação relativa.

O software permite ainda converter entre as unidades do LM629 e unidades mais amigáveis como as rps (rotações por segundo) e vice-versa.

Nas figuras seguintes encontram-se dois exemplos de perfis planeados e respectivos resultados. Na figura 4.22, foi pedido ao controlador para mover a direcção de 0 até 32000 pulsos (aproximadamente 22°) em 2 segundos e na figura 4.23 foi pedido o mesmo movimento mas com a duração de 4 segundos.

CAPÍTULO 4. SISTEMA DE CONTROLO DA DIRECÇÃO

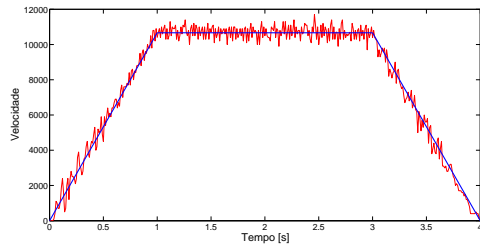


Figura 4.22: Movimento de 32000 pulsos em 4 segundos

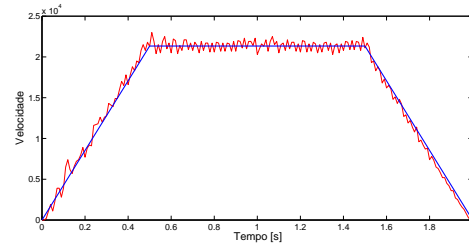


Figura 4.23: Movimento de 32000 pulsos em 2 segundos

Os valores de velocidade e aceleração a configurar no LM629 para o primeiro caso foram 357914 e 183 respectivamente e no segundo 715828 e 733. Nas imagens, a azul encontra-se o perfil de referência e a vermelho o resultado real.

Capítulo 5

Sistema de simulação

5.1 Simulador

A utilização da plataforma ISRobotCar nem sempre facilita o estudo e teste de novos algoritmos de navegação. Um ambiente de simulação com um grau de realismo elevado permite uma grande flexibilidade no desenvolvimento de novas soluções. Depois de simulado com sucesso é mais simples a implementação e debug na plataforma real.

Com este objectivo e dando seguimento ao trabalho já realizado anteriormente foi adaptado e actualizado um simulador em ambiente Matlab/Simulink para a plataforma ISRobotCar.

O simulador é constituído por um GUI (Graphical User Interface), um diagrama de Simulink e funções auxiliares.

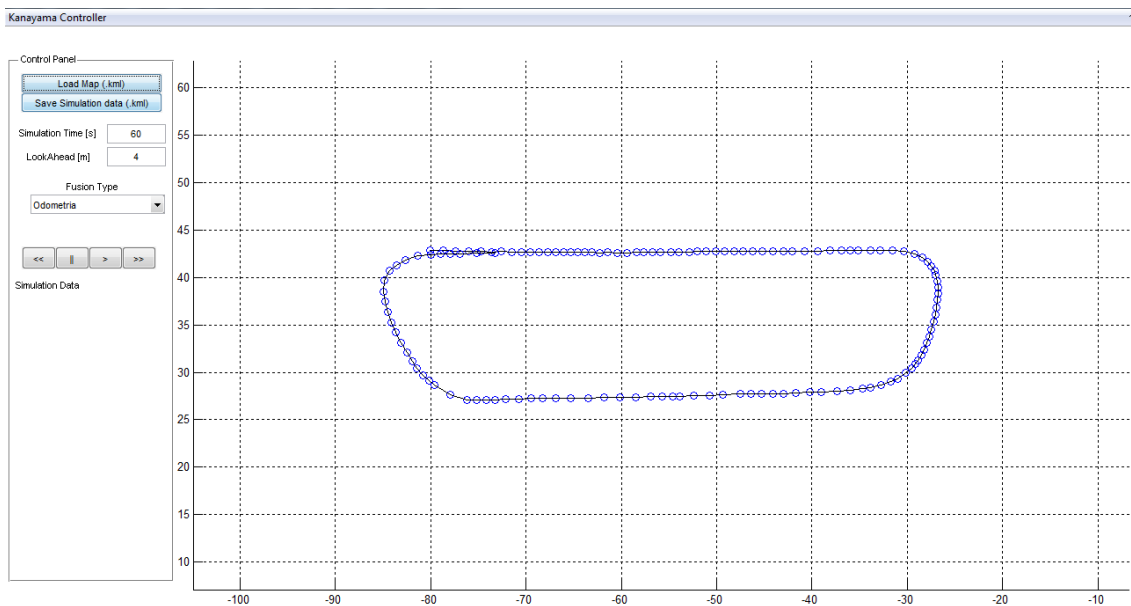


Figura 5.1: GUI do simulador

O simulador permite simular a navegação da plataforma com o controlador de movimento Kanayama. O diagrama de Simulink implementado para esta tarefa tem uma estrutura de controlo semelhante à que já foi apresentada no Capítulo 3.2.

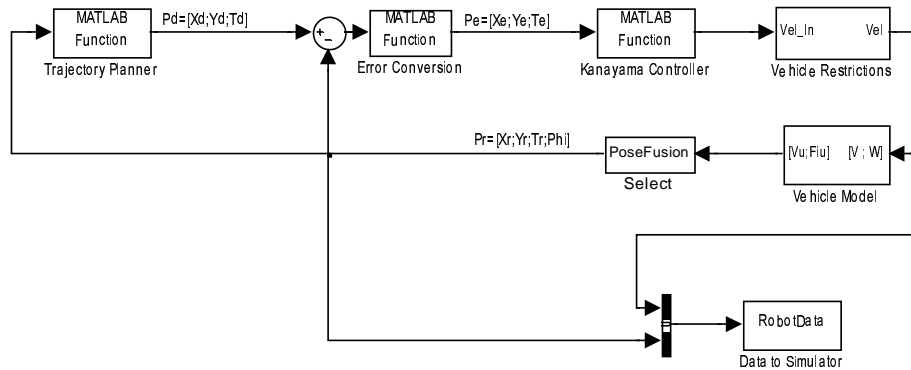


Figura 5.2: Diagrama de controlo em Simulink

De forma a dar realismo à simulação, o bloco representado por “Vehicle Model” contém os modelos do controlador lateral e longitudinal da plataforma. A qualidade de simulação depende em grande parte da qualidade destes modelos. O modelo do controlador lateral será descrito na secção seguinte.

5.1.1 Integração com o Google Earth

Uma das funcionalidades pretendida era a capacidade de definir os trajectos a percorrer utilizando a ferramenta “Google Earth”.

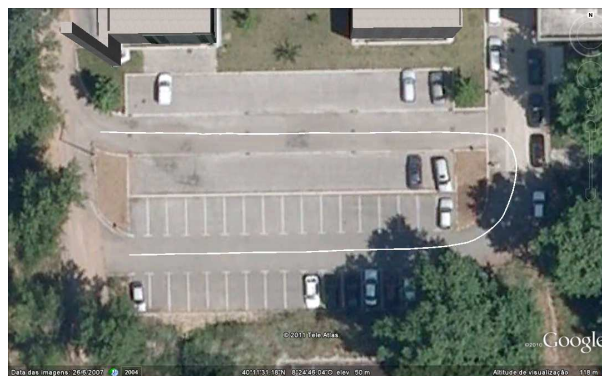


Figura 5.3: Trajecto definido no software Google Earth

Esta ferramenta permite a visualização da superfície terrestre através de imagens de satélite facilitando a definição dos trajectos e posteriormente guardá-los em ficheiros “KML”. Este tipo de ficheiro tem uma estrutura em XML e contém num dos campos as coordenadas Latitude Longitude e Altura de cada ponto definido. Após algumas conversões de coordenadas (detalhadas no capítulo 6.1.2) é possível utilizar o percurso para simulação.

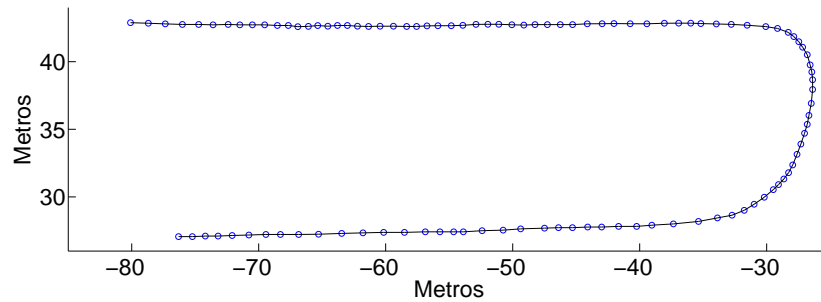


Figura 5.4: Percurso importado do Google Earth

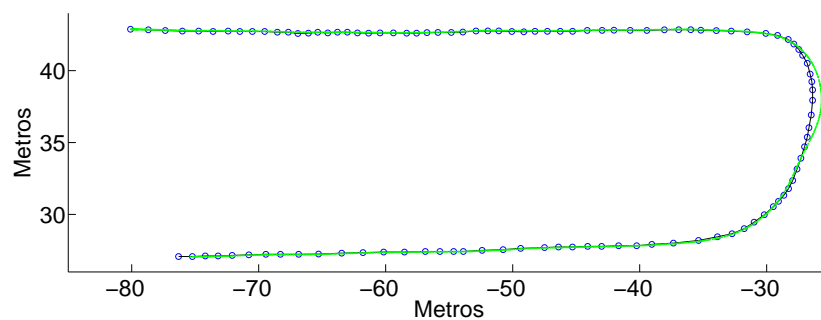


Figura 5.5: Simulação de navegação com controlador Kanayama

Na figura anterior a verde e sobreposto à trajetória desejada é possível verificar o resultado da simulação utilizando o controlador Kanayama.



Figura 5.6: Trajetória real e simulação no Google Earth

Após a simulação, o simulador permite exportar o resultado em formato KML para que seja possível sobrepor os resultados no Google Earth e obter uma noção mais realista da validade da simulação. Na imagem, a linha a branco representa a trajetória desejada e a vermelha a simulada.

5.2 Modelo simulação para controlador lateral

Obter dados realistas a partir da simulação obriga a construir modelos precisos do controlador lateral e longitudinal.

O sistema de controlo da direcção é constituído pelo controlador de movimento LM629, um motor DC de ímanes permanentes e toda a componente mecânica associada. De forma a obter o modelo para este sistema era necessário conhecer em detalhe todas as características quer eléctricas quer mecânicas. Tratando-se de um motor DC é relativamente fácil construir um modelo com base em parâmetros como a resistência, a indutância e o momento de inércia, no entanto desconhecendo-se os parâmetros associados à componente mecânica é difícil a criação de um modelo puramente matemático.

Outra abordagem possível é o agrupamento do sistema motor+mecânica e aplicar degraus de tensão para identificar a sua resposta.

Foi aplicado uma tensão de 12V directamente ao sistema de direcção e registada a evolução da velocidade, obtendo os seguintes resultados:

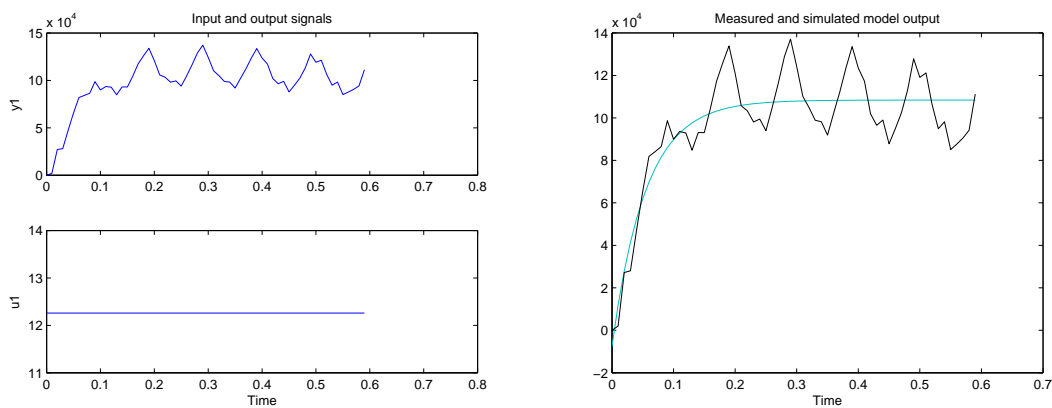


Figura 5.7: Resposta do sistema e comando aplicado
 Figura 5.8: Aproximação do sistema a um modelo de primeira ordem

Através da ferramenta “IDENT” do Matlab foi escolhido um modelo que melhor representa o sistema como se pode verificar na imagem 5.8.

A função de transferência obtida é:

$$G(s) = \frac{8842.6}{1 + 0.054694s} \quad (5.1)$$

Esta função de transferência relaciona tensão aplicada ao motor com velocidade de rotação (pulsos por amostra).

5.2.1 Cálculo dos ganhos do controlador PID

Uma vez identificada a função de transferência que caracteriza o sistema, é possível determinar os ganhos do controlador PID.

Para um sistema de posicionamento não são desejadas oscilações no sistema em torno do setpoint, por este motivo o valor escolhido para ζ é 1, forçando uma resposta criticamente amortecida. O sistema tem um pólo real em -18.2835 , e deseja-se o sistema mais rápido possível, para tal foi escolhido o valor de ω_n de 18.2835 .

Utilizando as equações 4.9 os valores para K_p , K_i e K_d são, $3.9581e-003$, $3.6641e-002$ e $1.0690e-004$ respectivamente.

5.2.2 Modelo de Simulação

Como foi visto no capítulo anterior, o LM629 cria perfis de velocidade trapezoidais que são os maiores responsáveis pela dinâmica do sistema. De forma a obter um modelo em Simulink do sistema, o gerador de rampas necessita de ser modelado.

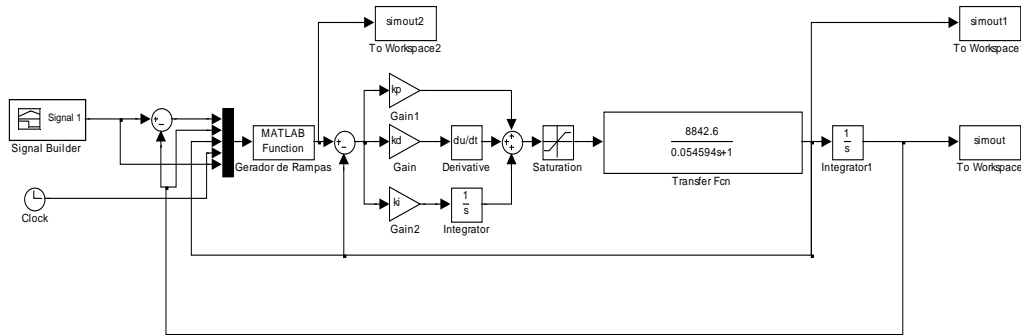


Figura 5.9: Diagrama simulink para modelo do controlador lateral

O sistema possui uma malha interior com um controlador PID de velocidade e os limites de saturação do sinal de comando (-12V e +12V). Esta malha é responsável por manter a velocidade do motor constante. À saída do PID existe um integrador convertendo ao longo do tempo velocidade em posição. Em torno desta malha existe outra exterior para controlo da posição. Entre as duas existe o gerador de perfis trapezoidais que vai calculando a cada instante qual a velocidade a aplicar à malha interior para cumprir os valores de velocidade e acelerações máximos.

Para validar o modelo criado foi definido no controlador da direcção um trajecto com vários setpoints a atingir. O percurso foi escolhido de forma a contemplar todas as situações, como por exemplo, ter ou não tempo suficiente para descrever a trajectória completa antes de lhe ser enviado um novo setpoint.

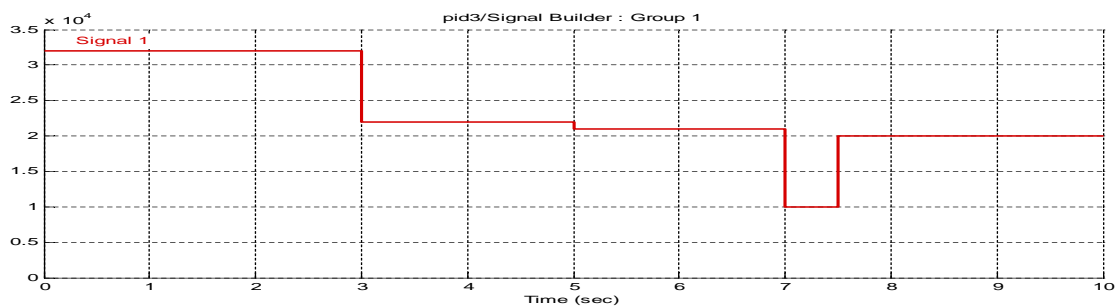


Figura 5.10: Diagrama temporal dos percursos a efectuar

O mesmo percurso foi efectuado utilizando o sistema de controlo real e o modelo criado. Na figura seguinte apresenta-se a sobreposição de ambos os resultados:

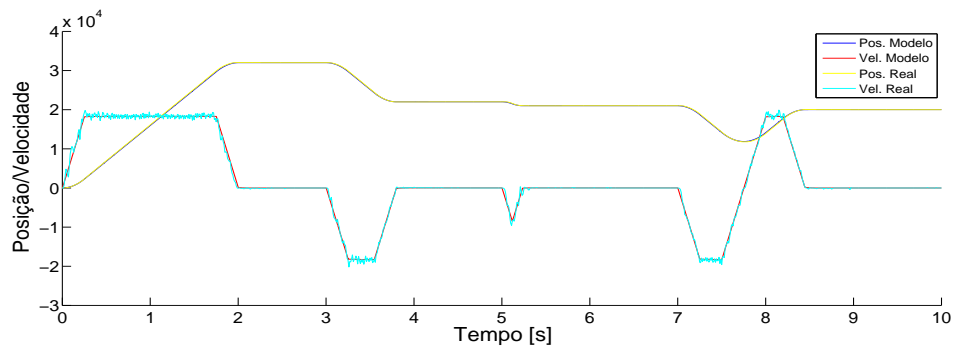


Figura 5.11: Sobreposição dos resultados reais e de simulação

Como é possível observar, o modelo consegue seguir de forma fiel à realidade os comandos de posição angular que lhe são impostos.

Capítulo 6

Robotic Operating System (ROS)

6.1 Sobre o Robot Operating System

O "Robot Operating System" (ROS) é uma estrutura de desenvolvimento de software para robôs (Framework) que fornece funcionalidades ao nível do sistema operativo. O ROS foi desenvolvido originalmente em 2007 sob o nome "Switchyard" pelo Laboratório de Inteligência Artificial de Stanford. A partir de 2008, o desenvolvimento continuou principalmente na Willow Garage, um instituto de pesquisa robótica/incubadora. O ROS presta serviços standards em sistemas operativos, como abstracção de hardware, transmissão de mensagens entre os processos e gestão de pacotes. Este sistema é orientado para sistemas Unix-like, sendo o Ubuntu o mais "apoiado". Outras variantes como Fedora e Mac OS X são até ao momento consideradas experimentais.

O ROS é organizado em pacotes e estes podem formar conjuntos chamados "stacks". Existem stacks capazes de implementar funcionalidades, tais como mapeamento e localização, planeamento de trajectórias, percepção, simulação, entre outros.

Alem das funcionalidades já mencionadas, o ROS tenta facilitar a gestão de pacotes/utilização através de utilitários como:

- roscd - Permite navegar directamente para as directorias onde se encontram os pacotes. Exemplo: "roscd commIO".
- rosmake - Compila automaticamente o pacote especificado. Exemplo: "rosmake commIO".
- rostopic - Permite publicar informações num tópico e mensagem específico através da linha de comandos. Exemplo: "rostopic pub -1 /IKinematicsISR/IKinematics IKinematicsISR/IKinematics 32000 25".
- rxgraph - Permite de forma gráfica visualizar os pacotes em execução no framework ROS bem como os tópicos que partilham entr si. Exemplo: "rxgraph".

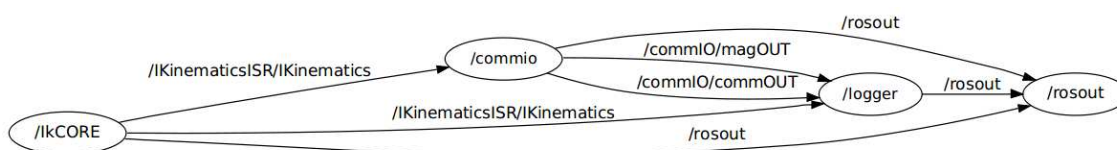


Figura 6.1: Diagrama produzido pelo rxgraph

CAPÍTULO 6. ROBOTIC OPERATING SYSTEM (ROS)

Em particular no trabalho realizado durante esta dissertação tirou-se partido da modularidade dos sistemas realizados usando ROS assim como do seu sistema de troca de mensagens. Os pacotes em execução dentro do framework ROS são conhecidos como nós. Os nós podem comunicar entre si através de tópicos numa filosofia de publicador/subscritor. Quando um nó pretende transmitir informações para outros, publica num tópico uma mensagem que pode ser standard ou personalizada. Por sua vez, se um dos nós estiver “interessado” nas informações presentes num determinado tópico subscrive-o. Um nó que tenha subscrito um determinado tópico contém uma função de “callback” que é executada cada vez que outro nó lá publique. O facto de serem usados tópicos para comunicar entre nós é útil, pois permite desacoplar os vários constituintes do software. A existência de uma função de callback também facilita o desenvolvimento pois permite a criação de uma malha de controlo síncrona com o nó anterior (semelhante ao uso do Simulink) como se pode verificar no diagrama seguinte:

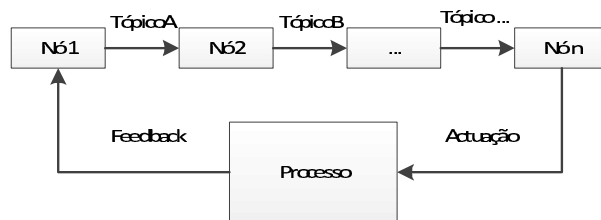


Figura 6.2: Exemplo de malha de controlo usando ROS

Durante esta dissertação foram criados três nós que providenciam funcionalidades como comunicação com a plataforma (6.2), comunicação com o GPS diferencial (6.3) e Log de todos os tópicos que circulem no sistema para posterior análise (6.4). A estrutura de controlo implementada é a seguinte:

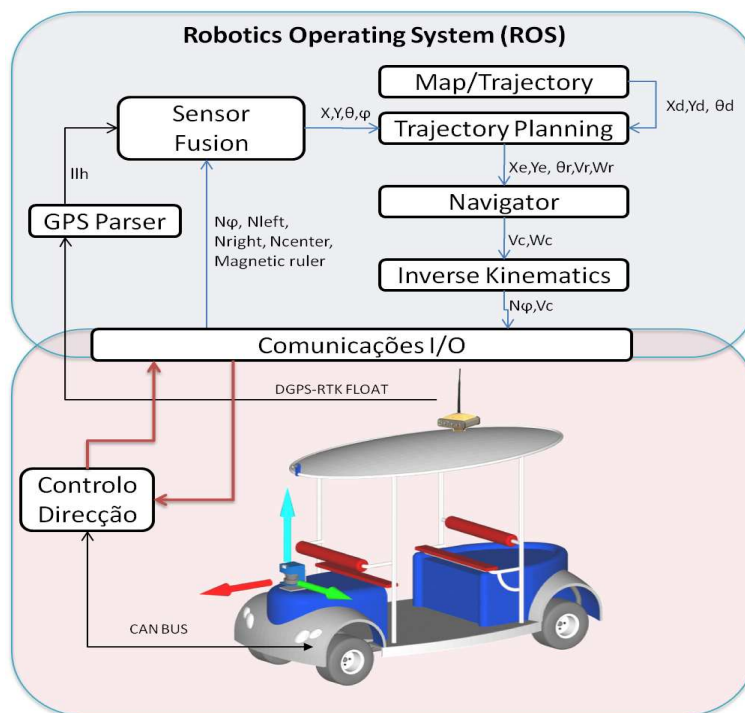


Figura 6.3: Integração do ROS com a plataforma

A imagem acima pretende mostrar a integração e comunicação do ROS na malha de controlo.

6.2 Nó de comunicação com plataforma commIO

Este nó é responsável pelo interface entre o hardware e o software da malha de controlo (ROS). A mensagem recebida por este nó contém informação sensorial enviada pelo módulo de controlo da direcção usando o protocolo já descrito. Por sua vez, o nó “commIO” publica no tópic “commIO/commOUT” uma mensagem personalizada capaz de enviar todos os dados para os nós seguintes. Outra funcionalidade do nó “commIO” é o envio de comandos de velocidade e ângulo de direcção calculados pelo navegador/cinemática inversa para os actuadores através do protocolo também já descrito.

6.3 Nó de comunicação com GPS diferencial TopCon

Montado na plataforma ISRobotCar existe um GPS diferencial do fabricante TopCon. De forma a integrar este dispositivo foi necessário desenvolver um módulo ROS que faça o parsing das strings NMEA e publique no sistema a informação que nelas esteja presente.

Para a aplicação desenvolvida e visto possuir todos os campos necessários foi usada a string GGA. Nesta string estão presentes informações como Latitude, Longitude, Altura, tipo de Fix (“Real Time Kinematic, Float RTK”), DOP (Dilution Of Precision) horizontal, entre outros.

O sistema de coordenadas usado pelo GPS é conhecido como LLH (Latitude Longitude Height). Este tipo de coordenadas utiliza um geóide de referência (superfície gravitacional equipotencial que define o nível médio do mar) sendo o mais comum, o WGS84.

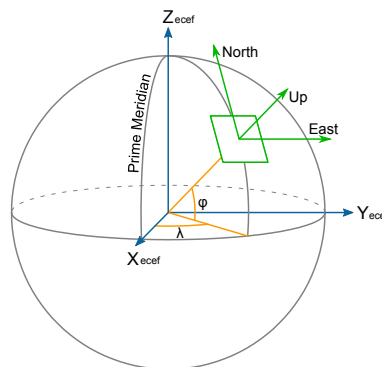


Figura 6.4: Vários sistemas de coordenadas (LLH, ECEF, ENU)

Os algoritmos de navegação estudados funcionam sobre um plano cartesiano, havendo a necessidade de conversão das coordenadas LLH para outro tipo de coordenadas. Para o efeito existe o sistema de coordenadas local ENU (East North UP). Este sistema cria um plano cartesiano virtual que é tangente à superfície da terra num determinado ponto escolhido como sendo a origem. Os eixos X, Y e Z deste plano ficam alinhados com o Norte, Este e cima (normal aos anteriores) respectivamente. A conversão entre estes sistemas de coordenadas não é directa, sendo necessária a utilização de um sistema de coordenadas intermédio conhecido como ECEF (Earth Centered Earth Fixed). Este sistema tem a sua origem no centro da terra.

Para além do interface com o GPS, este módulo também é responsável pela conversão entre os vários tipos de coordenadas, publicando para o sistema apenas coordenadas XYZ já em ENU. Para o efeito foram implementados os seguintes métodos de conversão:

6.3.1 Conversão LLH para ECEF

$$\begin{aligned}
 X &= \left(\frac{a}{\chi} + h \right) \cos \phi \cos \lambda \\
 Y &= \left(\frac{a}{\chi} + h \right) \cos \phi \sin \lambda \\
 e &= \sqrt{1 - \left(\frac{b}{a} \right)^2} \\
 Z &= \left(\frac{a(1 - e^2)}{\chi} + h \right) \sin \phi \\
 \chi &= \sqrt{1 - e^2 \sin^2 \phi}
 \end{aligned} \tag{6.1}$$

Os valores “a” e “b” são o semieixo maior e menor do elipsóide. Para a Terra estes valores são iguais a 6378137.0000[m] e 6356752.3142[m] respectivamente.

6.3.2 Conversão de ECEF para ENU

Para converter entre os sistemas de coordenadas ECEF e ENU é necessário um ponto de referência (X_r, Y_r, Z_r) em ECEF que se tornará o ponto de coordenadas (0,0,0) no sistema de coordenadas ENU. Desta forma, para converter o ponto genérico (X_p, Y_p, Z_p) em ECEF para o sistema de coordenadas local, aplica-se a seguinte transformação:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\sin Y_r & \cos Y_r & 0 \\ -\sin X_r \cos Y_r & -\sin X_r \sin Y_r & \cos X_r \\ \cos X_r \cos Y_r & \cos X_r \sin Y_r & \sin X_r \end{bmatrix} \cdot \begin{bmatrix} X_p - X_r \\ Y_p - Y_r \\ Z_p - Z_r \end{bmatrix} \tag{6.2}$$

6.4 Nó de Log do sistema loggerISR

Uma componente importante de qualquer sistema que se encontre em desenvolvimento é a capacidade de fazer registo de todas as operações realizadas para posterior análise.

Com este objectivo foi criado um nó destinado a “subscrever” todos os tópicos que circulam no ROS e armazená-los no disco rígido.

Posteriormente é possível importar estes registos através do Matlab e fazer diversos tipos de análise que seriam impossíveis de fazer em tempo real.

Capítulo 7

Contribuições no âmbito do projecto ISRobotCar

7.1 Sincronização de aquisição sensorial

A plataforma ISRobotCar possui uma rede CAN para transferência de dados e comandos entre os seus módulos.

Todos os pacotes transmitidos na rede seguem uma estrutura própria. O campo de ID do protocolo CAN é sub dividido para criar um identificador do emissor, do destinatário e da função da mensagem.

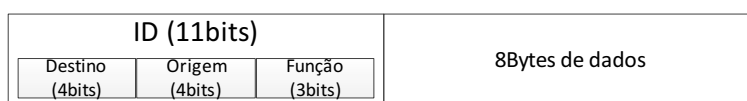


Figura 7.1: Estrutura de um frame CAN

Os seguintes identificadores estão actualmente atribuídos na rede CAN:

ID	Módulo	ID	Módulo
0	PC/WebSIM	8	...
1	MainBoard	9	...
2	Relé	10	...
3	Embraiagem	11	...
4	Gestão de Energia	12	...
5	Steering	13	...
6	Sinalética	14	...
7	...	15	Trigger

Integrado na rede, existe um nó que tem como função servir de Master e controlar o fluxo de dados. A rede CAN implementada, assenta sobre um sistema de “timeslots”. A cada milisegundo, o nó master transmite um pacote de dados (trigger) informando qual o módulo (ou grupos de módulos) que pode transmitir durante essa janela temporal.



Figura 7.2: Estrutura de mensagem de trigger

CAPÍTULO 7. CONTRIBUIÇÕES NO ÂMBITO DO PROJECTO ISROBOTCAR

Por sua vez todos os módulos reconhecem esta mensagem e se pertencerem aquela janela temporal podem transmitir.

Vários módulos fazem aquisição de informação de diversos sensores necessária ao processo de fusão sensorial.

Até ao início desta dissertação a aquisição e transmissão era feita de forma assíncrona obedecendo apenas aos timeslots impostos pela rede.

Este facto faz com que informação sensorial respeitante à mesma grandeza (por exemplo odometria e GPS) seja adquirida em instantes diferentes criando erros durante o processo de fusão. Foi necessária a implementação de um sistema de sincronização para os diversos módulos.

7.1.1 Sincronização CAN e GPS

A tecnologia usada no sistema GPS requer uma grande precisão nas temporizações dos seus sinais. Os relógios presentes nos GPS comerciais tenham uma base temporal precisa (sincronizada pelos satélites) e comum entre dispositivos.

São vários os receptores que possuem uma saída (onda quadrada) para possibilitar a sincronização ou a criação de uma base temporal com outros dispositivos.

O dispositivo ublox possui esta capacidade e foi utilizado para a sincronização da rede CAN. Se houver cobertura de GPS o dispositivo produz uma onda quadrada com período de 200 milissegundos. Por sua vez, o nó master, recebe esta onda e sincroniza o seu mecanismo de geração de triggers temporais. Caso não haja cobertura GPS o nó master continua a providenciar triggers ao sistema até nova sincronização.

Como a base temporal entre dispositivos GPS é comum, a rede CAN passa também a estar sincronizada com a aquisição do GPS diferencial existente na plataforma.

7.1.2 Sincronização entre módulos CAN

Apesar da rede CAN possuir agora um mecanismo para sincronização absoluta, os restantes módulos continuam a fazer as aquisições de forma assíncrona. Para contornar este problema foi adicionado à mensagem trigger outro campo. O período de amostragem escolhido para o funcionamento da plataforma são 100 milissegundos, assim sendo, foi adicionado um contador de 1 a 100 criando uma base temporal absoluta dentro da rede CAN.

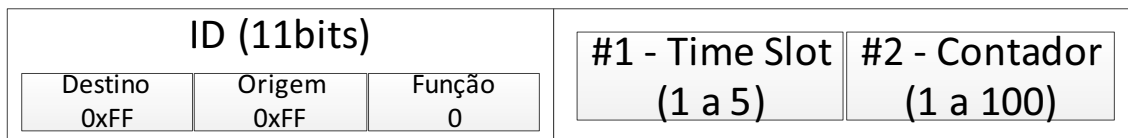


Figura 7.3: Estrutura de mensagem de trigger alterada

Os módulos que necessitam de fazer aquisição sensorial são sincronizados entre si da seguinte forma. Cada vez que o nó master envia uma mensagem em que o contador é igual a 100 todos os módulos adquirem.

CAPÍTULO 7. CONTRIBUIÇÕES NO ÂMBITO DO PROJECTO ISROBOTCAR

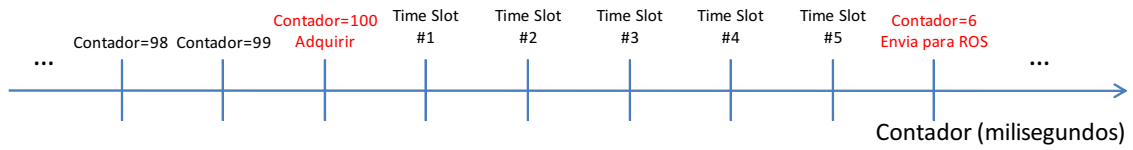


Figura 7.4: Diagrama temporal de aquisição

Durante os 5 milisegundos seguintes (percorre todos os timeslots) transmitem para a rede CAN e quando o contador for igual a 6 o módulo de controlo da direcção envia a informação sensorial para o ROS.

Capítulo 8

Validação dos resultados

De forma a validar o trabalho realizado durante a dissertação, neste capítulo serão apresentados os resultados do funcionamento do sistema de controlo da direcção bem como os resultados de um percurso realizado autonomamente pela plataforma ISRobotCar.

8.1 Sistema de Controlo da direcção

De forma a demonstrar o correcto funcionamento do sistema de controlo da direcção implementado, nesta secção são apresentadas duas figuras com dados reais do controlador em funcionamento.

A primeira destas figuras representa o posicionamento da direcção num determinado valor de referência (em pulsos de encoder).

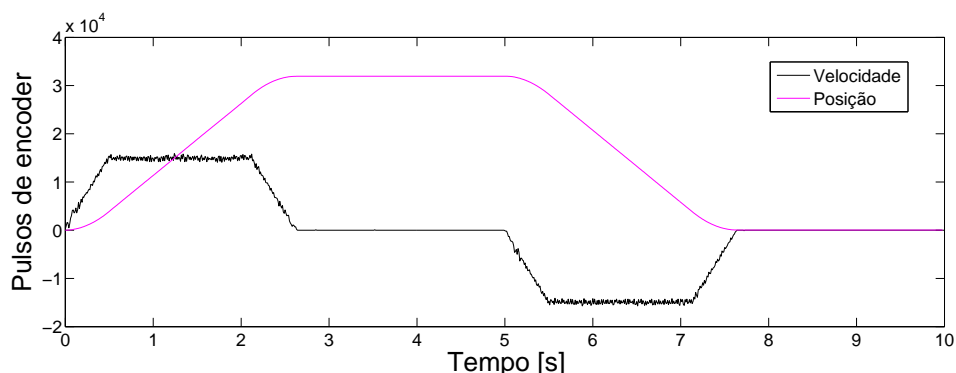


Figura 8.1: Posicionamento nos 32000 pulsos de encoder (22°)

Na figura anterior é possível observar a evolução da posição (a rosa) bem como da velocidade (a preto). O trajecto programado consistia em posicionar o sistema de direcção nos 32000 pulsos de encoder (aproximadamente 22 graus) e seguidamente voltar à posição inicial (0 graus).

Ao olhar para a curva da velocidade é possível identificar os perfis trapezoidais impostos pelo controlador. Nestes perfis é facilmente identificável a fase de aceleração e desaceleração bem como de velocidade constante.

O perfil de velocidade apresenta algum ruído, proveniente na sua maioria da interacção dos pneus da plataforma com o piso.

É necessário fazer referência ao facto de que estas curvas foram obtidas com a plataforma imóvel. Esta situação é a mais desfavorável, exigindo mais esforço por parte do

controlador.

A segunda figura representa a evolução do sistema de controlo da direcção durante um período de navegação autónoma da plataforma:

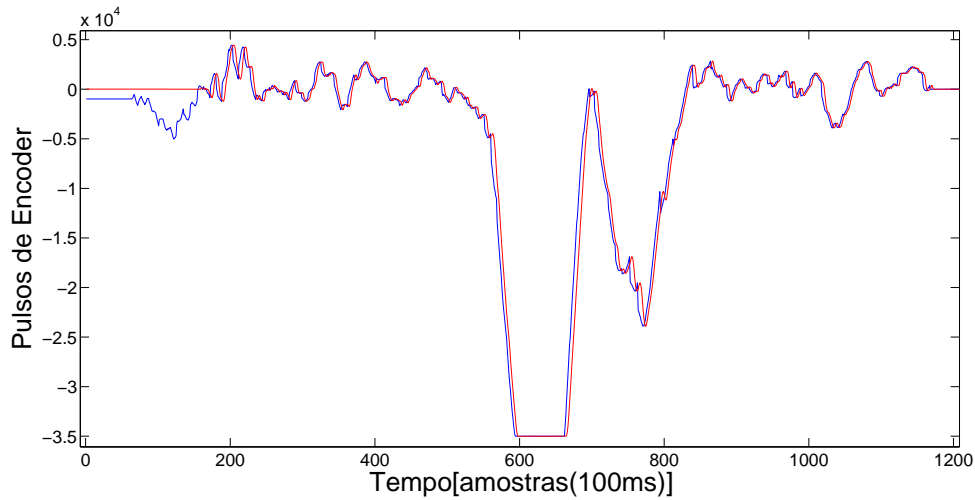


Figura 8.2: Resposta de posicionamento durante navegação autónoma (ref.-azul/real-vermelho)

O gráfico a azul representa os comandos de posicionamento criados pelo controlador Kanayama e cinemática inversa durante o percurso. Desfasada de 100 milisegundos (tempo de amostragem da plataforma) é representada a resposta do sistema de controlo da direcção. Verifica-se que o sistema consegue seguir eficazmente os comandos de posicionamento enviados.

No gráfico, até aproximadamente aos 20 segundos, o sistema de direcção não segue os comandos enviados, devido ao facto da chave da plataforma não se encontrar na posição AUTO e portanto a ponte-H se encontrar desactivada.

Como é possível verificar, em ambas as situações o controlador lateral desempenha correctamente a tarefa para a qual foi projectado.

8.2 Navegação Autónoma

Esta secção pretende mostrar os resultados obtidos durante um período de navegação autónoma.

Na imagem abaixo é possível observar a branco o percurso desejado, a vermelho o resultado da simulação utilizando o controlador Kanayama e a verde o percurso efectivamente realizado utilizando a plataforma ISRobotCar.

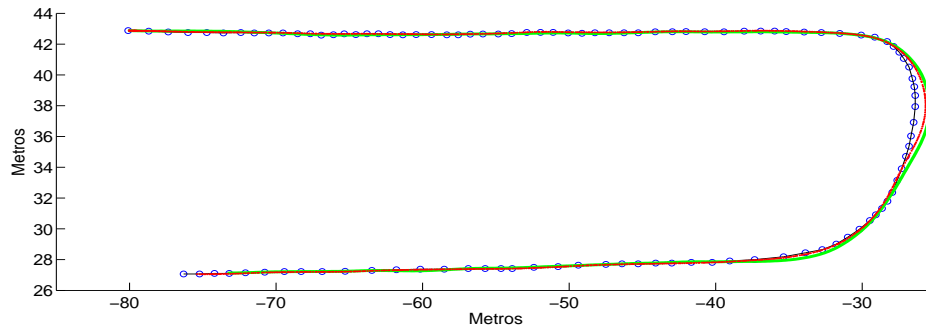


Figura 8.3: Trajetória desejada, simulada e real

Os resultados obtidos através da simulação e do teste real são bastante semelhantes e seguem correctamente a trajectória desejada. Estes resultados podem ainda ser melhorados através de um tuning mais fino aos ganhos do controlador kanayama.

Os mesmos resultados podem ser observados utilizando o Google Earth.



Figura 8.4: Sobreposição de resultados no Google Earth

Como anteriormente o traço branco representa a trajectória desejada, o vermelho a simulada e o verde a real.

Na figura abaixo é apresentada uma sequência de fotografias capturadas durante a execução do percurso.



Figura 8.5: SnapShots da execução do percurso

Pelos resultados apresentados, é possível verificar o correcto funcionamento do sistema de controlo da direcção, dos sistemas de simulação, bem como dos diversos constituintes da plataforma ISRobotCar em aplicações como os CyberCar.

Capítulo 9

Conclusões e Trabalho Futuro

O objectivo principal proposto para o trabalho a realizar durante esta dissertação era o desenvolvimento de um sistema de controlo da direcção para a plataforma ISRobotCar. Este projecto faz parte de um objectivo maior que é dotar a plataforma da capacidade de navegar autonomamente.

Pela observação dos resultados obtidos é possível verificar que o sistema desenvolvido cumpre de forma correcta os objectivos pretendidos. A plataforma ISRobotCar passa assim a conter um sistema de controlo da direcção, software de interface com a plataforma e GPS bem como novos métodos de sincronismo para aquisição sensorial.

Durante esta dissertação e juntamente com todo o trabalho da restante equipa do ISRobotCar, foi possível pela primeira vez, a navegação autónoma no dia 24 de Junho de 2011.

9.1 Trabalho Futuro

Como sugestão para trabalho futuro:

1. Nova forma de comunicação entre o ROS e a plataforma.

Bibliografia

- [1] Polícia de Segurança Pública. “ Actividade Operacional de Trânsito - Ano 2011”. Disponível em http://www.psp.pt/Pages/Transito/Transito_2011.aspx Visitado em 18 de Agosto de 2011.
- [2] Carsten and Cacciabue, 2007 Carsten, O. and Cacciabue, P. (2007). Modelling Driver Behaviour in Automotive Environments, chapter From driver models to modelling the driver: what do we really need to know about the driver?
- [3] Nunney, Malcolm James (2006). Light and Heavy Vehicle Technology. Elsevier Science. p. 521. ISBN 9780750680370.
- [4] Howe, Hartley E. (February 1956). “Mr. Power Steerings Ship Comes In” . Popular Science 168 (2)
- [5] Lamm, Michael (March 1999). “75 years of Chryslers”. Popular Mechanics 176 (3): 75. Retrieved 18 June 2010
- [6] Watson, Bill (22 March 2006). “History of Power Steering”. Imperial Automobile Club Archives. Retrieved 18 June 2010.
- [7] Oliveira, Luciano; Nunes, Urbano; Peixoto, Paulo; Silva, Marco and Moita, Fernando. “Semantic Fusion of Laser and Vision in Pedestrian Detection”, In: Pattern Recognition, pp. 3648-3659, vol. 43, issue 10, 2010.
- [8] C. Premebida; O. Ludwig; M. Silva and U. Nunes; “A Cascade Classifier applied in Pedestrian Detection using Laser and Image-based Features”, In: Proc. of the IEEE ITSC 2010, Madeira, Portugal.
- [9] Lopes, A.C.; Moita, F.; Nunes, U.; Solea, R.; “An outdoor guidepath navigation system for AMRs based on robust detection of magnetic markers”, In: Emerging Technologies and Factory Automation, 2007. ETFA.
- [10] Kanayama, Y.; Kimura, Y.; Miyazaki, F.; Noguchi, T.; “A stable tracking control method for an autonomous mobile robot”, In: Robotics and Automation, 1990. Proceedings.
- [11] Urbano Nunes and L. Conde Bento 2007. “Data Fusion and path-following controllers comparison for autonomous vehicles”, In Nonlinear dynamics Journal, Springer-Verlag, Vol.49, pp.445-462, September 2007.
- [12] Razvan Solea and Urbano Nunes 2007. “Trajectory planning and sliding-mode control based trajectory-tracking for cybercars”, Integrated Computer-aided Engineering, Int. Journal, IOS Press, vol.14, n.1, pp.33-47.

- [13] Feynman, Richard Phillips; Leighton, Robert B.; Sands, Matthew L. 2006. "The Feynman lectures on physics 3 vol." . Pearson / Addison-Wesley. ISBN 0-8053-9047-2: volume 2.