

# Reconversão da Plataforma Robuter num AGV com Guiamento Visual

Relatório preliminar de dissertação de mestrado

Bruno Vieira

*Universidade de Aveiro*  
*Mestrado em Engenharia de Automação Industrial*

Fevereiro de 2017

**Resumo** - O projeto de reconversão da plataforma Robuter II está a ser realizado no âmbito da UC Projeto/Dissertação e desenvolvido fundamentalmente no Laboratório de Automação e Robótica (LAR).

O objetivo principal passa por fazer o *retrofitting* da plataforma, que se encontra obsoleta. Para isso pretende-se substituir toda a parte de controlo por uma solução mais atual que permita a integração de diferentes módulos com base numa arquitetura distribuída baseada em ROS.

A plataforma será dotada de uma câmara integrada num sistema de perceção para que este possa ter navegação autónoma orientada por uma linha.

No final, espera-se obter um veículo com versatilidade suficiente para se poder adicionar diferentes módulos e ser capaz de ter aplicações à escala industrial, similares a um AGV.

Neste documento é feita uma descrição geral do projeto onde são identificadas as várias etapas do mesmo e o trabalho desenvolvido até ao momento. Para além disso, é apresentada a forma como se irá proceder para a implementação das metas propostas.

## 1 Introdução

Neste documento é feita uma apresentação do trabalho a desenvolver no âmbito do Projeto/Dissertação. Para além disso, encontra-se uma lista de tarefas a realizar onde cada tarefa é explicada detalhadamente.

## 2 Trabalhos relacionados

Existem vários projetos relacionados com a plataforma Robuter, não tanto de reconversão da plataforma, mas sim de integração de elementos na mesma para validar conceitos ou testar hipóteses. Inclusivamente, no LAR já existiram alguns projetos realizados nesta plataforma.

Um dos projetos realizados foi "Autonomous Navigation and Multi-Sensorial Real-Time Localization for a Mobile Robot"[1]. Esta tese teve como objetivo a navegação de ambientes utilizando uma sequência de instruções condicionadas nas observações feitas pelo robô, ou seja, adquiridas pelos sensores. Apesar desta tese recair muito sobre o planeamento de trajetórias (que não é o foco do projeto a desenvolver) também foi necessária a implementação de câmaras e técnicas de processamento de imagem que permitiam dar ordens de navegação. Foram instaladas duas câmaras, para aquisição de imagem frontal e lateral e um laser 2D para complementar a informação obtida. Para a extração de características das imagens foram utilizado o algoritmo SIFT (Scale Invariant Feature Transforms). Este algoritmo permite detetar e corresponder características (através de uma base de dados e distância Euclidiana) associadas a uma imagem.

Outro tipo de abordagem, realizada em 2004 [2], consistiu em adaptar a plataforma para navegação exterior (calçada e mosaico). Neste projeto foi realizada uma intervenção mecânica na plataforma e desenvolvidos algoritmos de navegação, sustentados por dados provenientes de um laser 2D e um sistema de GPS (para determinar a localização global).

**Nota:** esta fase ainda se encontra em desenvolvimento e será um dos temas a debater na reunião.

## 3 Tarefa A

Seleção de baterias para a plataforma.

As baterias existentes na plataforma não se encontram operacionais. Por este motivo, deve ser feito um estudo que permita efetuar dois orçamentos: um passará por colocar uma solução idêntica à atual (à partida mais económica), o outro por utilização de outro tipo de tecnologia (por exemplo baterias de lítio).

## 4 Tarefa B

Limpeza e montagem da plataforma numa bancada de trabalho.

Montagem de um *setup* funcional que permita ter a plataforma fixa numa posição onde possa rodar livremente. O *setup* irá ser constituído por apoios colocados na mesa, em cima de um cartão, para que a plataforma fique na sua posição horizontal mas que as rodas fiquem sem tocar na mesa. Para além disso no local das baterias serão colocadas as fontes de alimentação para otimizar espaço na bancada. Após a mudança, todas as

ligações anteriormente feitas devem ser testadas. Para além disso, deve efetuar-se uma limpeza geral.

## 5 Tarefa C

Elaboração de um drive de potência para acionar os travões a partir do Arduino. Os travões eletromagnéticos acoplados a cada uma das rodas são atuados por um sinal comum de 48V. Para além disso, para a plataforma estar destravada, os travões apresentam um consumo de cerca de 24W, ou seja, perto de 0.5A. Devido a este facto terá de ser desenvolvido um circuito de potência que permita a atuação dos travões a partir de uma saída digital do Arduino. Propõe-se um circuito como da Figura 1



Figura 1: Drive de potência para atuar travões a partir do Arduino.

Apesar da proposta do circuito, encontrou-se um módulo de relés próprios para ser integrados no Arduino. Se esse módulo se mostrar suficiente, irá ser utilizado.

## 6 Tarefa D

Controlar o sentido de rotação dos motores.

Os motores são acionados através do de um drive de potência (Copley 306) com um sinal PWM. Inicialmente, verificou-se que para o motor rodar no sentido horário, teria de ser aplicado um sinal entre  $[0;5]V$ , e para rodar no sentido anti-horário entre  $[-5;0]V$ . Para isso seria necessária a elaboração de um circuito para inverter o sinal PWM uma vez que o no Arduino este sinal apenas varia entre  $[0;5]V$ .

Após analisar detalhadamente, verificou-se que as duas entradas PWM, REF+ e REF- não necessitam de estar referenciadas à massa, mas sim que exista um diferencial entre

elas.

Utilizando uma abordagem por *software*, são utilizadas duas saídas de PWM por cada motor onde uma delas têm o sinal PWM que corresponde à velocidade de rotação desejada e a outra um sinal de 0V (para inverter o sentido basta inverter as entradas). Para melhor compreensão, na Figura 2 este conceito é ilustrado.



Figura 2: Ligação dos sinais PWM provenientes do Arduino para o drive de potência

## 7 Tarefa E

Substituição e melhoramento da cablagem e conectores da plataforma.

A cablagem existente na plataforma é antiga e desajustada. Por este motivo espera-se intervir na cablagem e utilizar etiquetas para identificar cada um dos cabos (Tabela 1).

Para além disso espera-se identificar cada um dos IO's utilizados pelo Arduino Leonardo ETH. Parte dessa tabela já começou a ser elaborada, Tabela 2, mas só será finalizada quando as ligações forem terminadas.

**Nota:**Faltam os sinais dos *encoders* que não estão aqui representados porque este sinal irá ser tratado externamente.

## 8 Tarefa F

Implementação de um contador para contar pulsos provenientes do *encoder*.

- Os *encoders* existentes na plataforma geram cerca de 8000 pulsos por volta.
- Nas características técnicas da plataforma, a velocidade máxima é de 1 m/s.
- O diâmetro de uma roda é de 75cm.

Tendo em conta os pontos *a,b,c* tem de se projetar uma solução capaz de dar resposta a uma frequência de 16kHz por cada motor (expressão 1). Utiliza-se um valor de referência consideravelmente superior por segurança, uma vez que a plataforma vai perder peso e conseqüentemente poderá atingir velocidades superiores.

$$RPS_{max} = \frac{1}{0.75} = 1.25rps \rightarrow f_{min} > 10kHz. \quad (1)$$

Após testes realizados, verificou-se que o Arduino não tem capacidade para lidar, por interrupção, com a quantidade de pulsos gerados pelo *encoder* enquanto lida com a comunicação Ethernet. Após esta verificação procedeu-se a outro teste. Colocou-se o Arduino exclusivamente a contar interrupções e consegue-se efetuar uma leituras de sinais quadrados com uma frequência de 145 kHz para uma entrada (uma fonte de

Tabela 1: Cable identification.

Code	Description	Signal	Color
48V	Supply for motor drives, DC/DC converters	+48V	Orange
12V	Supply for Arduino and switch	+12V	Blue
5V	Encoders supply	+5V	Red
GND	Ground	0V	Black
M1-V+	V+ Motor 1 supply, signal from drive 1	0-48V	Brown
M1-V-	V- Motor 1 supply, signal from drive 1	0-48V	White
M2-V+	V+ Motor 2 supply, signal from drive 2	0-48V	Brown
M2-V-	V- Motor 2 supply, signal from drive 2	0-48V	White
PWM1+	PWM signal from the controller to the drive 1, REF+	0-5V	White
PWM1-	PWM signal from the controller to the drive 1, REF-	0-5V	White
PWM2+	PWM signal from the controller to the drive 2, REF+	0-5V	White
PWM2-	PWM signal from the controller to the drive 2, REF-	0-5V	White
BK-en	Signal to enable/disable brakes	5V	Non defined
BK+	V+ Common supply for eletromagnetic brakes	48V	Black
BK-	V- Common supply for eletromagnetic brakes	0V	Black
E1-chA	Encoder 1 signal, channel A	0-5V	Light Blue
E1-chB	Encoder 1 signal, channel B	0-5V	Light Blue
E1-GND	Encoder 1 signal, ground	0-5V	Light Blue
E2-chA	Encoder 2 signal, channel A	0-5V	Light Blue
E2-chB	Encoder 2 signal, channel B	0-5V	Light Blue
E2-GND	Encoder 2 signal, ground	0-5V	Light Blue

Tabela 2: Lista de IO's - Arduino Leonardo ETH.

Pin	Description
2	Digital input. Emergency state
4	Digital input. Bumper state
3	Analog output. Connects to REF+ from Drive 1
5	Analog output. Connects to REF- from Drive 1
6	Analog output. Connects to REF+ from Drive 2
9	Analog output. Connects to REF- from Drive 2
7	Analog output. Brake actuation
8	Digital output. Chip select to SPI communication
MISO	ICSP connector. SPI communication, pin SDO
SCK	ICSP connector. SPI communication, pin SCK
MOSI	ICSP connector. SPI communication, pin SDI

interrupção) e 72kHz para duas entradas (duas fontes de interrupção). Para ultrapassar este problema, são propostas duas abordagens:

- Contador externo com interface de comunicação.

HCTL-2022 - utiliza 10 IO's.

LS7366R - SPI, 32 bits, 1 *encoder* por chip.

- Utilização de Arduinos de gama inferior dedicados para as leituras.

Com a utilização de Arduinos, podem ler-se os sinais dos *encoders* por interrupção e comunicar, via série ou SPI para o controlador principal (Arduino ETH).

## 9 Tarefa G

Seleção e instalação de um CPU.

A seleção do computador será feita em conjunto com os colegas do LAR e com aconselhamento do professor Miguel Oliveira. Projeta-se que será um PC da Intel da gama NUC.

No PC irá ser instalada a distribuição Ubuntu 16.04. Depois deste passo, serão instalados os *softwares* e bibliotecas necessários para este projeto (já foi elaborado uma lista de comandos para instalação).

**Questão:** Terá de se projetar o inversor DC/AC nesta fase? Alimentação do PC provavelmente será de 230VAC. Como fazer?

## 10 Tarefa H

Instalação e integração de um *switch* de rede.

Configuração dos IP's para integrar os dispositivos Ethernet numa rede.

## 11 Tarefa I

Instalação e integração de uma câmara.

Nesta tarefa será necessário integrar a câmara Flea3 GigE, modelo GE-28S4C-C [4]. A alimentação da câmara é 12-24V.

Para além da sua integração na rede, terá de ser desenvolvido um processo responsável pela aquisição de imagem e publicação para o ROS.

## 12 Tarefa J

Reorganização e fixação dos elementos da plataforma.

Após ter todos os elementos disponíveis, a sua instalação será completada para que se possa fechar a plataforma.

## 13 Tarefa K

Teste do *software* desenvolvido até ao momento com tudo montado.

Garantir que o controlo dos motores e a leitura das variáveis necessárias (para integração no ROS) se encontra terminado para se passar ao desenvolvimento de algoritmos de navegação. Nesta fase terá de ser finalizado o controlo diferencial, ou seja, as instruções para o Arduino já deverão ser dadas em velocidade e ângulo.

## 14 Tarefa L

Desenvolvimento de um algoritmo para navegar com recurso a uma linha.

Processamento de imagem para identificação da linha e conversão desses dados em coordenadas (ou variações de direção em ângulos).

## 15 Tarefa M

Estudo de formas alternativas de navegação.

Avistam-se como possíveis: utilização de marcadores especiais ou visão *stereo*.

## 16 Conclusões

O estudo do problema e decomposição em subtarefas auxilia bastante na fase de abordagem ao problema. Ao descrever em detalhe cada uma das subtarefas a realizar será possível planificar trabalho. Para além disso, pode-se prever questões que seriam um impedimento ao desempenhar a tarefa.

Após realizar este documento, verificou-se que existem algumas pontas soltas no projeto que devem de ser finalizadas o mais rapidamente possível para que todo o trabalho desenvolvido até ao momento não necessite de intervenção futura.

### ROS

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

```
$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEED01FA116
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install ros-kinetic-desktop-full
```

```
$ sudo rosdep init
```

```
$ rosdep update
```

```
$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
```

```
$ sudo apt-get install python-rosinstall
```

## Referências

- [1] C. F. C. d. N. Ferreira, “Autonomous navigation and multi-sensorial real-time localization for a mobile robot”, tese de doutoramento, Universidade de Aveiro, 2008. endereço: <https://ria.ua.pt/handle/10773/2468> (acedido em 23/02/2017).
- [2] C. Cabral, *Navegação em ambientes exteriores através da adaptação de uma plataforma Robotica Móvel*, 2004. (acedido em 23/02/2017).
- [3] U. S. B. Controller, *ATmega16u4/ATmega32u4*. endereço: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.620&rep=rep1&type=pdf> (acedido em 22/02/2017).
- [4] *Flea3 2.8 MP Color GigE Vision (Sony ICX687)*. endereço: <https://www.ptgrey.com/flea3-28-mp-color-gige-vision-sony-icx687-camera> (acedido em 22/02/2017).