

Faculdade de Engenharia da Universidade do Porto



FEUP

Implementação de um controlador de tracção num veículo eléctrico

Nelson Lima da Costa

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Automação

Orientador: Prof. Dr. Rui Esteves Araújo

Julho de 2009

Resumo

Nesta dissertação é feita uma revisão sobre o modelo longitudinal de um veículo terrestre, são apresentados alguns modelos empíricos e analíticos utilizados para representar a característica de fricção entre pneu/estrada, e é feito um estudo sobre o princípio de funcionamento de alguns controladores de tracção aplicados em veículos eléctricos, nomeadamente o Model Following Control (MFC) e o Slip Ratio Control (SRC). De seguida apresentam-se as fases de projecto, desenvolvimento e implementação dum sistema *DataLogger* com recurso a uma placa sbRIO-9631, devidamente caracterizada. Na fase de projecto são resumidas as principais características deste tipo de sistemas, são traçados os requisitos do sistema a desenvolver e apresentam-se de forma comparativa, algumas alternativas para o armazenamento de dados. Na fase de desenvolvimento são descritas as aplicações desenvolvidas em *Labview*, e em particular são apresentadas as vantagens e desvantagens das aplicações de leitura/escrita em ficheiros com formato .bin e .tdms. As aplicações de leitura de ficheiros têm como função ler os dados armazenados pelo *DataLogger* e fazer uma representação gráfica para uma análise rápida dos dados. Contudo é importante que haja a possibilidade destes dados serem analisados noutra tipo de ferramentas, além do *Labview*, e por isso é apresentada a forma utilizada para exportar os dados para o *Matlab*, que representa uma das ferramentas mais importantes no tratamento e análise de dados. Na implementação é mostrado o resultado final do sistema *DataLogger*.

Outra parte do trabalho é o projecto e implementação de um controlador de tracção. Na fase de projecto é determinado o modelo dos vários subsistemas que constituem o veículo uCar, são apresentados métodos experimentais que podem ser utilizados para se obter os parâmetros desconhecidos do veículo, e é formulado um controlador de tracção para o sistema uCar, com alguns resultados de simulação.

Por fim são apresentados os resultados experimentais, analisando a autonomia do veículo eléctrico uCar e o desempenho do controlador de tracção.

Abstract

In this dissertation a review of the longitudinal model of a land vehicle is done, and also are presented some empirical and analytical models used to represent the characteristic of friction between tire/road and a study about principle of operation of some traction controllers applied in electric vehicles, including the Model Following Control (MFC) and Slip Ratio Control (SRC). Afterwards, it is shown the stages of design, development and implementation of a *DataLogger* system using a sbRIO-9631 board, properly characterized. In the design stage, the main features of such systems are summarized, the requirements of the system to develop are outlined and are presented in a comparative way, several alternatives for data storage. In the development stage, the applications developed in *Labview* are described, in particular the advantages and disadvantages of read / write applications in format files .bin and .tdms are presented. The applications of reading files have the function of reading the data stored by the *DataLogger* and doing a graphical representation for rapid analysis of data. However it is important that there is the possibility of data being analyzed in another kind of tools, beyond the *Labview*, and therefore the form used to export the data to *Matlab* is presented, which represents one of the most important tools in the treatment and analysis of data. In the implementation stage the final result of *DataLogger* system is shown.

Another part of the work is the design and implementation of a traction controller. In the project stage, a model of the various subsystems which constitute the vehicle uCar is determined, experimental methods that can be used to obtain the unknown parameters of the vehicle are presented, and a traction controller for the uCar system is made, with some simulation results.

Finally, the experimental results are presented, analyzing the autonomy of electrical vehicle uCar and the performance of the traction controller.

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus pais pelo apoio e motivação, que me deram durante todo o meu percurso académico, e pela oportunidade de seguir em frente, tornando tudo isto possível.

De seguida agradeço ao meu orientador, Prof. Dr. Rui Araújo, pelo apoio, crítica e sugestões no desenvolvimento deste trabalho.

Por fim, não poderia deixar de agradecer ao Eng. Ricardo Castro, por toda a disponibilidade, apoio e boa vontade na partilha dos seus conhecimentos.

Nelson Costa

Índice

Resumo	i
Abstract.....	iii
Índice.....	vii
Lista de Figuras	xi
Lista de Tabelas	xv
Capítulo 1	1
Introdução.....	1
1.1 - Objectivos do trabalho.....	1
1.2 - Âmbito do trabalho.....	2
1.3 - Estrutura do documento.....	2
Capítulo 2	3
Revisão bibliográfica	3
2.1 - Introdução.....	3
2.2 - Modelo longitudinal de um veículo de estrada	3
2.3 - Modelo longitudinal de fricção Pneu/estrada	5
2.3.1 - <i>Piecewise Linear Model</i>	8
2.3.2 - <i>Burckhardt Model</i>	8
2.3.3 - <i>Rill Model</i>	8
2.3.4 - <i>Magic Formula</i>	9
2.3.5 - <i>Dahl Model</i>	9
2.4 - Controladores de tracção de veículos eléctricos.....	9
2.4.1 - <i>Model Following Control (MFC)</i>	10
2.4.2 - <i>Slip Ratio Control (SRC)</i>	11
2.5 - Conclusões	15
Capítulo 3	17
Projecto de um sistema <i>DataLogger</i>	17
3.1 - Introdução.....	17
3.2 - O que é, e qual a importância deste tipo de sistemas	17
3.3 - Caracterização de sistemas de aquisição de dados	18
3.4 - Caracterização da placa SbRIO-9631	21
3.4.1 - <i>Hardware</i>	22
3.4.2 - <i>Software</i>	22
3.5 - Requisitos	22
3.6 - Alternativas para armazenamento de dados.....	25

3.7 - Conclusões	27
Capítulo 4	29
Desenvolvimento do sistema <i>DataLogger</i>	29
4.1 - Introdução	29
4.2 - Sistema de aquisição de dados (HW)	30
4.2.1 - Variáveis a adquirir e caracterização dos sensores	30
4.2.2 - Projecto do circuito de condicionamento de sinal	32
4.2.2.1 - Sinais Digitais	32
4.2.2.2 - Sinais Analógicos	35
4.2.3 - Concepção do circuito de condicionamento em PCB	43
4.2.4 - Implementação do sistema de aquisição de dados	43
4.3 - Algoritmo do sistema <i>DataLogger</i> (SW)	45
4.3.1 - Aplicação de aquisição dos sinais (FPGA)	46
4.3.1.1 - Sinais Digitais dos <i>encoders</i>	46
4.3.1.2 - Sinais Analógicos	47
4.3.2 - Aplicação de processamento e armazenamento de dados (RTU)	47
4.3.3 - Aplicações de leitura de ficheiros e exportação de dados para <i>Matlab</i>	51
4.3.3.1 - Leitura de ficheiro <i>.bin</i>	51
4.3.3.2 - Leitura de ficheiro <i>.tdms</i>	51
4.3.3.3 - Exportação dos dados para o <i>Matlab</i>	52
4.4 - Implementação do <i>DataLogger</i>	53
4.5 - Conclusões	54
Capítulo 5	57
Projecto e implementação de um controlador de tracção	57
5.1 - Introdução	57
5.2 - Modelo do sistema uCar em <i>Simulink</i>	57
5.2.1 - Modelo longitudinal do veículo	57
5.2.2 - Modelo <i>Throttle</i> /inversor/motor	58
5.2.3 - Modelo motor/redutor/roda	58
5.2.4 - Modelo pneu/estrada	59
5.2.5 - Diagrama de blocos do sistema uCar	60
5.3 - Determinação dos parâmetros do veículo	60
5.3.1 - Determinação de K_t e τ	61
5.3.2 - Cálculo experimental de J_{eq} e B_{eq}	62
5.3.2.1 - Método em aceleração	62
5.3.2.2 - Métodos em desaceleração	62
5.4 - Resultados experimentais dos parâmetros do uCar	63
5.5 - Projecto do controlador de tracção	64
5.6 - Resultados de simulação do sistema em <i>Simulink</i>	66
5.6.1 - Comportamento do sistema sem controlo de tracção	67
5.6.2 - Comportamento do sistema com controlo de tracção	69
5.7 - Implementação do controlador de tracção	72
5.8 - Conclusões	72
Capítulo 6	75
Resultados experimentais	75
6.1 - Apresentação do veículo eléctrico uCar	75
6.2 - Caracterização do desempenho do uCar como veículo eléctrico	76
6.2.1 - Resultados gerais	76
6.2.2 - Aceleração	77
6.2.3 - Frenagem	78
6.2.4 - Potência	80
6.2.5 - Autonomia	81
6.3 - Comportamento do veículo em piso de reduzida aderência	82
6.3.1 - Sem controlo de tracção	83
6.3.2 - Com controlo de tracção	84

6.4 - Conclusões	89
Capítulo 7	91
Conclusões e trabalho futuro	91
7.1 - Conclusões	91
7.2 - Sugestões para trabalho futuro.....	91
Anexo A Circuito de condicionamento em PCB	93
<i>Multisim</i>	93
<i>Ultiboard</i>	97
Anexo B Aplicações em LabView	102
Projecto	102
Aquisição dos sinais	103
Processamento e armazenamento de dados.....	104
Leitura de ficheiros	105
Controlador de tracção.....	106
Referências	107

Lista de Figuras

Figura 2.1 - Modelo longitudinal do veículo [1]	4
Figura 2.2 - Perfis típicos de fricção pneu/estrada: veículo em diferentes condições de estrada, com velocidade de 20mph [2].	6
Figura 2.3 - Perfis típicos de fricção pneu/estrada: veículo no asfalto seco, com velocidades variadas [2].	6
Figura 2.4 - Diagrama de fricção e deslizamento [2].....	7
Figura 2.5 - Diagrama de blocos do MFC [4]	10
Figura 2.6 - Diagrama de blocos do SRC [4]	11
Figura 2.7 - Modelo do veículo [4]	12
Figura 2.8 - Razão de deslizamento nominal é dada por $a = 1$ [4].	13
Figura 2.9 - Controlador PI [4].....	13
Figura 2.10 - Lugar das raízes com variação dos parâmetros [4].....	14
Figura 3.1 - Diagrama de blocos da placa sbRIO-9631 [10]	21
Figura 3.2 - Placa sbRIO-9631 [10].....	21
Figura 3.3 - Arquitectura do sistema de aquisição de dados.....	23
Figura 3.4 - Diferentes soluções para o armazenamento de dados	25
Figura 4.1 - Princípio de funcionamento do LEM.....	31
Figura 4.2 - Característica do <i>Thermistor</i> NTC	32
Figura 4.3 - Sinal digital ideal	33
Figura 4.4 - Sinal digital com características reais	33
Figura 4.5 - Mudanças de estado múltiplas a um nível de <i>threshold</i> VT	33
Figura 4.6 - Curva ideal de histerese do <i>Schmitt trigger Buffer</i>	34
Figura 4.7 - Aplicação duma entrada com ruído a um Schmitt trigger Buffer	34

Figura 4.8 - Relação entrada/saída do 74HC14.....	35
Figura 4.9 - Circuito de condicionamento dos sinais do <i>encoder</i>	35
Figura 4.10 - Circuito de condicionamento do LEM.....	36
Figura 4.11 - Circuito de condicionamento da tensão da bateria.....	37
Figura 4.12 - Circuito de condicionamento do <i>Throttle</i>	37
Figura 4.13 - Circuito de condicionamento do <i>Thermistor</i> (1º método).....	38
Figura 4.14 - Aproximação linear de $V_{out}=f(T)$ (1º método)	40
Figura 4.15 - Circuito de condicionamento do <i>Thermistor</i> (2º método).....	41
Figura 4.16 - Aproximação linear de $V_{out}=f(T)$ (2º método)	42
Figura 4.17 - Integração do HW do Sistema de aquisição de dados.....	44
Figura 4.18 - Diagrama de blocos do sistema <i>DataLogger</i>	45
Figura 4.19 - Bloco FPGA I/O Node, Sinais digitais	47
Figura 4.20 - Bloco FPGA I/O Node, Sinais analógicos	47
Figura 4.21 - Excerto da tarefa de armazenamento de dados num ficheiro .tdms.	49
Figura 4.22 - Excerto da parte inicial da tarefa de armazenamento de dados num ficheiro .tdms, construção do cabeçalho.....	50
Figura 4.23 - Excerto da tarefa de armazenamento de dados num ficheiro .bin.	50
Figura 4.24 - Excerto da aplicação de leitura de ficheiros .bin	51
Figura 4.25 - Bloco <i>write to spreadsheet file</i>	52
Figura 4.26 - Integração do sistema <i>DataLogger</i>	53
Figura 4.27 - Configuração para armazenar os dados na sbRIO.....	54
Figura 5.1 - Acoplamento motor/roda	58
Figura 5.2 - Modelo pneu/estrada	59
Figura 5.3 - Diagrama de blocos do sistema	60
Figura 5.4 - Relação $T_m-V_{Throttle}$	61
Figura 5.5 - Arquitectura das camadas de controlo	65
Figura 5.6 - Característica das curvas $\mu-\lambda$ [3]	65
Figura 5.7 - Diagrama de blocos do controlador de tracção	66
Figura 5.8 - Curva $\mu-\lambda$ do piso com gelo utilizada na simulação	67
Figura 5.9 - Evolução das velocidades no arranque, sem controlo de tracção	68

Figura 5.10 - Evolução do deslizamento, sem controlo de tracção	68
Figura 5.11 - Evolução das velocidades no arranque, com controlo de tracção (ganho $K_{tcs}=1$)	69
Figura 5.12 - Efeito compensador no sinal do <i>Throttle</i> (ganho $K_{tcs}=1$)	70
Figura 5.13 - Evolução do deslizamento, com controlo de tracção (ganho $K_{tcs}=1$)	70
Figura 5.14 - Evolução do deslizamento para diferentes ganhos	71
Figura 5.15 - Evolução das velocidades no arranque, com controlo de tracção (ganho $K_{tcs}=10$)	72
Figura 6.1 - Veículo eléctrico uCar	75
Figura 6.2 - Percurso realizado (FEUP)	76
Figura 6.3 - Dados adquiridos (após filtragem digital)	77
Figura 6.4 - Veículo em aceleração	78
Figura 6.5 - Frenagem	79
Figura 6.6 - Tensão, Corrente e Potência nas baterias ao longo do teste	80
Figura 6.7 - Potências consumida e regenerada durante o teste	80
Figura 6.8 - Evolução das velocidades sem controlo de tracção (teste na subida)	83
Figura 6.9 - Evolução do deslizamento sem controlo de tracção (teste na subida)	84
Figura 6.10 - Evolução das velocidades com controlo de tracção (teste na subida, $K_{tcs}=0.5$) ..	85
Figura 6.11 - Evolução do deslizamento com controlo de tracção (teste na subida, $K_{tcs}=0.5$)	85
Figura 6.12 - Compensação do sinal de comando (teste na subida, $K_{tcs}=0.5$)	86
Figura 6.13 - Evolução da corrente das baterias, com controlo de tracção (teste na subida, $K_{tcs}=0.5$)	86
Figura 6.14 - Evolução das velocidades com controlo de tracção (teste na subida, $K_{tcs}=0.6$) ..	87
Figura 6.15 - Compensação do sinal de comando (teste na subida, $K_{tcs}=0.6$)	88
Figura 6.16 - Evolução da corrente das baterias, com controlo de tracção (teste na subida, $K_{tcs}=0.6$)	88

Lista de Tabelas

Tabela 3.1 - Comparação dos principais atributos dos sistemas de <i>DataLogger</i>	20
Tabela 3.2 - Características dos sinais.....	24
Tabela 3.3 - Estimativa do tempo máximo no armazenamento de dados	26
Tabela 4.1 - Sinais digitais e sensores.....	30
Tabela 4.2 - Variáveis analógicos e sensores	30
Tabela 4.3 - Características eléctricas do LEM.....	30
Tabela 4.4 - Valores nominais do NTC em diferentes temperaturas	32
Tabela 4.5 - Características do sinal do LEM para a Gama de corrente [-200,200] A	36
Tabela 4.6 - Temperaturas equidistantes escolhidas.....	39
Tabela 4.7 - Tensão de saída V_{out} para temperaturas 50-100°C (1ºmetodo).....	40
Tabela 4.8 - Tensão de saída V_{out} para temperaturas 50-100°C (2º método).....	42
Tabela 4.9 - Comparação dos resultados obtidos nos dois métodos	43
Tabela 4.10 - Características das tarefas.....	48
Tabela 5.1 - Parâmetros do modelo do sistema.....	61
Tabela 5.2 - Distribuição de peso no uCar	61
Tabela 5.3 - Valores por defeito atribuídos aos parâmetros desconhecidos	66
Tabela 5.4 - Coeficientes da <i>magic formula</i> (piso com gelo)	67
Tabela 6.1 - Características do uCar	75
Tabela 6.2 - Dados energéticos	81
Tabela 6.3 - Capacidade das baterias.....	81
Tabela 6.4 - Dados gerais.....	82

Abreviaturas e Símbolos

Lista de abreviaturas (ordenadas por ordem alfabética)

ABS	<i>Anti-lock Braking System</i>
DIO	<i>Digital input/output</i>
ESP	<i>Electronic Stability Control</i>
FPGA	<i>field-programmable gate array</i>
HW	<i>Hardware</i>
MFC	<i>Model Following Control</i>
NI	<i>National Instruments</i>
NTC	<i>Negative Temperature Coefficient</i>
PC	<i>Personal Computer</i>
SRC	<i>Slip Ratio Control</i>
SW	<i>Software</i>
TCS	<i>Traction Control System</i>
uCar	<i>microcar</i>

Lista de símbolos

A_f	Área frontal
a	Aceleração
C_D	Coeficiente de resistência aerodinâmica
F_f	Força de tracção no eixo dianteiro
F_r	Força de tracção no eixo traseiro
F_t	Força de tracção total
g	Aceleração da gravidade

h	Altura do centro de gravidade do veículo
h_a	Altura do ponto de aplicação da resistência aerodinâmica acima do nível da terra
h_d	Altura do <i>carga rebocada</i>
m	Massa do veículo
R_a	Resistência aerodinâmica
R_d	Carga rebocada
R_g	Resistência do peso no plano inclinado (<i>grade resistance</i>)
R_r	Resistência rolante
R_{rf}	Resistência rolante dos pneus dianteiros
R_{rr}	Resistência rolante dos pneus traseiros
V	Velocidade linear
W	Carga normal, peso
W_f	Carga no eixo dianteiro do veículo
W_r	Carga no eixo traseiro do veículo
θ_s	Ângulo de inclinação
μ	Coefficiente de fricção
μ_p	Valor de pico do coeficiente de fricção
μ_s	Valor do coeficiente de fricção de deslizamento
ρ	Densidade do ar
ω	Velocidade angular do veículo
ω_ω	Velocidade angular da roda

Capítulo 1

Introdução

Recentemente, muitos veículos eléctricos têm sido desenvolvidos, principalmente para resolver problemas a nível ambiental e energético, causados pelo uso de motores de combustão interna. Além das vantagens já enunciadas, estes veículos possuem características interessantes, que se encontram directamente ligadas ao controlo dos motores eléctricos. Estes motores, permitem uma maior precisão e rapidez no controlo do binário gerado, em comparação com os motores de combustão interna. Neste contexto, espera-se que as funcionalidades dos sistemas tradicionais (ABS, ESP, TCS, etc.), que asseguram a estabilidade e segurança do veículo, sejam estendidas e melhoradas face às soluções actuais utilizadas em veículos com motor de combustão interna. Portanto, esta dissertação destina-se a estudar, desenvolver e implementar um algoritmo de controlo de tracção, para integrar na plataforma de propulsão do veículo eléctrico uCar.

1.1 - Objectivos do trabalho

Neste trabalho foram definidos os seguintes objectivos:

- Projecto, desenvolvimento e implementação de um sistema *DataLogger*;
- Estudo do modelo do veículo uCar;
- Determinação experimental dos parâmetros do modelo do veículo;
- Projecto, desenvolvimento e implementação de um controlador de tracção;
- Caracterização da autonomia do veículo eléctrico, a partir da análise de dados, armazenados pelo sistema *DataLogger*;
- Análise de resultados experimentais obtidos no teste do controlador de tracção.

1.2 - Âmbito do trabalho

Este trabalho insere-se na disciplina de Dissertação do Mestrado Integrado em Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto, no 2º semestre do ano lectivo de 2008/2009.

1.3 - Estrutura do documento

Este documento, além deste capítulo inicial, encontra-se dividido em mais seis capítulos.

O capítulo 2 pretende fazer uma revisão bibliográfica sobre o modelo longitudinal dum veículo terrestre, fala sobre alguns modelos que podem ser utilizados para representar o coeficiente de fricção da interacção pneu/estrada, e por fim é feito um estudo sobre alguns controladores de tracção utilizados em veículos eléctricos, nomeadamente o MFC e o SRC.

O capítulo 3 descreve a fase de projecto do sistema *DataLogger*, onde inicialmente são apresentadas as características principais de alguns sistemas deste tipo, de seguida é feita uma caracterização a nível de HW e SW da plataforma utilizada para desenvolver o sistema, e são traçados os requisitos do *DataLogger*. Um dos requisitos mais importantes neste tipo de sistemas é a forma como os dados são armazenados, por isso este capítulo fecha com uma apresentação sobre algumas alternativas no armazenamento de dados, com uma análise comparativa das vantagens e desvantagens de cada uma.

O capítulo 4 descreve a fase de desenvolvimento do sistema *DataLogger*, quer na componente de HW quer de SW. Na componente de HW é apresentado o projecto do circuito de condicionamento de sinal, e na de SW os algoritmos desenvolvidos para adquirir e armazenar os sinais. É ainda feita uma comparação das vantagens e desvantagens na utilização de diferentes aplicações de leitura de ficheiros, e expõe o método utilizado para exportar os dados duma plataforma para o *Matlab*, que é uma ferramenta importante a utilizar na análise de dados. No final apresenta-se o resultado do sistema *DataLogger* implementado.

O capítulo 5 apresenta a fase de projecto e implementação do controlador de tracção, começando por se determinar a dinâmica dos vários subsistemas que constituem o veículo, depois são indicadas alguns métodos experimentais que podem ser utilizados para determinar os parâmetros que se desconhecem. De seguida é feito o projecto do controlador e são apresentados os resultados de simulação do sistema.

No capítulo 6 são mostrados os resultados experimentais obtidos na fase de validação do sistema *DataLogger* e do controlador de tracção. Na validação do *DataLogger* foi feito um teste de estrada que permitiu fazer uma análise em termos de autonomia e desempenho do veículo eléctrico uCar. Em relação ao controlador, são apresentados resultados do sistema num piso com reduzida aderência em situações com e sem controlo, e é feita a discussão dos resultados obtidos.

Capítulo 2

Revisão bibliográfica

2.1 - Introdução

Neste capítulo é apresentado o modelo longitudinal dos veículos de estrada, através da equação geral do movimento, e algumas formas de representar os modelos de fricção entre pneu/estrada. A fricção é um efeito que resulta do contacto pneu/estrada e é uma variável importante no modelo global do veículo. Estes modelos foram importantes, na fase de projecto e simulação do controlador de tracção (Capítulo 5).

Por fim, é feita uma revisão de alguns métodos de controlo de tracção aplicados em veículos eléctricos, como o *model following control* (MFC) e o *slip ratio control* (SRC), com a finalidade de demonstrar a estrutura, o princípio de funcionamento e as variáveis principais que estão em jogo nestes tipos de controlo.

2.2 - Modelo longitudinal de um veículo de estrada

As principais forças externas que actuam no veículo são mostradas na Figura 2.1. Na direcção longitudinal, são incluídas a resistência aerodinâmica R_a , resistência rolante dos pneus dianteiros e traseiros R_{rf} e R_{rr} , carga rebocada R_d , resistência do peso R_g ($W \sin \theta_s$), e a força de tracção dos pneus dianteiros e traseiros F_f e F_r . Para um veículo com tracção traseira, $F_f = 0$, enquanto um veículo com tracção dianteira, $F_r = 0$.

A equação do movimento ao longo do eixo longitudinal x do veículo é dada por

$$m\ddot{x} = \frac{W}{g} \ddot{x} = F_f + F_r - R_a - R_{rf} - R_{rr} - R_d - R_g, \quad (2.1)$$

onde $\ddot{x}$ é a aceleração linear do veículo ao longo do eixo longitudinal, g é aceleração da gravidade, e m e W são a massa e o peso do veículo, respectivamente [1].

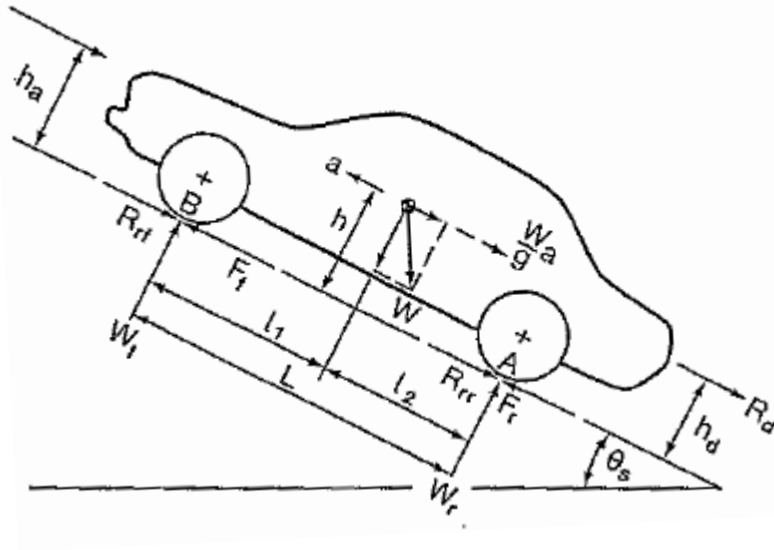


Figura 2.1 - Modelo longitudinal do veículo [1]

Por introdução do conceito da força de inércia, a equação (2.1) pode ser reescrita da seguinte forma

$$F_t = R_a + R_r + R_d + R_g + \ddot{x} \frac{W}{g} , \quad (2.2)$$

onde F_t é a força de tracção total e R_r é a resistência rolante total do veículo.

A resistência aerodinâmica R_a é uma força resistente que se opõe ao movimento do veículo, e é dada pela seguinte fórmula

$$R_a = \frac{\rho}{2} C_D A_f V_r^2 , \quad (2.3)$$

onde ρ é a densidade de massa do ar, C_D é o coeficiente de resistência aerodinâmica, A_f é a área característica do veículo, designada por área frontal, que é a área do veículo projectada na direcção do movimento, e V_r é a velocidade do veículo relativa ao vento [1].

As forças de tracção, F_f e F_r , dependem do coeficiente de fricção pneu/estrada μ , e da carga normal no eixo dianteiro W_f e traseiro W_r , ver Figura 2.1. As equações que permitem calcular essas forças são apresentadas a seguir

$$F_f = \mu W_f \quad (2.4)$$

e

$$F_r = \mu W_r , \quad (2.5)$$

onde μ é o coeficiente de fricção longitudinal, cujo valor depende das condições de contacto pneu/estrada. Esta grandeza é apresentada na secção seguinte, onde são indicados os factores que a influenciam, e as formas como esta pode ser modelizada.

2.3 - Modelo longitudinal de fricção Pneu/estrada

O próprio modelo de fricção pneu/estrada é essencial para descrever a dinâmica global do veículo, para simulação, análise, ou propósitos de controlo, uma vez que o movimento do veículo é principalmente determinado pelas forças de fricção (ou de tracção) transferidas da estrada para o veículo através dos pneus [2].

A tarefa principal de modelação do coeficiente de fricção pneu/estrada, é determinar a relação que existe entre as forças de fricção e as razões de deslizamento. Contudo, o fenómeno de fricção pneu/estrada é difícil de analisar por certas razões. Primeiro, o coeficiente de fricção é afectado por diferentes factores, incluindo, por exemplo, as condições de superfície pneu/estrada, pressão dos pneus, velocidade do veículo, etc. Segundo, as propriedades dinâmicas e não lineares da fricção pneu/estrada, i.e o fenómeno de histerese e viscosidade, são difíceis de descrever e analisar. Por isso, ao longo dos anos, têm sido desenvolvidos vários modelos que pretendem descrever o comportamento do coeficiente de fricção pneu/estrada. Alguns destes modelos, como *Piecewise Linear Model*, *Burckhardt Model*, *Rill Model*, *Magic Formula* serão apresentados na parte final desta secção.

Na maioria dos modelos propostos, o coeficiente de fricção assume-se, que é determinado pelo deslizamento da roda em consideração com alguns outros parâmetros [2].

Supondo que F_x é usado para denotar uma força de fricção longitudinal e F_{xn} uma força normal ao pneu, o coeficiente de fricção longitudinal pneu/estrada é normalmente definido como

$$\mu = \frac{F_x}{F_{xn}}, \quad (2.6)$$

onde por comparação, F_x e F_{xn} pode corresponder às forças, F_f e W_f de (2.4), e também às forças F_r e W_r de (2.5), do modelo do veículo.

O deslizamento da roda λ , é normalmente definido como [3]

$$\lambda = \frac{\omega_w - \omega}{\max(\omega_w, \omega)}, \quad \max(\omega_w, \omega) \neq 0, \quad (2.7)$$

onde ω_w é a velocidade angular da roda, e ω a velocidade angular do veículo. A velocidade angular do veículo é dada pela seguinte fórmula

$$\omega = \frac{V}{R_w} \quad (2.8)$$

onde V é a velocidade longitudinal do veículo e R_w o raio da roda.

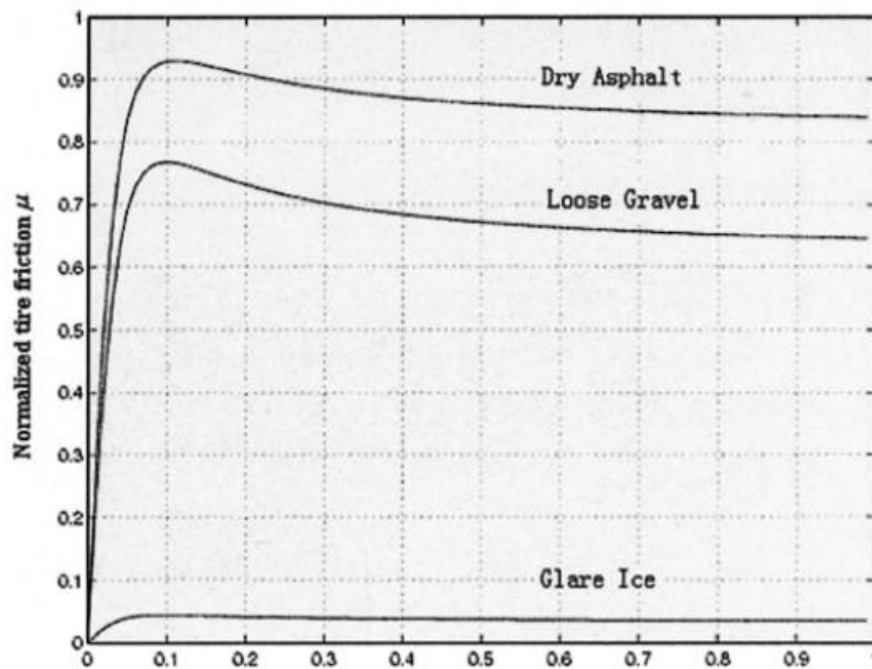


Figura 2.2 - Perfis típicos de fricção pneu/estrada: veículo em diferentes condições de estrada, com velocidade de 20mph [2].

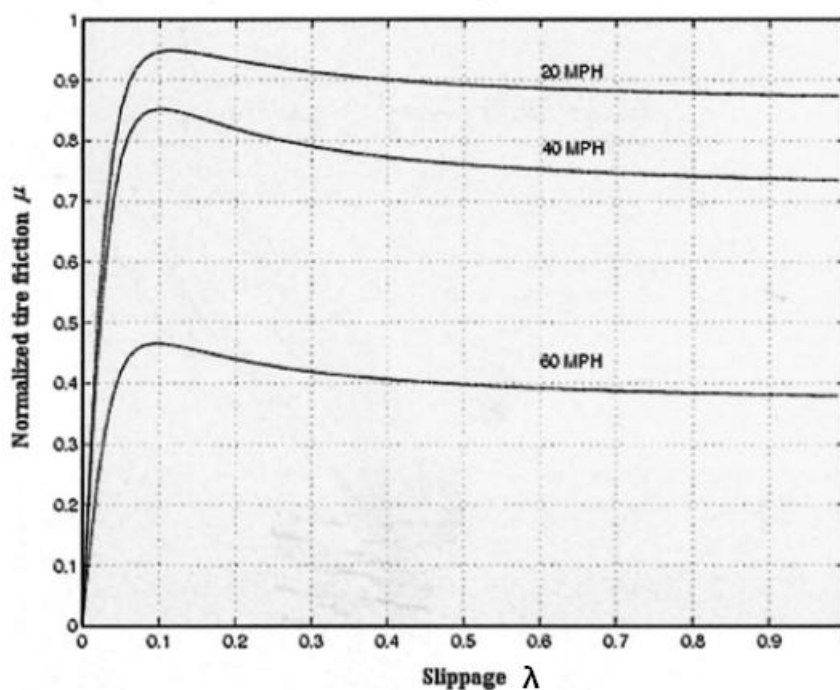


Figura 2.3 - Perfis típicos de fricção pneu/estrada: veículo no asfalto seco, com velocidades variadas [2].

Os gráficos, Figura 2.2 e Figura 2.3, demonstram a variação típica entre μ e λ , sob diferentes condições de estrada e diferentes velocidades do veículo, respectivamente. Como resultado da observação desses gráficos, conclui-se que μ normalmente aumenta, quando a velocidade do veículo decresce ou a superfície da estrada se torna mais áspera [2].

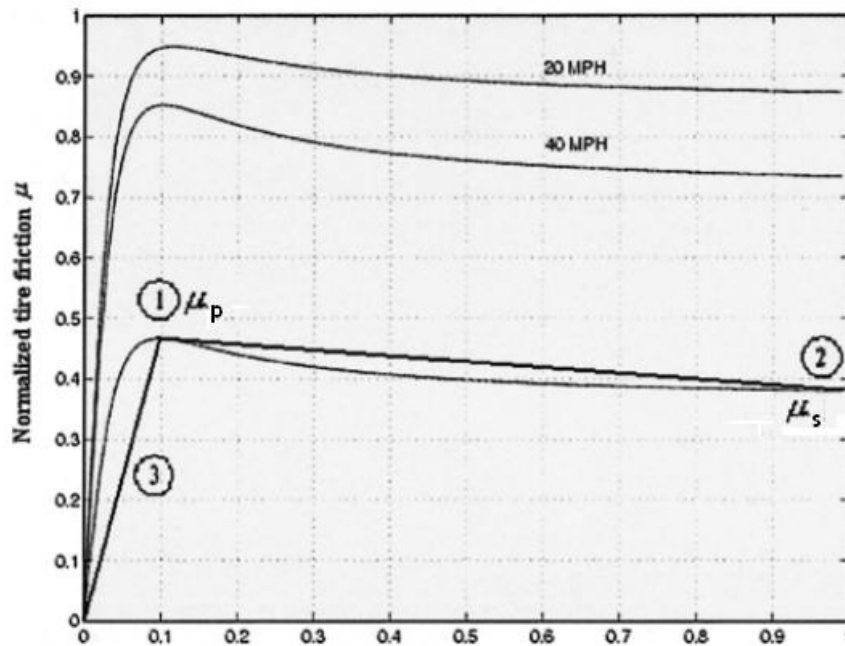


Figura 2.4 -Diagrama de fricção e deslizamento [2].

Como mostra a Figura 2.4, com as condições dadas, μ é uma função não linear de λ com um máximo, distinguido por μ_p (ponto 1, Figura 2.4), sendo o valor de pico do coeficiente de fricção. A relação de variação μ e λ pode ser claramente dividida em duas partes: a parte ascendente do gráfico μ - λ (3 na Figura 2.4) e a parte de deslizamento local, em que μ decresce gradualmente para μ_s (2 na Figura 2.4). Normalmente μ_s é mais pequeno que μ_p .

A seguir são apresentados as diferentes formas de representação de modelos do coeficiente de fricção longitudinal. Teoricamente estes modelos podem ser classificados em dois tipos: modelos empíricos (semi-empíricos) e modelos analíticos. Os modelos empíricos (semi-empíricos) são baseados em técnicas de *curve fitting* que, normalmente, formulam a curva μ - λ , numa função complexa de μ , principalmente em função de λ [2].

Os modelos empíricos (semi-empíricos) podem determinar com precisão as características do fenómeno de fricção pneu/estrada em regime permanente [2]. Contudo, não podem descrever vários comportamentos dinâmicos significativos, tal como a histerese. Por isso é que estes modelos também foram designados por modelos estáticos. Além disso, faltam-lhes interpretações físicas, e não reflectem directamente os efeitos de alguns factores tal como a humidade da estrada ou a pressão dos pneus. Assim, alguns modelos analíticos, especialmente o *brush model* tornou-se popular recentemente. Este modelo não será aqui apresentado com detalhe, por isso pode ser visto com mais pormenor em [2].

A maioria dos modelos analíticos usa equações diferenciais para descrever as propriedades de fricção pneu/estrada. No caso do *brush model*, de forma resumida, a ideia principal é ver o pneu como uma colecção de *elastic bristles* que tocam no plano da estrada e deflectem na direcção paralela à superfície da estrada, dando origem a um conjunto de forças que agem no

pneu, que finalmente conduzem a um conjunto de equações diferenciais, que podem ser usadas para interpretar a dinâmica do fenómeno pneu/estrada.

Dos modelos que são apresentados a seguir, os primeiros quatro são modelos empíricos, sendo o ultimo um modelo analítico. Mais modelos analíticos podem ser vistos com pormenor em [2].

2.3.1 - *Piecewise Linear Model*

Piecewise Linear Model é um modelo muito simples, onde a relação μ - λ é aproximada por uma função linear gradual que passa nos pontos μ_p e μ_s , ver Figura 2.4.

Apesar deste não poder capturar de forma precisa a característica dinâmica não linear de fricção pneu/estrada, este modelo recebe ainda atenções consideráveis devido à sua simplicidade.

2.3.2 - *Burckhardt Model*

Este é um modelo frequentemente mencionado, e originalmente é escrito como

$$\mu = (C_1(1 - e^{-C_2|\lambda|}) - C_3|\lambda|)e^{-C_4V} \quad (2.9)$$

onde C_i , $i=1, \dots, 4$ são os parâmetros que podem ser obtidos por dados experimentais. A parte e^{-C_4V} de (2.9), foi adicionada para representar a influência da velocidade.

Foi mostrado que este modelo pode descrever aproximadamente a curva μ - λ com os parâmetros C_i adequadamente escolhidos. Mas o custo de identificação dos parâmetros é relativamente alto [2]. Assim, várias formulas provenientes de (2.9), foram propostas. Por exemplo, uma delas é dada por

$$\mu = C_1 e^{-C_2|\lambda|} \cdot |\lambda|^{(C_3|\lambda| + C_4)} \cdot e^{-C_5 \cdot V}, \quad (2.10)$$

a qual pode ser manipulada para ter a seguinte forma

$$\ln \mu = C_1 - C_2|\lambda| + (C_3|\lambda| + C_4) \cdot \ln |\lambda| - C_5 V. \quad (2.11)$$

Como se pode observar, o modelo (2.11) é mais simples de ser identificado que o modelo original (2.9), pois os coeficiente C_i passaram a ser coeficientes lineares da equação.

2.3.3 - *Rill Model*

Este modelo é baseado nas características de força/binário do pneu, em regime permanente, junto com um simples modelo transitório de deflexão do pneu. Este calcula o deslizamento em regime permanente e a força correspondente com uma *curve fit* usando a inclinação inicial em $\lambda=0$, localização/magnitude de μ_p e μ_s como parâmetros [2].

2.3.4 - Magic Formula

Este método é originalmente definido como

$$\mu = C_1 \sin(C_2 \tan^{-1}(C_3 \lambda - C_4 (C_3 \lambda - \tan^{-1}(C_3 \lambda))))), \quad (2.12)$$

onde C_i , $i=1, \dots, 4$ são determinados por dados experimentais.

Este modelo tem recebido sucessivas transformações, e tornou-se o mais importante dos modelos de fricção semi-empíricos. Este pode ser aplicado para denotar aspectos tal como carga vertical e comportamentos transitórios [2].

Este modelo tem sido amplamente usado na simulação da fricção pneu/estrada e em projecto de sistemas de controlo. Contudo, está sobre parametrizada e é difícil de analisar.

2.3.5 - Dahl Model

Dahl Model é o primeiro modelo analítico de fricção pneu/estrada. *Dahl* desenvolveu um modelo simples que é uma generalização da fricção de *Coulomb*.

A histerese friccional durante o pré-deslizamento é aproximada por uma equação generalizada de primeira ordem. A equação proposta é dada por

$$\dot{F} = \sigma_0 \left(1 - \text{sgn}(v_r) \cdot \frac{F}{F_s} \right)^{\delta_d} v_r \quad (2.13)$$

onde σ_0 é a dureza inicial de contacto na inversão de velocidade, δ_d é o parâmetro do modelo que determina a forma da histerese, v_r é a velocidade do movimento relativo e F_s é a maior fricção em regime permanente [2].

2.4 - Controladores de tracção de veículos eléctricos

Como um dos objectivos deste trabalho é projectar e desenvolver um controlador de tracção para um veículo eléctrico, foi feita uma revisão bibliográfica sobre alguns controladores já desenvolvidos e implementados em veículos eléctricos, para perceber a estrutura dos algoritmos de controlo, e principalmente, para determinar quais as variáveis fundamentais no controlo de tracção. Desta forma, com base neste conhecimento podem surgir novas ideias para a formulação de um controlador, com um princípio de funcionamento diferente, como foi o caso neste trabalho (ver Capítulo 5).

Nesta secção são apresentados dois tipos de controlo de tracção, o *model following control* (MFC) e o *slip ratio control* (SRC), que foram desenvolvidos para um veículo eléctrico, com motores DC controlados em corrente. Estes controladores podem ser vistos com mais detalhe em [4], [5] e [6].

2.4.1 - Model Following Control (MFC)

O princípio de funcionamento do MFC encontra-se representado no diagrama de blocos da Figura 2.5. I_{com} é a corrente de comando proporcional ao ângulo do pedal do acelerador. ω_w é a velocidade angular da roda. Esta velocidade ω_w aumenta drasticamente quando o pneu desliza.

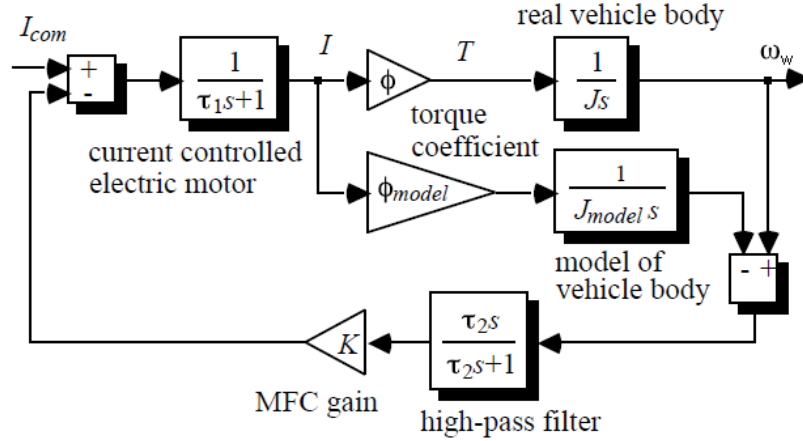


Figura 2.5 - Diagrama de blocos do MFC [4]

Embora a dinâmica do veículo, incluindo as características do pneu e a fricção da superfície de estrada, seja muito complicada, se introduzir a razão de deslizamento λ dado por (2.7), o corpo do veículo pode ser visto como um sistema de inércia, sendo o momento de inércia equivalente dado por

$$J = J_w + Mr^2(1 - \lambda) , \quad (2.14)$$

onde J_w , M , e r são o momento de inércia da roda, a massa do veículo, e o raio da roda respectivamente. A equação (2.14) significa que, quando o deslizamento ocorre, o veículo parece mais leve. Portanto, é usado o momento de inércia com $\lambda=0$ no modelo de referência, que é dado por

$$J_{model} = J_w + Mr^2 \quad (2.15)$$

Quando não há deslizamento, o actual J é quase igual ao J_{model} . Neste caso nenhum sinal é gerado pelo controlador MFC. Se o pneu deslizar, a velocidade actual da roda ω_w aumenta rapidamente, e como a velocidade do modelo de referência não aumenta, a diferença de velocidades contribui para uma diminuição do valor da corrente do motor através da malha de realimentação, logo o binário é reduzido rapidamente, permitindo que o pneu do veículo adira novamente à estrada.

Como a função de controlo é necessária só numa região, relativamente de alta frequência, é usado um filtro passa alto na malha de realimentação [4].

Resumindo, o controlo MFC actua no sistema quando há deslizamento da roda, compensando o sinal de comando de corrente que é enviado para o motor, diminuindo assim o deslizamento da roda. Este tipo de controlo possui o modelo de referência do veículo a correr em paralelo com o sistema físico, o veículo real, e por isso, apenas necessita da medida de velocidade da roda do sistema físico, já que a outra velocidade é gerada pelo modelo de referência.

Este controlo é simples mas é uma aproximação com pouco rigor, e por isso a seguir é apresentado o controlo SRC com uma aproximação mais precisa que permite manter o deslizamento numa zona desejada com maior exactidão.

2.4.2 - Slip Ratio Control (SRC)

A estrutura de controlo do SRC encontra-se representada no diagrama de blocos da Figura 2.6. Este controlador, em relação ao MFC, tem a vantagem de regular o deslizamento dentro de uma gama desejada [4].

O controlador é essencialmente constituído pelos blocos *slip ratio controller* e *road condition estimator*. O bloco *road condition estimator* tem como finalidade estimar a curva μ - λ , e determinar o deslizamento óptimo λ_{opt} . Alguns algoritmos de estimação de μ podem ser vistos com detalhe em [6].

O bloco *slip ratio controller* é um controlador de deslizamento, que recebe o sinal de referência λ_{opt} , e actua para que o sistema atinja o valor de referência λ_{opt} .

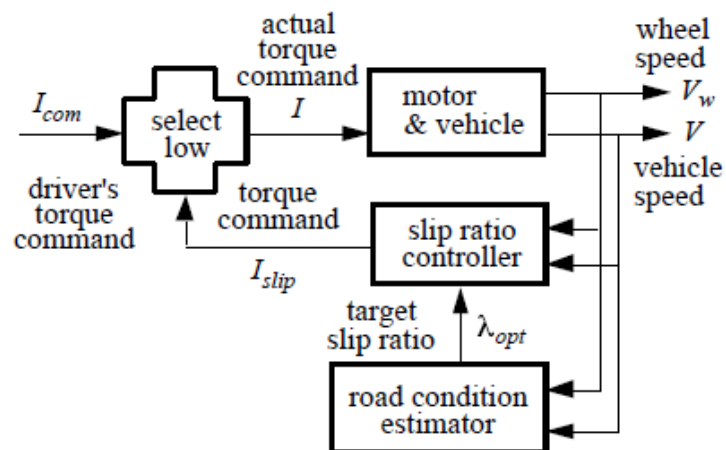


Figura 2.6 - Diagrama de blocos do SRC [4]

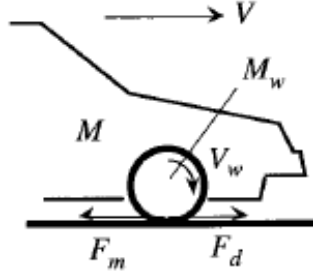


Figura 2.7 - Modelo do veículo [4]

O bloco *slip ratio controller* é projectado tendo em conta algumas considerações no modelo de veículo, feitas pelo autor em [4]. Assume-se que os dois binários motores e as forças de fricção nas duas rodas, esquerda e direita são iguais, e que as resistências do ar e rolantes são suficientemente pequenas. Desta forma, as equações da dinâmica da roda e do veículo da Figura 2.7 são dadas por

$$(F_m - F_d) \frac{1}{M_w} = \dot{V}_w \quad (2.16)$$

e

$$F_d \frac{1}{M} = \dot{V}, \quad (2.17)$$

onde F_m é o binário motor (força equivalente), F_d a força de fricção, M_w a inércia da roda (massa equivalente), e M a massa do veículo.

A força de fricção entre a roda e a estrada é dada por

$$F_d = \mu(\lambda)N \quad (2.18)$$

onde N é a força vertical dada por $N = Mg$.

Sabendo que em aceleração o deslizamento toma o seguinte valor

$$\lambda = \frac{V_w - V}{V_w} \quad (2.19)$$

então uma perturbação no deslizamento pode ser dada por

$$\Delta\lambda = \frac{\partial\lambda}{\partial V} \Delta V + \frac{\partial\lambda}{\partial V_w} \Delta V_w = -\frac{1}{V_{w0}} \Delta V + \frac{V_0}{V_{w0}^2} \Delta V_w \quad (2.20)$$

onde V_{w0} e V_0 são a velocidade da roda e do veículo no ponto de operação.

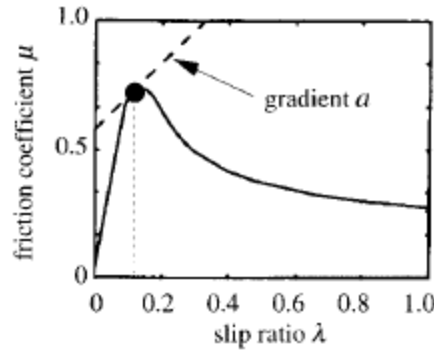


Figura 2.8 - Razão de deslizamento nominal é dada por $a=1$ [4].

A força de fricção é representada usando a o gradiente da curva $\mu-\lambda$, com a seguinte relação

$$\Delta\mu = a\Delta\lambda \quad (2.21)$$

Por combinação das equações (2.20) e (2.21) com (2.16) e (2.17), chega-se à função de transferência do binário motor e da razão de deslizamento, dada por

$$\frac{\Delta\lambda}{\Delta F_m} = \frac{1}{Na} \frac{M(1-\lambda)}{M_w + M(1-\lambda)} \frac{1}{1+\tau s} \quad (2.22)$$

onde τ é a constante de tempo dada por

$$\tau = \frac{1}{Na} \frac{MM_w V_{w0}}{M_w + M(1-\lambda)} \quad (2.23)$$

e é proporcional a V_{w0} .

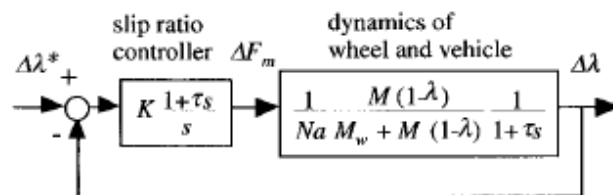


Figura 2.9 - Controlador PI [4].

O controlador do bloco *slip ratio controller* é um simples PI (ver Figura 2.9), com ganho variável e cujo numerador compensa o pólo de (2.22).

Finalmente a função de transferência da razão de deslizamento de comando e da razão de deslizamento actual é dada por

$$\frac{\Delta\lambda}{\Delta\lambda^*} = \frac{1}{1 + Na \frac{M_w + M(1-\lambda)}{M(1-\lambda)} \frac{1}{K} s} \quad (2.24)$$

Se $\lambda \ll 1$, (2.24) é um simples sistema de primeira ordem com uma constante de tempo ajustada pelo ganho K .

A Figura 2.8 mostra o valor nominal de deslizamento que é definido por $a = 1$, e fica do lado esquerdo do valor de pico que é a zona estável.

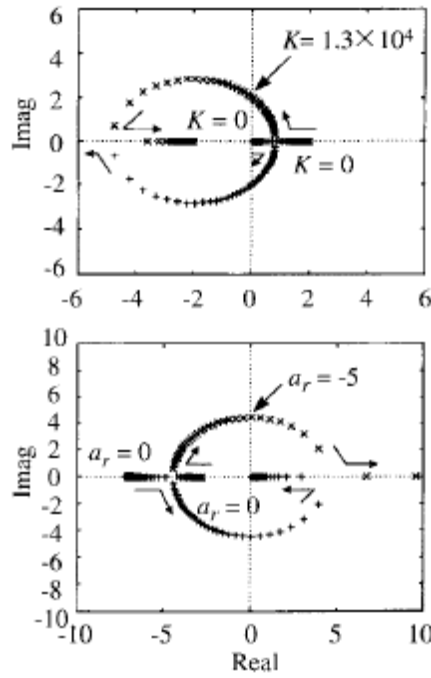


Figura 2.10 - Lugar das raízes com variação dos parâmetros [4]

Como o valor dos parâmetros do sistema actual variam largamente, a robustez do controlo é testada a partir do lugar geométrico das raízes (Figura 2.10), variando o valor de K e a_r (valor actual de a). Pode-se ver na Figura 2.10, que as raízes movem-se para o semi-plano esquerdo quando o ganho K aumenta. É importante que este ganho estabilize o sistema mesmo quando o valor actual de a se torna negativo, apesar de as raízes se moverem para a região instável.

Neste tipo de controlo SRC, é necessário a estimação do coeficiente de fricção que depende das condições de superfície de estrada, para se determinar o valor de deslizamento óptimo a fornecer ao controlador *slip ratio*. Alguns métodos de estimação sugeridos em [4] são

$$\mu = \frac{M}{N} \dot{V} \quad (2.25)$$

quando, a velocidade do veículo é medida directamente. Caso a velocidade do veículo não seja medida directamente pode-se usar

$$\mu = \frac{1}{N} (F_m - M_w \dot{V}_w) \quad (2.26)$$

2.5 - Conclusões

Neste capítulo, foi feita uma breve revisão sobre o modelo longitudinal de um veículo de estrada. Foram apresentadas algumas formas de modelizar o coeficiente de fricção pneu/estrada, baseados em modelos empíricos e analíticos, e por fim, dois tipos de controladores de tracção, MFC e SRC, foram analisados com base nos diagramas de blocos. O MFC segue um modelo pouco rigoroso, enquanto o SRC é um controlo com maior rigor que mantém o deslizamento numa determinada gama desejada, sendo também um controlador mais avançado a nível de desempenho.

A importância do modelo do veículo e do coeficiente de fricção provém da necessidade de estes serem utilizados para efeitos de simulação em projectos de controlo de tracção.

A conclusão principal que se pode retirar deste capítulo, é que o coeficiente de fricção pneu/estrada μ e o deslizamento λ são as variáveis mais importantes no controlo de tracção, pois é a característica μ - λ que define qual a zona perigosa, que se pretende evitar com o controlo de tracção. A zona da curva μ - λ , considerada perigosa, é principalmente a parte em que o coeficiente de fricção diminui com o aumento do deslizamento, é uma zona instável, pois o veículo entrando nessa zona, o deslizamento tende a aumentar rapidamente, e a aderência ao piso é perdida. Assim, o controlo de tracção consiste em controlar ou suprimir o deslizamento de tal forma que a perda de aderência seja evitada.

Com este capítulo ficamos a conhecer as variáveis importantes no controlo de tracção. Sendo o deslizamento uma dessas variáveis, projectou-se um sistema de aquisição de dados, que entre outras funcionalidades, é capaz de adquirir a velocidade do veículo e a velocidade das rodas, pois o deslizamento depende dessas duas velocidades (2.7), para serem utilizadas no algoritmo de controlo de tracção, projectado no Capítulo 5. Como um dos objectivos, após a implementação do controlador, é analisar os resultados dos testes experimentais de controlo, os dados relevantes durante os ensaios teriam de ser monitorizadas e armazenadas, para uma posterior análise. Por isso, foi projectado um sistema que é capaz de adquirir, processar e armazenar os dados, designado por *Sistema DataLogger*.

O próximo capítulo trata da fase de projecto do *Sistema DataLogger*.

Capítulo 3

Projecto de um sistema *DataLogger*

3.1 - Introdução

Este capítulo pretende sistematizar um conjunto de requisitos técnicos, nomeadamente as especificações do *software* (SW) e *hardware* (HW) necessários ao desenvolvimento de um sistema *DataLogger*, a ser incorporado na plataforma do veículo eléctrico *uCar*.

Outro objectivo, é avaliar diferentes formas no modo como os dados podem ser armazenados e respectiva facilidade de os exportar para outros suportes PC e/ou ferramentas de análise de dados (Excel, Matlab, etc).

Este capítulo inicia-se com uma definição geral do que é um sistema *DataLogger*, fala da importância da utilização deste tipo de sistemas em veículos terrestres, apresentando alguns exemplos de aplicação, e em particular o interesse que tem no desenvolvimento de veículos eléctricos. De seguida, é apresentado um “estudo de mercado” sobre alguns tipos de sistemas *DataLogger*, utilizados no desporto automóvel, que pretende essencialmente resumir as características/funcionalidades desses sistemas. Este estudo foi importante para ajudar a traçar os requisitos do sistema *DataLogger* a projectar.

Posteriormente, é apresentada a plataforma de desenvolvimento do sistema de aquisição de dados, com a caracterização da placa sbRIO-9631 da *National Instruments* (NI), e os requisitos do sistema. Um requisito importante do sistema está relacionado com o armazenamento dos dados. É neste contexto que são apresentadas algumas alternativas possíveis de serem implementadas, sendo discutidas as vantagens/desvantagens de cada uma delas. Pretende-se comparar as várias soluções e determinar qual delas é posta em prática.

3.2 - O que é, e qual a importância deste tipo de sistemas

O sistema *DataLogger* é um equipamento destinado a executar a aquisição e o armazenamento de dados para uma posterior análise. Estes dados são fornecidos por sensores ou equipamentos externos, dos quais se deseja obter um histórico de monitorização.

Normalmente, são equipamentos constituídos por um controlador (que pode ser um computador), memória para armazenamento dos dados, interface de aquisição e sensores.

Este tipo de sistemas pode incorporar funcionalidades que são importantes em diversas aplicações. Relativamente ao uso destes sistemas em veículos terrestres, podem ser utilizados por exemplo, para obter parâmetros do veículo, monitorizar o deslizamento ou bloqueamento das rodas, armazenar grandezas que permitam identificar alguma falha em caso de acidente, como acontece nos aviões, funcionando neste caso como uma caixa negra para veículos terrestres.

O *DataLogger* também representa uma ferramenta importante no desenvolvimento de veículos eléctricos, pois permite avaliar a autonomia e o desempenho do veículo, determinar a eficiência e as curvas de carga/descarga das baterias, quantificar em termos energéticos a percentagem de energia que é regenerada quando o veículo freia, quantificar as perdas no desempenho devido a variações paramétricas (tensões nas baterias, parâmetros dos motores, etc.), entre outras.

Neste trabalho, o *DataLogger* a desenvolver será importante no armazenamento dos dados durante ensaios experimentais, para determinar a autonomia e o desempenho do veículo eléctrico uCar, e avaliar o funcionamento do controlador de tracção

3.3 - Caracterização de sistemas de aquisição de dados

Nesta secção pretende-se sumariar vários tipos de sistemas *DataLogger* já existentes, nomeadamente os que são utilizados no desporto automóvel. O objectivo não é efectuar uma revisão completa destes sistemas, mas sim uma caracterização apenas dos que são considerados mais relevantes. A análise das funcionalidades destes sistemas é importante para ajudar a traçar os requisitos do sistema *DataLogger*.

Um dos sistemas analisados foi o DL1, é considerado o estado da arte dos sistemas *DataLogger* [7]. Em termos simples, este armazena uma gama de dados do veículo para mais tarde serem analisados num PC. As características mais importantes do DL1, são apresentadas a seguir:

- receptor GPS de 5Hz integrado, calcula posição e velocidade 5 vezes por segundo. As medidas do GPS e acelerómetros são combinados para calcular as posições e velocidades com alta precisão a 100 vezes por segundo;
- acelerómetros de 3 eixos com escala completa de 2g;
- armazena os dados em memória *compact flash* (CF). A CF é robusta, económica e ideal para uso em produtos de *datalogging*. As vantagens do seu uso consistem na rapidez nos tempos de *download* e na enorme capacidade de armazenamento;
- 8 entradas analógicas de 12bit;
- 2 entradas rpm;
- 4 entradas de velocidade, podem ser usadas para medir a velocidade das 4 rodas, ou razões de deslizamento;

- entrada série (RS232) para dados ECU/OBDii;
- Saída série (RS232) de dados;
- Operação simples. Um simples botão para *start/stop logging*.

Outro sistema analisado foi o *Digitek Cobra*, com altas taxas de aquisição e um grande número de canais disponíveis [8]. A sua única restrição é a velocidade de *download* 10mbit/seg por *Ethernet*. As características principais encontram-se a seguir:

- 64Mb de memória;
- 16 entradas analógicas directas de 12bit;
- 4 entradas digitais para velocidade ou RPM;
- 2 linhas CAN;
- Opção telemetria de tempo-real;
- Aquisição ate 1kHz;
- Endereço IP único;
- Expansão ate 288 canais;
- Comunicação *Ethernet*.

Ainda da mesma família do anterior existe o *Digitek Shark*, especificamente projectado para utilizadores profissionais que necessitam de remover os dados de forma rápida, e colocar o carro em movimento novamente [8]. As características apresentam-se a seguir:

- 256Mb de memória interna onde os dados podem ser duplicados por questões de segurança;
- Cartão de memória até 1Gb;
- 16 entradas analógicas directas;
- 4 entradas digitais para velocidade ou rpm;
- 3 linhas CAN;
- 2 linhas ARC Net;
- Endereço IP único;
- Expansão até 1024 canais;
- Comunicação *Ethernet* com PC;
- Opção telemetria de tempo-real;
- Aquisição até 4khz.

Outro sistema conhecido por RDL, um *DataLogger* desenvolvido pela *Magneti Marelli*, é utilizado especialmente para aplicações de competição [9]. Este sistema inclui uma linha *Ethernet* de alta velocidade para *download* dos dados. A interligação com a caixa pode ser feita usando 2 linhas CAN. No equipamento está presente um microcontrolador RISC de alto desempenho e uma FPGA para fins de diagnóstico. O dispositivo proporciona entradas para a velocidade das rodas, e é capaz de controlar qualquer gama de *displays* da *Magneti Marelli*. As principais características apresentam-se a seguir:

- 12 entradas analógicas

- Até 32 MB de memória interna
- Até 256 canais
- Até 40 kB/s taxa de *logging*
- Taxa de amostragem até 1kHz
- 2 barramentos de comunicação CAN e uma linha *Ethernet*

Na Tabela 3.1 encontram-se resumidas as principais características onde pode ser feita a comparação dos vários sistemas apresentados.

Tabela 3.1 - Comparação dos principais atributos dos sistemas de *DataLogger*

	DL1	<i>Digitek Cobra</i>	<i>Digitek Shark</i>	RDL
Entradas analógicas	8 entradas 12bit 0-12V	16 entradas 12bit 0-5V	16 entradas 12bit 0-5V	12 entradas
Entradas digitais	2 rpm 4 velocidade	4 entradas de velocidade e rpm	4 entradas de velocidade e rpm	
Sensores embutidos	GPS 5Hz, Acelerómetros 3 eixos			
Armazenamento de dados	<i>Compact Flash</i> 32MB-2GB	64Mb	256Mb memória interna, Cartão de memória ate 1Gb	32MB memória interna
Comunicação	RS232	CAN <i>Ethernet</i> RS232, RS485	CAN ARC Net <i>Ethernet</i>	CAN <i>Ethernet</i>
Alimentação	12V	7-18V	7-18V	

Os aspectos apresentados nos vários sistemas, tal como o armazenamento de dados, a capacidade de memória, o tipo de comunicações que dispõe para a transferência de dados, são aspectos importantes em sistemas *DataLogger*. Por isso, o estudo deste tipo de sistemas ajuda a traçar os requisitos do sistema a projectar (secção 3.5).

Na secção seguinte, é apresentada a plataforma de suporte ao desenvolvimento do sistema *DataLogger*, a placa sbRIO-9631 da National Instruments (NI).

3.4 - Caracterização da placa SbRIO-9631

Nesta secção descreve-se de forma sucinta a arquitectura geral da placa sbRIO-9631, do fabricante NI, a utilizar no projecto do sistema de aquisição de dados do veículo eléctrico uCar.

A placa sbRIO é especialmente concebida para aplicações de tempo-real, com requisitos temporais exigentes, e para sistemas de aquisição de dados. É considerado um dispositivo altamente confiável, de baixo custo para aplicações de grande volume que requerem flexibilidade, alto desempenho e confiança.

Na Figura 3.1 apresenta-se um diagrama de blocos da arquitectura da placa, com a identificação dos principais blocos de processamento e dispositivos de entrada/saída disponíveis. Em particular, os três principais componentes que incorporam a placa são: o processador de Tempo-Real, a FPGA (*field-programmable gate array*), e os módulos I/O [10].

A FPGA está ligada ao processador de Tempo-Real através de um barramento PCI de alta velocidade. Acresce que, os portos de I/O encontram-se directamente ligadas à FPGA, como se vê na Figura 3.1. No diagrama de blocos, a placa sbRIO possui ainda três *slots* de expansão que permitem a conexão de módulos adicionais de I/O da série C, proprietários, ou subsistemas especificamente desenvolvidos para uma determinada aplicação.

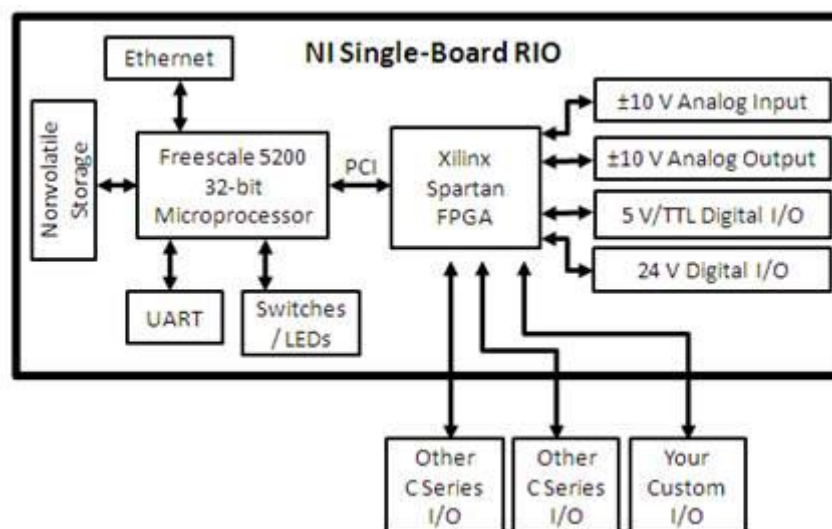


Figura 3.1 - Diagrama de blocos da placa sbRIO-9631 [10]

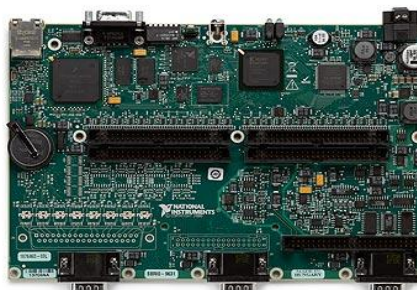


Figura 3.2 - Placa sbRIO-9631 [10]

A forma como esta placa se encontra projectada permite uma arquitectura de I/O reconfigurável (RIO).

A descrição geral da arquitectura da placa e o tipo de aplicações em que é utilizada foram apresentadas até aqui, a parte que se segue apresenta as características das duas componentes da placa, nomeadamente as de HW e SW.

3.4.1 - *Hardware*

As principais características de HW são as seguintes:

- Processador industrial integrado, FPGA reconfigurável, I/O analógicas e digitais numa simples PCB, ver Figura 3.2;
- 1M gate Xilinx Spartan-3 FPGA;
- Processador de tempo-real 266MHz;
- 110 linhas DIO de 3.3V, (para entradas no máximo 5V);
- 32 canais AI;
- 4 canais AO;
- suporta até 3 *slots* de expansão série C;
- 64MB DRAM, 128MB memória não volátil;
- Porta Ethernet 10/100BASE-T X com servidores FTP e HTTP integrados, e protocolos de comunicação TCP/IP, UDP, Modbus/TCP;
- Porta série RS232 para dispositivos periféricos.

3.4.2 - *Software*

Ao nível do SW a placa é suportada com a ferramenta de programação visual *Labview*, com os módulos dedicados às FPGA's e os módulos de tempo real, designados por *FPGA Module* e *Real-Time Module*, respectivamente.

O módulo da FPGA é uma ferramenta que permite, de forma fácil, personalizar o HW e os portos de I/O, o processamento em linha, e desenvolver aplicações de controlo avançado.

O módulo de tempo real é utilizado no desenvolvimento de aplicações embebidas, confiáveis e determinísticas, para processamento *floating-point*, *datalogging*, e comunicação em rede.

Após a descrição da plataforma a utilizar no desenvolvimento do *DataLogger*, pretende-se de seguida traçar os requisitos do sistema.

3.5 - Requisitos

Nesta secção, é apresentada a arquitectura do sistema de aquisição de dados, que caracteriza os sinais a adquirir, e mostra a forma como se pretende integrar todo o sistema ao

nível de HW. Do ponto de vista de *DataLogger*, são traçados os requisitos do sistema de acordo com a plataforma a utilizar.

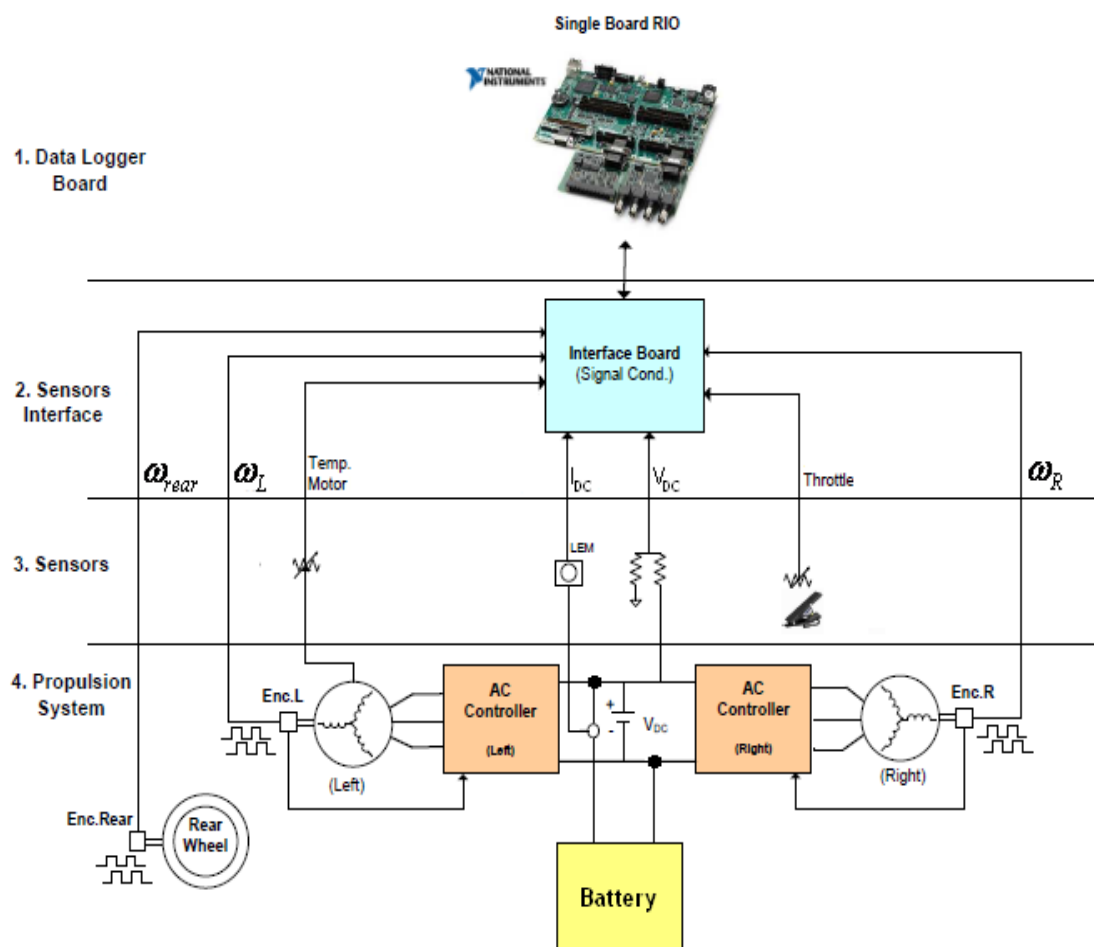


Figura 3.3 - Arquitectura do sistema de aquisição de dados

Como se pode ver na Figura 3.3, a arquitectura do sistema de aquisição de dados é constituída por três camadas principais. A camada 3 engloba os sensores, a camada 2 uma interface de condicionamento de sinal e a camada 1 a placa sbRIO-9631. A camada 2 faz a ligação entre os sensores e a placa sbRIO para fazer o condicionamento de sinal.

Na Tabela 3.2, é apresentada a forma como os sinais serão adquiridos pelo sistema, e as características de cada sinal (frequência mínima de amostragem e gamas de medida). Os sinais de corrente e tensão das baterias são importantes para avaliar o desempenho e a autonomia do veículo eléctrico uCar, enquanto os sinais de velocidade, a adquirir a partir de contadores (*encoders*), e o sinal do acelerador (*throttle*) são variáveis com um papel importante no controlo de tracção.

Tabela 3.2 - Características dos sinais

Analógicas			“Contadores”	
Nome	fMin(Hz)	Gama de medida	Nome	fMin(Hz)
Corrente I_{dc}	200	[-200,200] A	Vel. roda front L	100
Tensão V_{dc}	200	[0,80] V	Vel. roda front R	100
Throttle	50	[0,5] V	Vel. roda rear	100
Temp Motor	2	[50,100] °C		

No projecto do sistema *DataLogger* foram traçados os requisitos principais para a plataforma de desenvolvimento. Os requisitos apresentam-se a seguir:

- 4 entradas analógicas, 16bit e $\pm 10V$;
- 6 entradas digitais, 0-5V, para 3 *encoders* (cada *encoder* tem 2 canais A e B);
- Frequência de amostragem <1kHz;
- Memória para armazenamento de dados, discutida a seguir na secção 3.6;
- Alimentação: 19-30V, 28W;
- Iniciar ou parar aquisição com um simples botão *start/stop logging*;
- Comunicação: RS232 e Ethernet;
- Visualização de grandezas em tempo real.

No que diz respeito à alimentação, a placa sbRIO pode ser alimentada com 19-30V [11], e os requisitos de potência foram calculados admitindo que todos os recursos (portos de I/O) da placa são utilizados, para que no futuro todas possam ser utilizadas, se necessário. Assim, o valor de potência foi calculado da seguinte forma:

$$P_t = 1.2 * (P_{int} + P_{DIO} + P_{5V} + P_{CSer}) \quad (3.1)$$

onde, P_{int} é o consumo da operação interna da SB-RIO, incluindo I/O integradas, com o valor máximo de 7.75W, P_{DIO} é o consumo das DIO, com o valor máximo de 1.28W, P_{5V} é o consumo das saídas de tensão 5V, com valor máximo de 11.1W, e P_{CSer} é o consumo dos módulos C Séries instalados (no máximo 3 módulos), com valor máximo de 3.3W. Desta forma, o valor de potência resultante é $P_t = 1.2 * (7.75 + 1.28 + 11.1 + 3.3) \approx 28W$. É de notar que esta potência é calculada tendo em conta mais 20% que pode ser necessária nas condições de transição durante o funcionamento.

Um dos requisitos que é importante avaliar no projecto do sistema *DataLogger*, está relacionada com a forma como se pretende que os dados sejam armazenados. É nesta perspectiva, que são apresentadas de seguida algumas das possíveis alternativas, que foram analisadas.

3.6 - Alternativas para armazenamento de dados

Nesta secção, o objectivo é apresentar uma visão geral sobre algumas alternativas para o armazenamento de dados (ver Figura 3.4), e fazer uma análise comparativa das vantagens e desvantagens. Pretende-se com isto fazer um balanço de forma a escolher a melhor solução.

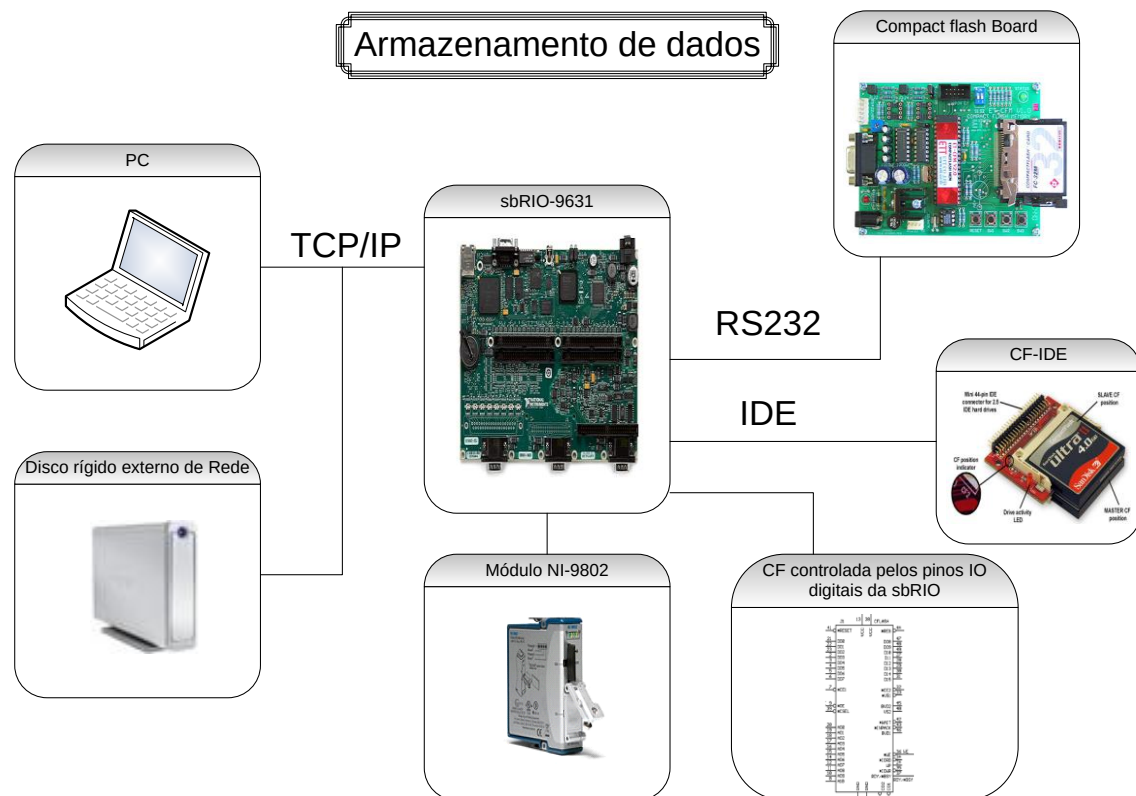


Figura 3.4 - Diferentes soluções para o armazenamento de dados

Um dos métodos mais simples seria utilizar a memória interna não volátil da própria placa sbRIO. Uma solução sem custos adicionais e com a vantagem principal de ser um dos métodos de armazenamento de dados mais rápidos. As taxas de transferência de dados podem tomar valores de 1.5 a 2.2MB/s. Contudo, esta alternativa tem a desvantagem da capacidade de memória interna estar limitada a 128MB.

Outro método, seria acrescentar um módulo adicional NI-9802. Este módulo NI 9802 (ver Figura 3.4) proporciona uma expansão de armazenamento de dados e é especialmente utilizada em aplicações de *datalogging* [12]. Tem dois *slots* onde são colocados cartões SD de 2GB cada. Estes cartões de memória utilizados são interessantes do ponto de vista de capacidade e rapidez nas operações de leitura/escrita. A velocidade de transferência de dados para o módulo é de 2MB/s. Além disto são flexíveis na medida em que é possível retirá-las e colocá-las num PC para analisar os dados. Uma das desvantagens desta solução é o custo relativamente elevado.

A *compact flash board*, designada por placa de interface de computador [13], é uma das possíveis soluções também apresentada. Possui um microcontrolador da PIC, suporta memória compact flash e tem interfaces série tipo RS232, RS422 e RS485 (ver Figura 3.4). Com base nestas características a ideia passaria por interligar esta placa via RS232 à placa sbRIO disponível. Desta forma, os dados seriam transferidos via RS232 da sbRIO para a placa, onde o microcontrolador da PIC se encarregava de armazenar os dados que recebia da porta série. Contudo esta alternativa tem uma limitação, a velocidade de transferência de dados ficaria limitada as taxas de transmissão disponíveis pela porta série.

Relativamente a soluções que integram memórias *compact flash*, são apresentadas mais duas além da anterior. Uma delas utiliza um adaptador CF-IDE *Hard Drive* cujo objectivo seria ligar este conector à placa sbRIO. O problema é que a sbRIO não está devidamente preparada para aceitar este tipo de conectores. A outra solução é ligar directamente pinos de IO digitais da sbRIO aos pinos da *compact flash*. Uma desvantagem desta solução é o facto de haver a necessidade de se desenvolver um programa adicional que teria como objectivo controlar o acesso à memória para ler/escrever dados e fazer o respectivo endereçamento.

Com base na descrição feita sobre estas últimas duas soluções, pode-se desde já dizer que para as desenvolver certamente seria necessário despender mais tempo em relação às restantes soluções, colocando as mesmas fora de hipótese de implementação, apesar de estas serem relativamente baratas.

As duas restantes soluções são: utilizar um disco rígido externo de rede ou um PC. O objectivo destas duas soluções seria ligar cada um dos dispositivos por cabo Ethernet à placa sbRIO e transferir os dados da placa para os dispositivos, por TCP/IP. Estas duas soluções permitem elevadas taxas de transferência de dados e possuem grandes capacidades de armazenamento na ordem de centenas de GB, mas têm um custo relativamente elevado.

Como resultado da análise das várias formas de armazenamento, o interessante seria utilizar uma das soluções com cartão *compact flash*, pelas principais vantagens de possuírem uma grande capacidade de memória e a flexibilidade de serem retiradas e colocadas num PC para analisar os dados. Contudo, estas soluções poderiam aumentar o tempo e o custo de desenvolvimento do sistema, e por isso deu-se especial interesse em avançar com o primeiro método apresentado, utilizando a memória interna não volátil de 128MB da placa sbRIO. Esta forma de armazenamento, embora tenha limitações na capacidade de memória será suficiente para os testes que se pretendem fazer. Para ter uma ideia, foi feita uma estimativa do tempo máximo de armazenamento de dados para as condições que se apresentam na Tabela 3.3.

Tabela 3.3 - Estimativa do tempo máximo no armazenamento de dados

Resolução	nº de canais	Frequência	Memória	Tempo
2Byte	20	1kHz	128MB	55min

A memória da placa sbRIO, é então suficiente, e por isso será utilizada, funcionando de forma *standalone* enquanto os dados são armazenados durante os ensaios. Além disso, a sbRIO possui uma interface *Ethernet* que permite extrair os dados rapidamente após os ensaios.

Assim, o objectivo será armazenar os dados de forma contínua, para a memória não volátil da placa sbRIO, enquanto o veículo estiver em movimento, e no final do ensaio a sbRIO é ligada à rede interna da FEUP ou a um PC, para extrair os dados por *Ethernet*.

3.7 - Conclusões

Neste capítulo, foi apresentada a importância da utilização dos sistemas *DataLogger*, especialmente em veículos eléctricos, e foram caracterizados alguns tipos de sistemas de aquisição de dados de topo, que são utilizados no desporto automóvel. Os sistemas caracterizados, proporcionaram uma base importante, na fase em que foram traçados os requisitos do sistema a desenvolver. A placa sbRIO-9631, foi apresentada como a plataforma de suporte ao desenvolvimento do sistema *DataLogger*, juntamente com a caracterização dos sinais a adquirir, e uma placa de interface de condicionamento de sinal que é necessário desenvolver, parte que será tratada no Capítulo 4.

Um dos requisitos importantes do sistema *DataLogger* é a forma como os dados são armazenados, neste contexto foram analisadas algumas alternativas, como por exemplo, a utilização de um PC ou de um disco externo, ligados por *Ethernet* à placa sbRIO, e algumas soluções que integravam cartões de memória *compact flash*. A vantagem comum destas soluções é a grande capacidade de memória que possuem, face à alternativa mais simples, a utilização da memória interna da própria placa sbRIO. Em particular, uma vantagem dos cartões de memória é a grande flexibilidade no transporte dos dados e colocação num PC para os analisar. No entanto, de todas as soluções, foi escolhida a utilização da memória não volátil da sbRIO, por vários motivos. Por um lado, é uma das soluções mais simples que evita a integração de mais HW, não acarretando com mais custos e tempo de desenvolvimento, por outro lado, a capacidade de memória é suficiente para os ensaios que se pretendem fazer e a interface *Ethernet* que a placa tem, permitirá descarregar os dados de forma rápida para um PC, após os ensaios.

Assim, ficaram definidos todos os requisitos do sistema *DataLogger*, sendo o próximo capítulo a fase de desenvolvimento do sistema.

Capítulo 4

Desenvolvimento do sistema *DataLogger*

4.1 - Introdução

Neste capítulo pretende-se descrever as fases de desenvolvimento do sistema *DataLogger*, explicando com detalhe a construção quer da parte de HW quer de SW.

A nível de HW, começa-se por caracterizar os tipos de sensores a utilizar em função das variáveis a adquirir, é feito o projecto do circuito de condicionamento de sinal que faz a interface entre os sensores e a sbRIO (Figura 3.3), e é apresentado o resultado de implementação de HW.

No que diz respeito ao SW, são apresentadas as aplicações desenvolvidas em Labview utilizando os módulos de FPGA e de tempo real. A aplicação que faz a aquisição dos sinais é processada na FPGA. A aplicação de tempo real tem três tarefas, 1) a estimação de velocidade das rodas do veículo, 2) a visualização em tempo real das variáveis num sinóptico, e o 3) armazenamento dos dados em memória, e são executadas no processador de tempo real da placa sbRIO. Para a tarefa de armazenamento de dados foram desenvolvidas duas formas diferentes, a escrita num ficheiro .bin e num ficheiro .tdms. Pretende-se avaliar as vantagens e desvantagens de cada um deles e indicar qual foi o escolhido para a versão final do *DataLogger*.

Após o armazenamento dos dados, o objectivo é descarregá-los num PC, por Ethernet e fazer a respectiva análise. Desta forma, é necessária uma aplicação que faça a leitura do ficheiro e represente os dados graficamente, por isso é apresentada a aplicação de leitura de ficheiros, desenvolvida em Labview e que é executada no PC. Normalmente, é importante que os dados sejam analisados noutro tipo de ferramentas, como por exemplo o *Matlab*, portanto será explicada a forma que foi utilizada para exportar os dados do Labview para o *Matlab*.

4.2 - Sistema de aquisição de dados (HW)

Nesta secção, apresenta-se o desenvolvimento de HW, desde o projecto do circuito de condicionamento de sinal, até ao resultado final da concepção do circuito numa placa em PCB.

4.2.1 - Variáveis a adquirir e caracterização dos sensores

No que diz respeito aos sensores de velocidade, foram escolhidos *encoders* Tabela 4.1, que geram impulsos digitais, a partir do qual se pode estimar (por software) a velocidade angular de cada uma das três rodas do uCar (secção 4.3). Cada um destes *encoders*, admite uma resolução de 64 pulsos por rotação.

Tabela 4.1 - Sinais digitais e sensores

“Contadores”		
Name	fMin(Hz)	Sensores
Vel. roda front L	100	<i>Encoders</i>
Vel. roda front R	100	
Vel. roda rear	100	

Os sensores escolhidos para cada uma das variáveis analógicas são indicados na Tabela 4.2.

Tabela 4.2 - Variáveis analógicas e sensores

Analógicas			
Nome	fMin(Hz)	Gama de medida	Sensores
Corrente I_{dc}	200	[-200,200] A	Efeito de Hall
Tensão V_{dc}	200	[0,80] V	Divisor resistivo
Acelerador	50	[0,5] V	Potenciometro
Temp Motor	2	[50,100] °C	Termistor NTC

O sensor para medir a corrente, é o LEM HAIS 100-P, com saída em tensão. Este sensor permite medir gamas de corrente máximas na ordem dos ± 300 A, como se pode ver na Tabela 4.3.

Tabela 4.3 - Características eléctricas do LEM

Corrente nominal do primário rms I_{PN} (A)	Nominal primário, gama de medida I_{PM} (A)	Tipo	Alimentação
100	± 300	HAIS 100-P	5V

O princípio de funcionamento deste sensor é baseado num fenómeno de indução magnética, criado pela corrente I_p que passa pela sua “espira”, dando origem a uma saída em tensão que é proporcional ao valor dessa corrente (ver figura 4.1).

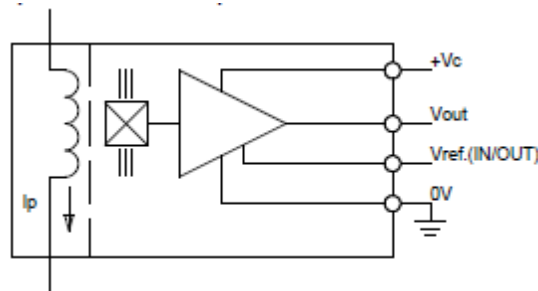


Figura 4.1 - Princípio de funcionamento do LEM

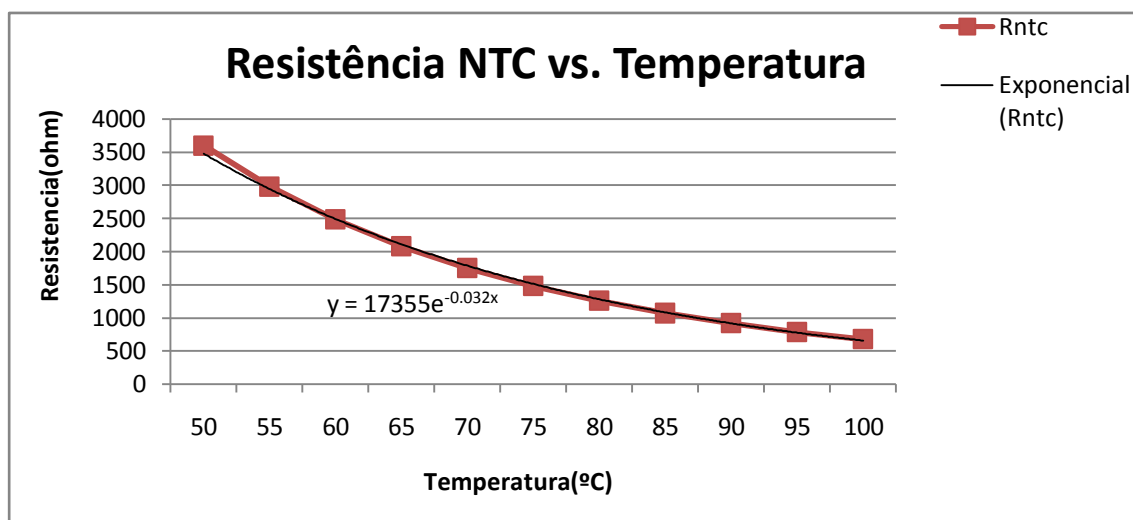
O valor de tensão de saída V_{out} é dado pela seguinte expressão

$$V_{out} = V_{ref} + 0.625 \cdot \frac{I_p}{I_{pn}}, \quad (4.1)$$

onde V_{ref} é uma tensão de referência de cerca de 2.5V, I_p a corrente que se pretende medir e I_{pn} a corrente nominal de 100A. Na medição do valor de tensão na bateria, é utilizado um simples divisor resistivo como sensor, em que a tensão de saída do divisor resistivo será directamente proporcional ao valor de tensão na bateria. A constante de proporcionalidade, é calculada a partir da relação entrada/saída que se pretende, e depende dos valores das resistências. O sinal do acelerador (*throttle*) é obtido a partir dum potenciómetro, cuja resistência varia proporcionalmente, com a posição do acelerador. O sensor escolhido para medir a temperatura do motor foi um *thermistor* NTC (*Negative Temperature Coefficient*). Este sensor é do tipo resistivo, e o valor da sua resistência varia, de forma não linear com a temperatura. A resistência tem o valor normalizado de 10kΩ a 25°C, tendo outros valores, para outras temperaturas, como se pode observar na Tabela 4.4. Nessa tabela, retirada do *datasheet* do NTC, apresentam-se alguns valores da sua resistência para determinadas temperaturas de teste (apenas dentro da gama de interesse de 50-100°C), que foram importantes para traçar a característica $R_{ntc}=f(T)$ deste sensor, como se pode ver na Figura 4.2.

Tabela 4.4 - Valores nominais do NTC em diferentes temperaturas

Temperatura T(°C)	Resistência $R_{ntc}(\Omega)$
50	3603
55	2986
60	2488
65	2083
70	1752
75	1481
80	1258
85	1072
90	917.7
95	788.5
100	680

Figura 4.2 - Característica do *Thermistor* NTC

A característica $R_{ntc}=f(T)$ é não linear e pode ser aproximada por uma função exponencial [14], como mostra a linha de tendência do gráfico da Figura 4.2.

4.2.2 - Projecto do circuito de condicionamento de sinal

4.2.2.1 - Sinais Digitais

Os sinais digitais, idealmente, são representados por transições instantâneas (Figura 4.3), do nível lógico 1 para o 0 e vice-versa. Na realidade, isto nunca acontece, devido à capacidade parasita, resistência e indutância inerentes aos circuitos.

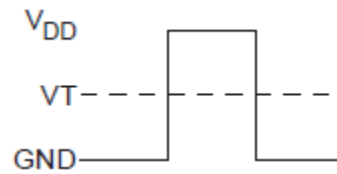


Figura 4.3 - Sinal digital ideal

Desta forma, os sinais passam a ter um tempo finito associado às transições *falling/rising edge* que são quantificados em termos de *slew rate* (tempo de propagação do sinal de 10 a 90%).

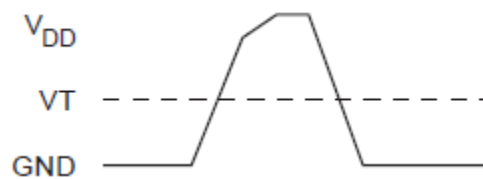
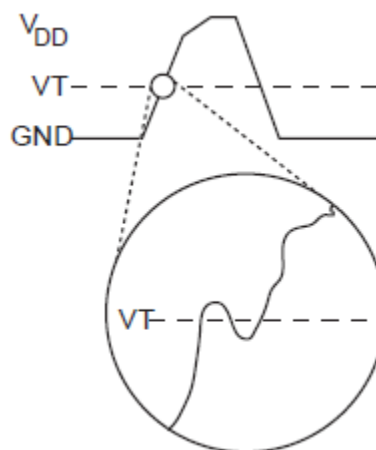


Figura 4.4 - Sinal digital com características reais

Neste contexto, se o sinal se tornar muito lento (tempos de propagação grandes), surge um problema. O ruído que se impõe sobre o sinal, pode dar origem a mudanças de estado múltiplas Figura 4.5, o que é algo indesejável.

Figura 4.5 - Mudanças de estado múltiplas a um nível de *threshold* VT

Para eliminar este tipo de problemas, utiliza-se um *Schmitt trigger*. O *Schmitt trigger* é um *buffer* usado para converter um sinal lento e ruidoso num sinal “limpo”, mais próximo do ideal, antes ser adquirido.

O *buffer Schmitt trigger* é caracterizado por uma função de transferência com histerese. A Figura 4.6 mostra uma curva de histerese, onde a tensão de entrada é no eixo dos xx e a de saída no dos yy.

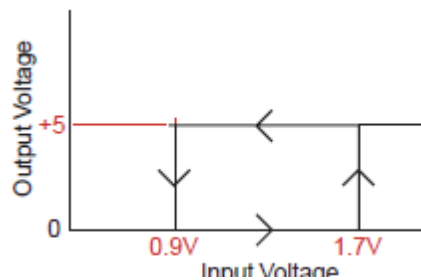


Figura 4.6 - Curva ideal de histerese do *Schmitt trigger Buffer*

Se a tensão de entrada sobe de 0V para 1.7V, a saída permanecerá em 0V. Só quando a entrada excede 1.7V, a saída tomará o valor 5V. Nesta situação, uma redução da tensão de entrada não causará a mudança da saída para 0V de forma imediata. Isto só acontece quando a tensão de entrada é reduzida abaixo dos 0.9V. Assim, pode-se observar que o nível de entrada para o qual a saída aumenta para o máximo, e o nível em que salta para o mínimo são diferentes. Esta característica é conhecida como histerese.

Se aplicarmos então a teoria da histerese ao sinal com ruído mostrado na Figura 4.7 pode-se observar, que uma saída quase perfeita é obtida, a partir duma entrada lenta e com muito ruído.

O *Schmitt trigger* utiliza o fenómeno de histerese com duas tensões de *threshold*. O *threshold* superior é usado para gerar a transição do nível baixo para o nível alto, e o inferior para a transição do nível alto para o nível baixo. Assim, o *trigger* é resistente ao ruído, e será mais resistente quanto menor for o valor pico a pico de ruído em relação à diferença entre as tensões de *threshold* (i.e. menor do que $1.7 - 0.9 = 0.8V$).

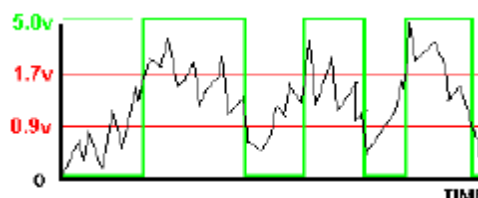


Figura 4.7 - Aplicação duma entrada com ruído a um *Schmitt trigger Buffer*

Neste trabalho, como se pretende adquirir sinais digitais provenientes de *encoders*, o ruído desses sinais será filtrado, utilizando a lógica do *Schmitt trigger Buffer*. O dispositivo escolhido para o efeito, é designado por *Hex inverting Schmitt trigger*, com referência 74HC14. Este integrado possui seis *Schmitt trigger*, a serem utilizados pelos três *encoders*, pois cada *encoder* tem dois canais A e B, e cada canal é ligado a um *buffer*.

A única diferença do funcionamento deste integrado é que as saídas são invertidas, como o próprio nome indica, “*inverting Schmitt trigger*” (Figura 4.8).

FUNCTION TABLE

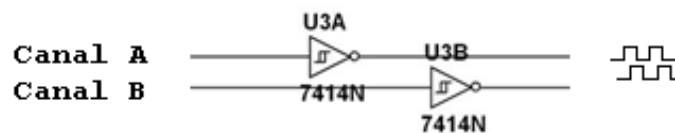
INPUT	OUTPUT
nA	nY
L	H
H	L

Notes

1. H = HIGH voltage level
L = LOW voltage level

Figura 4.8 - Relação entrada/saída do 74HC14

O esquema do circuito de condicionamento dos sinais digitais provenientes de cada *encoder*, encontra-se representado na Figura 4.9.

Figura 4.9 - Circuito de condicionamento dos sinais do *encoder*

Os sinais de saída do *encoder*, canal A e canal B, são ligados directamente aos *Schmitt trigger*, neste caso, U3A e U3B respectivamente. As saídas desses *buffers* são ligadas às entradas digitais da placa de aquisição de dados sbRIO. As tensões de alimentação dos *encoders* e do 74HC14 são 5V, gerando assim níveis de tensão compatíveis com os das entradas digitais da sbRIO que também são 5V.

4.2.2.2 - Sinais Analógicos

Aqui é feita uma descrição dos circuitos de condicionamento dos sinais analógicos, e em particular para o condicionamento de sinal do *Thermistor*, faz-se uma comparação sobre métodos possíveis de linearização da tensão de saída em função da temperatura.

Estes circuitos de condicionamento de sinal analógico, foram projectados para tensões de alimentação de 5V, para que a placa de condicionamento de sinal tivesse apenas um nível de alimentação, evitando assim o uso de mais HW para se gerar vários níveis de tensão.

Para a medição da corrente contínua das baterias (Figura 3.3), é utilizado um transdutor de corrente (LEM) com saída em tensão (Figura 4.10) proporcional ao valor da corrente, cuja expressão é dada por (4.1).

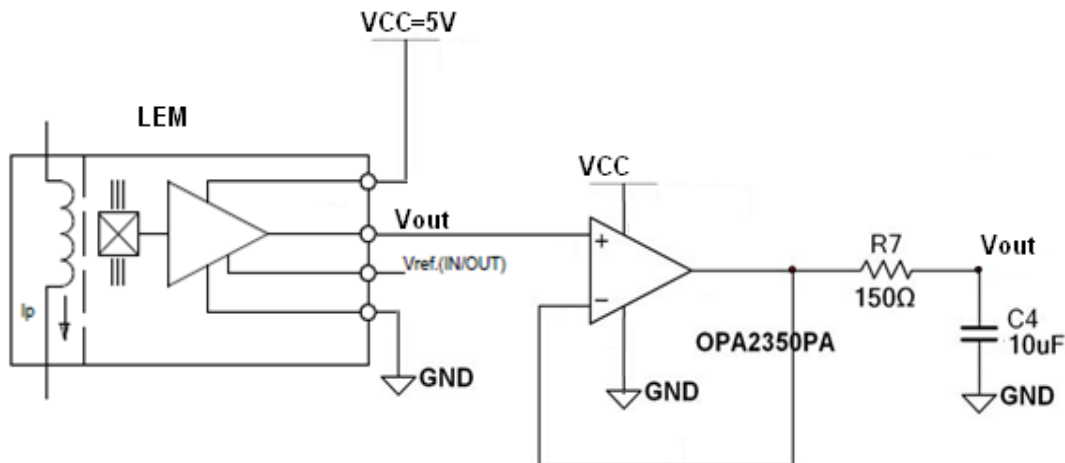


Figura 4.10 - Circuito de condicionamento do LEM

O circuito de condicionamento do LEM (figura 4.10), tem um *buffer* com alta impedância de entrada, seguido de um filtro passa baixo com frequência de corte $f_c = 1/(2\pi \cdot 10\mu \cdot 150) \approx 106\text{Hz}$, que é aproximadamente metade da frequência de amostragem do sinal (Tabela 4.2). Este filtro evita o fenómeno de aliasing além de filtrar ruído.

A sensibilidade do andar de condicionamento é dada por $S = 0.625/I_{pn}$ (V/A). Como a digitalização do sinal será feita pela placa sbRIO com ADCs de 16bit, e entradas analógicas de $\pm 10\text{V}$, significa que esta tem a capacidade de medir incrementos de tensão de $20/2^{16} = 305.2\mu\text{V}$, o que leva a que se tenha uma resolução $305.2\mu/0.00625 = 49\text{mA}$ na medida da corrente, o que é mais do que suficiente.

Na Tabela 4.5 encontram-se as características do sinal do LEM a medir, correspondente à gama de corrente $[-200, 200]\text{ A}$.

Tabela 4.5 - Características do sinal do LEM para a Gama de corrente $[-200, 200]\text{ A}$

	Gama de tensão de saída $V_{out}(V)$	Sensibilidade $S(V/A)$	Resolução (mA)
LEM	$[1.25; 3.75]$	0.00625	49

No condicionamento do sinal de tensão da bateria, é usado um divisor resistivo, seguido de um *buffer* e de um filtro passa baixo (Figura 4.11). A tensão de saída do circuito, em função da tensão da bateria, é dada pela seguinte expressão

$$V_{out} = V_{bateria} * \frac{R4}{R4 + R3} \quad (4.2)$$

onde R4 e R3 são calculados para que a gama de tensão na bateria $[0-80\text{V}]$ corresponda directamente a uma gama de saída V_{out} de $[0-5\text{V}]$.

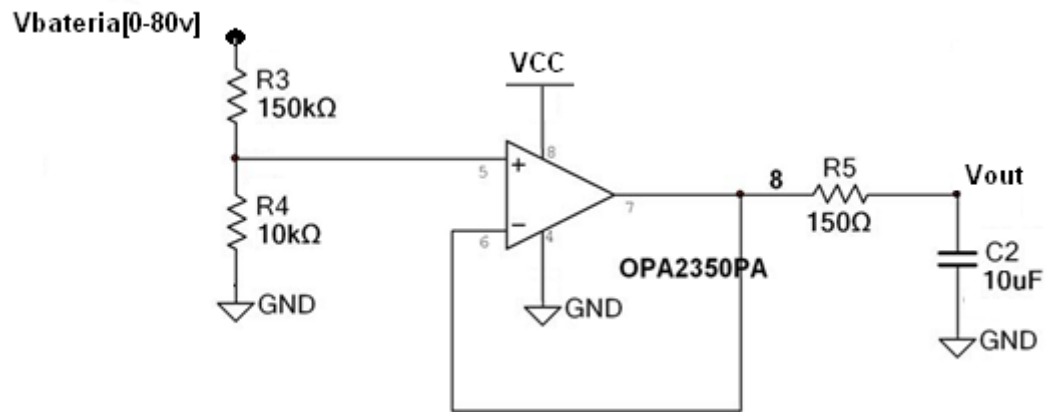


Figura 4.11 - Circuito de condicionamento da tensão da bateria

Ou seja, R4 e R3 devem ser calculados a partir da seguinte condição

$$\frac{V_{out}}{V_{bateria}} = \frac{R4}{R4 + R3} = \frac{5}{80} \quad (4.3)$$

obtendo-se R4=10kΩ para um R3=150kΩ. O filtro passa baixo tem as mesmas características do que foi projectado para o circuito do LEM, uma vez que a frequência de amostragem especificada para a corrente e tensão é a mesma (Tabela 4.2).

O sinal do acelerador provém de um potenciômetro, cuja resistência entre dois dos seus terminais varia em função da posição do acelerador. A variação dessa resistência R_{pot1} (Figura 4.12) é directamente proporcional à posição do acelerador, e como a tensão de saída é proporcional à resistência, também o é em relação à posição. Isto é importante pois a posição do acelerador dá uma referência em velocidade para o veículo, sendo esta proporcional ao sinal de tensão.

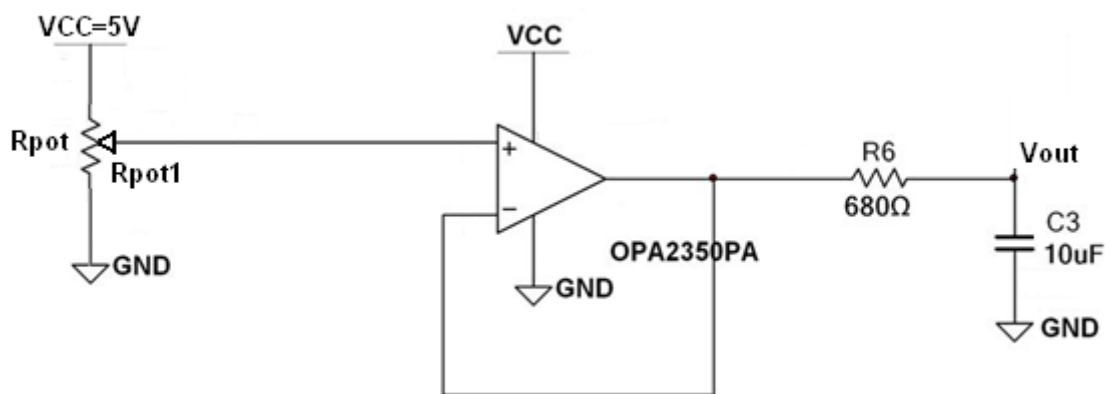


Figura 4.12 - Circuito de condicionamento do acelerador

A tensão de saída em função das resistências entre os terminais do potenciômetro é dada por

$$V_{out} = V_{CC} * \frac{R_{pot1}}{R_{pot}} \quad (4.4)$$

onde R_{pot1} é a resistência que varia linearmente com a velocidade de referência, e R_{pot} é a resistência total do potenciômetro, que se mantém constante. O filtro passa baixo tem uma frequência de corte $f_c = 1/(2\pi * 10\mu * 680) \approx 23.4\text{Hz}$, que é aproximadamente metade da frequência de amostragem definida para este sinal.

No condicionamento de sinal do *Thermistor* NTC, em que o comportamento da resistência é não linear com a temperatura, foram analisadas e comparadas duas soluções possíveis na linearização da tensão de saída em função da temperatura.

O 1º método tem como base um divisor resistivo (Figura 4.13) em que o NTC está em série com uma resistência.

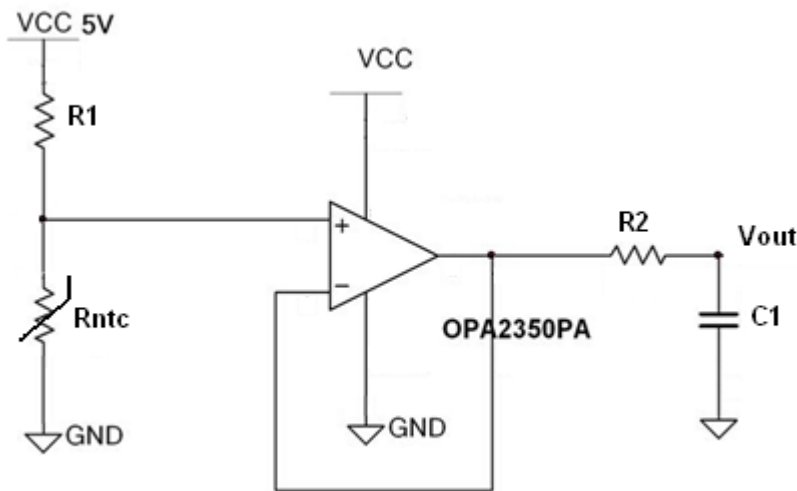


Figura 4.13 - Circuito de condicionamento do *Thermistor* (1º método)

No condicionamento é utilizado um *buffer*, seguido de um filtro passa baixo com frequência de corte (f_c) de aproximadamente 1Hz, que é metade da frequência de amostragem especificada. Este filtro tem valor de $C1 = 10\mu\text{F}$ e de $R2 = 16\text{k}\Omega$, verificando-se $f_c = 1/(2\pi * 10\mu * 16\text{k}) \approx 1\text{Hz}$.

A tensão de saída do circuito tem a seguinte expressão

$$V_{out} = V_{CC} \frac{R_{ntc}}{R_{ntc} + R1} \quad (4.5)$$

onde $V_{CC} = 5\text{V}$ é a tensão de alimentação, e $R1$ é determinado de tal forma que a relação $V_{out} = f(T)$ seja praticamente linear na gama de temperaturas de interesse 50-100°C.

O método analítico para calcular $R1$, é obtido a partir da condição que força três pontos equidistantes de temperatura a corresponder a tensões de saída também equidistantes [14]. Ou seja, $R1$ é obtido a partir da seguinte condição

$$V_{out1} - V_{out2} = V_{out2} - V_{out3} \quad (4.6)$$

onde V_{out1} é o valor da tensão de saída para a temperatura T_1 , V_{out2} para T_2 e V_{out3} para T_3 .

Da condição anterior, retira-se o valor de $R1$, que é dado pela seguinte expressão

$$R1 = \frac{R_{ntc2} R_{ntc1} + R_{ntc3} - 2R_{ntc1}R_{ntc3}}{R_{ntc1} + R_{ntc3} - 2R_{ntc2}} \quad (4.7)$$

onde R_{ntcX} é o valor da resistência do *thermistor* à temperatura T_x . Notar que as temperaturas T_1 , T_2 e T_3 têm de ser escolhidas de forma a satisfazer a seguinte condição

$$T_1 - T_2 = T_2 - T_3, \quad (4.8)$$

para assegurar a equidistância entre os três pontos de temperatura.

Neste caso, foram escolhidas as temperaturas mínima, máxima e intermédia, da gama de temperaturas a medir, 50-100°C (Tabela 4.6) E para esses valores de temperatura, foram retirados os valores da resistência do *thermistor* (valores fornecidos pelo *datasheet*).

Tabela 4.6 - Temperaturas equidistantes escolhidas

Temperatura (°C)	R_{ntc} (Ω)
$T_1=50$	3603
$T_2=75$	1481
$T_3=100$	680

Com os valores da Tabela 4.6, e aplicando a equação (4.7) obteve-se $R1=1.092k\Omega$. No entanto, a resistência disponível com o valor mais próximo normalizado é $1k\Omega$, e por isso no processo de linearização foi tomado $R1=1k\Omega$. Na tabela 4.7 encontram-se os valores da tensão de saída do circuito da Figura 4.13, obtidos com $R1=1k\Omega$. Com esses valores foi traçada a relação $V_{out}=f(T)$ para analisar a sua característica, e fazer a respectiva aproximação linear Figura 4.14.

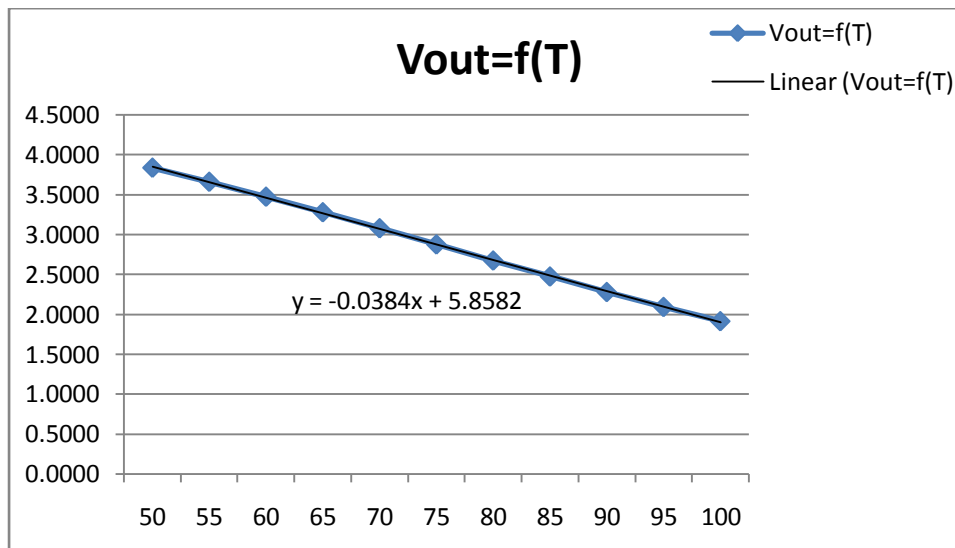
Tabela 4.7 - Tensão de saída V_{out} para temperaturas 50-100°C (1º método)

Temperatura T(°C)	Resistencia $R_{ntc}(\Omega)$	$V_{out}(V)=V_{cc}*R_{ntc}/(R_{ntc}+R1)$
50	3603	3.9138
55	2986	3.7456
60	2488	3.5665
65	2083	3.3782
70	1752	3.1831
75	1481	2.9847
80	1258	2.7857
85	1072	2.5869
90	917.7	2.3927
95	788.5	2.2044
100	680	2.0238

Com os dados da tabela 4.7, foi representado o gráfico $V_{out}=f(T)$, e como se pode observar na Figura 4.14, a característica é praticamente linear. Com este resultado procedeu-se ao cálculo da expressão da linha de tendência obtendo-se a seguinte relação

$$V_{out} = -0.0384T + 5.8582 \quad (V) \quad (4.9)$$

que faz a interpolação linear dos pontos calculados na Tabela 4.7.

Figura 4.14 - Aproximação linear de $V_{out}=f(T)$ (1º método)

A sensibilidade do andar de condicionamento é dado por $S = -0.0384V/^{\circ}C$. Como a digitalização do sinal será feita pela placa sbRIO com ADCs de 16bit, e entradas analógicas de

$\pm 10V$, significa que esta tem a capacidade de medir incrementos de tensão de $20/2^{16}=305.2\mu V$, o que leva a que se tenha uma resolução $305.2\mu/0.0384\approx 7.9m^{\circ}C$ na medida de temperatura, o que é mais do que suficiente para a aplicação em causa.

O 2º método de condicionamento de sinal do NTC tem como base uma fonte de corrente que alimenta a resistência NTC em paralelo com outra resistência (ver Figura 4.15).

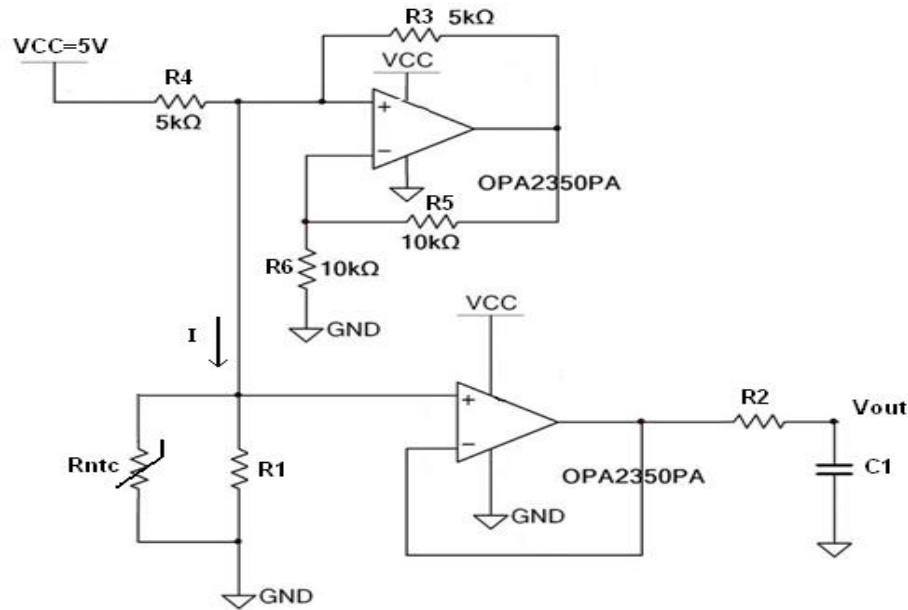


Figura 4.15 - Circuito de condicionamento do *Thermistor* (2º método)

A fonte de corrente é feita a partir duma configuração com um ampop, fornecendo uma corrente $I = V_{cc}/R_4 = 1mA$, ao paralelo da resistência NTC com R_1 . Tal como no 1º método, é utilizado um *buffer* seguido de um filtro com as mesmas características. A tensão de saída V_{out} , neste método, é dada pela seguinte expressão

$$V_{out} = R_p * I \quad (4.10)$$

onde R_p é a resistência equivalente do paralelo entre R_{ntc} e R_1 , e é dada por

$$R_p = \frac{R_{ntc} * R_1}{R_{ntc} + R_1} \quad (4.11)$$

onde R_1 é determinada seguindo o mesmo princípio da equação (4.6), mas com uma diferença do 1º método, a tensão V_{out} passa a ser dada pela equação (4.10). Apesar das equações de V_{out} nos dois métodos serem diferentes, a expressão que se chega para a determinação de R_1 neste método, é igual à expressão obtida no método anterior, portanto o valor de $R_1 = 1.092k\Omega$.

Na Tabela 4.8 apresentam-se os valores da tensão de saída, neste 2º método com $R_1=1k\Omega$ na gama de 50-100°C.

Tabela 4.8 - Tensão de saída V_{out} para temperaturas 50-100°C (2º método)

Temperatura T(°C)	$R_p(\Omega)$	$V_{out}(V)=R_p \cdot I$
50	838.24	0.78275
55	799.79	0.74912
60	759.10	0.71330
65	716.59	0.67564
70	672.86	0.63663
75	628.68	0.59694
80	584.68	0.55713
85	541.05	0.51737
90	498.73	0.47854
95	457.95	0.44087
100	419.11	0.40476

Com os dados da Tabela 4.8, foi traçada a relação $V_{out}=f(T)$, para o 2º método, ver Figura 4.16. Com base nessa relação obteve-se a linha de tendência de $V_{out}=f(T)$ dada pela expressão que se segue

$$V_{out} = -0.00768T + 1.1716 \quad (V) \quad (4.12)$$

Que faz a interpolação linear dos pontos calculados na Tabela 4.8.

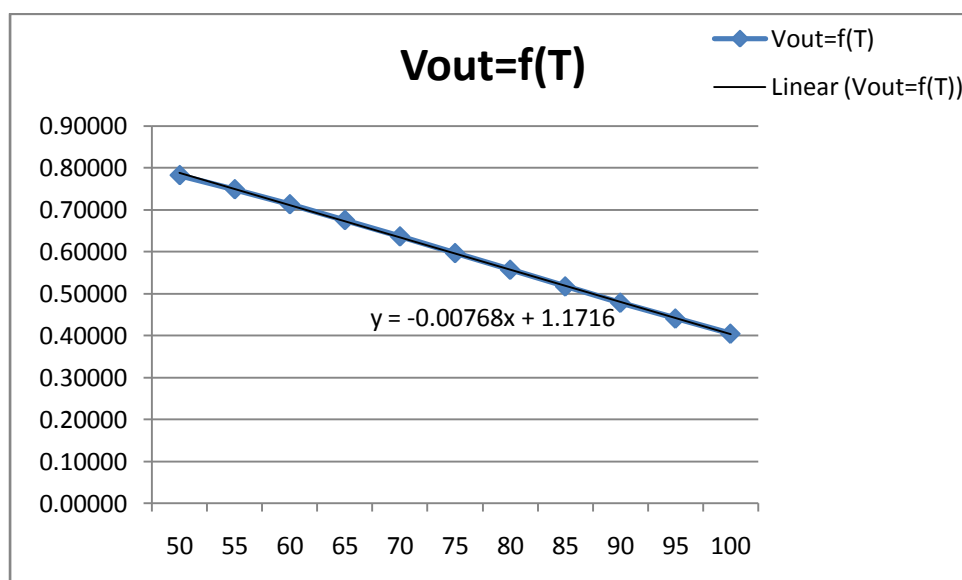


Figura 4.16 - Aproximação linear de $V_{out}=f(T)$ (2º método)

Neste 2º método, a sensibilidade do andar de condicionamento é $S = -0.00768 \text{ V/}^\circ\text{C}$, obtendo-se uma resolução $305.2 \mu\text{V} / 0.00768 \approx 39.7 \text{ m}^\circ\text{C}$ na medida da temperatura.

Tendo em conta os resultados que se obtiveram na análise de cada um dos métodos, pode-se fazer uma comparação em termos de gama de tensão de saída V_{out} , da sensibilidade da tensão com a variação da temperatura, e a resolução conseguida na medida da temperatura (Tabela 4.9).

Tabela 4.9 - Comparação dos resultados obtidos nos dois métodos

	Gama de $V_{\text{out}}(\text{V})$	Sensibilidade $S(\text{V/}^\circ\text{C})$	Resolução (m°C)
1º método	[2.0238; 3.9138]	-0.039	8
2º método	[0.41911; 0.83824]	-0.00851	35.9

O 1º método permite uma gama de tensões de saída mais alargada, obtendo-se uma maior resolução na medida de temperatura e uma maior sensibilidade em relação ao 2º método. Além disto, o facto de a gama de tensões ser maior permite também uma medida de temperatura mais imune ao ruído, ou seja, supondo que é induzido ruído com o mesmo nível nos dois métodos, o valor de temperatura medido no 1º método terá uma componente adicional devido ao ruído que será inferior ao que se obtém no 2º método. Pelas razões apresentadas, foi escolhido o 1º método para a medição de temperatura na gama de 50-100°C.

Após o projecto dos circuitos apresentados, foi desenvolvida a placa de interface de condicionamento de sinal em PCB.

4.2.3 - Concepção do circuito de condicionamento em PCB

A 1ª fase do desenvolvimento do circuito de condicionamento de sinal, foi explicada anteriormente, a 2ª fase foi projectar esse circuito de condicionamento de sinal numa placa PCB, recorrendo a ferramentas da NI, nomeadamente o *Multisim* e o *Ultiboard* (versão 10). Esta fase englobou uma componente de aprendizagem destas ferramentas, no que diz respeito à concepção de PCBs. Foi feito um pequeno tutorial sobre as fases de concepção da PCB e que se encontra no anexo A.

4.2.4 - Implementação do sistema de aquisição de dados

A placa PCB que foi desenvolvida, faz a interface entre os sensores e a placa sbRIO para o condicionamento de sinal. O resultado final obtido na integração deste HW pode ser visto na Figura 4.17.

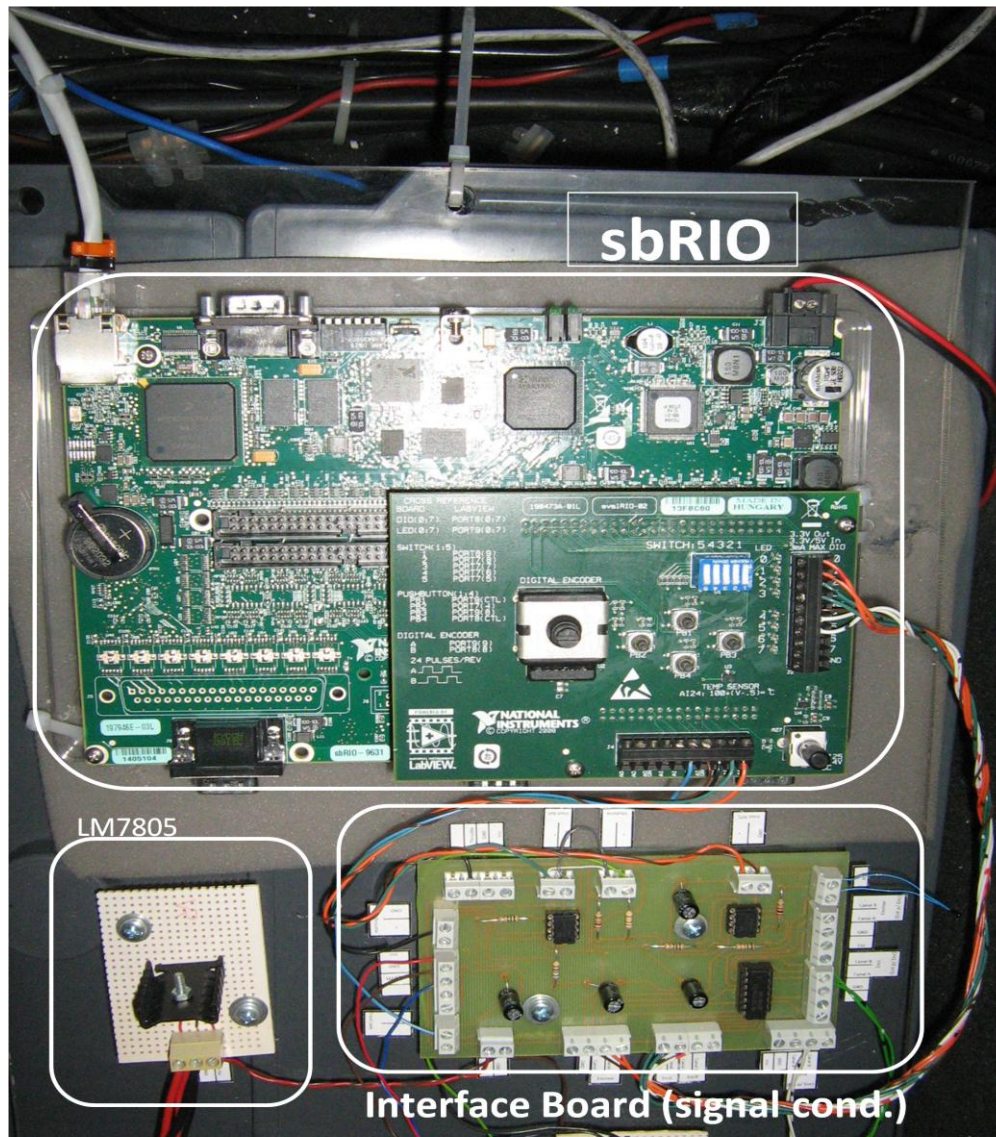


Figura 4.17 - Integração do HW do Sistema de aquisição de dados

Nessa figura encontra-se a placa sbRIO que adquire os dados dos sensores, devidamente condicionados pela placa *Interface Board*.

Neste sistema foi necessário ter dois níveis de tensão, 24V para alimentar a sbRIO e 5V para a placa de condicionamento de sinal. Os 5V foram obtidos a partir dos 24V recorrendo a um LM7805.

O desenvolvimento de HW do sistema *DataLogger* foi apresentado com detalhe, pretende-se na secção seguinte descrever a parte correspondente ao SW.

4.3 - Algoritmo do sistema *DataLogger*(SW)

Nesta secção pretende-se apresentar o SW que foi desenvolvido em *Labview*, explicar como este se encontra estruturado, descrever a funcionalidade e as características de cada aplicação, e mostrar a forma como algumas aplicações interagem entre si.

O SW desenvolvido pode ser dividido em diferentes aplicações, em função do bloco onde são executadas. Desta forma, as aplicações são as seguintes:

- Aplicação de aquisição dos sinais (executa na FPGA);
- Aplicação de processamento e armazenamento de dados (executa no RTP);
- Aplicações de leitura de ficheiros e exportação de dados para *Matlab* (executa no PC).

O diagrama de blocos da Figura 4.18, representa em termos gerais o funcionamento do sistema *DataLogger* desenvolvido.

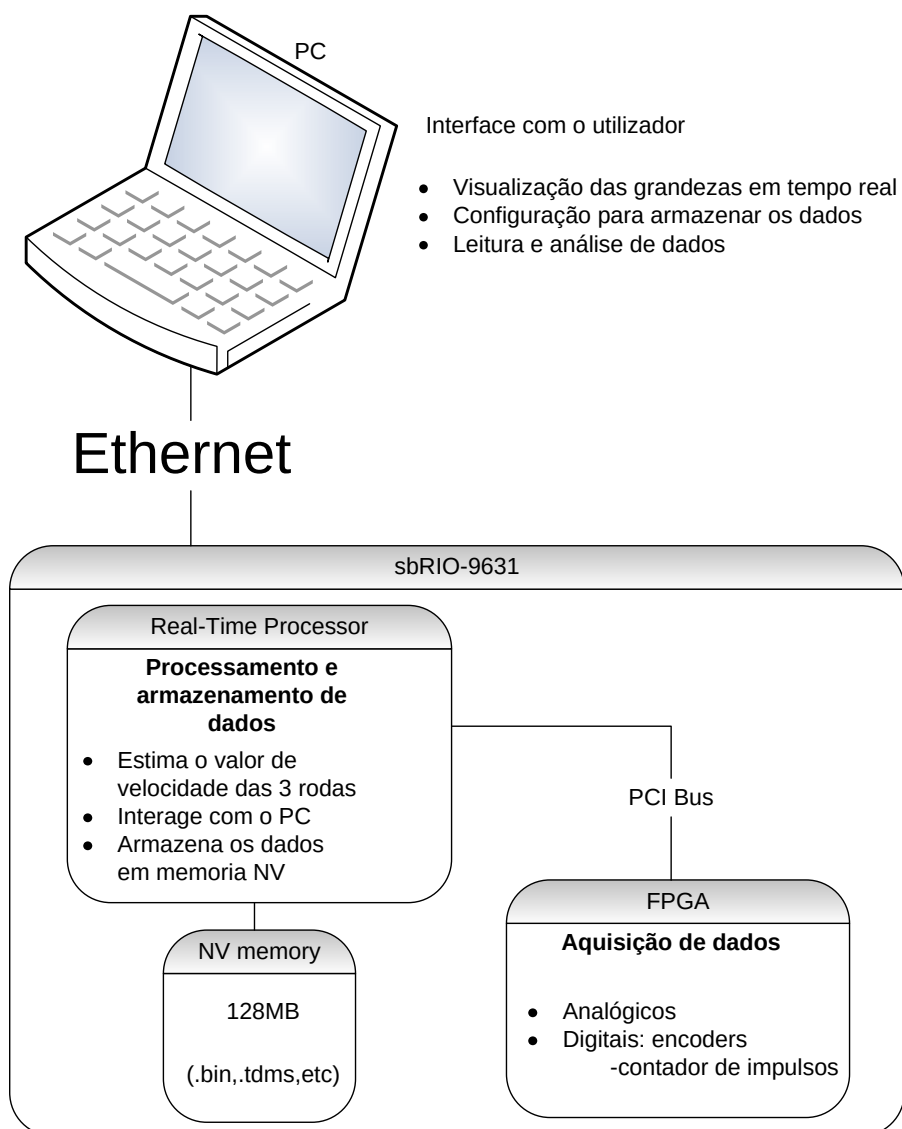


Figura 4.18 - Diagrama de blocos do sistema *DataLogger*

Nessa figura estão descritas resumidamente as funções de cada aplicação. A aplicação que é executada na FPGA, tem como objectivo adquirir os sinais analógicos e digitais. Ao adquirir os sinais digitais (provenientes dos 3 *encoders*), faz uma contagem ascendente/descendente do número de impulsos que cada *encoder* gera durante um certo intervalo de tempo, para que estes sejam utilizados na estimação de velocidade.

As tarefas que executam no RTP têm como função, estimar de forma periódica o valor de velocidade a partir do nº de impulsos que são calculados na FPGA, representar os dados graficamente enviando essa informação para o PC e por fim armazenar os dados na memória NV (*non-volatile*) da placa sbRIO.

O PC encontra-se ligado à placa sbRIO por *Ethernet* permitindo visualizar as grandezas em tempo real. No entanto, caso não se pretenda a utilização do PC, este pode ser ligado apenas para fazer uma configuração inicial no que diz respeito ao armazenamento de dados (especificar a frequência de amostragem, o nº de amostras, etc), e depois pode ser desligado, ficando na mesma a placa sbRIO a armazenar os dados. No final dos ensaios liga-se o PC à sbRIO e descarregam-se os dados por FTP (*file transfer protocol*). Para analisar os dados foi desenvolvida uma aplicação de leitura de ficheiros, que faz a representação gráfica dos dados e além disso exporta-os para outra ferramenta de análise, o *Matlab*.

As aplicações foram desenvolvidas recorrendo aos módulos FPGA e de tempo real do *labview*. A estrutura do projecto do programa com as aplicações pode ser vista no anexo B.

A seguir apresentam-se cada uma das aplicações explicadas com mais detalhe.

4.3.1 - Aplicação de aquisição dos sinais (FPGA)

Esta aplicação encarrega-se de fazer a aquisição dos sinais analógicos e digitais, e é executada na FPGA. Além disso, foi desenvolvido um *subvi* (ver Anexo B) nesta aplicação, para fazer a contagem do nº de pulsos de cada *encoder*, utilizando uma estratégia que aumenta em quatro vezes a resolução dos *encoders*.

Para se avaliar as restrições desta aplicação, a nível de frequência máxima dos pulsos que os *encoders* podem gerar, foi determinado experimentalmente o tempo de execução da aplicação, que representa o período com que os sinais de onda quadrada são amostrados. O resultado desse tempo foi cerca de 32us. Como a estratégia de contagem exige que se amostra pelo menos 4 vezes em cada período da onda quadrada significa que a frequência máxima permitida para os sinais dos *encoders* é de $f_{\max}=1/(4*32\mu s)\approx 7.8\text{kHz}$. Para esta frequência a velocidade correspondente dos motores seria $N=7.8\text{k}*60/64\approx 7.3\text{krpm}$, o que significa que a aplicação funciona sem problemas já que a velocidade dos motores não excede as 3000rpm.

4.3.1.1 - Sinais Digitais dos *encoders*

A aplicação adquire os sinais digitais devidamente endereçados a partir dum bloco de alto nível que faz a interface entre a FPGA e os portos de entrada/saída, ver Figura 4.19. De

seguida os sinais de cada *encoder* entram num bloco que faz a contagem do nº de pulsos com 4 vezes maior resolução. A lógica de contagem é feita da seguinte forma (Anexo B), de cada vez que é detectado um *rising/falling edge* em cada canal (A ou B), o contador é incrementado/decrementado, dependendo do sentido de rotação do *encoder* que é determinado pela relação de quadratura entre os dois sinais A e B.

A contagem do nº de pulsos é colocada em registos, que são lidos periodicamente, pela tarefa que calcula o valor de velocidade na aplicação de processamento e armazenamento de dados.

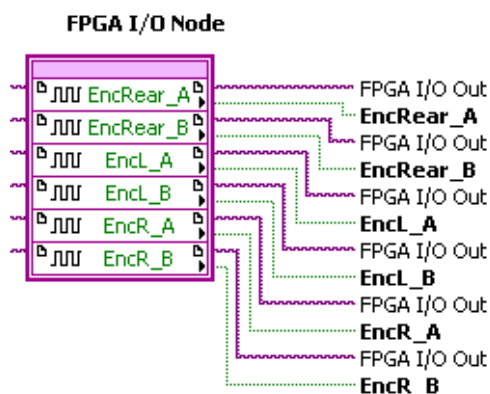


Figura 4.19 - Bloco FPGA I/O Node, Sinais digitais

4.3.1.2 - Sinais Analógicos

Os sinais analógicos são adquiridos a partir do bloco da figura Figura 4.20. Estes sinais são adquiridos e colocados em registos a partir dos quais a aplicação de tempo real tem acesso para os armazenar.

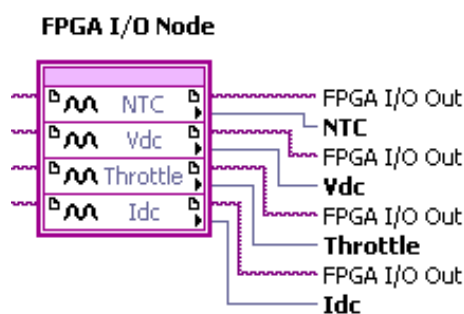


Figura 4.20 - Bloco FPGA I/O Node, Sinais analógicos

4.3.2 - Aplicação de processamento e armazenamento de dados (RTU)

Esta aplicação tem três tarefas a serem executadas em paralelo¹: a estimação de velocidade (A), o armazenamento de dados (B), e a visualização das grandezas em tempo real (C). Estas tarefas têm um conjunto de requisitos temporais, que necessitam de ser cumpridos.

¹O processamento paralelo dos sistemas de tempo real com um único processador é uma expressão em termos conceituais.

Para isso foi necessário determinar o tempo de execução (C) de cada uma das tarefas, e foram especificados os prazos (D) que devem ser cumpridos, (ver Tabela 4.10).

Tabela 4.10 - Características das tarefas

Tarefa	Período (T) (ms)	Prazo (D) (ms)	Tempo de execução (C)	Prioridade
A	10	0.5	0.44	300
B	10	3.5	3	200
C	200	200	4	100

A prioridade que foi afectada a cada tarefa é directamente proporcional à sua importância. Por exemplo a tarefa mais importante é a estimação de velocidade que será utilizada no controlo de tracção, cujo prazo é muito apertado, e por isso sempre que estiver pronta a executar é executada sem interferência das outras tarefas. Assim, evita que hajam erros na medida de velocidade que poderiam ocorrer se esta tarefa fosse de menor prioridade.

Para verificar se o conjunto de tarefas cumpre com os requisitos temporais foi utilizado o algoritmo que calcula o tempo de resposta de cada tarefa no pior caso ($Rwci$).

No que diz respeito à tarefa de maior prioridade, esta cumpre sempre o prazo $RwA=0.44 < D_A$. Pretende-se verificar se as restantes também cumprem com o prazo definido, com o calculo do RwB e RwC . O algoritmo utilizado para determinar estes valores é recursivo e utiliza as expressões (4.13) e (4.14).

$$a_0 = \sum_{j=1}^i C_j \quad (4.13)$$

$$a_{n+1} = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{a_n}{T_j} \right\rceil C_j \quad (4.14)$$

Na determinação de $Rwci$ o funcionamento do algoritmo é o seguinte:

1. Determinar o valor de a_0 , a partir de (4.13), que representa a soma dos tempos de execução de todas as tarefas excepto as de menor prioridade que a tarefa i .
2. Aplicar recursivamente (4.14), até $a_n = a_{n+1}$ ou $a_{n+1} > D_i$. Se $a_n = a_{n+1}$, então $Rwci = a_n$. Se $a_{n+1} > D_i$ então não cumpre com os requisitos temporais.

Para a tarefa C,

$$a_0 = 4 + 3 + 0.44 = 7.44 \quad (4.15)$$

$$a_1 = 4 + \left\lceil \frac{7.44}{10} \right\rceil 3 + \left\lceil \frac{7.44}{10} \right\rceil 0.44 = 4 + 1*3 + 1*0.44 = 7.44 \quad (4.16)$$

logo, o tempo de resposta no pior caso de C é $RwcC=7.44 < D_C$, por isso a tarefa cumpre com os requisitos temporais.

Para a tarefa B,

$$a_0 = 3 + 0.44 = 3.44 \quad (4.17)$$

$$a_1 = 3 + \left\lceil \frac{3.44}{10} \right\rceil 0.44 = 3 + 1*0.44 = 3.44 \quad (4.18)$$

logo, $RwcB=3.44 < D_B$, e cumpre com o prazo.

Desta forma fica provado que os requisitos temporais de todas as tarefas são cumpridos contribuindo para o bom funcionamento do sistema.

Uma das tarefas mais importantes do sistema *DataLogger* é o armazenamento de dados, por isso pretende-se explicar os métodos que foram utilizadas para a escrita de dados na memória NV da sbRIO. Foram desenvolvidas duas formas de armazenamento de dados, a escrita de dados em ficheiros com formatos *.tdms e *.bin.

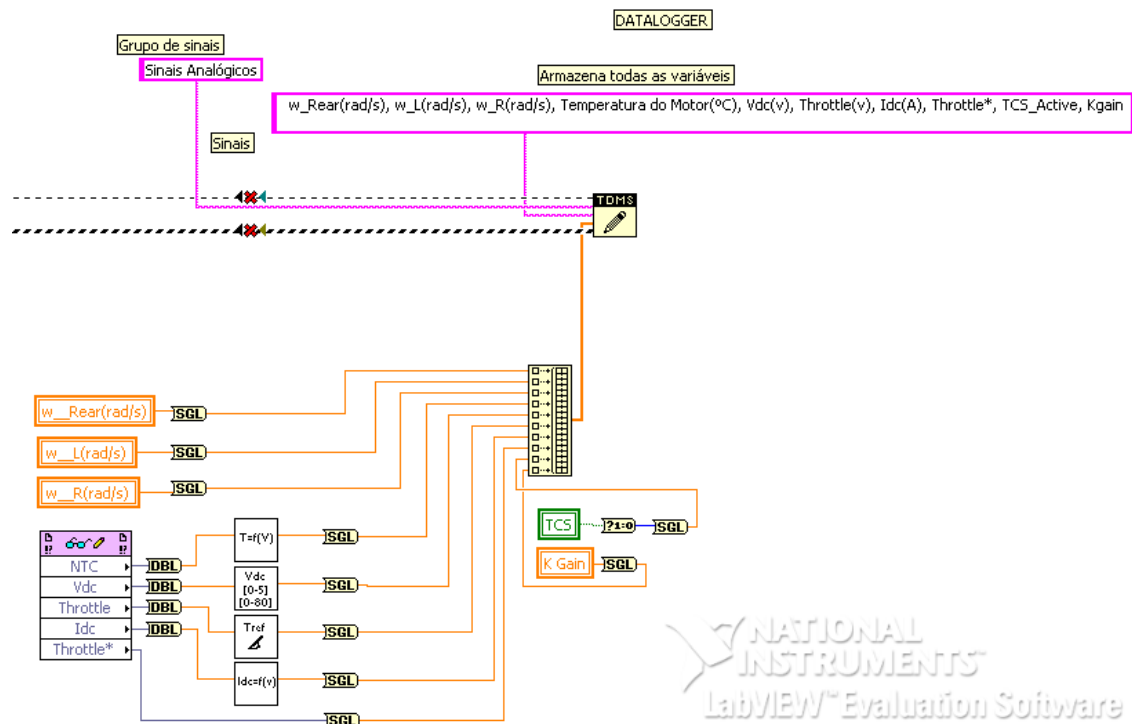


Figura 4.21 - Excerto da tarefa de armazenamento de dados num ficheiro .tdms.

A escrita de dados num ficheiro .tdms é feita de forma periódica, executando o programa que se encontra na Figura 4.21. Nessa figura, pode-se ver um bloco de escrita de alto nível (TOMS), que permite além de armazenar os dados, definir as propriedades dos sinais,

nomeadamente, o nome do sinal e o grupo de sinais a que pertence. Esta informação é importante para ser utilizada na identificação dos sinais quando é feita a leitura do ficheiro. Outra característica é a possibilidade de se poder atribuir propriedades ao ficheiro, ver Figura 4.23. Neste trabalho foi importante a utilização destas propriedades, para definir o cabeçalho do ficheiro no início da tarefa, como por exemplo a hora de início do armazenamento de dados, a frequência de amostragem dos dados, que é uma informação relevante a obter quando é feita a análise de dados.

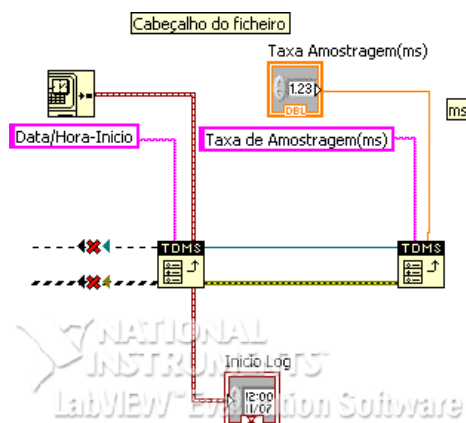


Figura 4.22 - Excerto da parte inicial da tarefa de armazenamento de dados num ficheiro .tdms, construção do cabeçalho

A escrita de dados num ficheiro .bin, é feita de forma periódica executando o programa da Figura 4.23. Esta forma de escrita não permite definir directamente propriedades dos sinais ou do ficheiro, como a utilização dos blocos TDMS, sendo apenas armazenados os dados.

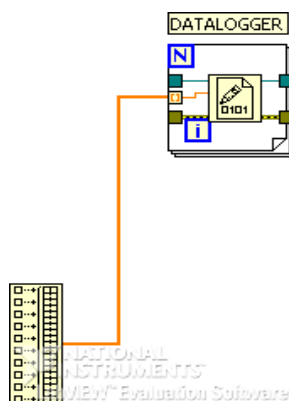


Figura 4.23 - Excerto da tarefa de armazenamento de dados num ficheiro .bin.

No entanto o programa poderia ser alterado de forma a que a informação adicional (cabeçalho e nome dos sinais), fosse armazenada no ficheiro .bin, mas isso iria complicar demasiado a solução quer na escrita quer na leitura em relação à que foi desenvolvida com o ficheiro .tdms.

Com todas as vantagens enunciadas quer na leitura quer na escrita dos ficheiros .tdms, optou-se por se utilizar esses métodos na versão final do *DataLogger*.

4.3.3.3 - Exportação dos dados para o *Matlab*

O *Matlab* é uma ferramenta muito importante na análise de dados, pois permite fazer cálculos matriciais, representar gráficos, desenvolver scripts, enfim, uma série de funcionalidades que são importantes utilizar para processar os dados e apresentar os resultados. Nesse sentido, foi acrescentado à aplicação de leitura o bloco da Figura 4.25, que permite exportar os dados para o *Matlab*. Esse bloco escreve os dados num ficheiro com formato ASCII, que é carregado no *Matlab* como um *array* de dados.

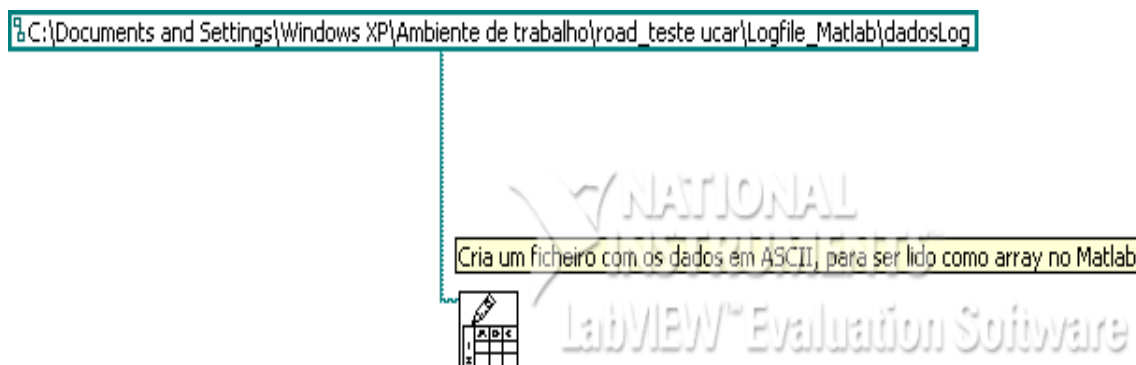


Figura 4.25 - Bloco *write to spreadsheet file*

4.4 - Implementação do *DataLogger*

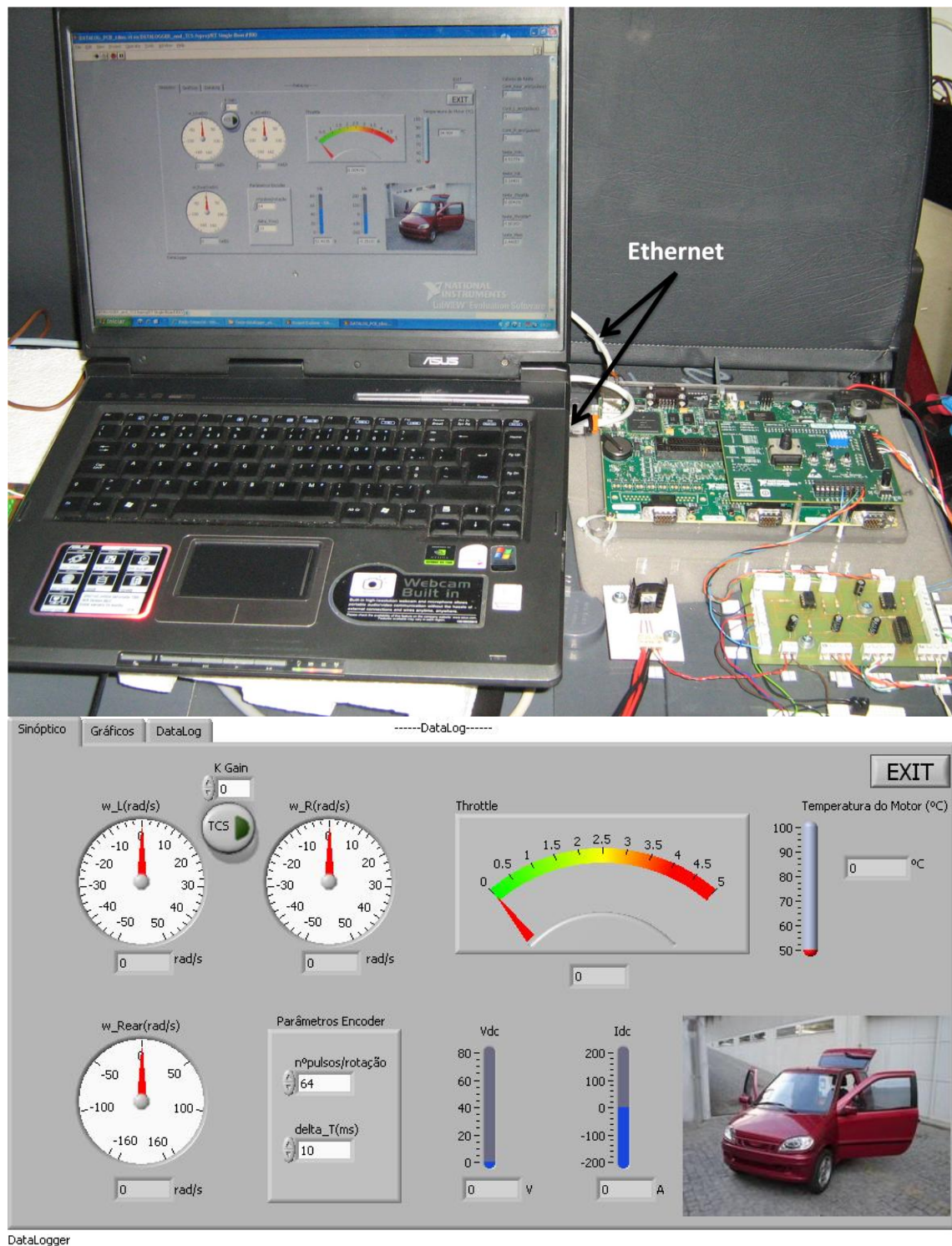


Figura 4.26 - Integração do sistema *DataLogger*

A Figura 4.26 mostra o resultado da integração do sistema *DataLogger*, em que o PC se encontra ligado por cabo *Ethernet* à placa sbRIO. Nesta situação é possível visualizar as grandezas em tempo real, mas se o objectivo é apenas armazenar os dados durante os

ensaios, o PC é ligado inicialmente apenas para fazer a configuração do armazenamento de dados (ver Figura 4.27), e de seguida o cabo pode ser desligado.

No final dos ensaios o PC é ligado novamente à sbRIO, e os dados armazenados na sbRIO são descarregados por FTP.

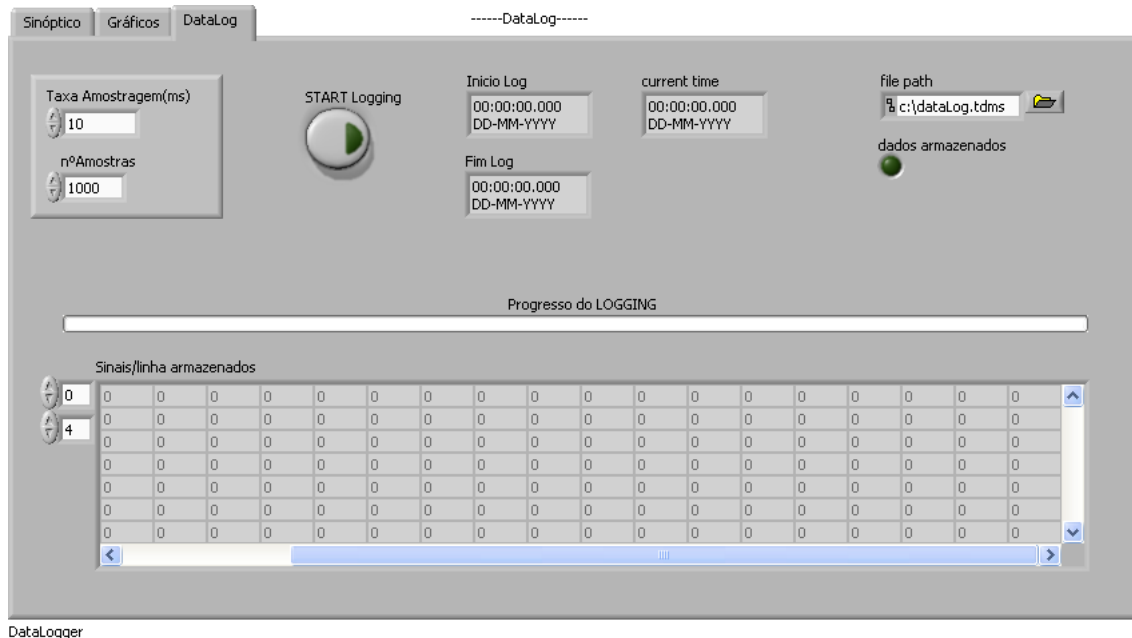


Figura 4.27 - Configuração para armazenar os dados na sbRIO

4.5 - Conclusões

Neste capítulo foram apresentadas as partes mais relevantes no desenvolvimento do sistema *DataLogger*. Em termos de HW foi descrito o projecto do circuito de condicionamento de sinal, e apresentado o resultado final da placa em PCB, integrada com a sbRIO e os sensores.

A nível de SW, foram descritas as várias aplicações desenvolvidas, indicando as suas funcionalidades, e a forma como elas interagem entre si.

Na aplicação de aquisição de dados foi determinada a velocidade máxima que a aplicação pode medir de forma correcta, chegando à conclusão que o sistema não terá problemas. Na aplicação de processamento e armazenamento de dados foi feita uma análise para verificar se as tarefas cumpriam com os requisitos temporais, para validar o bom funcionamento do sistema. Em relação às aplicações de leitura/escrita, foram apresentadas as principais vantagens da utilização das aplicações com formato .tdms face aos ficheiros .bin, como sendo mais robustos às alterações efectuadas, e permitindo dar informação sobre o nome dos sinais, a frequência de amostragem, e a data do armazenamento dos dados. Foi acrescentado mais uma funcionalidade à aplicação de leitura, para permitir exportar os dados para o *Matlab*, uma vez que representa uma das principais ferramentas de análise de dados.

Por fim apresentou-se o resultado final do desenvolvimento do sistema *DataLogger*. Com esta plataforma desenvolvida, procederam-se aos testes para validar o funcionamento do sistema e que são apresentados na secção 6.2, com a análise do desempenho do veículo eléctrico.

O capítulo seguinte trata do projecto e implementação de um controlador de tracção no veículo eléctrico uCar.

Capítulo 5

Projecto e implementação de um controlador de tracção

5.1 - Introdução

Neste capítulo pretende-se descrever a fase de projecto, simulação e implementação de um controlador de tracção. Começa-se por determinar a dinâmica dos vários subsistemas que constituem o veículo uCar, de seguida apresentam-se métodos que podem ser utilizados para determinar certos parâmetros do veículo, como momentos de inércia e coeficientes de atrito viscoso. Por fim é apresentada a estratégia de controlo de tracção, com alguns resultados de simulação.

5.2 - Modelo do sistema uCar em *Simulink*

Nesta secção pretende-se determinar a dinâmica do veículo eléctrico uCar, começando por dividi-lo em vários subsistemas. Desta forma, torna-se mais fácil a análise da dinâmica de cada um e é mais intuitiva a ligação que existe entre eles. Este modelo será utilizado para simular o comportamento do sistema, permitindo a avaliação do desempenho do controlador e a respectiva validação antes da sua implementação.

5.2.1 - Modelo longitudinal do veículo

O modelo utilizado para simular a dinâmica longitudinal do uCar pretende reter o essencial do comportamento do veículo, e portanto a expressão utilizada é dada por

$$M \cdot \frac{dV}{dt} = F_t - R_a \quad (5.1)$$

onde M , V , F_t e R_a são, a massa do veículo, a velocidade, a força de tracção total e a resistência aerodinâmica do veículo. Esta expressão deriva de (2.1), com as seguintes

considerações: o veículo move-se em piso plano, as resistências rolantes dos pneus são suficientemente pequenas.

5.2.2 - Modelo Throttle/inversor/motor

O uCar possui já um sistema de controlo de baixo nível no que diz respeito ao controlo dos motores. Existe um inversor entre o sinal do acelerador (*Throttle*) e cada um dos motores, que se encarrega de fazer o respectivo controlo.

A grande dificuldade na determinação deste modelo, é que se desconhece por completo a estrutura de controlo por não ter sido facultada pelo fabricante, apenas é sabido que o controlo é feito em velocidade. Esta foi uma das grandes restrições na fase de modelização do sistema uCar.

No entanto, para uma primeira aproximação considerou-se que numa situação em aceleração a variação do sinal do acelerador poderá corresponder a uma variação do binário motor de forma proporcional e com um determinado atraso. Assim, nessa situação aproxima-se o sistema por um de 1ª ordem, dado por

$$\frac{T_m(s)}{V_{Throttle}(s)} = \frac{K_t}{\tau s + 1} \quad (5.2)$$

onde K_t representa um ganho e τ um atraso imposto pela electrónica de controlo.

5.2.3 - Modelo motor/redutor/roda

O motor está acoplado à roda através de um redutor, havendo assim a necessidade de modelizar toda a cadeia de transmissão.

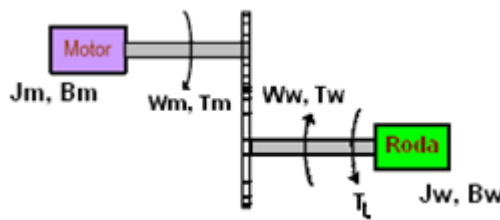


Figura 5.1 - Acoplamento motor/roda

Para o motor a roda pode ser vista como uma carga com uma determinada inércia e coeficiente de atrito, e que impõe um determinado binário resistente (ver Figura 5.1).

Considerando que a J_w e B_w são a inércia e o coeficiente de atrito da carga, neste caso a roda, e J_m e B_m os do motor, a dinâmica deste subsistema pode ser dada por

$$J_{eq} \frac{d\omega_m}{dt} = T_m - B_{eq} \omega_m - a T_L \quad (5.3)$$

onde $J_{eq} = (J_m + a^2 J_w)$, $B_{eq} = (B_m + a^2 B_w)$, ω_m , T_m , T_L são, a inércia e o coeficiente de atrito equivalentes de todo o conjunto, a velocidade angular do motor, o binário motor, o binário devido à força de tracção exercida na roda, e a razão de acoplamento, respectivamente. A razão de acoplamento, considerando um rendimento de 100% no sistema de engrenagens, é dada por

$$a = \frac{T_m}{T_w} = \frac{\omega_w}{\omega_m} \quad (5.4)$$

onde ω_w e T_w são a velocidade angular da roda e o binário aplicado à roda.

No que diz respeito ao binário de carga T_L , este está relacionado com o efeito da força de tracção que é exercida na roda, que resulta da interacção pneu/estrada. Desta forma o binário de carga é dado por $T_L = R_w F_{tp}$, em que R_w é o raio da roda e F_{tp} é a força de tracção exercida no pneu (explicada a seguir).

5.2.4 - Modelo pneu/estrada

A força de tracção F_{tp} exercida no pneu depende do coeficiente de fricção μ que existe entre o pneu e a estrada, e é dada pela seguinte fórmula:

$$F_{tp} = \mu N_p \quad (5.5)$$

onde N_p é a força normal que o plano exerce na roda e é dado por $N_p = W_p$ (ver Figura 5.2).

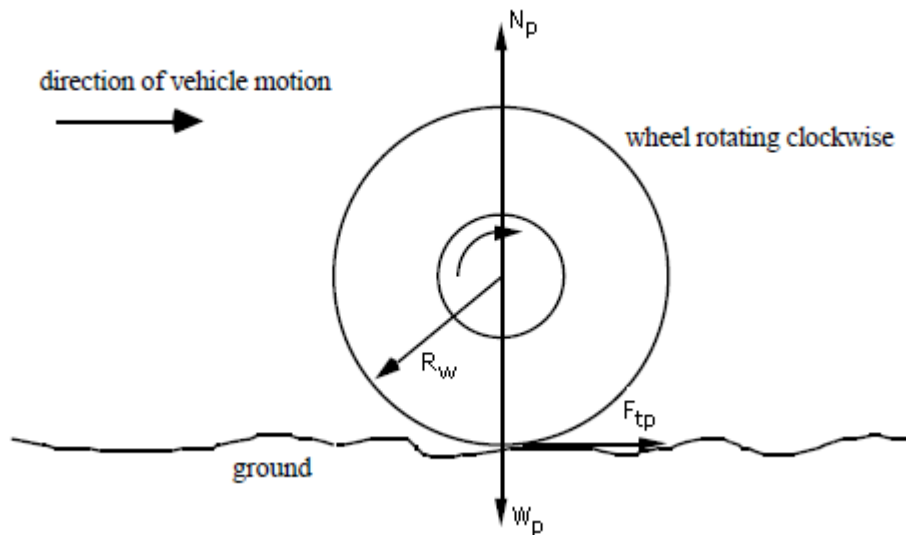


Figura 5.2 - Modelo pneu/estrada

O coeficiente de fricção μ , como foi visto na secção 2.3, depende principalmente do deslizamento λ da roda e das condições de superfície de estrada. Foi utilizada a *Magic Formula* dada por (2.12), para modelizar a relação μ - λ a utilizar na simulação.

5.2.5 - Diagrama de blocos do sistema uCar

Com as dinâmicas dos vários subsistemas, foi construído o diagrama de blocos do sistema uCar em *simulink* (ver Figura 5.3). Este modelo representa o uCar com apenas uma roda frontal com tracção, ($F_t = F_{tp}$) pois o controlo será feito apenas a uma roda.

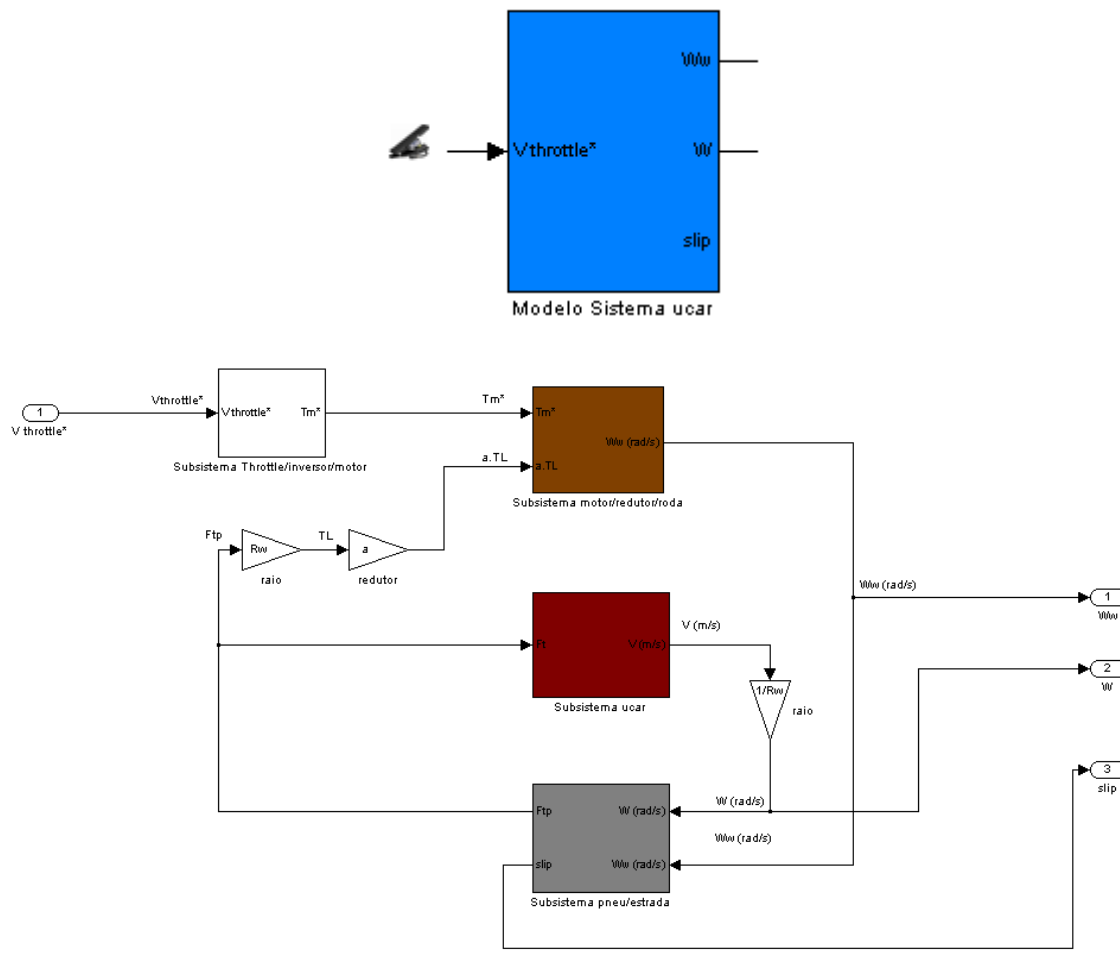


Figura 5.3 - Diagrama de blocos do sistema

Depois de determinada a dinâmica do sistema global, é necessário conhecer os parâmetros do veículo. A secção seguinte indica os parâmetros que são necessários para a simulação, e possíveis formas de os determinar.

5.3 - Determinação dos parâmetros do veículo

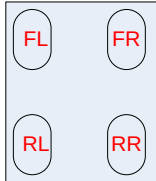
Na Tabela 5.1 encontram-se os parâmetros necessários para a simulação do modelo do sistema, em que estão definidos os valores dos parâmetros conhecidos, e especifica os que são necessários determinar.

Tabela 5.1 - Parâmetros do modelo do sistema

M	683kg
a	1/7
J_{eq}	Determinar
B_{eq}	Determinar
R_w	0.26m
N_p	1543.5N
K_t	Determinar
τ	Determinar

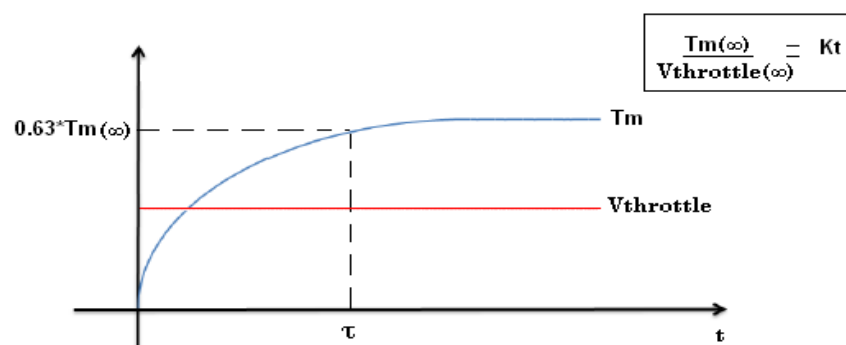
O controlo de tracção será feito apenas a uma roda, na roda esquerda frontal, portanto, conhecendo a distribuição do peso do uCar em cada roda, como é mostrado na Tabela 5.2, o cálculo da reacção normal nessa roda é dado por $N_p = 157.5 \cdot 9.8 = 1543.5\text{N}$. A massa M do uCar, foi obtido a partir da soma dos pesos distribuídos pelas 4 rodas.

Tabela 5.2 - Distribuição de peso no uCar

Secção	Peso (kg)	Porcentagem	Distribuição do peso (kg)	
Nose	311	45,5%	157,5	
Rear	372	54,5%		
Cross	351	51,4%	198,0	174,0
Total	683			

5.3.1 - Determinação de K_t e τ

Para se determinar a função de transferência que define a relação do sinal do acelerador e do binário motor, o ganho K_t e a constante de tempo τ devem ser obtidos experimentalmente. Para isso, poder-se-ia aplicar um sinal em degrau do acelerador e avaliar o comportamento do binário tal como se pode ver na Figura 5.4, sendo necessário para o efeito a medida do binário motor.

Figura 5.4 - Relação T_m - $V_{Throttle}$

Desta forma, o valor do ganho K_t seria dado pelo quociente do valor do binário e do sinal do acelerador em regime permanente, e a constante de tempo pelo instante em que o binário atinge cerca de 63% do seu valor em regime permanente.

5.3.2 - Cálculo experimental de J_{eq} e B_{eq}

Os valores de inércia e coeficiente de atrito viscoso equivalentes do conjunto motor/redutor/roda, são importantes para a simulação. De seguida apresentam-se alguns métodos que podem ser aplicados para determinar estes valores de forma experimental.

5.3.2.1 - Método em aceleração

Os parâmetros, J_{eq} e B_{eq} , podem ser obtidos a partir dum ensaio feito com o uCar suspenso, com a roda a rodar livremente, sem estar em contacto com a estrada ($T_L=0$). Assim, como se sabe, a equação (5.3) da dinâmica do conjunto deste subsistema, nestas condições passa a ser dada por

$$J_{eq} \frac{d\omega_m}{dt} = T_m - B_{eq}\omega_m, \quad (5.6)$$

pois deixa de haver força de tracção exercida na roda. Desta forma, conhecida a resposta de velocidade a um degrau de binário motor aplicado, obtém-se a constante de tempo J_{eq}/B_{eq} e o ganho $1/B_{eq}$ da função de transferência, obtendo-se assim os valores dos parâmetros J_{eq} e B_{eq} . Para este ensaio é necessário a medida de velocidade e de binário motor.

5.3.2.2 - Métodos em desaceleração

O momento de inércia J_{eq} pode ser determinado por desaceleração livre [15]-[16], desde que a energia cinética das partes rotativas seja dissipada por perdas lentas (*idling losses*) durante o tempo t_1 de desaceleração

$$J_{eq} \frac{\omega_m^2}{2} = \int_0^{t_1} P_{id} dt \quad (5.7)$$

onde J_{eq} é o momento de inércia do conjunto motor/redutor/roda, ω_m a velocidade do motor no início da desaceleração, e P_{id} são as perdas lentas (*idling losses*).

Para usar este método, regista-se a curva $\omega_m(t)$ no período de desaceleração, onde t é o tempo corrente, junto com a curva $P_{id}(\omega_m)$, o qual representa as perdas lentas, e estas curvas são usadas juntas para determinar $P_{id}(t)$, que são as perdas lentas relativas ao tempo de desaceleração. A integração gráfica dessas perdas dá então o trabalho A consumido, e portanto o momento de inércia é dado por

$$J_{eq} = \frac{2A}{\omega_m^2} \quad (5.8)$$

Este método é um dos mais precisos [16], mas o uso prático é restringido pela complexidade de integração gráfica, por esta razão, é prática comum empregar-se um método simplificado na base em que o binário dinâmico é igual ao binário representativo das perdas lentas (M_{id}) durante o período de desaceleração, esse método é dado por

$$J_{eq} \frac{d\omega_m}{dt} = M_{id} \quad (5.9)$$

A derivada do lado esquerdo da equação, é obtida por construção gráfica da tangente à curva de desaceleração $\omega_m(t)$ para um dado ω_m . A intersecção entre a tangente e a abcissa é usada com a perpendicular do ponto de contacto para definir o tempo nominal T de desaceleração, onde

$$\frac{d\omega_m}{dt} = \frac{\omega_m}{T} \quad (5.10)$$

O valor de M_{id} para a velocidade ω_m pode ser dada por

$$M_{id} = \frac{P_{id}}{\omega_m} \quad (5.11)$$

Desta forma, obtém-se o valor da inércia dada por

$$J_{eq} = P_{id} \frac{T}{\omega_m^2} \quad (5.12)$$

No entanto, este método de determinação de J_{eq} é aproximado, porque há alguma incerteza na posição da tangente à curva, para melhorar a medida da inércia, determina-se a mesma para vários valores de velocidade e calcula-se a média.

No que diz respeito à obtenção do valor de B_{eq} , este poderia ser estimado a partir da curva de $P_{id}(\omega_m)$, aproximando-a pela relação $P_{id}(\omega_m) = B_{eq} \omega_m^2$.

5.4 - Resultados experimentais dos parâmetros do uCar

As constantes da relação do binário e do acelerador não foram obtidas por não haver condições para medir o binário motor.

Em relação aos parâmetros do subsistema motor/redutor/roda, o objectivo era utilizar um dos métodos experimentais em desaceleração, explicados anteriormente, para determinar os parâmetros de inércia e coeficientes de atrito equivalentes do conjunto. Acabou-se por não

efectuar os ensaios, por falta de meios, nomeadamente a impossibilidade de se medir a potência de perdas $P_{id}(\omega_m)$.

Neste contexto, foram utilizados *parâmetros* por defeito no modelo do veículo.

5.5 - Projecto do controlador de tracção

Nesta secção pretende-se descrever a forma como foi formulado o problema para projectar um controlador de tracção, apresentar o modelo de controlo e o seu funcionamento.

O modelo do sistema uCar que foi obtido na secção 5.2 pretende representar a dinâmica do veículo com 1 entrada, o acelerador (*throttle*), e duas saídas, a velocidade da roda e do veículo. O acelerador (*throttle*) é o sinal de comando do veículo uCar que é manipulado pelo condutor, e que é enviado para o controlo de baixo nível do motor. O veículo com esta dinâmica pode estar sujeito à perda de aderência que pode levar à instabilidade no movimento longitudinal. Desta forma surge a necessidade de se projectar e integrar um controlador de tracção no sistema uCar. Para isso é necessário englobar uma camada de controlo de alto nível (ver Figura 5.5) que processe o sinal de comando do acelerador e o envie para a camada de controlo de baixo nível, de forma a que o sistema se torne estável. De seguida é explicado a forma como foi desenvolvida a camada de controlo de tracção.

Como foi visto na secção 2.4, as variáveis importantes no controlo de tracção de um veículo, são o coeficiente de fricção μ e o deslizamento λ . Estas variáveis definem uma curva μ - λ que depende essencialmente das condições de superfície de estrada (ver Figura 5.6). Essa curva tem uma característica que define a zona em que o sistema perde a aderência e portanto deve ser evitada. Essa zona é a parte da curva em que o coeficiente de fricção diminui com o aumento do deslizamento, e o veículo ao entrar nessa zona a velocidade das rodas aumenta drasticamente, fazendo com que o veículo perca a aderência.

Neste contexto, o controlador a formular, tem como objectivo actuar no veículo de forma a suprimir o deslizamento para evitar que entre na zona perigosa ou se entrar, controlar o sistema para se afastar rapidamente dessa zona.

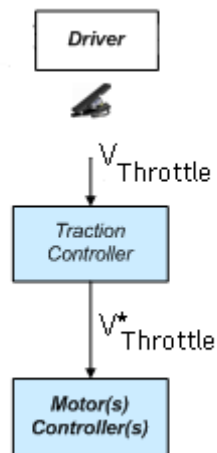
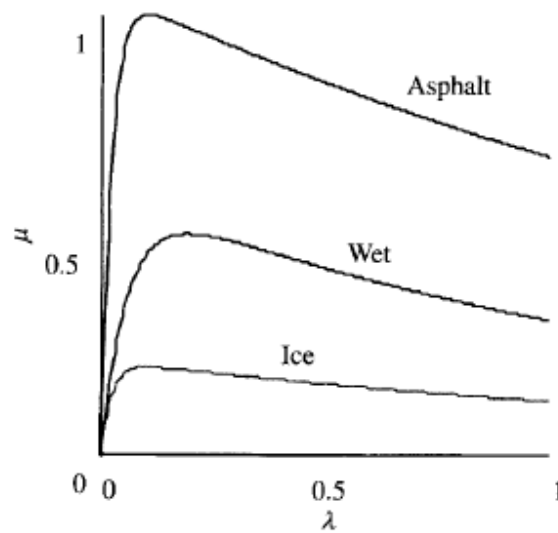


Figura 5.5 - Arquitectura das camadas de controlo

Figura 5.6 - Característica das curvas μ - λ [3]

Como o deslizamento é uma variável que depende da velocidade da roda e da velocidade do veículo, como já foi visto na secção 2.3, e é dada por

$$\lambda = \frac{\omega_w - \omega}{\max(\omega_w, \omega)}, \quad \max(\omega_w, \omega) \neq 0 \quad (5.13)$$

o controlador actuará no sistema a partir da informação das duas velocidades. Assim sendo a estratégia de controlo pensada foi a seguinte: colocar uma malha de controlo com um ganho proporcional, que processa a diferença das duas velocidades medidas, velocidade da roda e do veículo, e compensa essa diferença no sinal de comando do acelerador. O princípio de compensação é muito semelhante ao controlo MFC apresentado na secção 2.4, havendo um ganho que é ajustável. O diagrama do sistema de controlo em malha fechada pode ser visto na Figura 5.7.

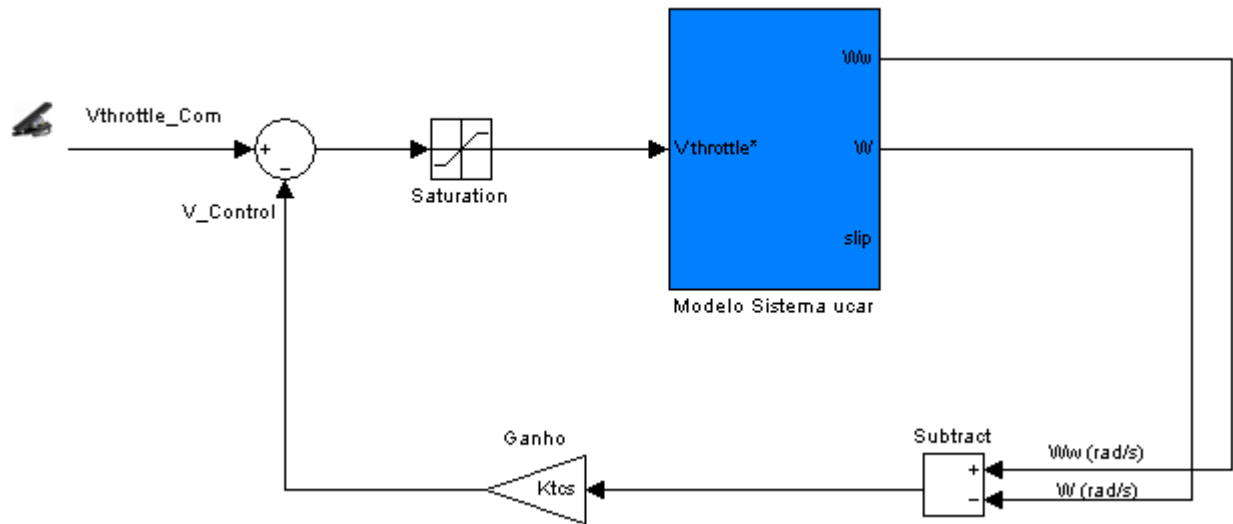


Figura 5.7 - Diagrama de blocos do controlador de tracção

Determinada a dinâmica do sistema e a estratégia de controlo, pretende-se de seguida apresentar resultados de simulação para validar o funcionamento do controlador.

5.6 - Resultados de simulação do sistema em *Simulink*

Nesta secção pretende-se demonstrar alguns resultados de simulação do sistema. Como foi dito na secção 5.4, não foi possível determinar experimentalmente os parâmetros do veículo que estavam em falta, e portanto foram utilizados valores por defeito (ver Tabela 5.3), ficando a simulação limitada a estes valores. De qualquer forma, como o controlador tem um ganho que é ajustável, na teoria pode-se avaliar a forma como a estratégia de controlo actua com a variação deste ganho, para ter uma ideia de como o sistema na prática se pode comportar.

Tabela 5.3 - Valores por defeito atribuídos aos parâmetros desconhecidos

J_{eq} (kgm ²)	B_{eq} (Nm.s/rad)	K_t	τ (ms)
0.1102	0.2	4	10

Para avaliar o desempenho do controlador, foi simulado uma situação de perda de aderência. A situação que foi considerada, simula o comportamento do veículo, quando se encontra num piso de reduzida aderência com gelo e este arranca com o binário máximo.

A curva utilizada na simulação, para modelizar a característica do coeficiente de fricção do piso com gelo, pode ser vista na Figura 5.8. Esta curva foi obtida utilizando a *magic formula*, já apresentada no capítulo 2, dada por

$$\mu = C_1 \sin(C_2 \tan^{-1}(C_3 \lambda - C_4(C_3 \lambda - \tan^{-1}(C_3 \lambda)))) \quad (5.14)$$

em que os coeficientes foram calculados de forma a aproximar a função aos valores típicos tabelados. A ferramenta utilizada no *Matlab* foi a *curve fitting*, que permitiu determinar os coeficientes que se encontram na Tabela 5.4.

Tabela 5.4 - Coeficientes da *magic formula* (piso com gelo)

C_1	C_2	C_3	C_4
0.2212	1.3059	27.7221	1.0426

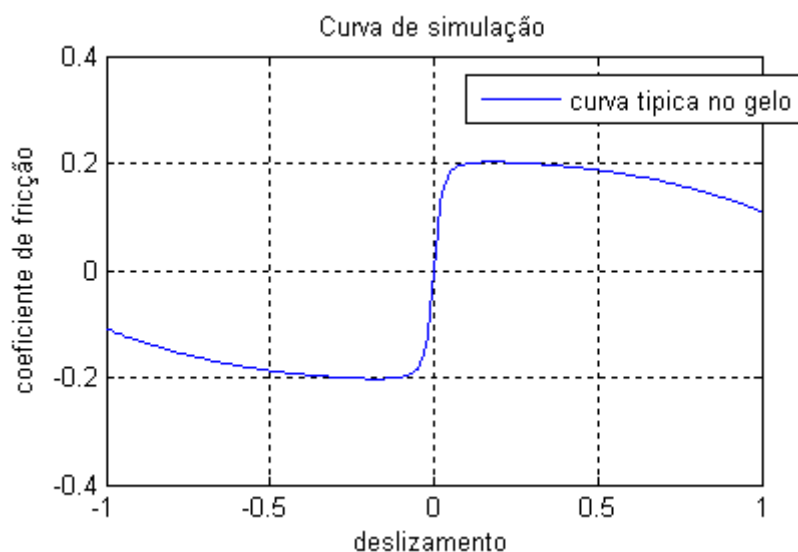


Figura 5.8 - Curva μ - λ do piso com gelo utilizada na simulação

De seguida pretende-se demonstrar e analisar os resultados de simulação, avaliando o efeito do controlador a partir da comparação dos resultados em malha aberta e em malha fechada.

5.6.1 - Comportamento do sistema sem controlo de tracção

Nesta situação o sistema encontra-se em malha aberta não havendo actuação do controlo de tracção. O veículo arranca com o acelerador no máximo, e como se pode observar na Figura 5.9, a velocidade da roda dispara drasticamente em relação à do veículo, e durante aproximadamente 2.7s o veículo encontra-se numa zona instável da curva μ - λ , atingindo nesse instante de tempo o valor de deslizamento (≈ 0.1) correspondente ao valor de fricção máxima (ver Figura 5.10). Nesse instante de tempo rapidamente, o sistema tende a entrar numa zona segura, aderindo ao piso. Face a este comportamento, é importante que o controlador de tracção projectado, actue no sistema suprimindo o deslizamento para que rapidamente este adira à estrada. De seguida vão ser apresentados resultados de simulação nas mesmas condições, mas com controlo de tracção.

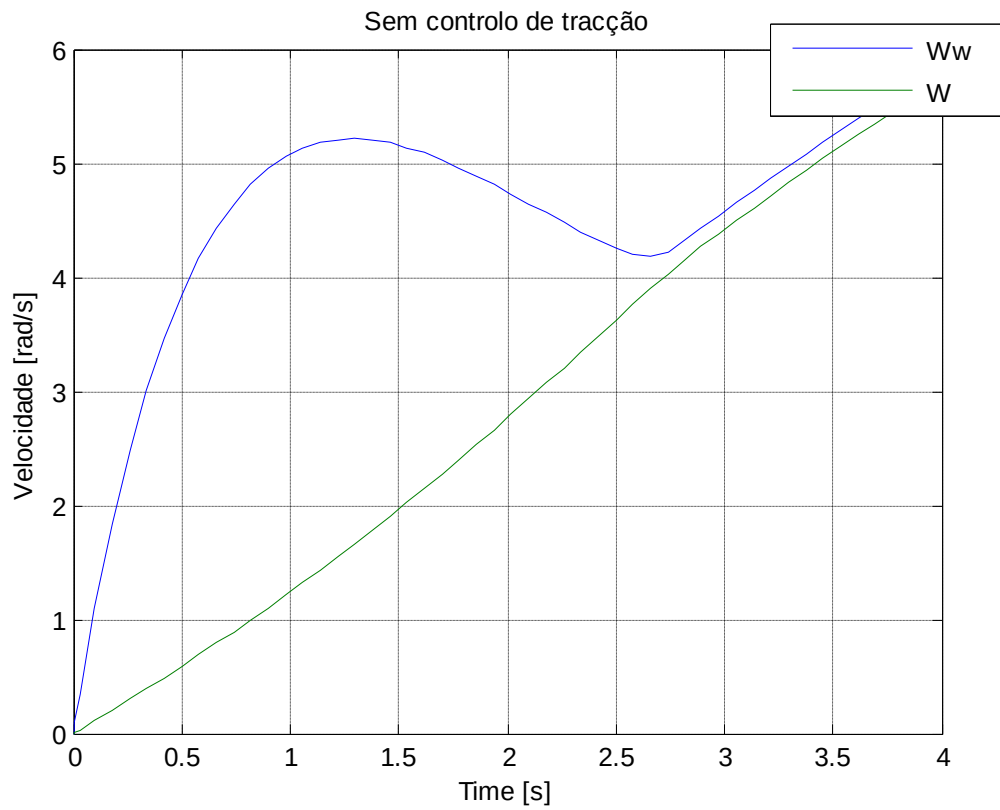


Figura 5.9 - Evolução das velocidades no arranque, sem controlo de tracção

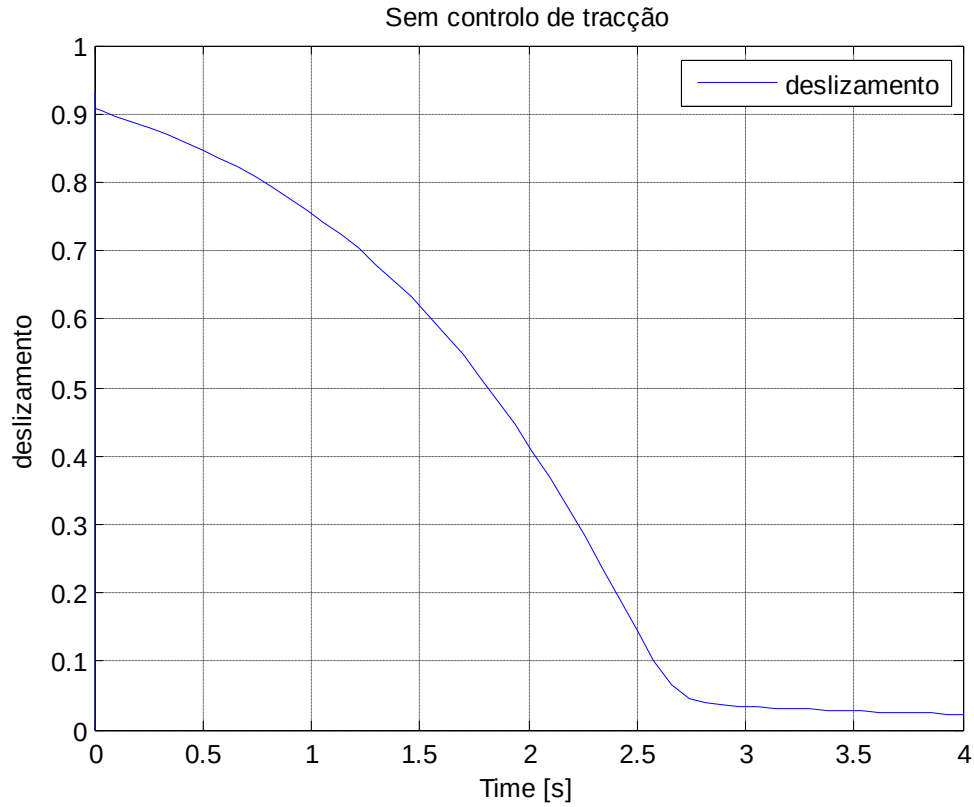


Figura 5.10 - Evolução do deslizamento, sem controlo de tracção

5.6.2 - Comportamento do sistema com controlo de tracção

Como foi visto anteriormente, o sistema em malha aberta numa fase de arranque com piso de reduzida aderência tende a manter-se na zona instável durante relativamente muito tempo. Com o controlador pretende-se mostrar o efeito compensador, quando este actua no sistema em malha fechada, avaliar o desempenho do controlo e comparar os resultados com diferentes ganhos.

Na Figura 5.11, pode-se ver a evolução das velocidades com a influência do controlo com ganho $K_{tcs}=1$, e comparando o comportamento resultante, com o da Figura 5.9, a velocidade da roda não aumenta de forma tão descontrolada, e o sistema atinge mais rapidamente a zona estável. Na Figura 5.13, pode-se ver que o deslizamento com o controlo é suprimido e diminui mais rapidamente em relação ao da Figura 5.10, entrando na zona estável aproximadamente no instante 1.4s. Na Figura 5.12, temos o sinal de comando ($V_{Throttle}$) e o sinal que efectivamente é enviado para o controlo dos motores ($V_{Throttle}^*$). O sinal de controlo dos motores $V_{Throttle}^*$ depende do sinal de comando e da diferença das velocidades multiplicadas pelo ganho do controlador de tracção. Desta forma o que acontece, é que quando a diferença de velocidades aumenta, o sinal $V_{Throttle}^*$ diminui, fazendo diminuir o binário dando origem a uma diminuição na variação de velocidade. Assim, como resultado obtém-se uma aderência ao piso mais rápida do que sem controlador de tracção. Nos resultados da Figura 5.11, além da aderência ao piso ser mais rápida, o desempenho em termos de aceleração do veículo foi ligeiramente melhorado.

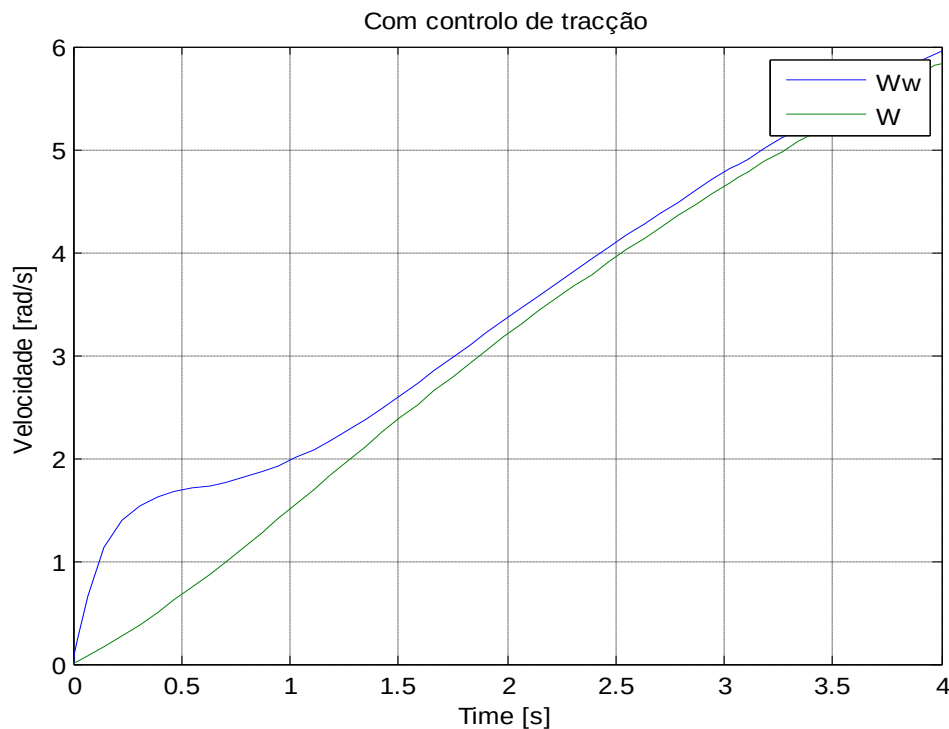
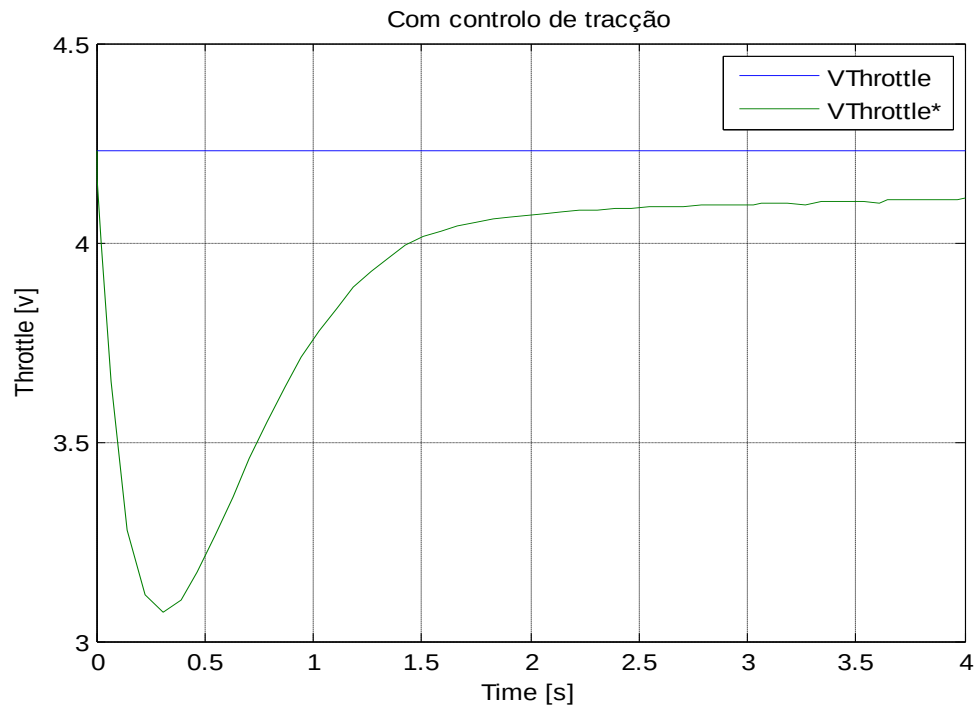
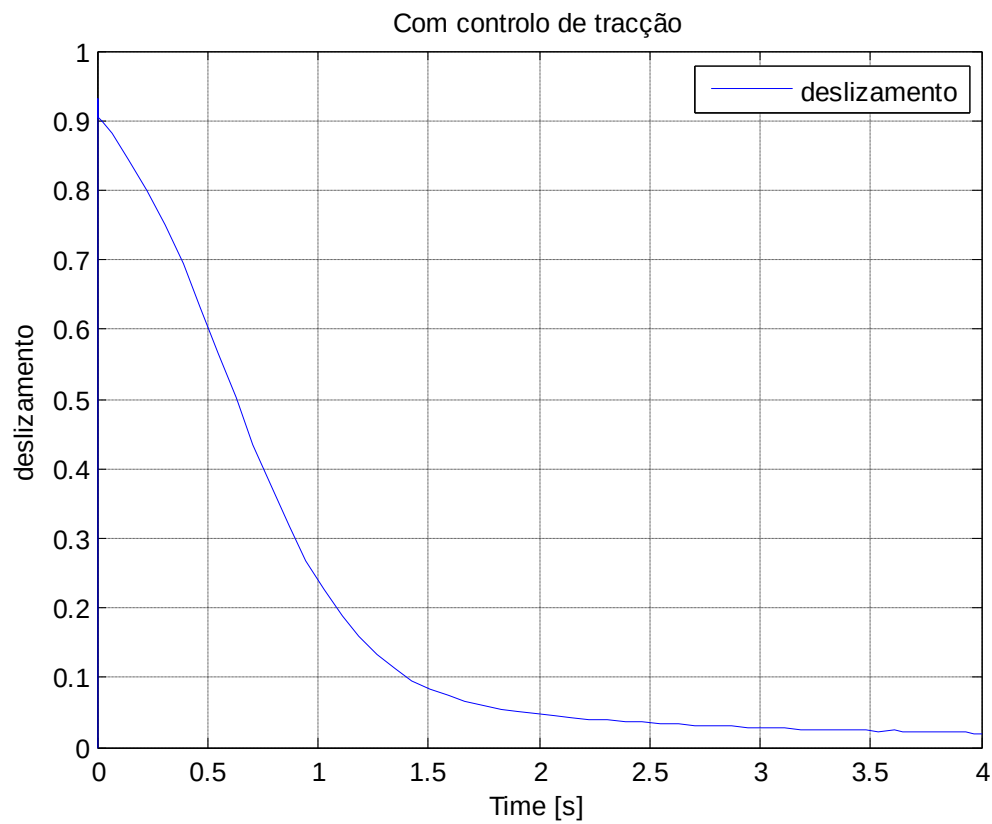


Figura 5.11 - Evolução das velocidades no arranque, com controlo de tracção (ganho $K_{tcs}=1$)

Figura 5.12 - Efeito compensador no sinal do *Throttle* (ganho $K_{tcs}=1$)Figura 5.13 - Evolução do deslizamento, com controlo de tracção (ganho $K_{tcs}=1$)

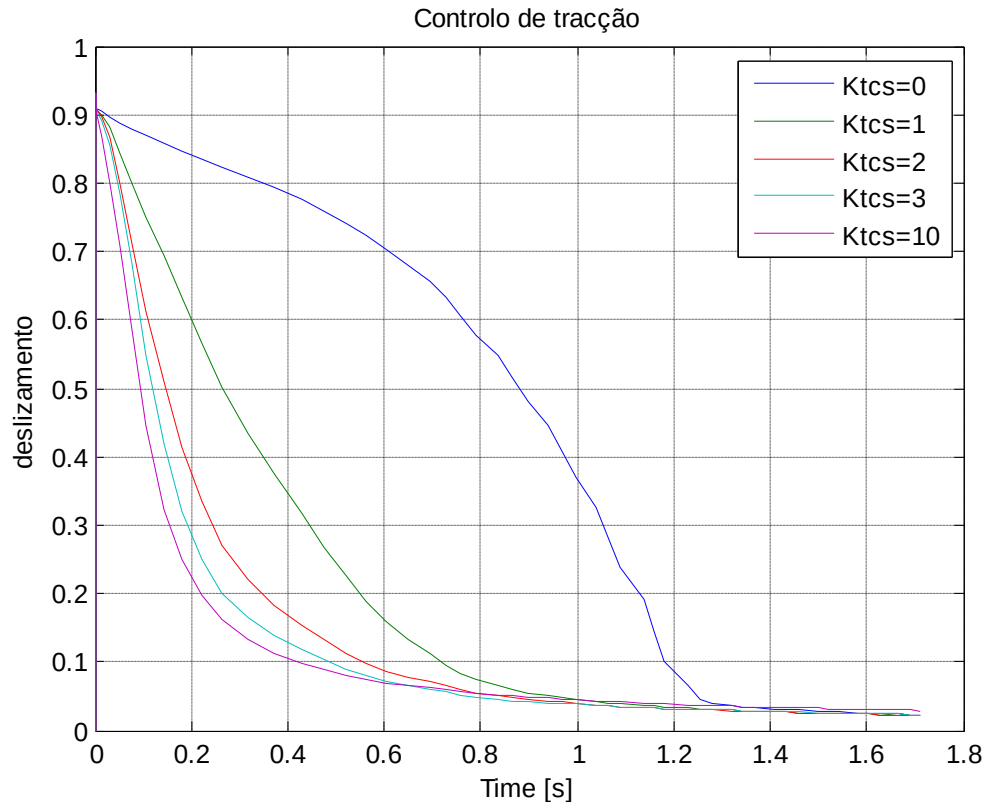


Figura 5.14 - Evolução do deslizamento para diferentes ganhos

Na Figura 5.14, temos a evolução do deslizamento para diferentes valores de ganho, e como se pode ver à medida que se foi aumentando o ganho, o deslizamento foi suprimido e o sistema tendeu a aderir mais rapidamente. Com ganho 10 o sistema em aproximadamente 400ms adere ao piso, e foi analisado que à medida que se aumentava o ganho acima de 10 para diminuir o tempo de resposta, havia uma diminuição no desempenho em aceleração do veículo. O que faz todo o sentido pois tende a suprimir demais o deslizamento na zona estável, diminuindo a força de tracção. Assim sendo, o ganho 10 permite uma boa aderência ao fim de 400ms e não prejudica fortemente o desempenho do veículo em aceleração depois de aderir ao piso (ver Figura 5.15).

Concluindo, o aumento do ganho do controlador permite diminuir o tempo de resposta, mas a partir de certo ponto começa a influenciar fortemente o desempenho do veículo em aceleração, pois suprime de tal forma o valor do deslizamento que a força de tracção “disponível” diminui.

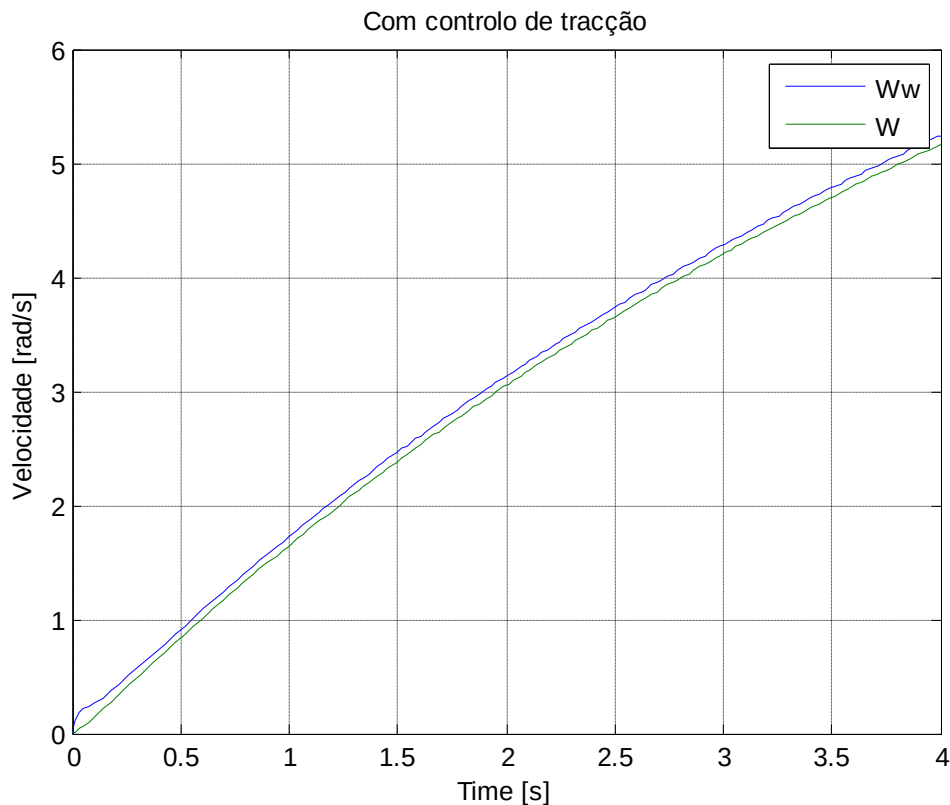


Figura 5.15 - Evolução das velocidades no arranque, com controlo de tracção (ganho $K_{tcs}=10$)

5.7 - Implementação do controlador de tracção

O controlador de tracção foi desenvolvido em *Labview* (anexo B) e implementado na FPGA da sbRIO, integrando o controlador no sistema *DataLogger*. O controlo foi feito apenas a uma roda, por não ter sido possível integrar um sensor de velocidade na roda traseira, que supostamente daria a velocidade do veículo. Assim desactivou-se o motor da roda direita frontal dando essa roda a velocidade do veículo, e fez-se o controlo na roda esquerda.

A frequência de controlo aplicada foi de 10ms, e foi necessário colocar filtros de média com janela de comprimento 3 para diminuir o nível de ruído das velocidades adquiridas.

5.8 - Conclusões

Neste capítulo foi apresentada toda a fase de projecto, e implementação do controlador de tracção no veículo eléctrico uCar. O primeiro passo foi determinar o modelo global do sistema, dividindo-o em vários subsistemas. Depois foram analisados os parâmetros do veículo que eram necessários para a simulação do sistema, indicando possíveis métodos para os determinar. No entanto por falta de meios, esses parâmetros não foram determinados, sendo utilizados parâmetros por defeito. Outra grande limitação nesta fase foi determinar o modelo do subsistema *throttle/motor*, pois há um controlo de baixo nível em velocidade que é

totalmente desconhecido e que não foi de maneira nenhuma facultado pelo fabricante. Desta forma assumiu-se um modelo de 1ª ordem entre o sinal do acelerador e o binário motor.

No entanto com todas estas limitações, foram feitas simulações do sistema num piso de reduzida aderência com gelo, numa situação de arranque, e avaliada a estratégia de controlo. Concluiu-se que o controlo de facto actua no sistema compensando o sinal de comando sempre que existe um deslizamento, e suprime esse deslizamento para que o sistema volte a aderir ao piso. Foram avaliadas várias situações com ganhos de controlo diferentes, e observou-se que o aumento do ganho diminuía o tempo de resposta do sistema para aderir ao piso, no entanto a partir de certo ganho o controlador passava a ter uma forte influência no desempenho do veículo em aceleração na zona estável, pois o deslizamento era de tal forma suprimido que a força de tracção disponível “diminuía”.

Após as simulações e avaliação da estratégia de controlo esta foi implementada no sistema uCar, para fazer o controlo a uma roda, fazendo experiências com o ajuste do ganho.

O capítulo seguinte trata da apresentação e análise dos resultados experimentais, quer para avaliar o efeito do controlador, quer para validar o funcionamento do sistema *DataLogger* desenvolvido.

Capítulo 6

Resultados experimentais

6.1 - Apresentação do veículo eléctrico uCar

Nesta secção o objectivo é apresentar as especificações do veículo eléctrico uCar, onde foram integrados o sistema *DataLogger* e o controlador de tracção.

O uCar é um veículo eléctrico multi-motor, com um chassis MicroCar, modelo Virgo, movido por 2 motores AC de indução, de baixa tensão (2.2kW, 26V, 63Arms, 1410rpms), instalados na parte dianteira do veículo (ver Figura 6.1), que são alimentados por um conjunto de 4 baterias de 12V, ligadas em serie.



Figura 6.1 - Veículo eléctrico uCar

Tabela 6.1 - Características do uCar

uCar		Motor (/Unidade)		Baterias	
		Pot. Max.	4kW	Tipo	Sonnenschein GF12110V-SON
Dimensões (LxWxH)	2580x1375x1390 mm	Binário max.	20N.m	Capacidade	110Ah
Peso	590 kg (com baterias)	Velocidade max.	3900 rpm	Tensão	12x4V
Raio da Roda (sem carga)	0,26m	Redutor	7:1	Massa	50kg/bateria

6.2 - Caracterização do desempenho do uCar como veículo eléctrico

Nesta secção pretende-se mostrar os resultados que foram obtidos em fases de teste, para validar o funcionamento do *DataLogger*, e fazer uma análise em termos de desempenho e de autonomia do veículo eléctrico uCar. O teste de estrada apresentado nesta secção, foi feito no parque da FEUP, cujo percurso pode ser visto na Figura 6.2.

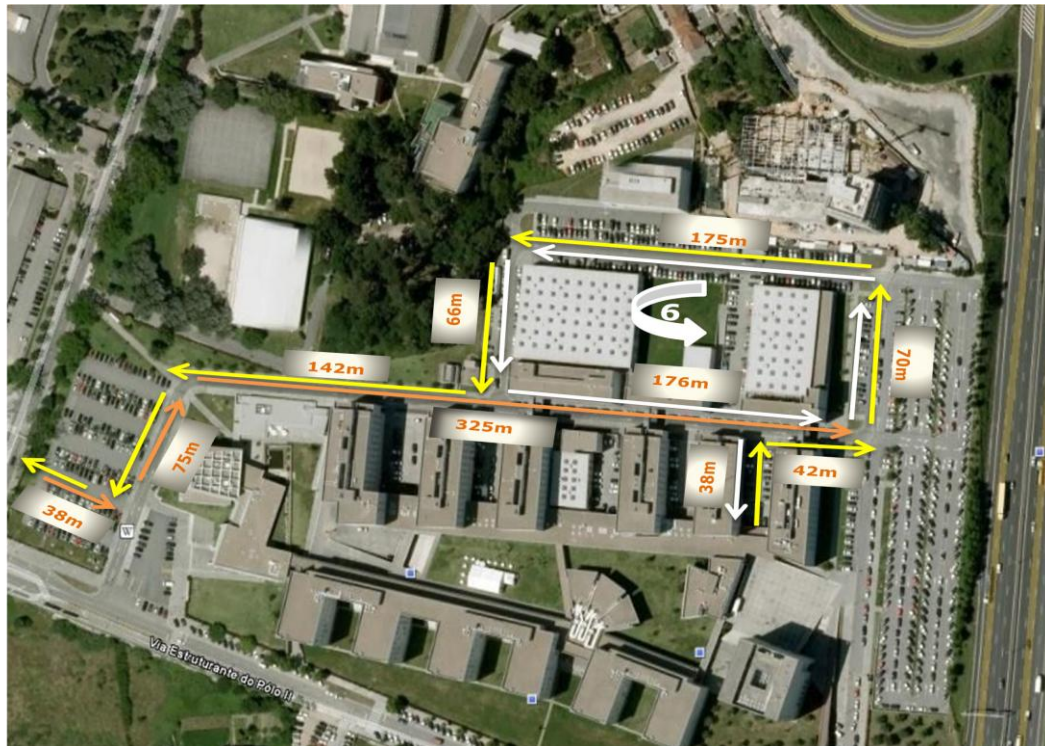


Figura 6.2 - Percurso realizado (FEUP)

O percurso inicia-se com as setas a amarelo depois, as laranjas e por fim as brancas. O local circunscrito pelas setas brancas foi repetido 6 vezes. As distâncias representadas foram calculadas de forma aproximada pelo *GoogleEarth*. A distancia total será comparada com a distancia que se obteve experimentalmente, a partir do integral da velocidade do veículo obtida durante o percurso. A distância que se obteve pelo *GoogleEarth* foi de aproximadamente 4km.

6.2.1 - Resultados gerais

Velocidade máxima: 38.1km/h

Velocidade média (com paragens): 8.3km/h

Velocidade média (sem paragens): 14.9km/h

Distancia percorrida (30min com paragem): 4.2km (através do integral de velocidade)

A distância percorrida em km foi determinada experimentalmente a partir da seguinte equação

$$\text{distancia} = \text{trapz}(\text{abs}(\frac{\omega_L + \omega_R}{2} R_w)) \frac{T_a}{1000} \quad (6.1)$$

onde ω_L , ω_R são a velocidade das rodas, R_w o raio da roda, T_a o período de amostragem, e $\text{trapz}()$ é uma função do *Matlab* que permite fazer integração gráfica.

Duração da aquisição: 30min

Tempo em movimento: 16.7min

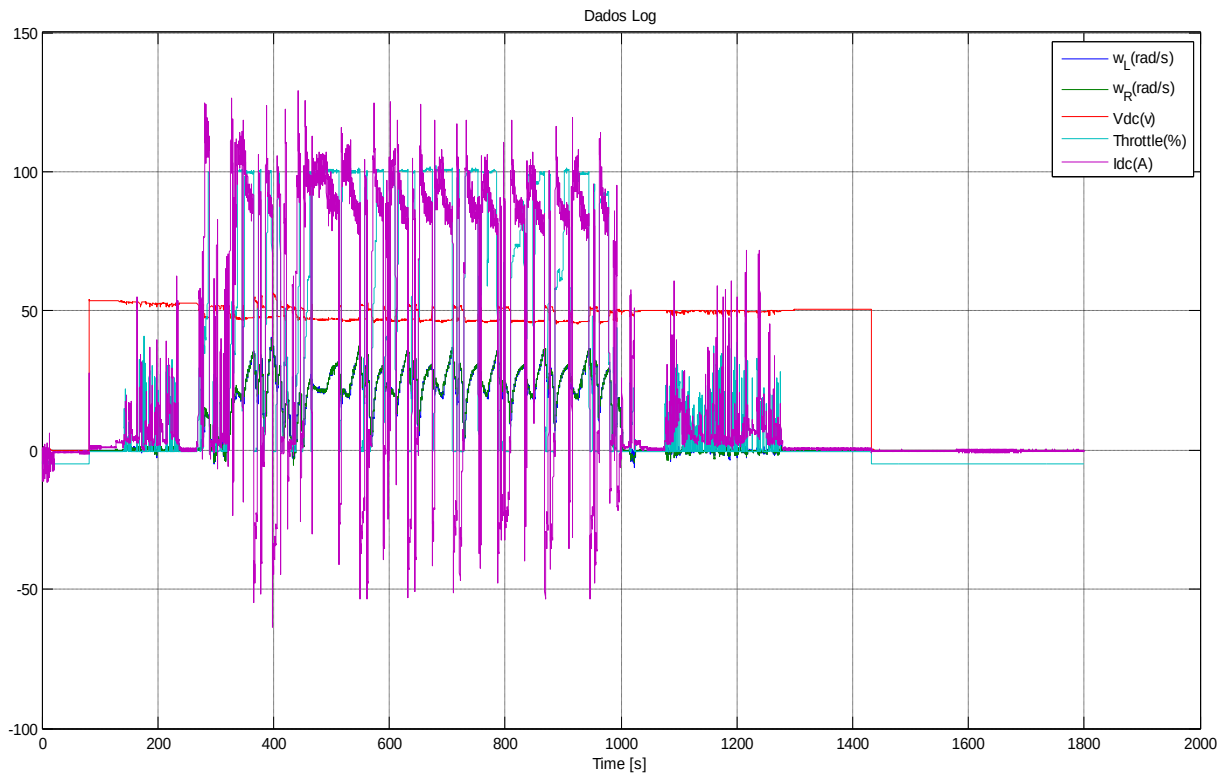


Figura 6.3 - Dados adquiridos (após filtragem digital)

Um filtro digital de média móvel com comprimento de janela 5, foi aplicado ao sinal I_{dc} , e um de comprimento 3 ao sinal V_{dc} .

6.2.2 - Aceleração

No teste de estrada que foi feito, uma das situações em que o veículo se encontra em aceleração pode ser vista na Figura 6.4. Nesse gráfico pode-se analisar o comportamento dos sinais do sistema, por exemplo, quando o sinal do acelerador (*Throttle*) aumenta até ao máximo, o sinal da corrente das baterias dispara, para o veículo acelerar. À medida que a corrente aumenta, o sinal de tensão nas baterias diminui pois a bateria atinge o limite de potência. Outro aspecto importante que convém salientar, é o facto de a corrente das

baterias diminuir, à medida que a velocidade aumenta a partir de determinado valor. Esta situação está relacionado com o funcionamento dos motores eléctricos na zona de enfraquecimento de campo que corresponde à zona de potência constante. Este controlo é feito pelos inversores, e à partida o motor deve entrar nessa zona a partir dos 50Hz. Para confirmar esta teoria, pode-se calcular essa frequência de forma aproximada, a partir da velocidade das rodas. A corrente começa a decrescer quando a velocidade das rodas ultrapassa, aproximadamente 22 rad/s, como a velocidade do motor é 7 vezes maior por causa do redutor, a velocidade mecânica dos motores é cerca de 154 rad/s, que representa $154/(2\pi) * 2 \approx 49\text{Hz}$ em termos de velocidade eléctrica, em que o factor 2 é o nº de pares de pólos dos motores.

Outro aspecto que se pode observar na parte inicial do gráfico, é a diferença do valor de velocidade nas duas rodas, que significa que o veículo nesta altura encontrava-se a fazer uma curva à esquerda, pois a velocidade da roda direita (ω_R) é maior do que a esquerda (ω_L).

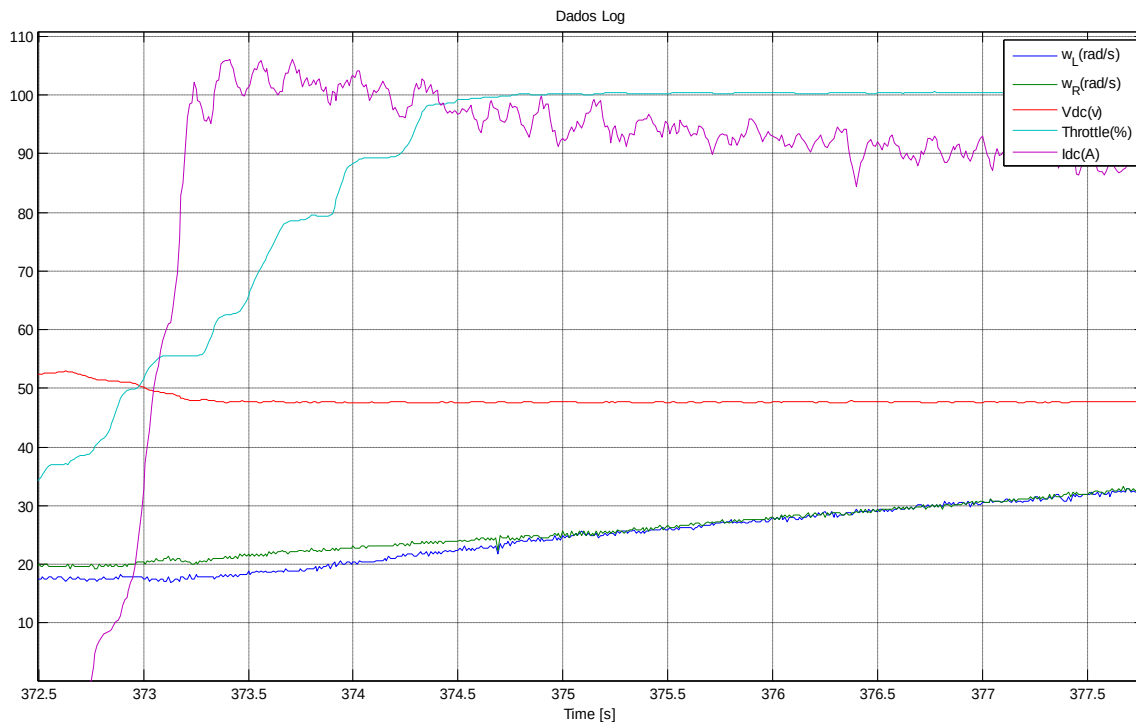


Figura 6.4 - Veículo em aceleração

6.2.3 - Frenagem

Uma das situações em que o veículo regenera energia nas baterias, pode ser observada na Figura 6.5. Nesta situação os motores do veículo encontram-se num estado de frenagem em que aproveitam a energia cinética do veículo para a transformar em energia eléctrica, recarregando desta forma as baterias.

A frenagem ocorre quando o sinal do acelerador diminui com o veículo em movimento, pois o controlo dos inversores tende a diminuir rapidamente a velocidade, tendo para isso que aplicar um binário de travagem. Por isso, na figura, quando o sinal do acelerador diminui bruscamente a corrente inverte o sentido. As baterias começam a recarregar aumentando a tensão das mesmas. À medida que ocorre a frenagem, a velocidade vai diminuindo e a corrente que recarrega as baterias também, pois a energia cinética do veículo diminui.

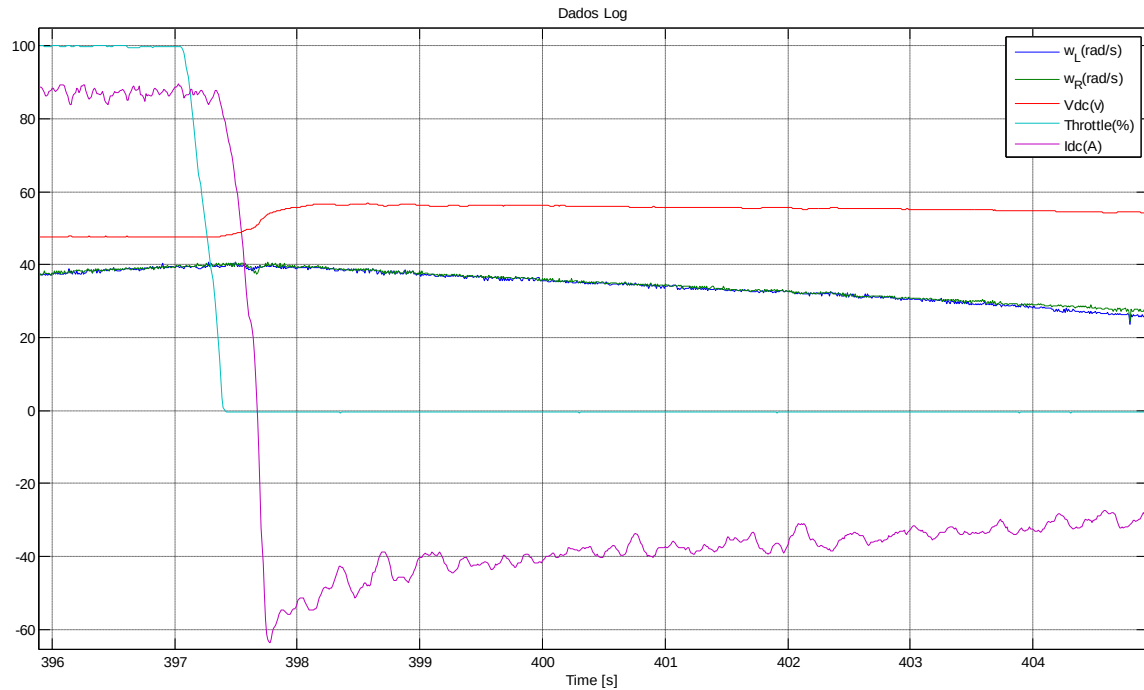


Figura 6.5 - Frenagem

6.2.4 - Potência

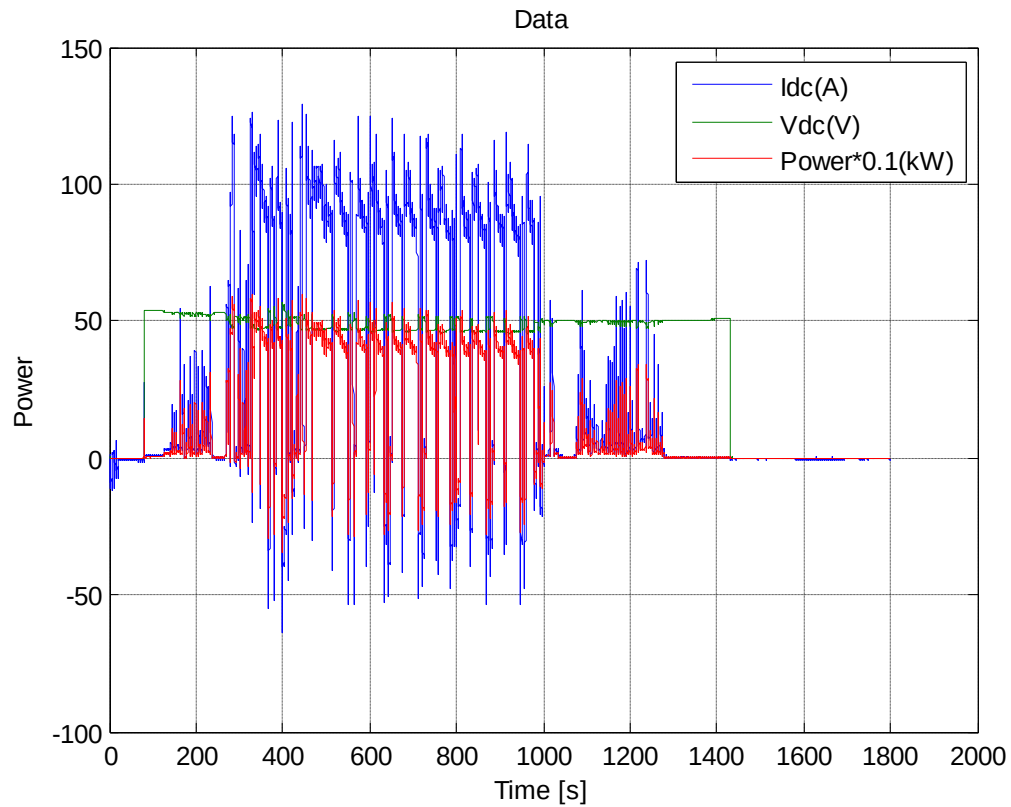


Figura 6.6 - Tensão, Corrente e Potência nas baterias ao longo do teste

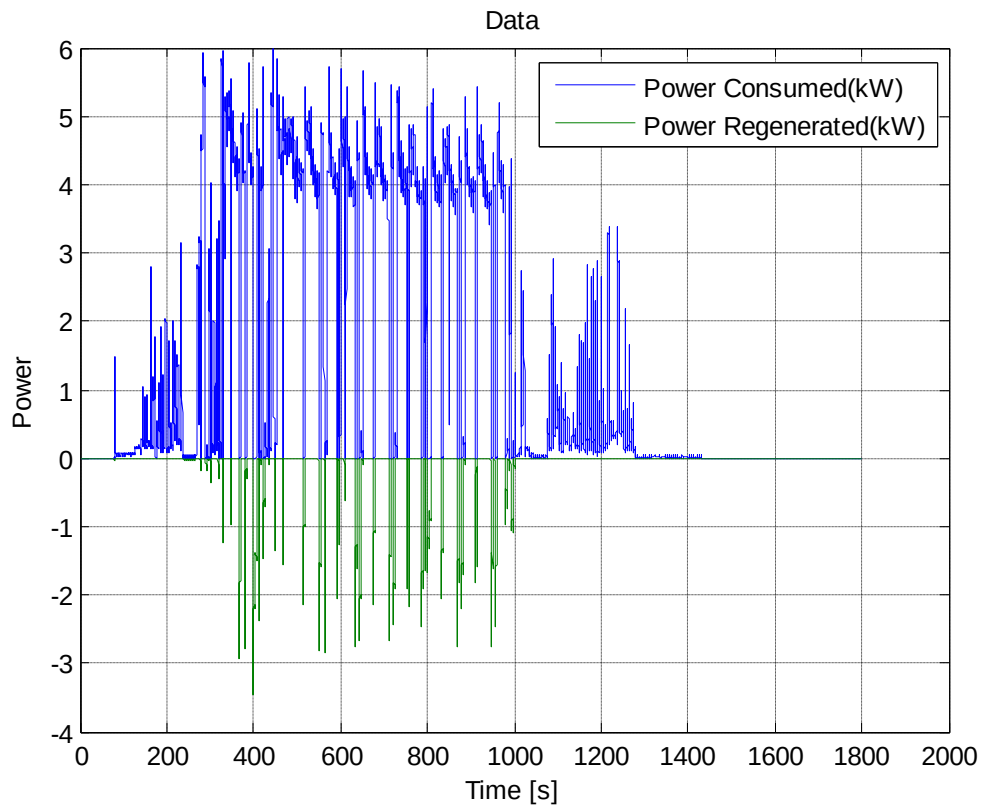


Figura 6.7 - Potências consumida e regenerada durante o teste

6.2.5 - Autonomia

Na Tabela 6.2 são apresentados os resultados energéticos, e uma estimativa do custo por 100km considerando a tarifa BTN (baixa tensão normal). No cálculo da energia fornecida pela rede, é tido em conta o coeficiente (0.8) que pretende de forma aproximada quantificar as perdas entre a energia que é fornecida pela rede à bateria, e a que efectivamente é fornecida pela bateria.

Tabela 6.2 - Dados energéticos

Energia consumida (kWh)	Energia regenerada (kWh)	Energia fornecida pelas baterias (kWh/100km)	Energia fornecida pela rede (kWh/100km)	€/100km
0.657	0.045	14.45	14.45/0.8	2.08

Com os dados da energia, nomeadamente a energia fornecida pelas baterias aos 100km, e com a informação da capacidade energética das baterias (ver Tabela 6.3), pode-se estimar a autonomia do veículo eléctrico uCar. Notar que esta estimativa depende fortemente do percurso realizado.

Tabela 6.3 - Capacidade das baterias

Baterias	
Capacidade (Ah)	Tensão (V)
110	12x4

A energia total que as baterias podem fornecer, no máximo, é aproximadamente $110\text{Ah} \cdot 48\text{V} = 5.28\text{kWh}$, e como a energia consumida pelo veículo é de $14.45 \text{ kWh}/100\text{km}$, significa que tem uma autonomia de aproximadamente $5.28\text{kWh} \cdot 100\text{km} / 14.45\text{kWh} = 36.5\text{km}$. Os dados resultantes da análise encontram-se resumidos na Tabela 6.4. O veículo apresentou neste teste de estrada, uma regeneração de cerca de 7%, e com um custo de consumo estimado em $2.08\text{€}/100\text{km}$. A autonomia do uCar foi estimada em cerca de 36.5km, mas é de notar que esta depende fortemente do percurso realizado. O sistema *datalogger* comportou-se de forma adequada durante o teste de estrada fornecendo, no final, os dados que aqui foram analisados. O sistema fica validado, por análise dos sinais que estão de acordo com o esperado.

Tabela 6.4 - Dados gerais

roadTest	
3/06/09	
Duração (min)	30
Distancia percorrida (km)	4.2
Vel. Média (com paragens) (km/h)	8.3
Vel. Média (sem paragens) (km/h)	14.9
Energia consumida (kWh)	0.657
Energia regenerada (kWh)	0.045
Eficiência (kWh/100km)	18.1
Regeneração (%)	6.8
Custos (€/100km)	2.08
Vel. Máxima (km/h)	38.1
VDC inicial (V)	53.9
VDC máxima (V)	56.8
VDC mínima (V)	45.4
IDC máxima (A)	128.8
IDC mínima (A)	-63.6
Pot. Máxima (kW)	5.99

6.3 - Comportamento do veículo em piso de reduzida aderência

Nesta secção apresentam-se os resultados dos testes do controlador de tracção. A situação criada para simular um piso com aderência relativamente baixa, foi utilizar uma chapa metálica onde a roda do veículo foi posta a patinar. Um método sistemático em piso plano para colocar a roda a deslizar melhor, foi inicialmente andar com o carro para trás dando algum balanço, e de repente inverter o sentido do movimento com a roda sobre a chapa, aplicando o sinal de comando máximo. Um outro teste que permitiu fazer com que a roda deslizesse foi colocar a chapa numa subida e subir a rampa com a roda por cima da chapa.

Para avaliar o efeito e desempenho do controlador, foram feitos testes nessas condições, com e sem controlo de tracção.

6.3.1 - Sem controlo de tracção

Na Figura 6.8, são mostrados os resultados da evolução das velocidades, a da roda (ω_w) e a do veículo (ω), nos testes que foram feitos com a chapa de metal numa subida. Nesta situação o veículo parte no início da subida com o sinal de comando no máximo, e sem controlo de tracção.

Como se pode observar o veículo arranca e quando passa por cima da placa de metal a velocidade da roda dispara rapidamente em relação à velocidade do veículo.

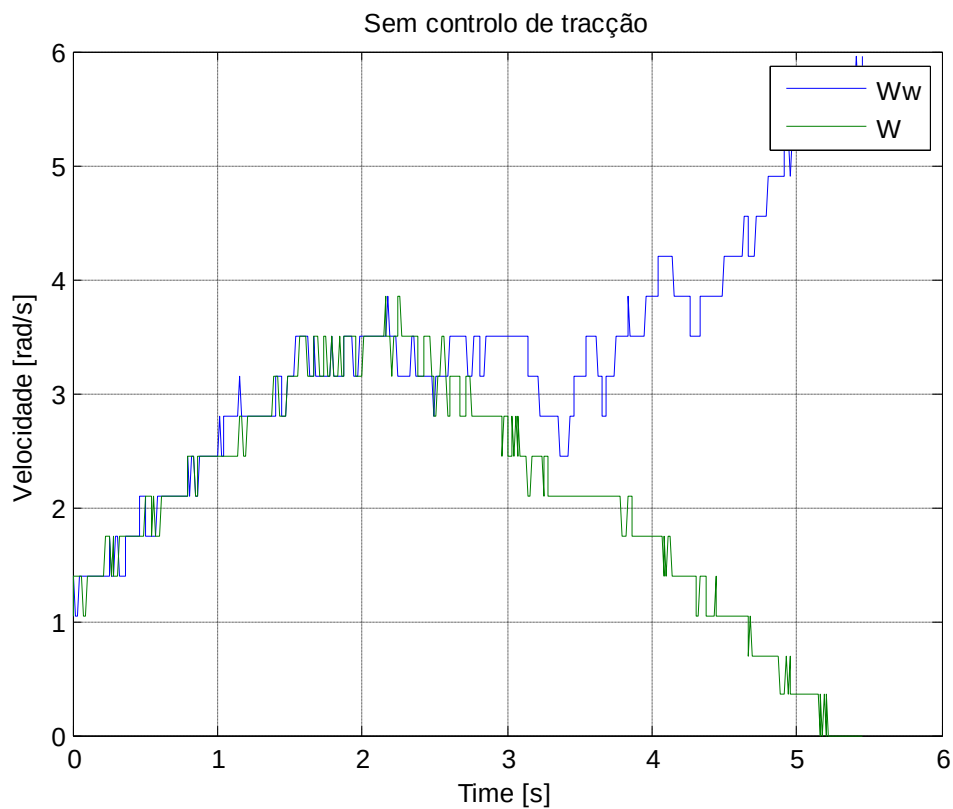


Figura 6.8 - Evolução das velocidades sem controlo de tracção (teste na subida)

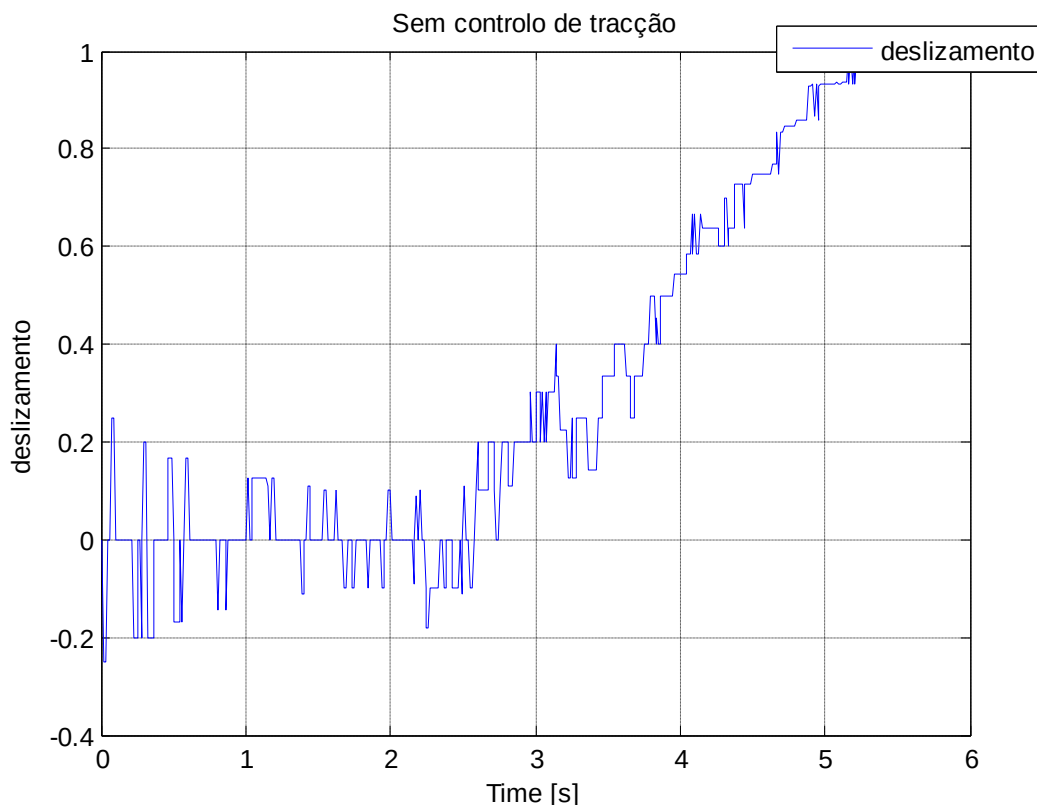
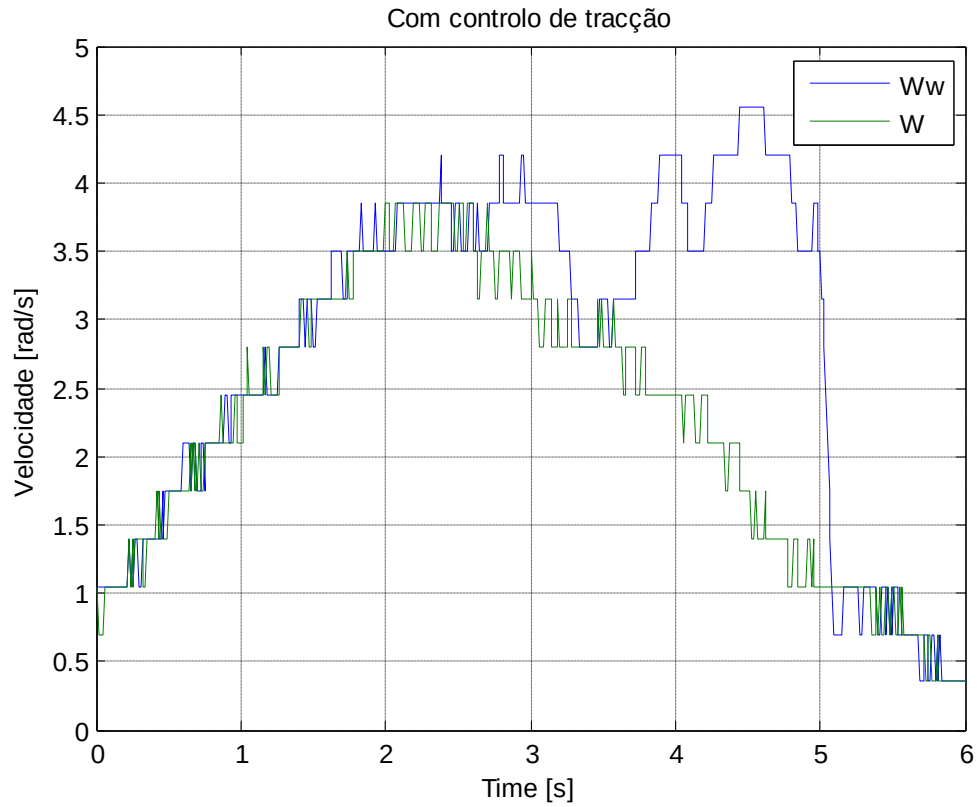
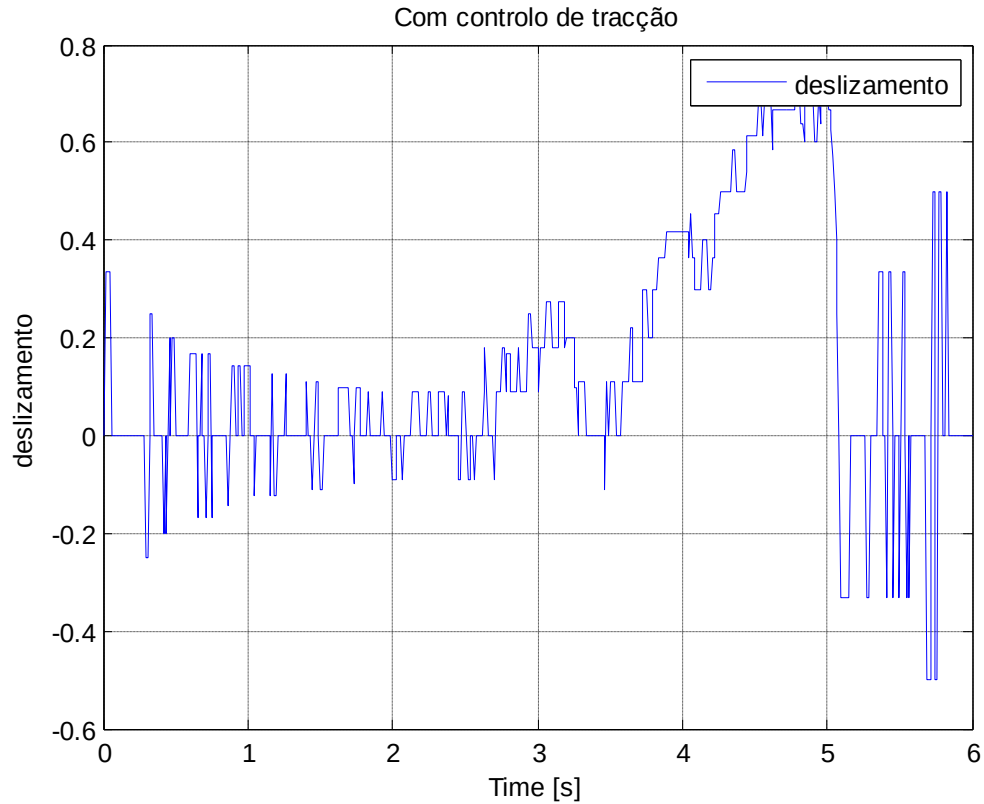


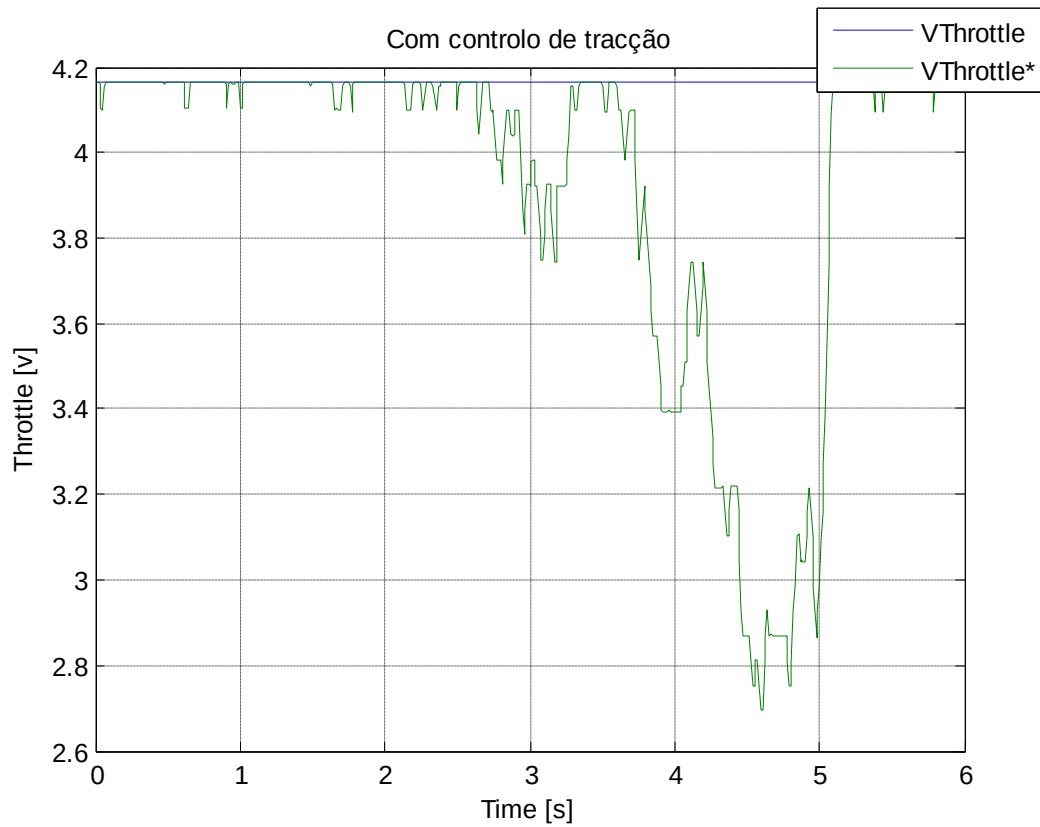
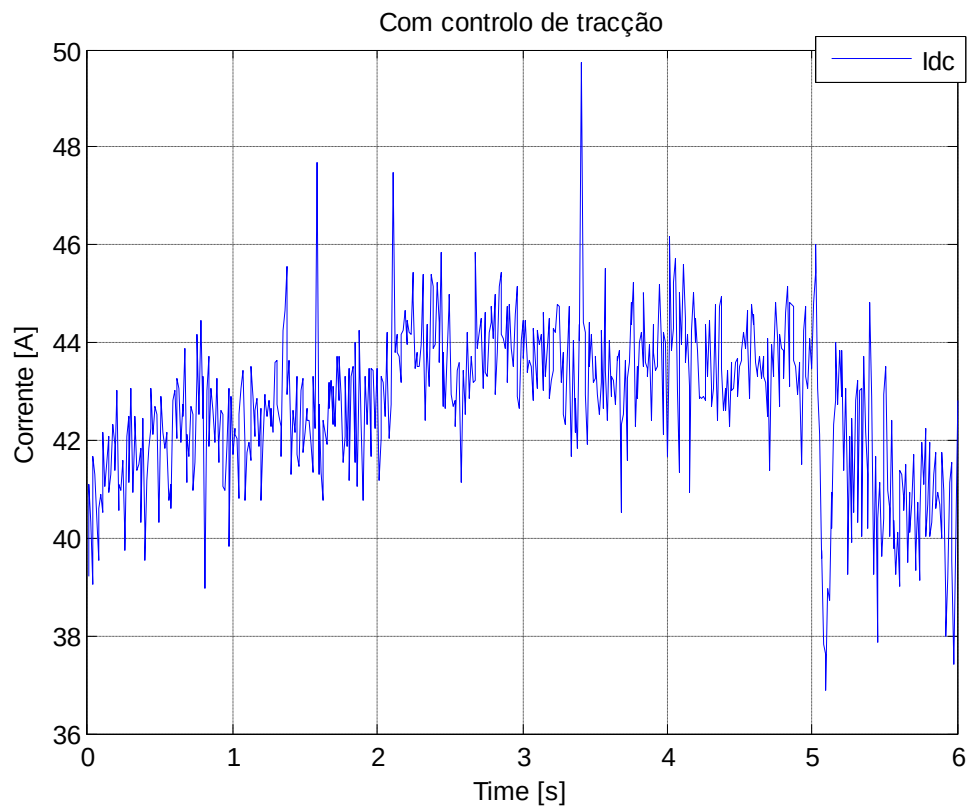
Figura 6.9 - Evolução do deslizamento sem controlo de tracção (teste na subida)

A roda do veículo dispara rapidamente porque o veículo passa por um piso com uma aderência menor, e entra na zona instável da característica $\mu-\lambda$ (diminuição do coeficiente de fricção com o deslizamento) do piso. Na Figura 6.9, é visível a entrada na zona instável, pode-se ver que a partir de determinado instante (aproximadamente 2.8s) o deslizamento aumenta rapidamente. Como é sabido, em termos de valores típicos o valor de deslizamento para o qual o coeficiente de fricção é máxima em diferentes pisos está aproximadamente entre os [0.1-0.2], e portanto pode-se ver que realmente é em torno desse valor que o veículo começa a perder a aderência aumentando rapidamente o deslizamento.

6.3.2 - Com controlo de tracção

Na Figura 6.10, está representada a evolução das velocidades mas com o controlador de tracção activo, no teste feito na subida, e pode-se observar a influência do controlo, que actua suprimindo o deslizamento (ver Figura 6.11), fazendo o sistema aderir ao piso. Uma observação que foi feita é que o controlador só começa a ter mais efeito a partir de determinado instante (aproximadamente 4.5s). Para tentar perceber este efeito, foi analisado o comportamento da corrente nas baterias, uma vez que esta está de certa forma relacionada com o binário que é aplicado ao motor.

Figura 6.10 - Evolução das velocidades com controlo de tracção (teste na subida, $K_{tcs}=0.5$)Figura 6.11 - Evolução do deslizamento com controlo de tracção (teste na subida, $K_{tcs}=0.5$)

Figura 6.12 - Compensação do sinal de comando (teste na subida, $K_{tcs}=0.5$)Figura 6.13 - Evolução da corrente das baterias, com controlo de tracção (teste na subida, $K_{tcs}=0.5$)

Na Figura 6.13 encontra-se a evolução da corrente, e analisando o comportamento da corrente em conjunto com o sinal de controlo ($V_{Throttle}^*$) que é enviado para a camada de controlo do motor, chega-se à conclusão que a compensação no sinal de comando dentro de determinados valores influencia muito pouco na variação de corrente, por exemplo no intervalo de tempo [3.5;5]s, na Figura 6.12, o sinal tende a diminuir enquanto que o sinal de corrente praticamente não sofre alteração, mantendo-se aproximadamente constante. No entanto só a partir dos 5s é que é notável um efeito brusco de corrente que resulta provavelmente como resposta a todo o decrescimento do sinal de controlo. Conclui-se que há no fundo uma relação de atraso relativamente elevado entre o sinal de controlo e a resposta em corrente. Ora esta situação dá origem a que o sinal de deslizamento não comece a ser suprimido mais cedo, sendo suprimido apenas no instante $t=5s$, por influência da variação da corrente.

Perante esta análise aumentou-se o ganho para $K_{tcs}=0.6$ para analisar o efeito do ganho no controlo. Com este ganho nitidamente o sistema ficou instável. Olhando para a Figura 6.14, pode-se ver a evolução da velocidade da roda com muita oscilação, não conseguindo fazer um controlo de tracção adequado. Isto acontece por causa do tal grande atraso que há na resposta do sinal de corrente na diminuição do sinal de controlo, e por causa da variação brusca e elevada que a corrente tem no final do tempo de atraso (ver Figura 6.16 no instante 4.7s), levando à oscilação do sistema.

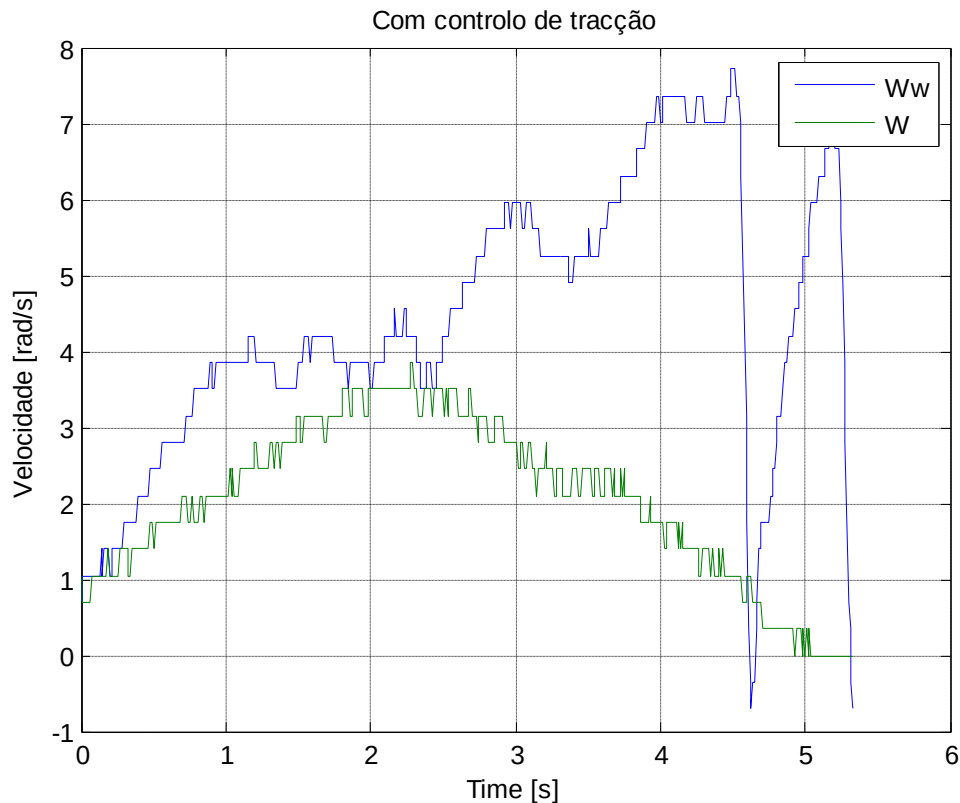


Figura 6.14 - Evolução das velocidades com controlo de tracção (teste na subida, $K_{tcs}=0.6$)

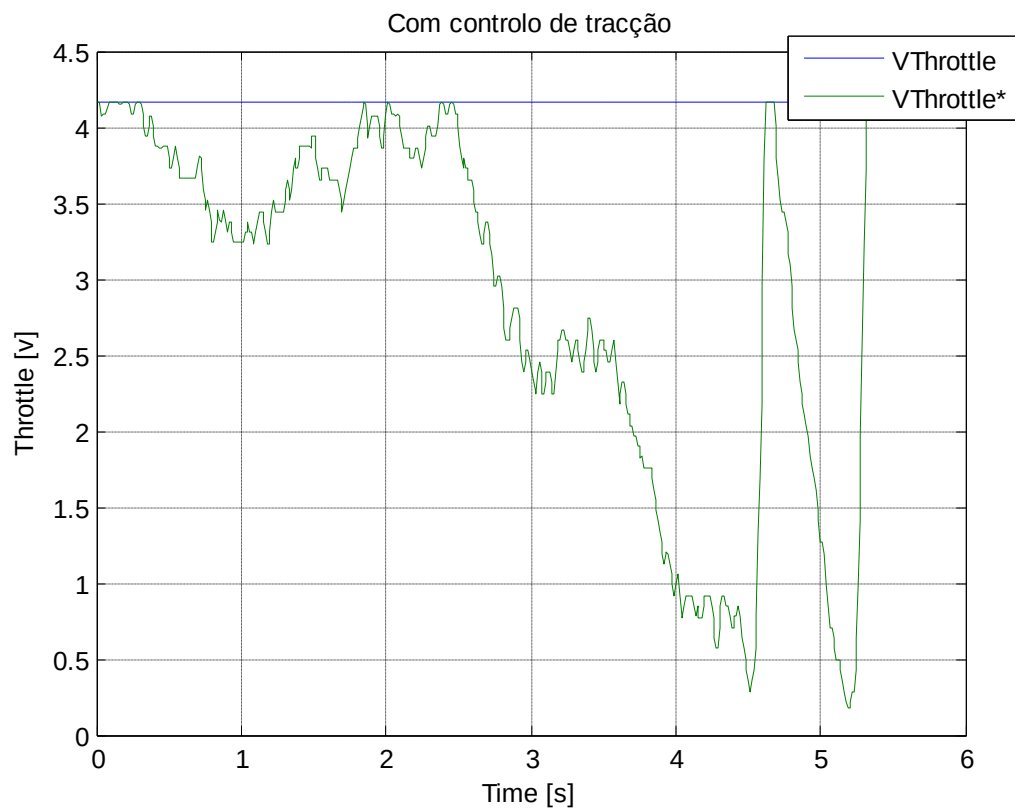


Figura 6.15 - Compensação do sinal de comando (teste na subida, $K_{tcs}=0.6$)

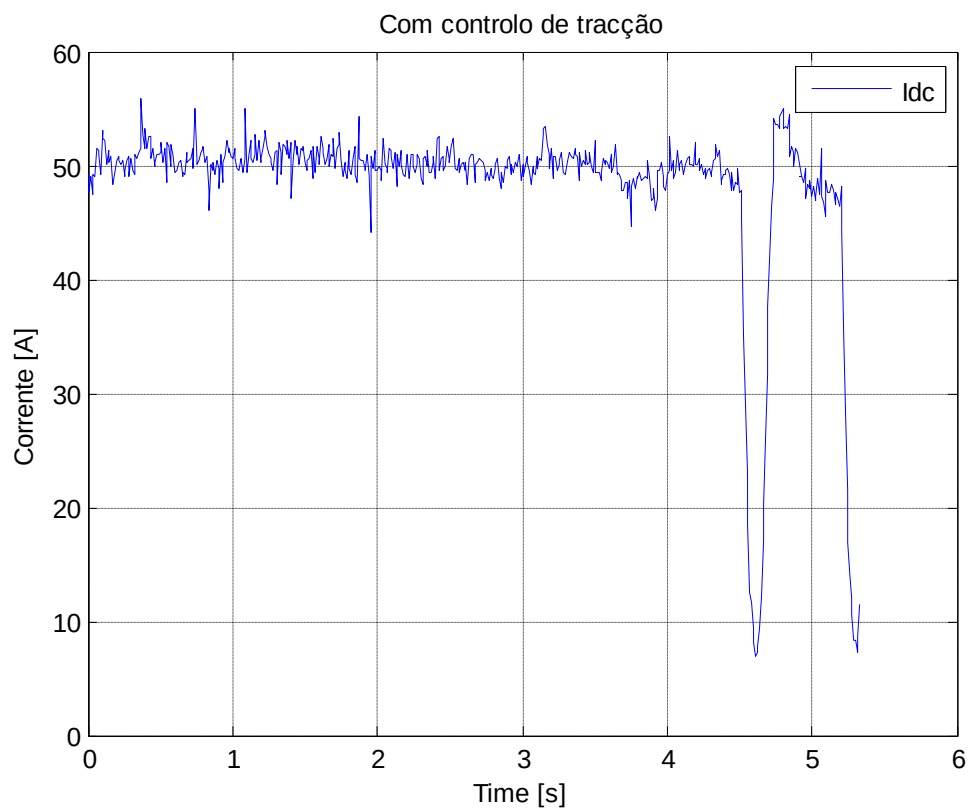


Figura 6.16 - Evolução da corrente das baterias, com controlo de tracção (teste na subida, $K_{tcs}=0.6$)

Com estas observações o controlo fica de certa forma limitado ao comportamento da corrente, pois enquanto o sinal de controlo tenta compensar o deslizamento diminuindo o seu valor, a corrente não responde de imediato mas quando varia actua bruscamente no sistema. Desta forma o ganho a que ficamos limitados no controlador é $K_{tcs}=0.5$, embora tenha um atraso na supressão do deslizamento, o sistema é capaz de aderir novamente ao piso sem oscilações.

6.4 - Conclusões

Neste capítulo foi feita uma apresentação do veículo eléctrico com as especificações do uCar. O objectivo foi mostrar toda a plataforma do uCar onde o sistema *DataLogger* e o controlador de tracção foram integrados.

Um aspecto importante nos veículos eléctricos é a sua autonomia e o desempenho, por isso, foi feito um teste experimental de estrada durante o qual foram armazenados vários sinais pelo sistema *DataLogger*, e que permitiram no final do ensaio fazer a análise respectiva. Com os dados foi feita uma estimativa da autonomia do veículo obtendo como resultado 36.5km, no entanto é importante dizer que esta variável depende fortemente do percurso.

Na parte final deste capítulo mostraram-se os resultados experimentais obtidos na fase de testes do controlador de tracção. Foram mostrados os resultados com e sem controlo de tracção, e foram comparados. Com controlo de tracção foram ainda comparados dois resultados com diferentes ganhos. Com todos estes resultados chegou-se à conclusão que o controlador funciona para um determinado ganho, mas com algumas limitações devido ao comportamento da corrente. Além disto concluiu-se que o efeito do comportamento da corrente com ganhos maiores prejudica o controlo de tracção levando o sistema a uma oscilação de velocidade drástica. Um aspecto importante que deve ser aqui referido é que o comportamento desta corrente depende fortemente da camada de controlo de baixo nível, o qual é completamente desconhecido (não facultado pelo fabricante), e a situação que foi simulada na secção 5.6, admitiu uma aproximação de 1ª ordem entre o sinal do acelerador e o binário motor, simulando o subsistema como se de um controlo em binário se tratasse na camada de baixo nível. Desta forma e como foi analisado na realidade o subsistema não tem esse comportamento. Embora o desempenho do controlador na realidade não seja tão bom como o da simulação, pelas restrições já descritas, o controlador funciona permitindo que o sistema adira novamente à estrada.

Capítulo 7

Conclusões e trabalho futuro

7.1 - Conclusões

Neste trabalho foi desenvolvido um sistema *DataLogger*, utilizando uma placa sbRIO-9631 com características de HW e SW interessantes. Foi integrado junto com a placa sbRIO um circuito de condicionamento de sinal que tinha como objectivo fazer a interface entre os sensores e a sbRIO, para condicionar o sinal. O sistema *DataLogger* foi dotado com funcionalidades que permitiram avaliar a autonomia e o desempenho do veículo eléctrico uCar, e obter os resultados de teste do controlador de tracção.

Neste trabalho houve algumas limitações, nomeadamente na determinação do modelo completo do veículo eléctrico uCar, por haver parâmetros que se desconheciam, e que não puderam ser determinados por falta de meios. No entanto ficaram registados possíveis métodos que podem ser utilizados para os determinar. Uma outra grande limitação foi o controlo da camada de baixo nível se desconhecer por completo, não sendo este um simples controlo em binário, o que dificultou um pouco o ajuste do controlador de tracção e o seu desempenho na prática. Relativamente aos testes práticos deste controlador foram reportadas alguns resultados, analisando e comparando situações com e sem o efeito do controlador. Além disso, foram justificados os resultados obtidos no que diz respeito ao comportamento do sistema com diferentes ganhos no controlador. Conclui-se então que o desempenho do controlador à medida que se aumenta o ganho, é afectado fortemente com o comportamento da corrente. Esta corrente apresenta um grande atraso a uma diminuição do sinal de controlo, e quando responde varia de forma brusca provocando instabilidade no controlo para ganhos elevados.

7.2 - Sugestões para trabalho futuro

Nos dias de hoje a tecnologia evolui a passos galopantes, são desenvolvidos frequentemente sistemas novos com tecnologia de ponta, ou até mesmo sistemas que já

existem mas que são melhorados e optimizados acrescentado algumas funcionalidades. Neste contexto, pretende-se deixar aqui algumas sugestões para trabalho futuro:

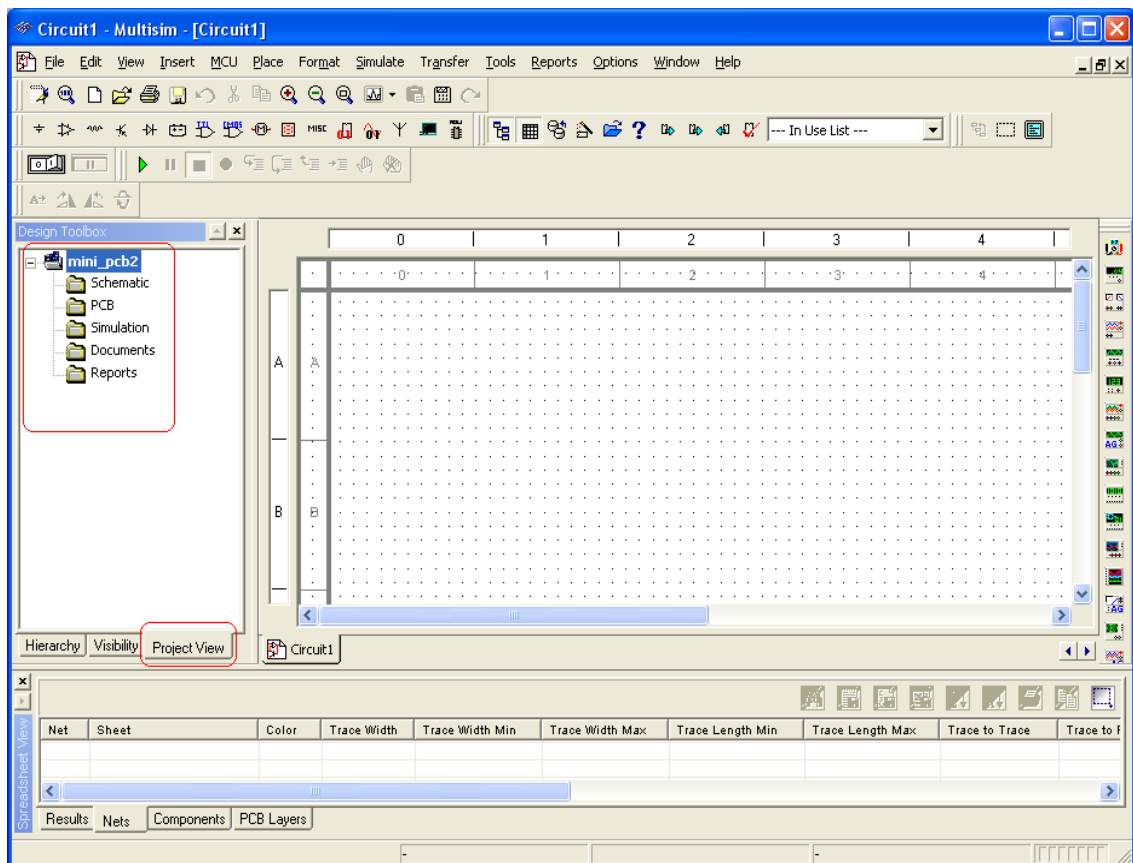
- Integrar um módulo GPS com interface RS232 no sistema *DataLogger*, com taxas de actualização de 5Hz, para armazenar os dados relativos à posição e à velocidade do veículo em cada instante;
- Integrar um módulo UMI com interface SPI, e acelerómetros com medida em três eixos;
- Integrar um cartão de memória *compact flash*;
- Integrar um display ligado ao sistema *DataLogger*, com uma interface para o condutor, como um computador de bordo
- Desenvolver um algoritmo de estimação da característica de fricção e com base nessa característica determinar o ganho ideal para o controlador, fazendo desta forma um controlo adaptativo de tracção às diferentes condições de estrada;
- Desenvolver um algoritmo para estimar o valor da velocidade do veículo, pois a informação da velocidade duma roda sem motor, deixa de ser válida em situações em que a roda bloqueia por travagem;
- Desenvolver um controlador de estabilidade numa camada superior ao de controlo de tracção;

Anexo A Circuito de condicionamento em PCB

Neste anexo pretende-se descrever de forma sucinta o processo da construção da PCB, que se inicia com o desenho do esquemático do circuito de condicionamento no *Multisim*, e posteriormente, é feita a transferência do circuito para o ambiente de desenvolvimento do *layout* da PCB, designado por *Ultiboard*

Multisim

O processo de concepção da PCB inicia-se com o desenho do esquemático, do circuito de condicionamento de sinal. Para isso, começa-se por criar um projecto novo, fazendo **File/New Project**, designado por exemplo *mini_pcb2*, obtendo-se o resultado da figura que se segue.

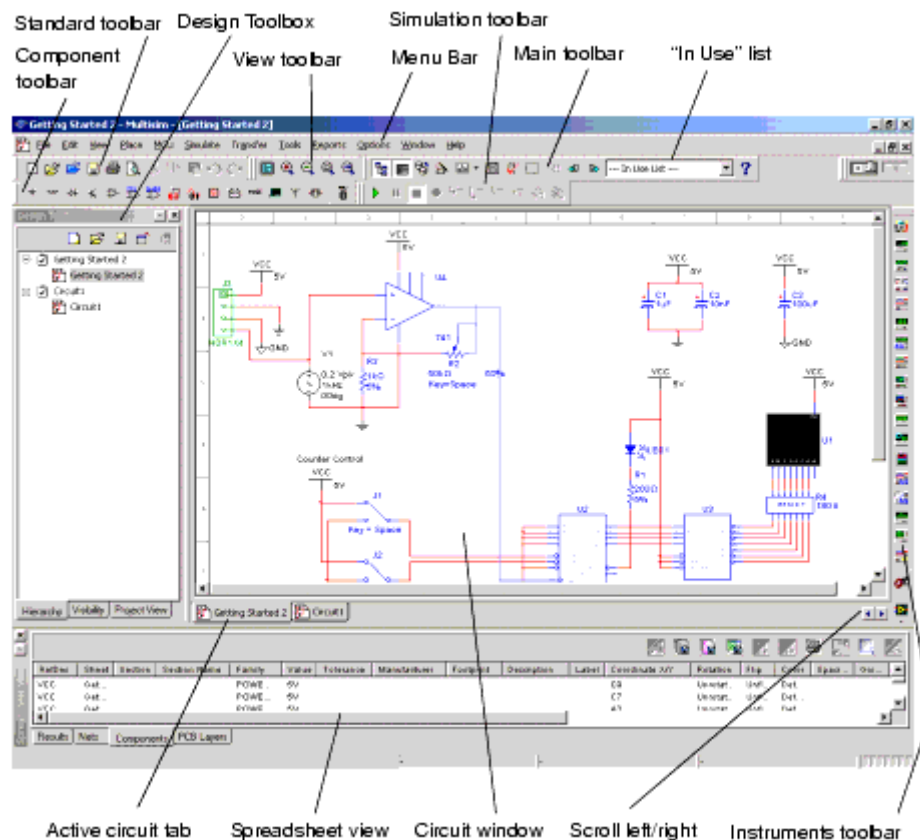


Na janela *Design Toolbox* em *Project View*, aparece o projecto criado e as pastas que contém. Estas pastas são criadas automaticamente com a seguinte finalidade:

- **Schematic folder:** para adicionar circuitos desenhados como *schematics* no *Multisim*. Pode haver múltiplos ficheiros de circuitos num só projecto, e um ficheiro de circuito pode fazer parte de mais do que um projecto;

- **PCB folder:** para os circuitos desenvolvidos no *Ultiboard*;
- **Documents folder:** para documentos do projecto que se deseja coleccionar. Por exemplo, descrição do projecto em *Word* , etc;
- **Reports folder:** para relatórios gerados pelo *Multisim*, por exemplo, *Netlist Report*, etc.
- Desta forma, obtém-se uma estrutura que permite organizar todos os ficheiros criados durante a fase de projecto.

O passo seguinte é desenhar o esquemático, mas para isso, primeiro é necessário conhecer as funcionalidades/comandos que o programa disponibiliza para o efeito. Por isso, é apresentado na figura que se segue, os elementos básicos da interface de utilizador do *Multisim*.

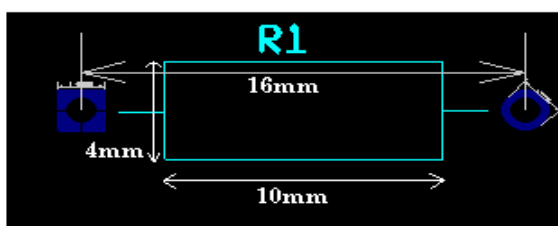


Esses elementos têm as seguintes funcionalidades:

- **Menus:** são onde se encontram comandos para todas as funções;
- **Standard toolbar:** contém botões para executar funções comuns, como guardar/abrir um ficheiro;
- **Simulation toolbar:** contém botões para iniciar, parar, e outras funções de simulação;

- **Instruments toolbar:** contém botões para cada instrumento, por exemplo um multímetro para utilizar numa simulação;
- **Component toolbar:** contém botões que permite seleccionar componentes da base de dados do *multisim* para coloca-lo no esquemático;
- **Circuit Window (ou workspace):** é onde se constrói o esquemático;
- **Design Toolbox:** permite navegar ao longo dos diferentes tipos de ficheiros no projecto (*schematics*, PCBs, *reports*), e ver a hierarquia do esquemático;
- **Spreadsheet View:** permite uma vista rápida dos parâmetros dos componentes, dos detalhes como *footprints*, *RefDes*, atributos e restrições de desenho. Além disso, os utilizadores podem mudar parâmetros dos componentes.

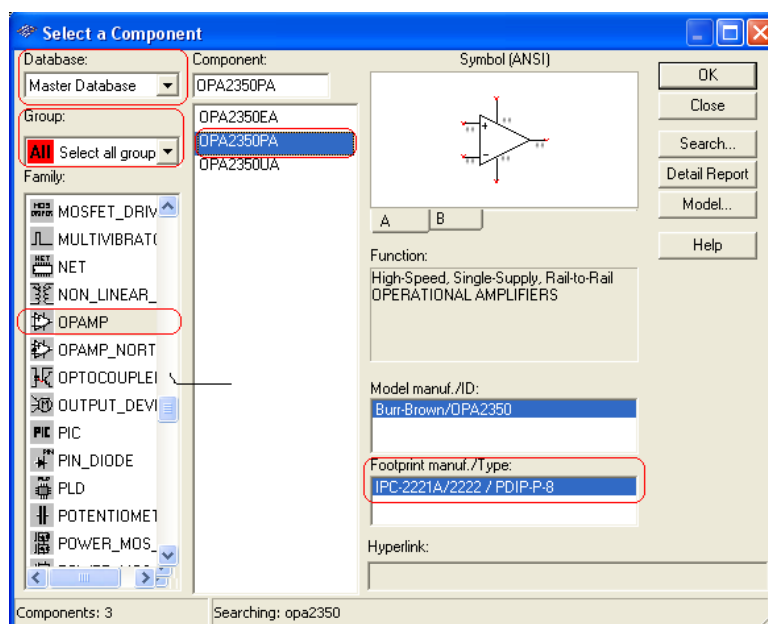
Depois da caracterização dos elementos, pode-se dar início à construção do esquemático. Para construir o circuito deve ser conhecida a referência dos componentes, e o *footprint* de cada componente a utilizar no circuito de condicionamento. O *footprint* é basicamente um atributo, que define o formato do componente a nível de dimensões e do tipo de pinos (ex: tipo DIP). Por exemplo as resistências que foram escolhidas para integrar na PCB têm *footprint* RES1600-1000X400 (ver tabela 7), onde 1600 significa que a distância entre pinos é de 16mm e 1000X400 é a dimensão da resistência, como se pode ver na figura que se segue.



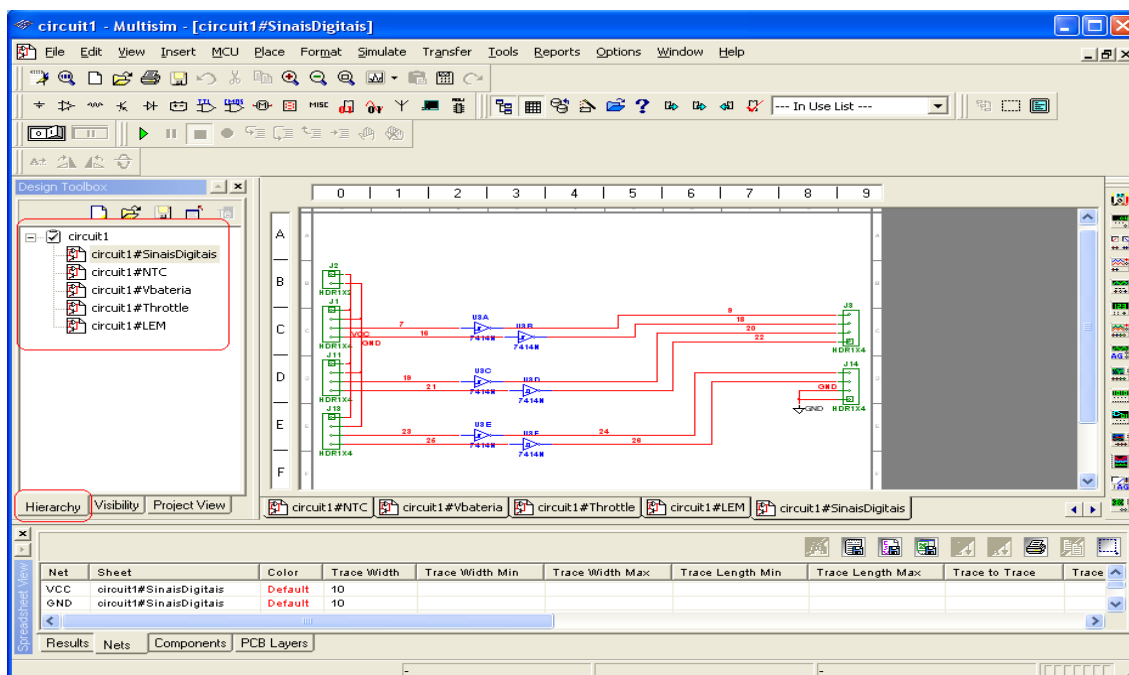
Os detalhes dos componentes escolhidos, para o circuito de condicionamento de sinal, encontram-se na tabela que se segue. Notar que, os detalhes do LEM e do NTC, sensores que pertencem ao circuito de condicionamento, não aparecem na tabela abaixo, porque estes não são integrados na PCB, mas sim no local onde vão fazer a medida das grandezas (ex. o NTC é colocado na carcaça do motor).

Componente	Referencia	Footprint Manuf. / Type
Hex inverting Schmitt trigger	74HC14N	IPC-2221A/2222 / NO14
Ampops	OPA2350PA	IPC-2221A/2222 / PDIP-P-8
Resistências	xk Ω	IPC-2221A/2222 / RES1600-1000X400
Condensadores	x μ F	IPC-2221A/2222 / CAPPR1000-2200X2500
Conectores	HDR1X2, HDR1X4	Generic/HDR1X2, Generic/HDR1X4

Agora, com os dados da anterior pode-se escolher os componentes respectivos na base de dados do *Multisim* a partir do **Component toolbar** (na figura que se segue). Por exemplo, para colocar Ampops OPA2350PA no esquemático, recorre-se à base de dados dos componentes, aparecendo a janela **Select a Component** como se pode ver na que se segue. De seguida, nessa janela selecciona-se **Database/Master Database**, **Group/Select all group** e em **Family** o **OPAMP**, depois em **Component** escolhe-se o OPA2350PA. Na parte inferior direita da janela, aparece o **Footprint manuf./Type**, onde se escolhe o footprint do componente de acordo com o especificado. Escolhido o componente, este deve ser colocado no **Circuit Window**, onde é desenvolvido o esquemático. A escolha dos restantes componentes segue a mesma lógica. Depois, basta fazer a ligação entre os componentes (criando as *Nets*), de forma a se obter o esquemático pretendido.



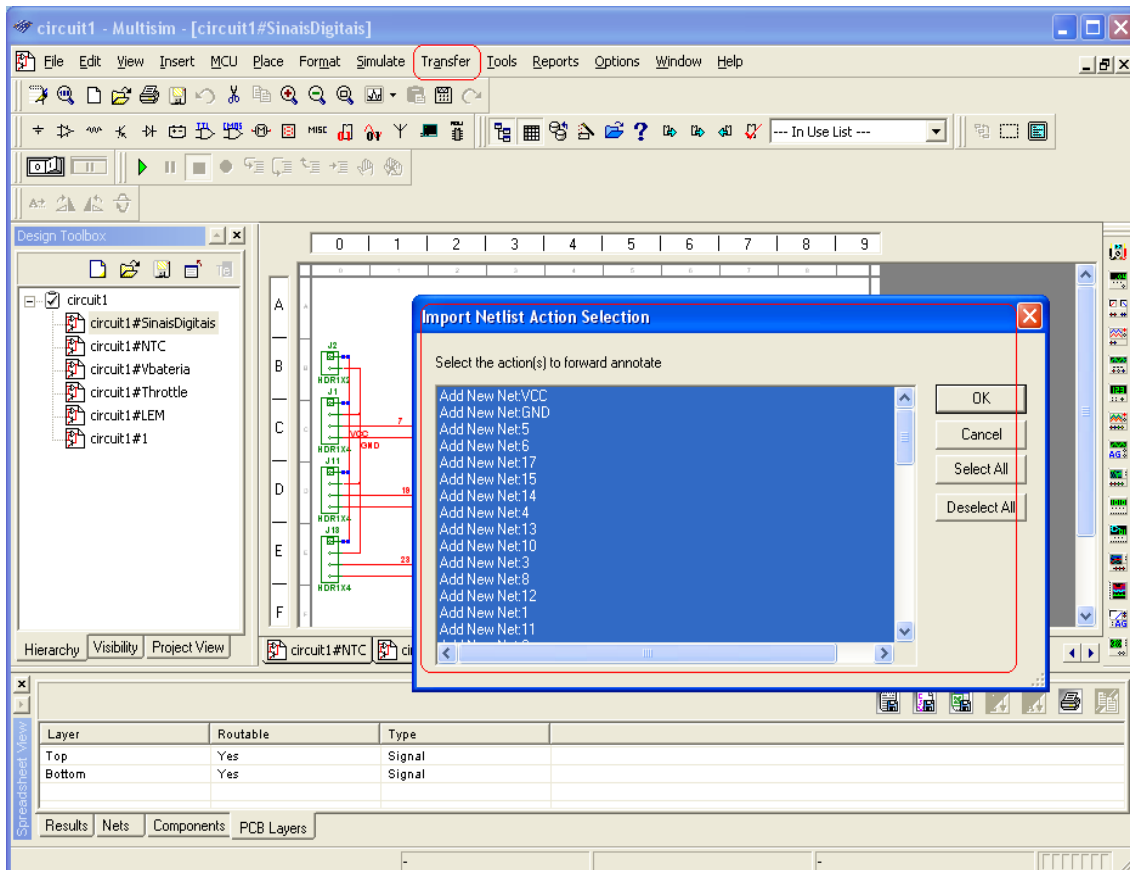
No caso do circuito de condicionamento de sinal que se pretende desenvolver, como este encontra-se dividido em pequenos circuitos, cada um deles com a finalidade de medir grandezas diferentes, achou-se conveniente, até por uma questão de organização, dividi-los em várias páginas (ver figura 4.21), mas pertencendo todas elas ao mesmo circuito. A forma como se criam essas páginas é a seguinte, faz-se **Place/Multi-Page** e é acrescentada automaticamente uma nova página ao circuito.



O resultado final da estruturação do esquemático apresenta-se de forma resumida na figura acima, onde se pode observar o nome que se atribuiu a cada uma das 5 páginas que pertencem ao **circuit1**. Em relação aos circuitos que foram construídos em cada uma das páginas, estes podem ser vistos no anexo A com pormenor.

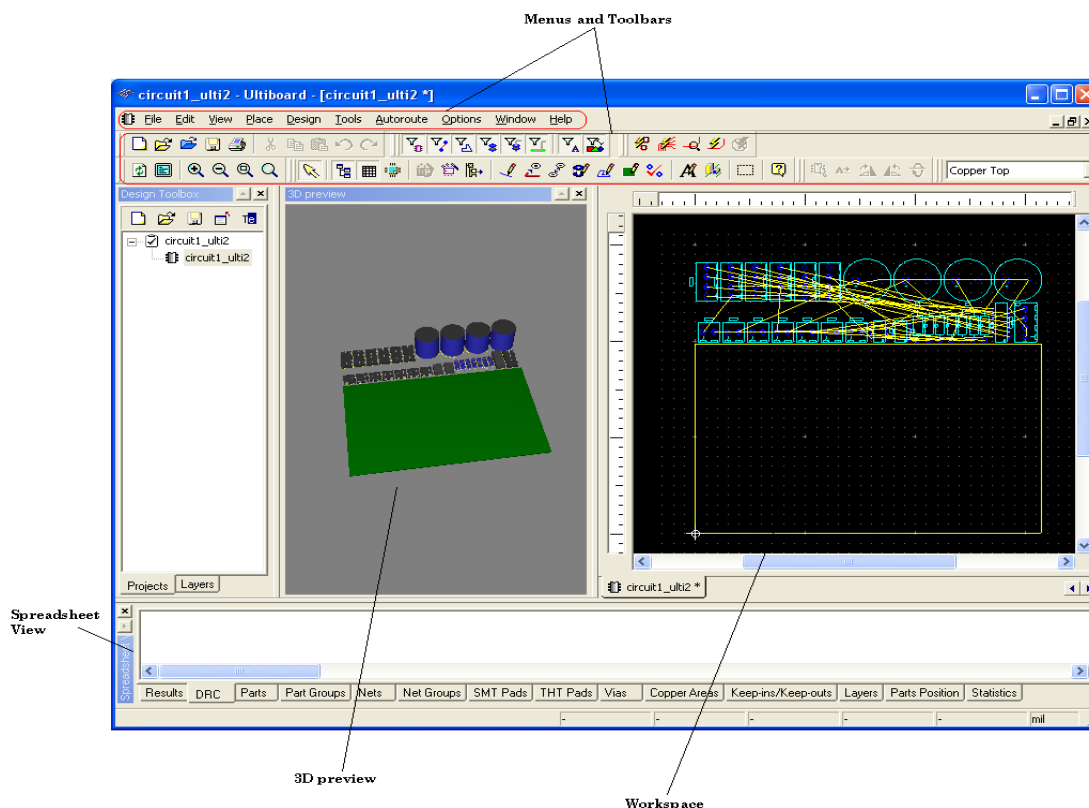
Ultiboard

Após a construção do esquemático no *Multisim*, deve ser feita a transferência da *NetList* para o *Ultiboard*, da seguinte forma, **Transfer/Transfer to Ultibord 10**, e aparece a janela com todas as Nets do esquemático a transferir, como se pode ver na figura que se segue.



Ao clicar em **OK** é então importada a *NetList* para o *Ultiboard*, e é criado automaticamente um projecto com o circuito transferido, obtendo-se o resultado da figura seguinte. Nessa figura está representada a interface do *Ultiboard*, com a indicação dos elementos básicos que é necessário conhecer, para iniciar o design da placa PCB. Os elementos mais importantes e as suas funcionalidades são as seguintes:

- **Menus and Toolbars:** dão acesso aos comandos do *Ultiboard*;
- **3D Preview:** fornece uma visão tridimensional da placa;
- **Spreadsheet View:** tal como no *Multisim*, permite uma vista rápida dos parâmetros dos componentes, dos detalhes como *footprints*, *RefDes*, atributos e restrições de desenho. Além disso, os utilizadores podem mudar parâmetros dos componentes;
- **Workspace:** é onde se constrói o design.

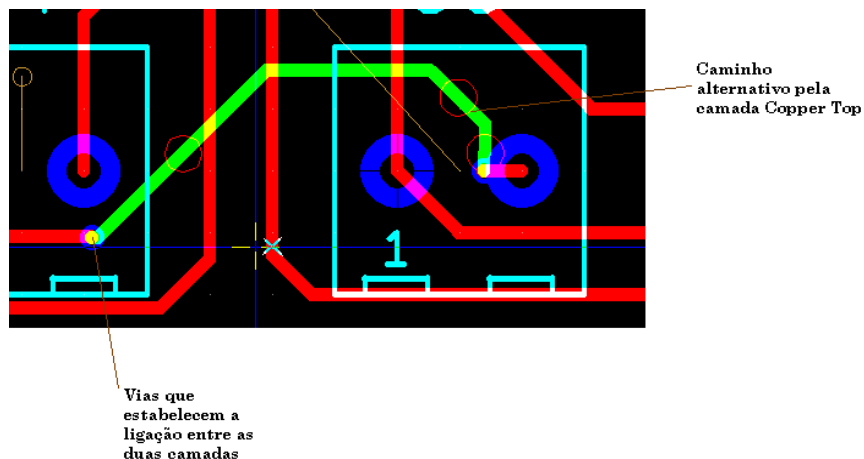


Depois de se conhecer os comandos mais importantes do *Ultiboard*, pode-se então, começar alocar os componentes do circuito sobre a placa no **Workspace**. À medida que isso vai sendo feito pode-se ter a percepção em 3D da placa com os componentes, no **3D Preview**.

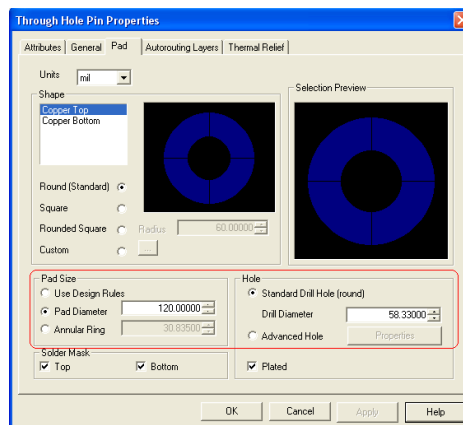
Após a alocação dos componentes, o passo seguinte é configurar as restrições das Nets, se estas ainda não tiverem sido feitas no *Multisim*, definindo a largura dos traços de cada Net no atributo **Trace Width**, o espaçamento mínimo permitido entre traços no **Trace Clearance**, e a camada (ex. *Copper Top* ou *Copper Bottom*) onde as Nets serão dispostas durante o “route”, em **Routing Layers**. Estes atributos encontram-se no **Spreadsheet View** do *Ultiboard* em **Nets** (ver figura que se segue). Na placa que foi desenhada, foram impostas as seguintes restrições: 20mil para o **Trace Width** uma vez que as linhas das Nets são de sinal, 10mil para **Trace Clearance**, e o código 01 para **Routing Layers** que define a camada Copper Bottom.

Net Na...	Locked	Trace Width	Max Width	Min Width	Trace L...	Max ...	Min...	Trace Clearance	Routing Layers
VCC	No	20.0000	393.7008	0.0394	20068.9...	N/A	N/A	10.0000	01
GND	No	20.0000	393.7008	0.0394	23403.9...	N/A	N/A	10.0000	01
26	No	20.0000	393.7008	0.0394	2190.56...	N/A	N/A	10.0000	01
25	No	20.0000	393.7008	0.0394	2595.43...	N/A	N/A	10.0000	01
24	No	20.0000	393.7008	0.0394	2388.88...	N/A	N/A	10.0000	01
23	No	20.0000	393.7008	0.0394	2122.20...	N/A	N/A	10.0000	01
22	No	20.0000	393.7008	0.0394	2122.44...	N/A	N/A	10.0000	01
21	No	20.0000	393.7008	0.0394	1259.80...	N/A	N/A	10.0000	01
20	No	20.0000	393.7008	0.0394	1851.49...	N/A	N/A	10.0000	01
19	No	20.0000	393.7008	0.0394	638.4282	N/A	N/A	10.0000	01

De seguida, inicia-se o processo de *Autorouting*, em *Autoroute/Start/Resume Autorouter*, para que os traços das *Nets* sejam desenhados na placa de forma automática cumprindo com as restrições impostas. Este processo automático nem sempre encontra uma solução completa, como foi o caso neste projecto, sendo necessário terminá-lo com um *route* manual. Desta forma, algumas ligações foram necessárias completar, e uma delas em particular, foi obrigada a ter um caminho alternativo pela camada *Copper Top* através de vias que faziam a ligação entre as duas camadas (ver a figura que se segue) visto não haver solução possível para que todas elas ficassem na camada *Copper Bottom* sem se cruzarem, como tinha sido imposto nas restrições das *Nets*.

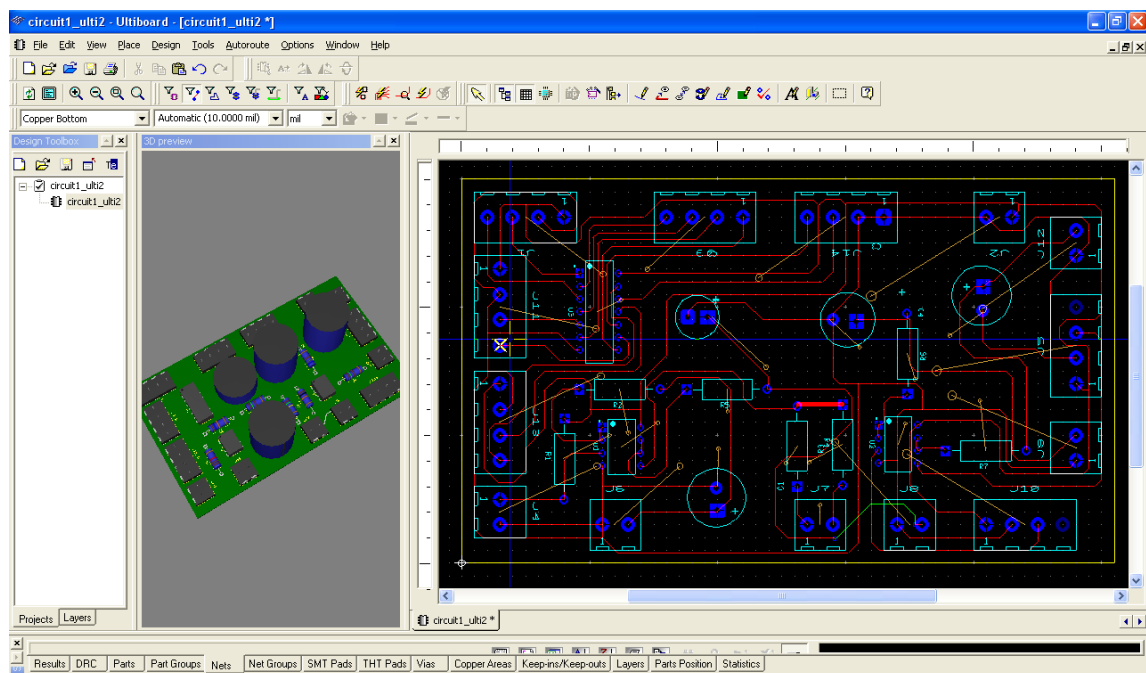


Depois de terminado o processo de *routing*, foram editados os parâmetros dos *Pads* dos componentes, nomeadamente o *Pad Diameter* e o *Drill Diameter*, para que o *Anular Ring* fosse suficiente para dar margem na solda dos pinos dos componentes (ver figura seguinte).



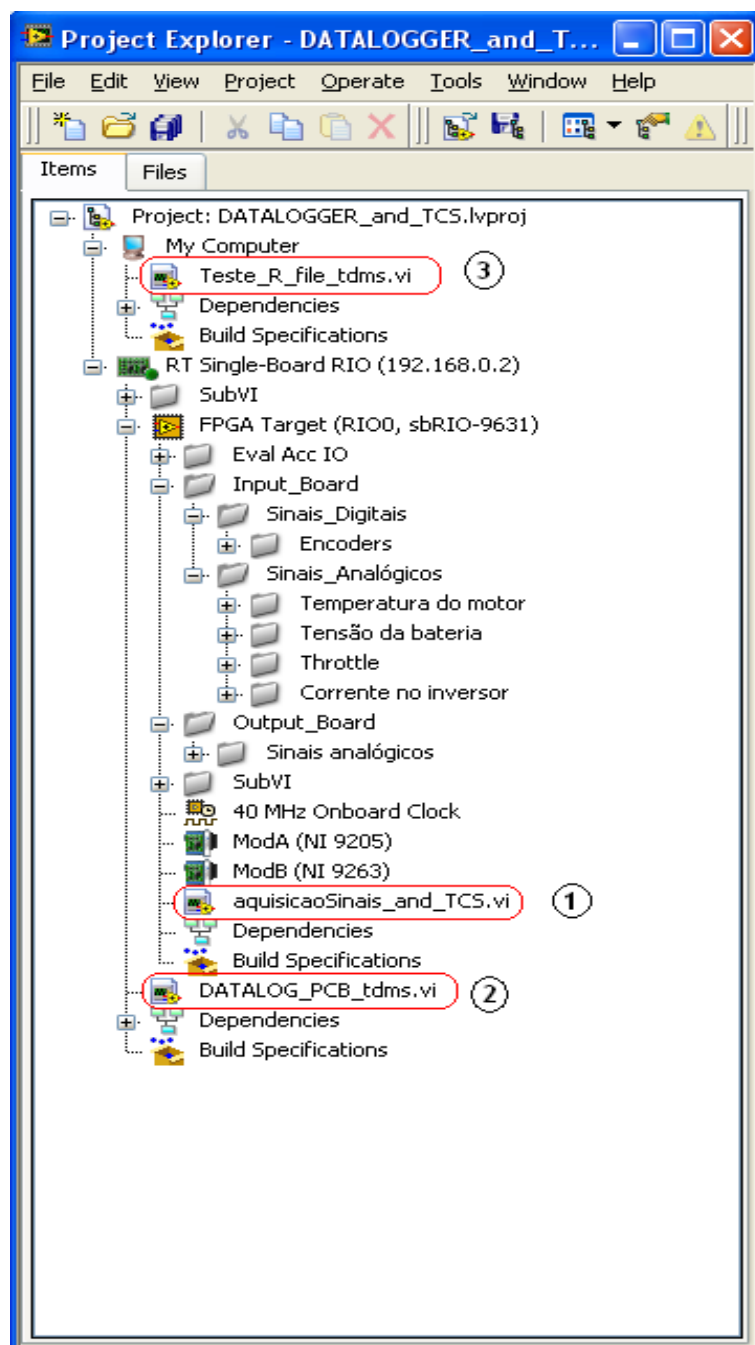
E assim termina o processo de design e concepção do circuito na placa PCB. Para analisar se existem erros de design, ou se por acaso foi violada alguma restrição imposta, pode-se executar o **DRC (Design Rule Error)** em **Design/Netlist & DRC Check**, e ver o resultado em **SpreadSheet View/DRC**.

O resultado final obtido no projecto do circuito de condicionamento encontra-se na figura que se segue. Nessa figura podem-se ver os conectores que se encontram alocados ao longo da periferia da PCB, para permitirem a interface de ligação com os sensores e a placa sbRIO, como foi inicialmente especificado na arquitectura do sistema.



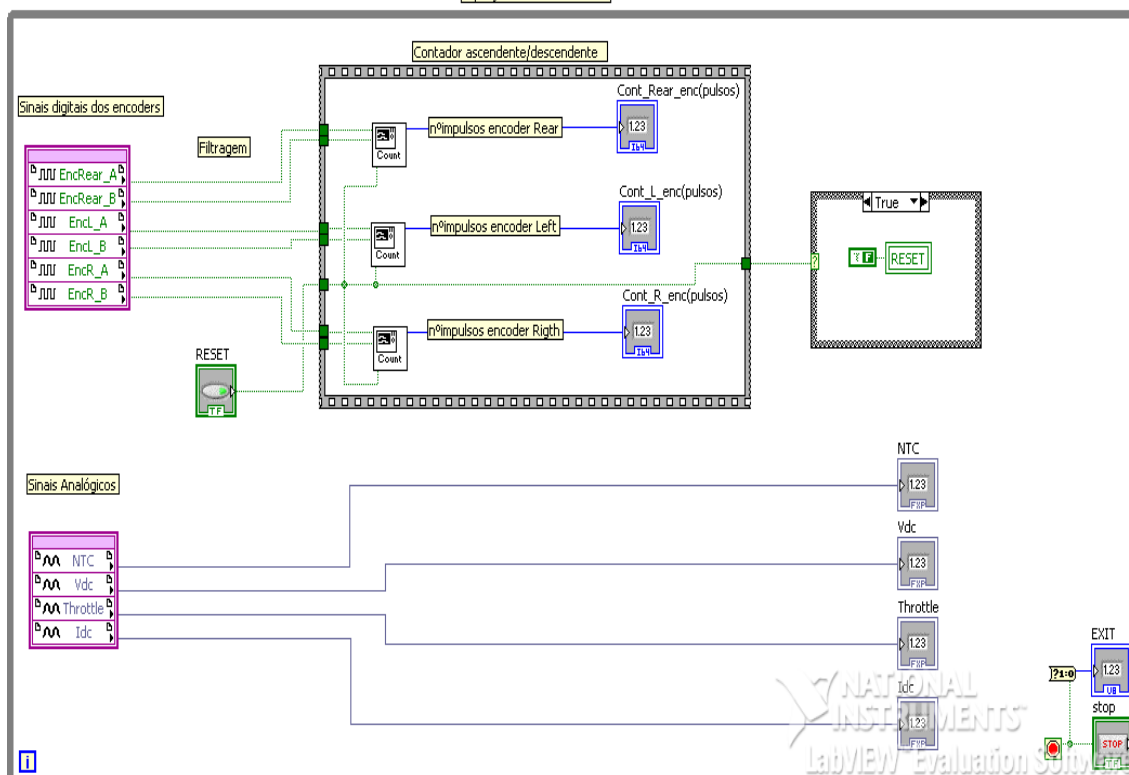
Anexo B Aplicações em LabView

Projecto



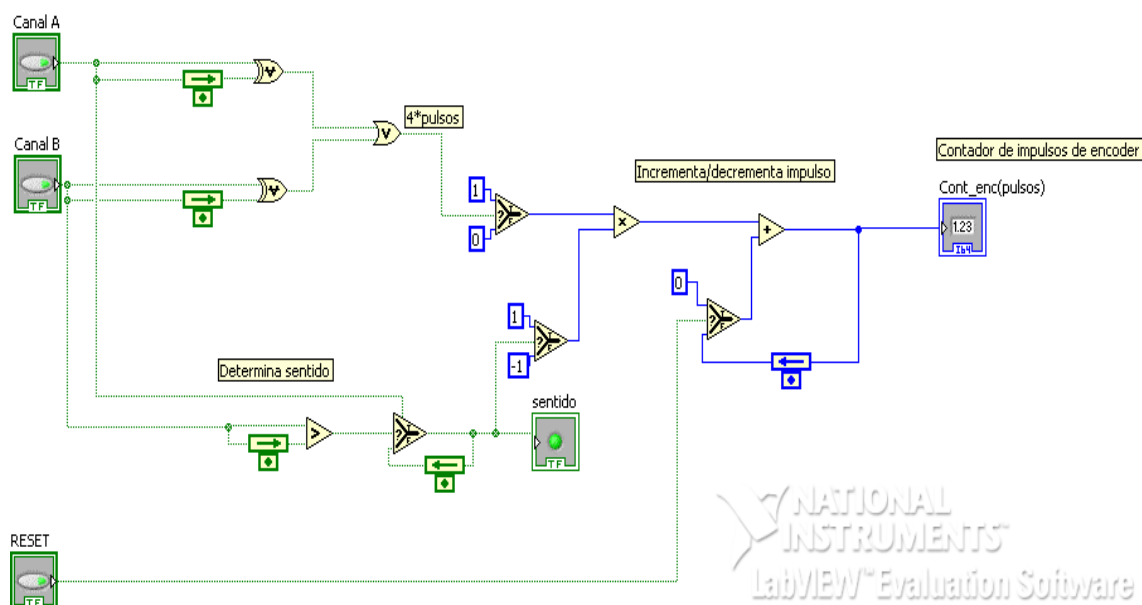
Aquisição dos sinais

Aquisição de Sinais na FPGA



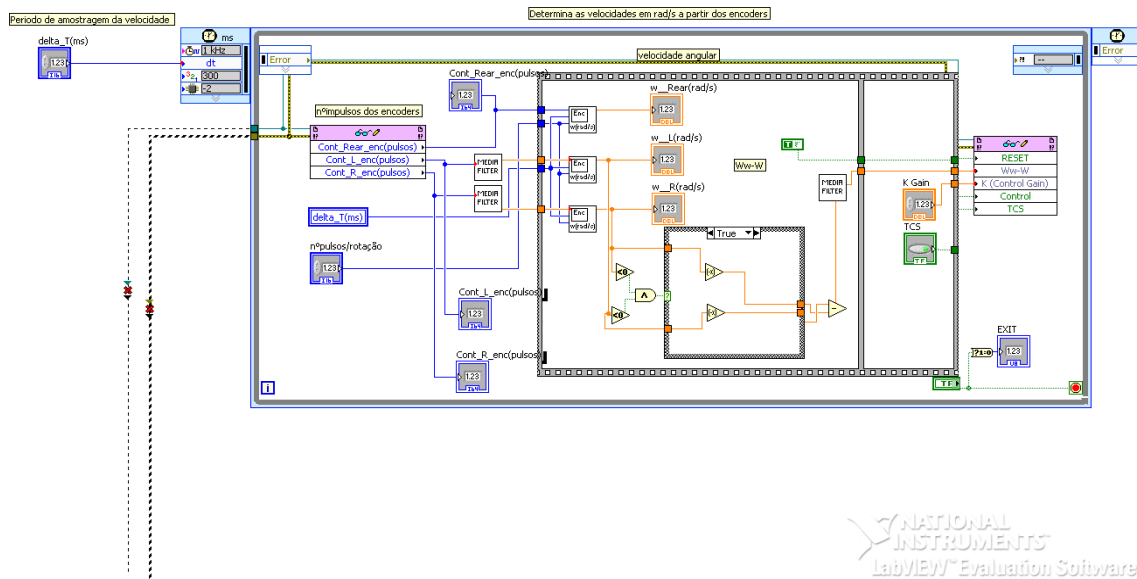
Subvi Conta impulsos com 4* maior resolução

Contador ascendente/descendente de impulsos do encoder com 4* maior resolução

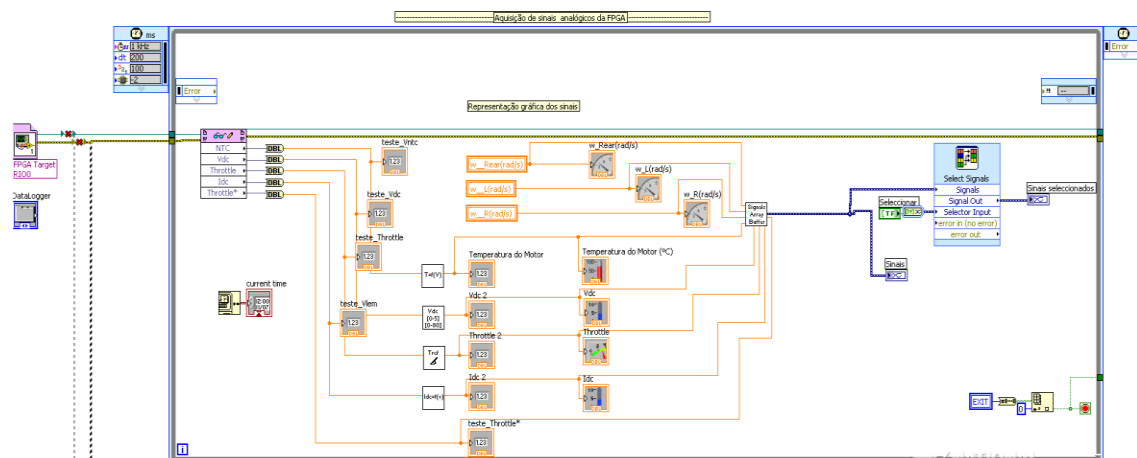


Processamento e armazenamento de dados

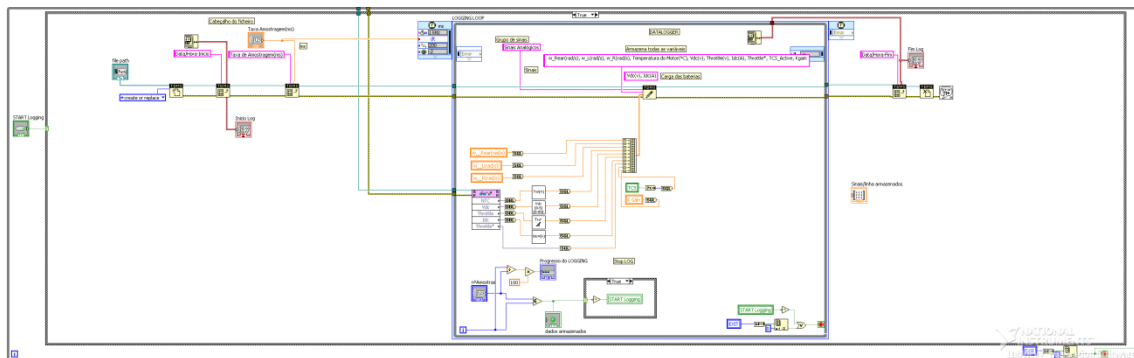
Tarefa de cálculo da velocidade



Tarefa de representação gráfica de dados

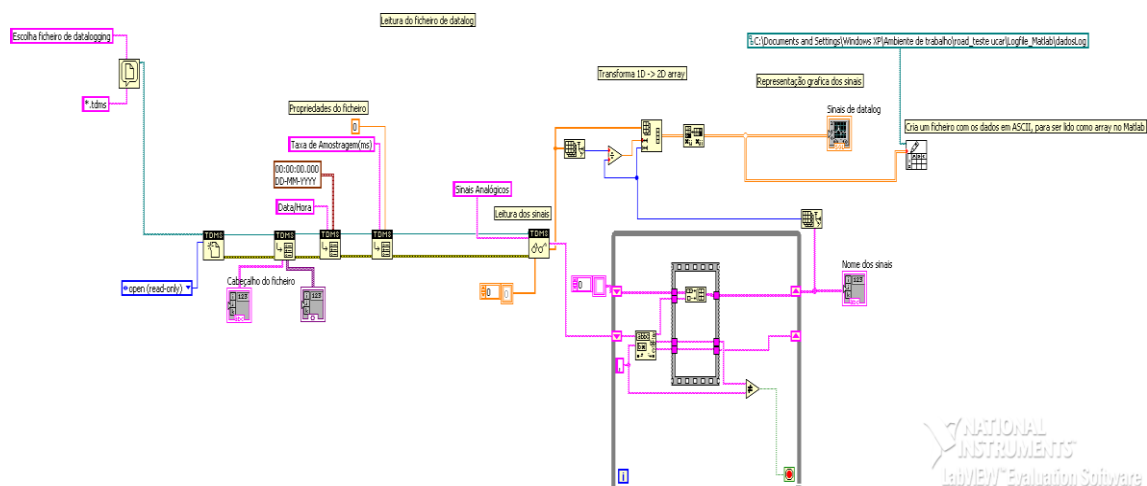


Tarefa de armazenamento de dados .tdms

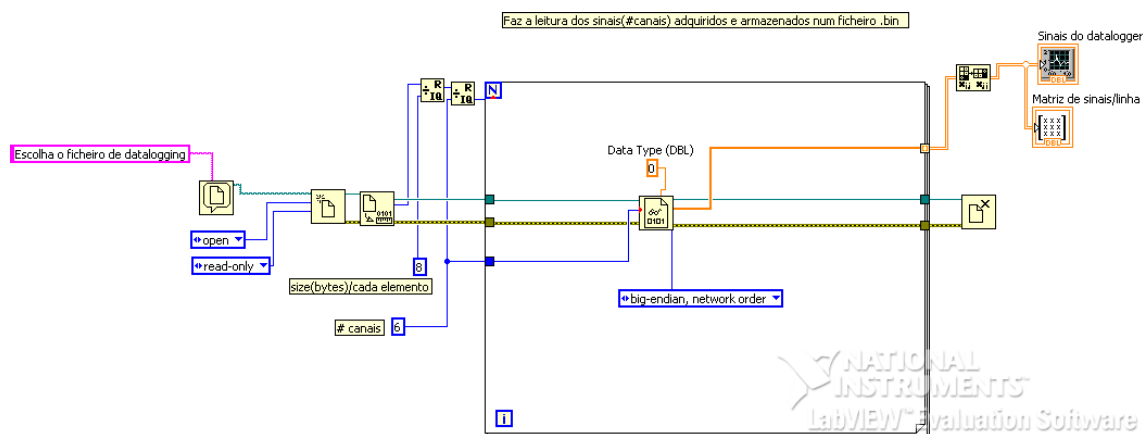


Leitura de ficheiros

Leitura de ficheiros .tdms



Leitura de ficheiros .bin



Referências

- [1] J.Y.Wong, *Theory of Ground Vehicles*. Ottawa: John Wiley & Sons, Inc, 2001, pp. 203-289.
- [2] F.Y.Wang, L.L., *Advanced Motion Control and Sensing for Intelligent Vehicles* New York: Springer Science, 2007, pp. 33-46
- [3] Chapter 4 Vehicle Dynamics. Disponível em http://scholar.lib.vt.edu/theses/available/etd5440202339731121/unrestricted/CHAP3_DOC.pdf. Acesso em 3/junho/2008.
- [4] Hori, Y., Y. Toyoda, and Y. Tsuruoka, *Traction control of electric vehicle: basic experimental results using the test EV "UOT electric march"*. Industry Applications, IEEE Transactions on, 1998. 34(5): p. 1131-1138.
- [5] Hori, Y., *Future vehicle driven by electricity and Control-research on four-wheel-motored "UOT electric march II"*. Industrial Electronics, IEEE Transactions on, 2004. 51(5): p. 954-962.
- [6] Hori, Y., Y. Toyoda, and Y. Tsuruoka. *Traction control of electric vehicle based on the estimation of road surface condition-basic experimental results using the test EV "UOT Electric March"*. in *Power Conversion Conference - Nagaoka 1997., Proceedings of the*. 1997.
- [7] Race Technology. Disponível em http://www.race-technology.com/description_8_956.html. Acesso em 17/Junho/2009.
- [8] Motorsports. Disponível em <http://www.magnetimarelli.com/motorsportsdigitekdatalog.html>. Acesso em 17/Junho/2009.
- [9] Magneti Marelli DataLogger RDL. Disponível em http://www.compsystems.com.au/index.php?option=com_content&task=category§ionid=9&id=33&Itemid=69. Acesso em 17/Junho/2009.
- [10] National Instruments sbRIO. Disponível em <http://sine.ni.com/nips/cds/view/p/lang/en/nid/205722>. Acesso em 17/Junho/2009.
- [11] Datasheet, "NI Single-Board RIO Embedded Control and Acquisition Devices", National Instruments, 2008.
- [12] National Instruments. Disponível em <http://sine.ni.com/nips/cds/view/p/lang/en/nid/204328>. Acesso em 20/Junho/2009.

- [13] Futurlec. Disponível em http://www.futurlec.com/CompBoard_Technical.shtml. Acesso em 20/Junho/2009.
- [14] R.Pallàs-Areny, *Sensors and signal conditioning*. New York: John Wiley, 2000.
- [15] W.Leonhard, *Control of Electrical Drives*. New York: Springer, 1997, pp.11-13.
- [16] S.K.Pillai, *A First Course on Electrical Drives*. New York: John Wiley & Sons, 1989, pp. 18-19.