



universidade
de aveiro

Relatório Intermédio da Dissertação em
Mestrado em Engenharia de Automação Industrial

Reconversão da Plataforma Robuter num AGV com Guiamento Visual

Bruno Vieira N°82094

Orientador: Dr. Vitor Santos

Aveiro, Janeiro 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.

Palavras-chave: Robuter II, *retrofitting*, arquitetura modular, ROS, sistema de visão, AGV.

Abstract

The retrofitting of the Robuter II platform project is being carried out as a final Project/Dissertation and mainly developed in the Automation and Robotic Laboratory.

The main goal is centered in the platform retrofitting. For that, it is intended to substitute the old control part, for a modern solution that allows the integration of different modules based on a ROS distributed architecture.

The platform will have a camera to integrate a perception system so that it can navigate autonomously, guided by a line.

In the end, it is expected to get a vehicle with enough versatility so that different modules can be added and still be capable of having several uses on an industrial scale, such as an AGV.

In this document, a general description is made where all the different steps are identified, as well as all the work developed until the present moment. Furthermore, the guideline to follow in order to achieve the proposed goals, will be presented.

Keywords: Robuter II, retrofitting, modular architecture, perception system, ROS.

Índice

1. Introdução	1
1.1. Motivação e objetivos	2
1.2. Técnicas de navegação.....	3
1.3. Cronograma	4
2. Estudo da plataforma.....	7
2.1. Robuter II	7
2.2. Identificação de elementos	8
3. Descrição genérica do sistema.....	13
3.1. Tipos de arquitetura de controlo.....	13
3.2. Comunicação entre subsistemas.....	14
3.3. Unidades de processamento.....	15
4. Soluções propostas.....	17
4.1. Diagrama geral.....	17
4.2. <i>Hardware</i>	18
5. Conclusões e plano imediato	21
6. Referências Bibliográficas	23

Lista de Figuras

Figura 1.1 – Primeiro AGV, desenvolvido por Arthur Barret [1].....	2
Figura 2.1 – Robuter II original.	7
Figura 2.2 – Torre de controlo principal original.	8
Figura 2.3 – Setup atual da plataforma.	9
Figura 2.4 – Roda com elementos acoplados. 1 – Motor; 2 – Travão eletromagnético; 3 – Encoder.	10
Figura 2.5 – Conector circulador em perspetiva (esquerda) e vista frontal (direita).	10
Figura 3.1 – Esquema ilustrativo de sistemas centralizados [A] vs. distribuídos [B].	13
Figura 3.2 – Exemplo do funcionamento dos nodos em ROS.....	15
Figura 4.1 – Topologia de ligação do hardware para a integração dos elementos do AGV. 1 – Controlo e monitorização; 2 – Unidade central de processamento; 3 – Sinais do bumper detetor de colisões; 4 – Outros IO's; 5 – Câmara para aquisição de imagem; 6 – Switch de rede; 7 – Arduino; 8 – Encoder; 9 – Controladores DC; 10 – Dois motores DC 300W.	17
Figura 5.1 – Aplicação desenvolvida em GTK+ com programação em C para atuar e monitorizar os motores da plataforma Robuter.....	22

Lista de Tabelas

Tabela 1.1 – Cronograma detalhado do projeto com as principais tarefas.....	5
Tabela 2.1 – Especificações técnicas da plataforma Robuter II original.	9
Tabela 2.2 – Pinout da ficha circular proveniente dos motores DC.....	11

1. Introdução

É do conhecimento geral que as exigências do mercado estão a crescer de forma exponencial, quer a nível qualitativo quer quantitativo. A forte competição faz com que as empresas elevem constantemente os seus padrões de qualidade, reduzam os seus tempos de resposta e façam planeamento de tarefas. A automação industrial tem desempenhado um papel crucial para colmatar estas necessidades, sistematizando tarefas que podem ser deixadas para as máquinas. Uma dessas tarefas, comum a praticamente toda a indústria e bastante consumidora de tempo, é o transporte de material dentro das instalações fabris.

Na última década, os veículos guiados autonomamente (AGV) têm ganho expressão porque são capazes de dar resposta a esta problemática. É neste enquadramento que surge a importância deste tipo de veículos, capazes de desempenhar tarefas de forma autónoma, permitindo libertar operários de tarefas de baixo nível, maximizando a eficiência no meio industrial.

Atualmente a maioria dos AGV's utilizados à escala industrial são similares a empilhadores ou reboques, e têm a vantagem de poderem ser integrados nas próprias linhas de montagem, transportando matéria entre estações sem que seja necessária qualquer tipo de interação humana. Outra das suas principais características é a segurança. Para que possa ser utilizado como elemento colaborativo, obedece a uma série de normas que previnem os acidentes, reduzindo de forma significativa estas ocorrências no desempenho destas tarefas que, segundo a OSHA (*Occupational Safety and Health Administration*), superam as 90,000 por ano nos Estados Unidos da América.

O desenvolvimento desta área começou em 1954, quando o engenheiro Arthur "MAC" Barret criou o primeiro veículo conduzido autonomamente na sua empresa, Barret Electronics. Este veículo foi denominado de *Guide-O-Matic* (Figura 1.1) e foi desenvolvido para ser utilizado num armazém. Consistia num veículo elétrico, semelhante a um reboque, mas de dimensões inferiores que, para navegar, utilizava condutores no teto da instalação (que rapidamente passaram para pistas no chão da fábrica). Para se poderem deslocar autonomamente, na sua parte de baixo, continham sensores de campo magnético que procuravam o campo induzido pela corrente que atravessava os condutores das pistas. Para parar em cada estação, existiam códigos magnéticos que o veículo era capaz de processar [1] [2].



Figura 1.1 – Primeiro AGV, desenvolvido por Arthur Barret [1].

1.1. Motivação e objetivos

“Qualquer grande empresa no mundo que possamos pensar já utiliza AGV’s” (Scott Kwilinski, diretor de soluções de engenharia na Egemin Automation) e as previsões apontam para que cada vez mais este tipo de equipamentos seja integrado de forma rápida e versátil na indústria. As suas aplicações são vastas e permitem aos trabalhadores concentrarem-se nas tarefas de alto nível. O desenvolvimento de uma plataforma deste tipo vai de encontro à demanda atual e futura do mercado.

Existe ainda uma forte motivação interna, ou a nível académico mais concretamente. Com este projeto será possível converter por completo um equipamento obsoleto e inutilizado numa plataforma de aprendizagem moderna. Considera-se que será uma mais-valia para o LAR, e que de outra forma, a escola necessitaria de dispendir uma verba elevada para adquirir um equipamento com características similares.

Trata-se de uma plataforma muito robusta, com uma estrutura metálica maciça, preparada para suportar até 120 kg de *payload* mas que foi encontrada num estado de completa inutilização. Este facto deveu-se a algum do seu *hardware* não se encontrar operacional e do seu método de programação ser arcaico e pouco prático. Para além de que seria quase impossível a sua expansão devido a ser um solução fechada e proprietária.

O principal foco será então a reestruturação do sistema de controlo e aplicação de um sistema de perceção, nomeadamente uma câmara. A nível de *hardware*, implica reorganizar/refazer toda a instalação elétrica e melhorar o aspeto físico da plataforma. Ao nível de *software*, implementar uma arquitetura versátil e aberta, como o ROS, que permitirá a integração de diferentes unidades de controlo e aquisição de dados.

No final, espera-se dotar o laboratório de um equipamento modular e expansível ao ponto de permitir que este seja desenvolvido por futuros estudantes ou até mesmo integrado nas atividades letivas. Um caso concreto será a aplicação do braço robótico Fanuc LRMate 200iD no topo da plataforma. Esta última aplicação está a ser desenvolvida em paralelo por outro colega, no âmbito da sua dissertação, e espera-se então, no final, obter um robô móvel com um *gripper* associado.

1.2. Técnicas de navegação

Este sub-capítulo foi elaborado com base no livro “Navigation Mobile Robots Systems and Techniques” [3].

A navegação é uma área de estudo que procura encontrar soluções para três questões fundamentais: “Onde estou?”, “Para onde vou?”, “Como chego lá?”. (Leonard and Durrant-Whyte, 1991).

Relativamente à primeira questão, que pretende responder à posição do robô no seu meio, pode dizer-se que as soluções existentes têm dificuldades em dar uma resposta exclusiva e elegante. Isto é, existem muitas soluções parciais e/ou combinadas, que apesar de não poderem ser categorizadas são geralmente divididas em dois grupos: posição relativa e absoluta. Devido à falta de um método único, os AGV’s integram, por norma, tecnologias de ambos os grupos.

Atualmente, as técnicas utilizadas podem ir desde soluções que implicam a reestruturação do *layout* da fábrica/armazém até métodos não invasivos:

Localização absoluta

- Localizadores ativos – processamento da posição absoluta do robô através de três ou mais localizadores ativos para poder ser feita a triangulação. Estes localizadores podem utilizar tecnologia laser ou radio-frequência e devem ter a sua posição conhecida no meio.
- Referências passivas – estas referências podem ser naturais (inerentes ao meio de navegação) ou artificiais (colocadas propositadamente para permitir a sua fácil deteção). São geralmente detetados com técnicas de perceção e à semelhança dos localizadores ativos é necessário ter três identificadores no campo de visão para que se possa triangular a posição.
- Correspondência de modelo – comparação da informação adquirida pelos sensores com um modelo ou mapa do meio de navegação.

Como principais vantagens, pode dizer-se que este método é pouco invasivo e apresenta boas precisões no posicionamento quando corretamente referenciado. As principais desvantagens, nomeadamente nas técnicas que envolvem perceção, são a difícil programação e a robustez fortemente condicionada pelo meio de navegação.

Localização Relativa

- Odometria – utilização de *encoders* ou sensores inerciais como giroscópios e acelerómetros para determinar a direção e distância percorrida a partir de uma referência.

Uma das desvantagens é a acumulação de erro, uma vez que são tecnologias que utilizam um método integrativo (adicionam-se valores de direção e deslocamento a um dado ponto inicial). Como ponto positivo, é de destacar que estes sistemas funcionam independentemente das condições de visibilidade e podem operar em condições adversas, como por exemplo altitudes elevadas.

Guiamento

- Fio – ranhura no chão que contém um condutor detetado por um sensor na parte de baixo do veículo. O fio é posto ao longo do percurso.
- Fita magnética ou colorida – deteção da fita magnética através de sensores ou colorida (incluindo preto e branco) através de visão artificial ou sensores de cor. É colocada ao longo do percurso.

As principais vantagens são a robustez de navegação, fácil programação e boa precisão no posicionamento. Como desvantagens podem-se enumerar a necessidade de alteração do *layout* da fábrica, e no caso do fio a abertura de ranhuras no chão da mesma, o que a torna uma técnica pouco versátil.

1.3. Cronograma

Para permitir uma melhor organização, planeamento e delineamento de estratégias para a execução deste projeto, foi efetuado um cronograma, correspondente à Tabela 1.1, que contém as atividades a desempenhar e a sua duração ao longo do ano letivo.

Tabela 1.1 – Cronograma detalhado do projeto com as principais tarefas.

Tarefas	Out.	Nov.	Dez.	Jan.	Av Inter.	Fev.	Mar.	Abr.	Mai.	Jun.	Av Final
Pesquisa bibliográfica, levantamento de trabalhos relacionados e análise da plataforma Robuter II	X	X	X	X							
Desmontagem da plataforma atual e identificação dos diversos componentes e suas interligações		X	X								
Estudo concetual da solução a implementar para substituir os diversos componentes				X	X						
Seleção e instalação do <i>hardware</i> de controlo e unidades computacionais						X	X	X			
Programação de controladores e unidades de processamento						X	X	X			
Criação de <i>software</i> para as tarefas de navegação em ambiente ROS						X	X	X			
Validação experimental e demonstração das soluções desenvolvidas									X	X	
Escrita de documentação		X	X	X		X	X	X	X	X	

2. Estudo da plataforma

Neste capítulo é feita uma apresentação do Robuter II original e são expostos alguns passos intermédios do reconhecimento e identificação dos elementos plataforma.

2.1. Robuter II

O conceito de *Robuter* foi criado por uma empresa francesa, a Robosoft, e consiste numa plataforma robótica móvel (Figura 2.1), com um CPU *onboard*, dedicada a pesquisa e desenvolvimento. Na década de 80 este foi um conceito inovador que permitia aos seus utilizadores uma grande flexibilidade, uma vez que permitia não só a programação da plataforma (com recurso ao ALBATROS) como também a integração de *hardware* adicional [4].



Figura 2.1 – Robuter II original.

O ALBATROS é um sistema operativo de tempo real desenvolvido com o intuito de controlar, de forma eficiente, sistemas multi-eixo e multi-sensoriais. Tem uma organização multi-tarefa e pode facilmente integrar sistemas em rede.

Ao longo dos anos esta plataforma teve um papel importante na evolução da robótica móvel. Alguns dos temas seguidamente listados são exemplos disso:

- Estratégias de rastreamento de trajetórias em aplicações robóticas [5]
- Algoritmos de aprendizagem e reação e planeamento [6]

- Navegação robótica em espaços interiores com uma Câmara [7]
- Manipuladores robóticos autónomos controlados por M.A.S (Multi Agent System) [8]

2.2. Identificação de elementos

Para melhor compreensão desta plataforma foi efetuado um levantamento do seu equipamento e procedeu-se à respetiva desmontagem para perceber a localização e ligações físicas entre o *hardware*. De notar que toda esta fase do processo foi catalogada caso fosse necessário a sua reconstrução.

Em primeiro lugar foi retirado o módulo principal de controlo. Na Figura 2.2 observa-se que este possui uma montagem que permite a inserção de módulos com encaixe *Eurocard*. Foram identificados vários módulos: dois conversores DC/DC (para 5V e 12V); um módulo de segurança; dois controladores de motores DC e um CPU principal.

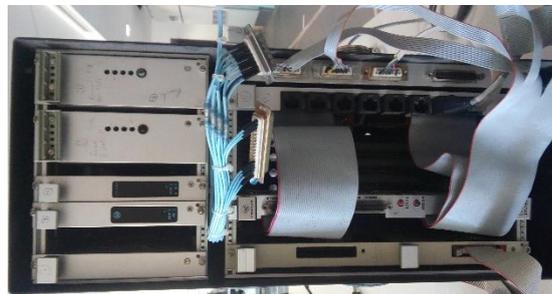


Figura 2.2 – Caixa de controlo principal original.

Após a análise da torre de controlo e o caminho de cabos, foram retirados todos os elementos que não serão utilizados na plataforma e refeitas as ligações dos restantes. Com esta ação, conseguiu-se obter o *setup* da Figura 2.3.

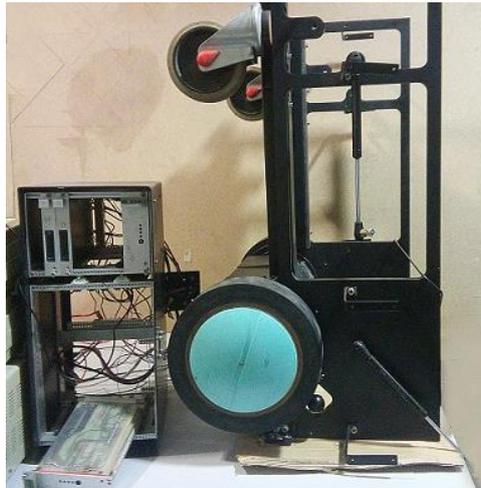


Figura 2.3 – Setup atual da plataforma.

No decorrer desta etapa, conseguiram-se apurar características da plataforma e ligações relevantes para os componentes a utilizar. Existiram várias versões, sendo que a utilizada neste projeto é o Robuter II (disponível no LAR).

Tabela 2.1 – Especificações técnicas da plataforma Robuter II original.

Comprimento	102.5 cm
Largura	68 cm
Altura	44 cm
Peso	150 kg
Carga/Payload	até 120 kg
Velocidade	até 1 m/s
4 Baterias de chumbo	12V 60Ah
2 Motores	DC ímanes permanentes 300W; 48V; 3000 rpm primário, 130 rpm saída
Controlo	Diferencial
Precisão no posicionamento	Encoders óticos < 1mm
Sistema Operativo	ALBATROS
Programação	Controlo interativo com mensagens ASCII via modem; Programas em linguagem C

Como é apresentado na Tabela 2.1, o controlo é diferencial (com duas rodas traseiras atuadas) e duas rodas dianteiras livres. Detetou-se ainda, ao desmontar as rodas, travões eletromagnéticos acoplados a cada um dos motores, que são desabilitados com um sinal de 48V, e *encoders* de 2 canais. Na Figura 2.4 podem ser observados estes elementos com mais detalhe.

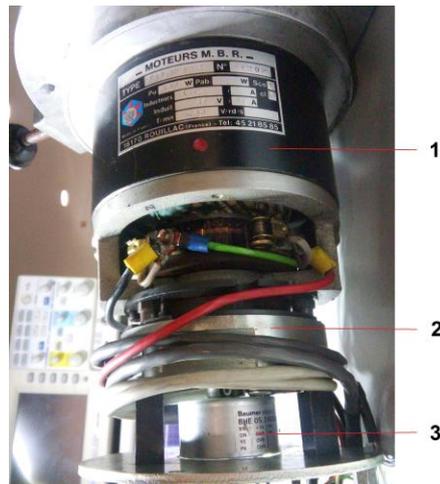


Figura 2.4 – Roda com elementos acoplados. 1 – Motor; 2 – Travão eletromagnético; 3 – *Encoder*.

Foi ainda identificado um conector circular principal (Figura 2.5) que permite interligar a parte de controlo aos motores.



Figura 2.5 – Conector circular em perspectiva (esquerda) e vista frontal (direita).

O conector encontra-se numerado com pinos de 1 a 16 e a identificação dos motores esquerdo (ME) e direito (MD) é feita na perspectiva de um condutor, ou seja, colocando-se de frente para a traseira do veículo. A associação dos sinais a cada um dos pinos é feita na Tabela 2.2.

Tabela 2.2 – *Pinout* da ficha circular proveniente dos motores DC.

Sinal	ME(1)	MD(2)
V- Motor	15	14
V+ Motor	16	13
V+ Encoder	2 (comum)	
V- Encoder	1 (comum)	
Channel A	7	3
Channel B	8	4
Travão	11 e 12 (sem polaridade)	

3. Descrição genérica do sistema

Neste capítulo é feita a distinção entre sistemas centralizados e modulares e apresentada a forma como os subsistemas vão comunicar entre si.

3.1. Tipos de arquitetura de controlo

Os sistemas centralizados utilizam, de forma geral, um CPU que é responsável pelo processamento de todas as tarefas. Ou seja, todos os sensores/atuadores comunicam diretamente com o CPU, através dos seus periféricos I/O, o que exige que este tenha uma elevada capacidade de processamento de forma a ser capaz de dar respostas atempadas (tempo real). Como está tudo centralizado, por norma, estas são soluções mais compactas e leves do que as modulares.

Por outro lado, numa arquitetura distribuída, os sistemas são divididos em outros subsistemas, ou módulos. Cada um desses módulos é responsável pelo controlo de uma determinada tarefa e tem, por norma, o seu próprio CPU associado. Em conjunto, todos os módulos atuam individualmente para o funcionamento do sistema. Nesta topologia podem estar todos interligados entre si e funcionar de forma independente ou todos ligados a um CPU central (*Master*).

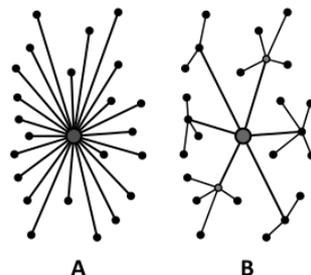


Figura 3.1 – Esquema ilustrativo de sistemas centralizados [A] vs. distribuídos [B].

A Figura 3.1 representa esquematicamente a diferença entre os dois tipos de sistemas.

É assim fácil de perceber que um sistema modular é mais robusto, facilita o desenvolvimento (nomeadamente na fase de testes) e é bastante mais versátil para se efetuarem melhorias, do que um sistema centralizado. Por estas razões e por se tratar de um projeto com âmbito académico, com aplicação direta no LAR e que está destinado a ser continuamente melhorado, a opção por um sistema modular torna-se evidente.

3.2. Comunicação entre subsistemas

De forma a dar continuidade à ideia de modularidade e facilidade de integração de subsistemas, irá utilizar-se o protocolo de comunicação TCP/IP através do meio *Ethernet*. Outro fator decisivo é o facto de no mercado existirem várias unidades de microcontroladores e sensores capazes de comunicar através deste protocolo. Este tipo de comunicação é amplamente usado nas redes industriais e podem ser implementadas configurações onde todos os equipamentos comunicam entre si.

A arquitetura de suporte e interface entre os diversos módulos será o ROS (*Robot Operating System*). O ROS é uma *framework* de *software* que possui um vasto leque de ferramentas e bibliotecas direcionadas para a robótica. É *open-source* e promove a reutilização de código. Esta última característica é potenciada pela capacidade de executar códigos desenvolvidos de forma independente (mesmo em linguagens diferentes) e integrá-los no funcionamento geral do sistema, ou seja, existem vários processos a decorrer em separado mas que comunicam entre eles. [9]

Cada um destes programas ou processos é chamada de nodo. Os nodos comunicam entre eles através de publicações (*publish*) e subscrições (*subscribe*) e utilizam tópicos para o efeito (canais atribuídos para comunicação). Na Figura 3.2 encontra-se um exemplo desta topologia.

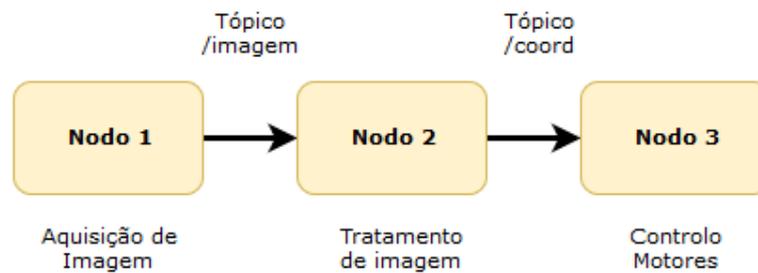


Figura 3.2 – Exemplo do funcionamento dos nodos em ROS.

Da Figura 3.2, o nodo 1, que é responsável pela aquisição de imagem, publica os dados (neste caso a imagem adquirida) no tópico **/imagem**. O nodo 2 vai ao mesmo tópico recolher essa informação e utilizá-la para a poder processar. Este nodo vai ainda transmitir para o tópico **/coord** a informação por ele processada para que possa ser recolhida por outro nodo (por exemplo as coordenadas de um determinado objeto para que o sistema de controlo de motores transmita a informação que é para se deslocar para essa posição). Apesar de não estar esquematicamente representado na Figura 3.2, é de notar que cada tópico pode ser acedido por diferentes nodos (tanto para *publish* como para *subscribe*).

3.3. Unidades de processamento

Seguindo a linha de raciocínio apresentada na secção anterior, irão existir unidades de processamento distintas. Assim, tarefas individuais como o controlo dos motores DC são feitas exclusivamente por um controlador, que irá transferir os dados e ser atuado pela unidade principal. Neste momento só se projeta ter um controlador, apesar de poderem ter de ser adicionados módulos extra (dependente do número e complexidade de tarefas a desempenhar).

Como a parte de processamento de imagem é muito pesada computacionalmente, irá optar-se pela escolha de um PC de dimensões reduzidas mas com uma capacidade de processamento muito superior à de um *chip* comum. Com a integração de um PC consegue-se ter um sistema operativo associado e consequentemente os serviços e aplicações nele incluídos.

4. Soluções propostas

Com a familiarização da plataforma e com os objetivos finais do projeto em vista, começou-se a idealizar a arquitetura de controlo do robô.

4.1. Diagrama geral

A comunicação entre os componentes da plataforma será feita com recurso ao protocolo TCP/IP. Irá ser estabelecida através de um meio físico (*Ethernet*) em vez de sem fios porque apenas é necessária uma solução local. Assim, um *switch* é responsável pela gestão das comunicações. Na Figura 4.1 é possível observar um exemplo da topologia de conexão proposta.

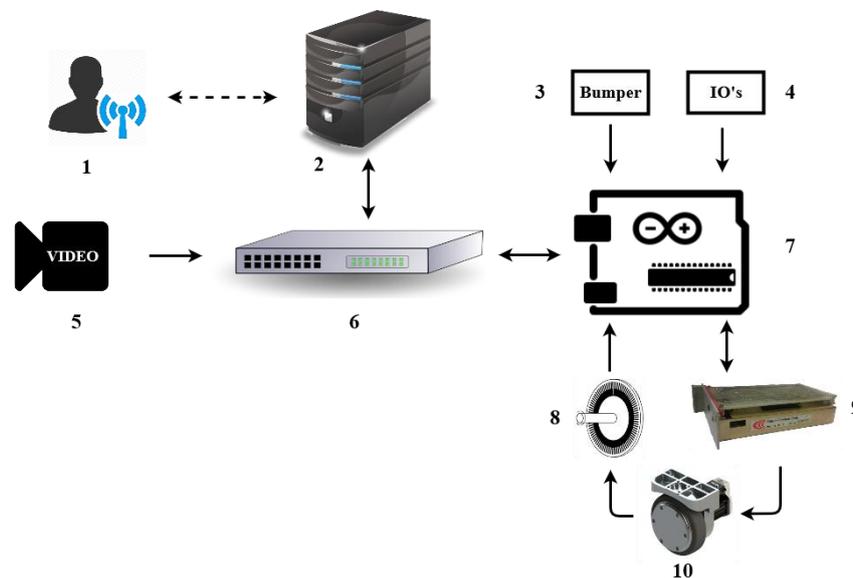


Figura 4.1 – Topologia de ligação do *hardware* para a integração dos elementos do AGV. 1 – Controlo e monitorização; 2 – Unidade central de processamento; 3 – Sinais do *bumper* detetor de colisões; 4 – Outros IO's; 5 – Câmera para aquisição de imagem; 6 – *Switch* de rede; 7 – Arduino; 8 – *Encoder*; 9 – *Drive* DC; 10 – Dois motores DC 300W.

Uma câmara de vídeo é responsável pela aquisição da imagem que irá ser recebida e processada pela unidade central. Consoante a informação nela contida, serão transmitidas mensagens ao controlador para atuar sobre os motores de acordo com a ordem de navegação.

Os motores são acionados através de *drives* DC que recebem sinais em PWM de acordo com a velocidade pretendida de rotação. Cada um dos motores está equipado com um *encoder* que gera um sinal quadrado de frequência variável com a sua velocidade de rotação. Este sinal será amostrado e processado pelo controlador para que se possa determinar a velocidade, em rpm. O valor da velocidade é então transmitido para a unidade central, fechando assim a malha de controlo.

No diagrama apresentado, existe ainda um sistema remoto que será utilizado para monitorizar e controlar a plataforma. Será desenvolvida uma aplicação, do tipo cliente TCP/IP que possibilitará a troca de mensagens e comandos entre o Robuter e o utilizador. Como este elemento não está incluído na plataforma, terá de se estabelecer uma rede entre o PC remoto e o PC principal, para que se possa comunicar via *wireless*.

4.2. *Hardware*

Neste subcapítulo é listado o material projetado para a realização deste projeto. Esta lista é provisória uma vez que está sujeita a aprovação de orçamento pela escola. Outros dos elementos mencionados advêm da sua disponibilidade no laboratório e de projetos anteriores, mas tal não implica que sejam definitivos porque podem revelar-se insuficientes.

Câmara

A câmara que irá ser utilizada para este projeto será a Flea3 GigE, modelo GE-28S4C-C. São câmaras de cor com 2.8MP e está equipada com um sensor: Sony ICX687 CCD, 1/1.8", 9.69µm; Resolução 1928 x 1448 *pixels* a 15 FPS.

Switch de rede

Switch de rede TL-SG1008P. Contém 8 portas GigE (1000 Mbps) onde em 4 têm PoE (*Power Over Ethernet*). Alimentação de 48V DC .Potência de 5.8W sem PoE. Potência máxima de 58W.

Microcontrolador

Chip ATmega32U4 integrado na placa de desenvolvimento Arduino Leonardo ETH. 20 pinos IO (entre eles 7 saídas PWM e 12 entradas analógicas). Cristal de 16MHz. Tensão de alimentação entre 7V e 12 V. Potência até 600mW. Arduino disponível no LAR.

CPU

Nesta fase, o CPU ainda não se encontra selecionado uma vez que há outros colegas a realizar projetos no LAR que também irão necessitar de um equipamento igual, pelo que a seleção será conjunta para que a encomenda possa ser feita no mesmo local e com o consentimento de todos. Apesar disso, aponta-se um CPU na linha da Intel® NUC. Este será a unidade central de processamento, onde irá estar o ROS.

Drives DC

Modelo 306 da Copley Controls. Tensão de alimentação 16 a 80V DC. Corrente máxima de saída 10A. Controlo com sinal PWM externo com frequência de *switching* de 22kHz e possibilidade de inversão do sentido de marcha. Estes são os *drivers* originais do motor que após a reparação de um deles ficaram operacionais.

Motores DC

Dois motores DC de ímanes permanentes. Potência 300W. Alimentação 48V. Redução de 3000 rpm no primário para 130 rpm no secundário. *Encoders* de dois canais com 8000 pulsos por rotação (resolução de 0.045°). **Nota:** O sinal dos *encoders* ainda deve ser revisto porque apresentava algum ruído que pode ter interferido nas medições (testes feitos por interrupção no Arduino). Espera-se melhorar a cablagem e soldar conetores apropriados para medições mais precisas.

Travões eletromagnéticos integrados KEB 48V.

Baterias

Relativamente às baterias, foi efetuada uma pesquisa para verificar se compensava alterar as baterias existente para tecnologia Lítio. Verificou-se que os preços são muito maiores para a mesma capacidade.

As 4 baterias contidas na plataforma são do mesmo modelo que as originais, Shapir 700, mas não puderam ser aproveitadas por estarem em fim de vida. Tensão de 12V e capacidade de 54 a 70Ah cada uma, o que totaliza uma energia de cerca de 3kWh. As baterias ainda não foram selecionadas mas espera-se conseguir obter algo dentro da mesma gama de valores para que o AGV possa ter alguma autonomia e capacidade de carga.

Fontes de alimentação

Conversor DC-DC Q1000 com tensão de alimentação até 150V. Saídas variáveis de 3.3V a 48V e saída fixa de 5.1, 12, 15 e 24V. Rendimento 90%. Potência máxima 132W.

Conversor DC-DC 24Q2000 com tensão de alimentação até 75V. Saídas variáveis de 3.3V a 48V e saída fixa de 5.1, 12, 15 e 24V. Rendimento 86%. Potência máxima 106W.

Bumper

Pára-choques para detetar colisões e desencadear a paragem imediata da plataforma.

Botoneira de emergência

Está prevista a colocação de uma botoneira de emergência em série com a alimentação do circuito.

5. Conclusões e plano imediato

Uma das dificuldades sentidas até ao momento é a existência de um vasto leque de opções que permitem dar solução ao trabalho proposto. A oferta é vasta, o que é algo positivo, mas torna a escolha mais difícil.

Até ao momento, considera-se que o projeto segue o seu curso natural, uma vez que cumpre o planeamento inicialmente proposto, mais concretamente a realização das três principais tarefas apresentadas no cronograma da Tabela 1.1. Porém, no terceiro ponto, é importante referir que nem todas as propostas de material estão totalmente fechadas.

Durante o primeiro semestre letivo foram desenvolvidas competências importantes para o desenvolvimento da plataforma, nomeadamente em *Linux* (distribuição Ubuntu 16.04):

- Elaboração de projetos em linguagem C em ambiente *cmake*;
- “Linkagem” e inclusão de bibliotecas externas;
- Comunicação entre processos;
- Documentação de código de programação através de Doxygen;
- Breve introdução ao C++.

Para além do proposto, foi elaborado no âmbito do projeto final da disciplina de Projeto de Automação e Robótica Industrial, uma interface em GTK+ gerido por um programa em linguagem C que permite atuar os motores do Robuter. Esta aplicação tem um processo paralelo à interface gráfica, um cliente TCP/IP que comunica com o servidor (Arduino Leonardo ETH), responsável por adquirir o sinal de velocidade de cada motor, em RPM e atuar sobre os mesmos. Está também preparada para adquirir o estado e atuar sobre os travões eletromagnéticos. Na Figura 5.1 é apresentada a aplicação desenvolvida que servirá como ponto de partida para testes iniciais. De notar que esta pode não ser a opção adotada para controlar/monitorizar a plataforma de forma remota.

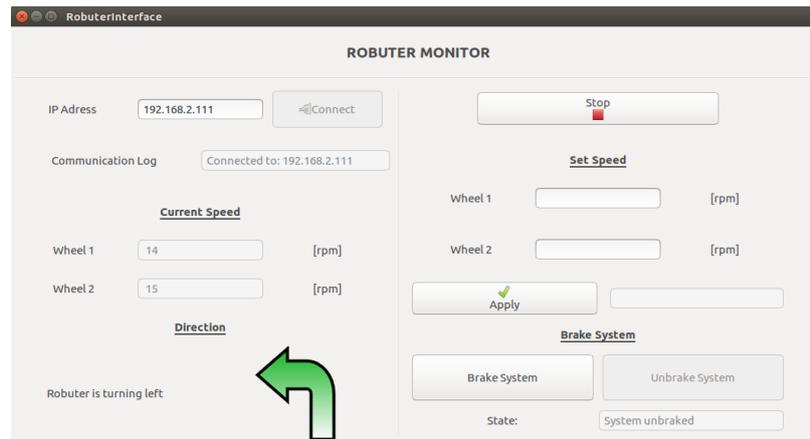


Figura 5.1 – Aplicação desenvolvida em GTK+ com programação em C para atuar e monitorizar os motores da plataforma Robuter.

Como plano imediato espera-se fechar a plataforma nos inícios de Março, isto é, terminar a fase de *hardware* para que a interligação dos subsistemas e integração de algoritmos de visão artificial possam ser integrados por *software*. Desta forma, a seleção e montagem dos equipamentos na plataforma avizinha-se como a tarefa mais importante nesta fase porque todas as outras dependem dela.

Ainda decorrer do mês de Fevereiro espera-se frequentar uma formação em ROS para potenciar a sua utilização neste projeto.

6. Referências Bibliográficas

- [1] “The History of AGV's,” Savant Automation, [Online]. Available: <http://www.agvsystems.com/history-agvs/>. [Acedido em 1 de Dezembro 2016].
- [2] *Electrical Engeneering, Volume 77, Issue 7*, p. 664, 1958.
- [3] J. Borenstein, H. R. Everett e L. Feng, *Navigation Mobile Robots Systems and Techniques*, A K Peters, 1996.
- [4] Robosoft, “B2B service robots robosoft,” Robosoft, [Online]. Available: <http://www.lokarria.com/corporate/museum.html>. [Acedido em 30 de Outubro 2016].
- [5] P. Pomiers, A. Semerano e V. Dupourque, “Trajectory tracking strategies in service robotics applications,” 1998.
- [6] A. Weiser e P. Lima, “An Integrated Learning, Planning and Reacting Algorithm Applied to a Real Mobile Robot,” 1996.
- [7] G. Gini e A. Marchi, “Indoor Robot Navigation with Single Camera Vision,” 2002.
- [8] A. Hentout, M. A. Messous e B. Bouzouia, “Multi-agent Control Approach for Autonomous Mobile Manipulators: Simulation Results on RobuTER/ULM,” 2014.
- [9] “ROS/Introduction,” ROS, [Online]. Available: <http://wiki.ros.org/ROS/Introduction>. [Acedido em 02 01 2017].