Instituto de Sistemas e Robótica

Pólo de Lisboa

# Acostagem de uma plataforma móvel a uma estação de trabalho fixa[1]

Pedro Mira Vaz Renato Ferreira Vasco Grossmann
Maria Isabel Ribeiro

Setembro de 1996
RT-601-96

ISR-Torre Norte
Av. Rovisco Pais
1096 Lisboa CODEX
PORTUGAL

[1]Este trabalho foi parcialmente financiado pela NATO Scientific Affairs Division no âmbito do programa NATO Science for Stability, projecto PO-ROBOT.

**Resumo**

Neste trabalho descreve-se uma estratégia de acostagem de um robot móvel não holonómico a uma estação de trabalho fixa, suportada num sistema sensorial com base em infra-vermelhos que permite uma localização relativamente ao ponto de acostagem. O facto de a plataforma ser não-holonómica obriga a que sejam geradas trajectórias suaves pois existem raios mínimos de curvatura.

Para identificar o ponto de acostagem foram utilizados dois cilindros de material reflector posicionados a uma distância conhecida um do outro. Sobre a plataforma foram colocados dois emissores-receptores de infravermelhos montados sobre motores passo-a-passo que permitem, a cada um dos sensores, varrer o espaço em redor, de modo a serem localizados os reflectores.

A estratégia de movimentação implementada está dividida em duas fases distintas. A primeira, designada por **aproximação**, tem como objectivo posicionar a plataforma numa posição onde seja possível obter uma leitura com o sistema sensorial. A segunda - **acostagem** - inicia-se quando se dispõe de uma medida dos sensores e leva a plataforma até ao ponto de acostagem, sendo a malha de controlo de movimentação fechada com dados sensoriais.

Todo o sistema sensorial é controlado por um microcontrolador PIC. Este comanda os sensores segundo as ordens recebidas do computador de bordo da plataforma, devolvendo ao processo os dados que recolheu. Com os ângulos medidos deste modo fazem-se, através de triangulação, os cálculos que permitem a localização da plataforma.

Apresentam-se resultados experimentais sobre a precisão e exactidão do procedimento de acostagem desenvolvido. Conclui-se que se consegue uma acostagem com um erro em ($x$, $y$, $t$) de ($\pm 0.5$ cm, $\pm 0.5$ cm, $\pm 1°$).

**Keywords:** Robótica Móvel, *Docking*, *Sensor Based Motion*, Sensores de Infravermelhos, Trajectórias para Veículos não Holonómicos

# Índice

# Lista de Figuras

# Lista de Tabelas

# 1 Introdução

Desde há alguns anos a esta parte, os AGVs (*Automated Guided Vehicles*), têm vindo a assumir uma importância crescente em numerosos ambientes industriais e comerciais, tais como linhas de montagem, armazéns, e outros locais onde o transporte automático de materiais, peças e produtos traz vantagens económicas relevantes pelo aumento da capacidade produtiva e pela redução de gastos.

Nas implementações mais comuns, os AGVs seguem uma rede de trajectórias pré-definidas, instaladas no solo sob a forma de pistas (eléctricas, ópticas, magnéticas, carris, etc).

Uma instalação deste tipo é pouco flexível dado que, para alterar a rede de trajectórias possíveis, é necessário também alterar a colocação física das pistas no ambiente de trabalho, podendo isso levar à suspensão temporária da actividade industrial ou comercial, com inconvenientes óbvios. Além do mais, é impossível, com AGVs clássicos, fazer o contorno de obstáculos, dado que é imperativo que estes sigam uma pista, sendo portanto a única solução esperar que o obstáculo seja removido, ou, no caso de o AGV ser bidireccional, voltar atrás e tentar seguir uma outra trajectória que esteja eventualmente livre; põe-se exactamente o mesmo problema com o cruzamento de AGVs em sentidos opostos: ou o planeamento das trajectórias entra em linha de conta com esse problema, evitando que eles se cruzem, ou são instaladas duas pistas em cada troço, uma para cada direção, ou são colocados, de onde em onde, locais que permitam o cruzamento. Qualquer destas soluções exige um esforço económico acrescido ou uma muito maior complexidade no planeamento das trajectórias.

Por tudo isto, tem-se vindo a sentir a necessidade de passar dos AGVs, que fazem seguimento de pistas, para robots móveis, que circulam mais livremente, utilizando técnicas de navegação mais flexíveis. No entanto este aumento de flexibilidade traz problemas acrescidos: enquanto que no caso dos AGVs bastava controlar uma variável (deslocamento ao longo da pista), com um robot móvel é necessário controlar três variáveis: posição (duas variáveis) e orientação. Para que se possa fazer este controlo é evidentemente necessário que se faça localização do robot móvel no espaço de trabalho. Diversas técnicas, como por exemplo emissores/receptores de rádio instalados no ambiente de trabalho, pontos de referência artificiais (cuja localização pode ser facilmente alterada) e visão, têm vindo a ser desenvolvidas e testadas para a localização absoluta de robots móveis.

Um problema particularmente difícil na navegação de robots móveis é a acostagem (*docking*). Quando se fala de AGVs, seguindo uma pista, com um grau de liberdade, basta a existência de uma qualquer marca na estação de trabalho onde se quer fazer a acostagem, para muito facilmente se saber quando parar o AGV. Com um robot móvel, fazer a acostagem a uma estação de trabalho implica conhecer a localização do robot móvel em relação à estação de trabalho (posição e orientação), pelo que se torna necessário um sistema de sensores mais complexo; exige um planeamento cuidadoso da trajectória, mormente no caso de robots não-holonómicos de maneira a que a acostagem se possa fazer na posição e orientação desejadas; implica ainda que se faça controlo de movimento para o seguimento correcto da trajectória planeada.

É neste contexto que se insere o objectivo do presente trabalho: projectar, implementar e testar um sistema de sensores capaz de fornecer os dados necessários à correcta localização de um robot móvel relativamente a uma estação de trabalho fixa, e, com esta informação, conseguir a acostagem do robot à referida estação de trabalho, através do correcto planeamento das trajectórias a seguir, assim como do controlo de movimento corrigido pela informação obtida a partir do referido sistema sensorial. Assumem-se conhecidas, a menos de um determinado erro, a posição e a orientação iniciais da plataforma móvel em relação à estação de trabalho.

A organização do relatório é a seguinte.

No Capítulo 2 descreve-se o ambiente de trabalho e o material à disposição para a consecução do objectivo, assim como alguns condicionamentos a que o desenvolvimento deste trabalho esteve sujeito.

No Capítulo 3 discutem-se algumas soluções possíveis para um sistema sensorial a implementar na plataforma móvel capazes de fornecer os dados de localização necessários à movimentação correcta da plataforma; o sistema escolhido é então descrito, quer quanto ao princípio geral de funcionamento, quer quanto à implementação que foi feita. Termina-se o capítulo com a apresentação dos resultados do estudo estatístico que foi feito sobre o sistema completo.

No Capítulo 4 descreve-se o algoritmo desenvolvido para o planeamento das trajectórias de acostagem e para a acostagem de precisão.

Resultados experimentais relevantes são apresentados no Capítulo 5.

Finalmente no Capítulo 6 apresentam-se as conclusões e discutem-se perspectivas futuras na continuação do presente trabalho.

# 2 Descrição do ambiente de trabalho

O presente trabalho foi maioritariamente desenvolvido no Laboratório de Robótica Móvel do Instituto de Sistemas e Robótica, Pólo do Instituto Superior Técnico.

Neste laboratório, utilizou-se:

- uma plataforma móvel Robuter, da Robosoft
- uma estação de trabalho gráfica Sun Sparc Classic
- um PC para apoio ao desenvolvimento de *hardware* e para simulação e aperfeiçoamento de algoritmos
- um *kit* de desenvolvimento para microcontroladores PIC - PICStart 16C

Todo o desenvolvimento da electrónica e montagem dos circuitos utilizados, foi levada a cabo nos Laboratórios para Trabalhos Finais de Curso da Secção de Sistemas e Controlo, do Departamento de Engenharia Electrotécnica e de Computadores, no Instituto Superior Técnico, onde foram utilizados os aparelhos de medida usuais (multímetro, osciloscópio, frequencímetro, etc) e as ferramentas necessárias à montagem dos circuitos nas placas de circuito impresso (berbequim, ferro de soldar, chupa-soldas, etc.).

## 2.1 Plataforma Robuter

Especificações técnicas relevantes [11]:

- Dimensões: 1025 x 680 x 440 mm
- Peso: cerca de 150 Kg
- Baterias: 4x12 V, 60Ah
- Motores: 2 motores de corrente contínua (magnetos permanentes) de 300 W.
- Direcção diferencial, por controlo independente de velocidade das duas rodas motrizes
- Computador de bordo:
  - Bus VME, 68020, 16 MHz
  - 4 portas série
  - Alimentação: 5 V 16 A, +/- 12 V 1.7 A
- Sistema Operativo em Tempo Real - Albatros - [2], desenvolvido pela Robosoft, especificamente para a Robuter
- Programação: Package de *Software* de desenvolvimento para compilação e *download* de *software* para o computador de bordo [3].

Na Figura 1 apresenta-se a plataforma móvel Robuter.

## 2.2 Estação de trabalho gráfica Sun Sparc Classic

Esta estação foi utilizada para desenvolver, compilar e fazer o *download* para o computador de bordo da Robuter, do processo destinado a gerir o sistema de sensores, e a definir e fazer executar pela plataforma as trajectórias necessárias à acostagem pretendida.

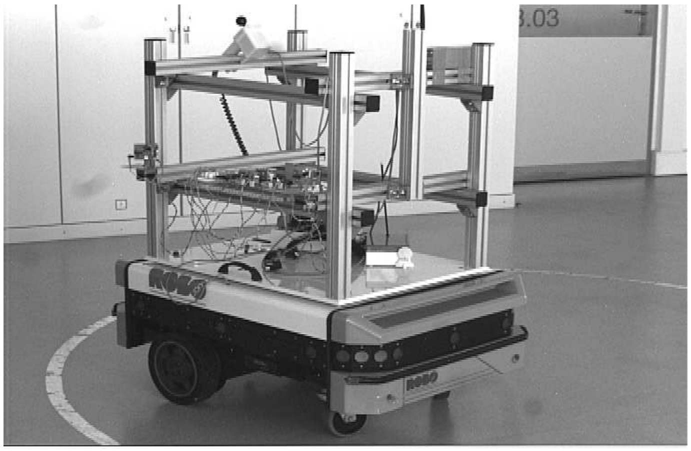

Figura 1: Aspecto geral da Robuter

## 2.3 PC (Computador Pessoal)

Utilizou-se para apoio a este projecto um PC 486DX2 66 MHz, fornecido pelo Departamento de Engenharia Electrotécnica e de Computadores, com o sistema operativo Windows 95. Utilizaram-se, em fases distintas do projecto, diferentes programas, cuja utilização se descreve em seguida.

### 2.3.1 PSpice Electrical Circuit Simulation, v.5.4 (Evaluation)

Este programa destina-se à simulação de circuitos electrónicos. A versão *evaluation* utilizada tem como única limitação o facto de não se poder exceder um número máximo de componentes em cada circuito simulado.

Esta versão do programa foi desenvolvida para utilização no ambiente Windows, e inclui um pacote de edição gráfica de esquemáticos, pelo que não é necessária a descrição do circuito num ficheiro de texto.

Este programa inclui numerosas bibliotecas de componentes, que se utilizaram para a descrição do circuito simulado.

Os resultados da simulação são apresentados mais à frente, aquando da descrição do sistema sensorial (Figura 13).

### 2.3.2 Tango Schematic Editor, v.1.20

O Tango Schematic Editor é um módulo do pacote de *software* Tango destinado ao projecto de placas de circuito impresso.

Este módulo permite a utilização de uma biblioteca extensiva de componentes, num ambiente gráfico, para que se faça a descrição do circuito, e se crie um ficheiro com a "*net list*" (lista de ligações), que será utilizado por outros módulos descritos em seguida, para a continuação do projecto das placas.

### 2.3.3 Tango PCB Plus, v.2.00

Neste módulo, são colocados os componentes sobre a placa de circuito impresso, já nas suas verdadeiras dimensões.

Carrega-se então o ficheiro de ligações criado pelo Schematic Editor, para depois se poder utilizar o módulo seguinte (*router*).

### 2.3.4 Tango-Route Plus, v.2.00

Este último módulo do pacote de *software* Tango destina-se a determinar automaticamente o melhor traçado das pistas da placa de circuito impresso, a partir da informação gerada pelos 2 módulos precedentes. Voltando ao módulo anterior, estas pistas podem ser corrigidas pelo utilizador.

Termina-se com este módulo o projecto das placas de circuito impresso.

### 2.3.5 Borland Delphi, v.1.0

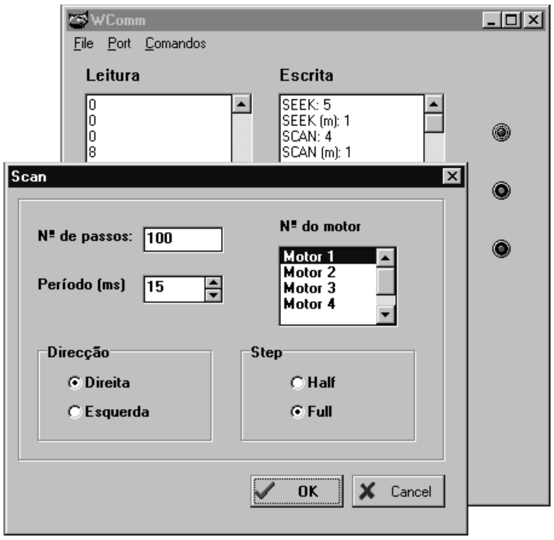


Figura 2: Wcomm em envio de comandos

O Delphi é uma linguagem de programação visual, em Object Pascal, para o desenvolvimento de aplicações em Windows.

Desenvolveu-se com este *software* uma aplicação - WComm -, com o objectivo de se enviarem comandos para o sistema sensorial, recebendo os dados devolvidos, numa fase preliminar do desenvolvimento do mesmo. Esta aplicação fornece uma *interface* muito simples, permitindo uma fácil visualização dos resultados.

Nas figuras 2 e 3 apresentam-se algumas potencialidades desta aplicação.

### 2.3.6 Borland C++, v. 3.1

Utilizou-se para o desenvolvimento de uma aplicação destinada ao aperfeiçoamento, por simulação, do planeamento de trajectórias.

Desenvolveu-se uma aplicação para DOS, em modo gráfico, que simula o movimento da Robuter, mostrando a trajectória seguida, utilizando exactamente os comandos que são enviados para a plataforma - Traject. Permitiu um muito rápido aperfeiçoamento dos algoritmos desenvolvidos, a par de um sistema de *debugging* muito poderoso.

## 2.4 PICStart 16C

Este *kit* de desenvolvimento [10] destina-se à programação dos microcontroladores produzidos pela MicroChip. É utilizado em computadores pessoais, sobre o sistema operativo DOS.

A família de microcontroladores PIC da MicroChip, descritos com maior detalhe no capítulo seguinte, permite uma fácil programação em *assembly*, utilizando o PICStart 16C, que inclui:

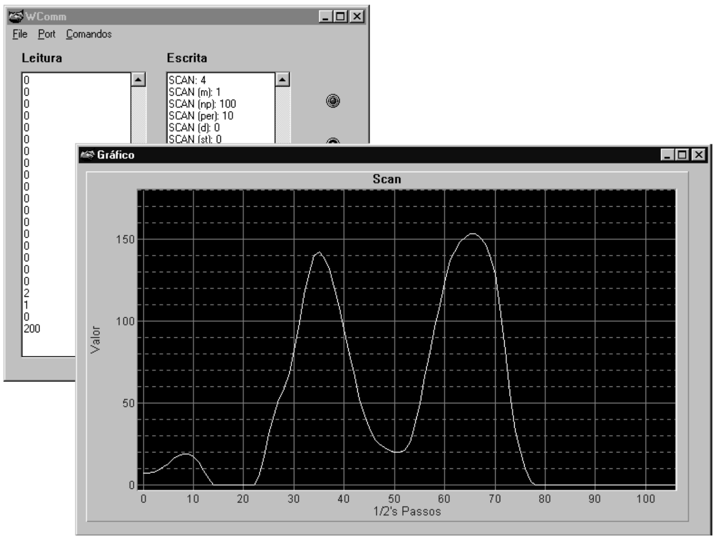


Figura 3: WComm - Gráfico de recepção de sinal

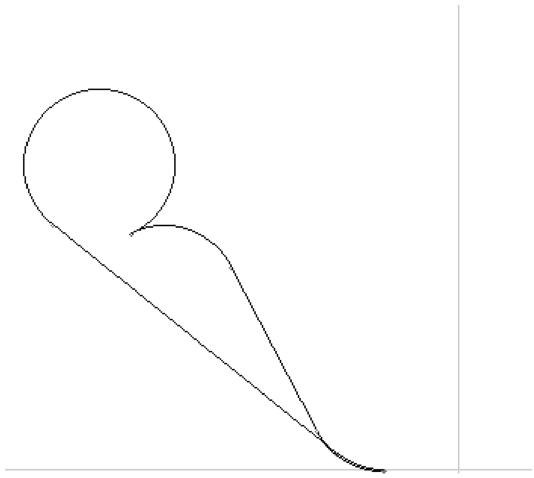

Figura 4: Traj - 2 trajectórias possíveis

- MPasm - *Assembler* para os microcontroladores PIC
- MPSim - ferramenta de simulação e visualização de resultados
- MPS16C - ferramenta para programação dos microcontroladores
- Programador com ligação série, a ser utilizado conjuntamente com o MPS16C.

# 3 Sistema Sensorial

Para a resolução do problema apresentado, ou seja, realizar a acostagem de um robot móvel a uma estação de trabalho fixa, é necessário projectar um sistema capaz de obter a informação necessária à correcta localização do robot em relação à estação de trabalho.

Como se verá com mais detalhe no Capítulo 4, o problema pode subdividir-se em dois problemas distintos. O primeiro refere-se à situação em que o robot móvel está demasiado longe da estação de trabalho para que o sistema sensorial, seja ele qual for, detecte correctamente a sua localização relativamente à estação de acostagem. Neste caso, utilizam-se as condições iniciais dadas sobre uma estimativa da localização num referencial global, para fazer uma aproximação do robot móvel à estação de trabalho, de modo a que se possa vir localizar através do sistema sensorial. O segundo problema, já à vista da estação de trabalho, consiste em planear as manobras de acostagem, complicadas pelo facto de o veículo ser não-holonómico.

Este tipo de solução exige que as condições iniciais dadas - localização e orientação da plataforma móvel em relação à estação de trabalho fixa - tenham um erro suficientemente pequeno para que os primeiros movimentos de aproximação permitam que o robot móvel se aproxime da estação de trabalho de modo a que o sistema sensorial funcione correctamente.

O sistema sensorial a escolher deve apresentar precisão, não sendo no entanto necessário que tenha grande alcance, visto ser possível fazer uma primeira aproximação da plataforma à estação de trabalho, até que aquela se encontre mais próxima.

Assim, consideraram-se diversas soluções, as quais se apresentam de seguida, com uma muito breve análise das vantagens e inconvenientes de cada uma.

Como já foi dito, o seguimento de pistas não é uma solução a considerar, pelo facto de se querer precisamente aumentar a flexibilidade e autonomia da plataforma em relação aos AGVs clássicos.

Sistemas de navegação baseados em odometria podem ser extremamente precisos, através do uso de *encoders*, giroscópios, bússolas, etc. No entanto, para se obter essa precisão, é necessário o emprego de equipamentos demasiado dispendiosos para o trabalho em curso. A grande desvantagem que apresentam é o facto de a localização ser feita num referencial global do mundo, e não fazer uso de dados externos.

Outra hipótese que se põe é a utilização de sistemas de visão e processamento de imagem. Estes sistemas, para além de apresentarem sempre um peso computacional elevado, são também dispendiosos. Permitem um alcance grande, dependendo da qualidade do dispositivo de aquisição de imagem utilizado.

A última possibilidade estudada foi a utilização de pontos de referência artificiais, de fácil colocação e grande flexibilidade. Por uma questão de viabilidade económica, decidiu-se que todos os pontos de referência deveriam ser passivos, mantendo toda a parte activa do sistema concentrada na plataforma móvel.

Deste modo, eliminou-se a possibilidade de ter emissores de rádio, infra-vermelhos, laser, ou de qualquer outro tipo, como pontos de referência. Decidiu-se então utilizar como sensor um conjunto emissor/receptor de infra-vermelhos capaz de detectar o sinal emitido, reflectido nos pontos de referência, que seriam assim reflectores. O sensor assim concebido foi montado sobre um motor, colocado sobre a plataforma móvel. Utilizando-se dois destes conjuntos, juntamente com dois reflectores como pontos de referência, é suficiente conhecer as direcções de cada um dos reflectores em relação a cada um dos sensores, a distância entre reflectores e a distância entre sensores, para conhecer a posição e orientação da plataforma relativamente à estação de trabalho (ver Figura 5).

Optou-se pela utilização de infra-vermelhos, pelo baixo custo dos componentes de emissão e recepção, pensando-se que estes teriam o alcance suficiente para o objectivo pretendido.

Fizeram-se diversos testes com circuitos conhecidos de emissão/recepção de infra-vermelhos, que não se mostraram de um modo geral muito satisfatórios. Projectou-se por isso um circuito com as características desejadas, que, após alguns testes, se revelou adequado ao objectivo.

Para os motores, optou-se imediatamente por motores passo-a-passo, por serem de relativamente baixo custo e não apresentarem problemas de controlo de posição.

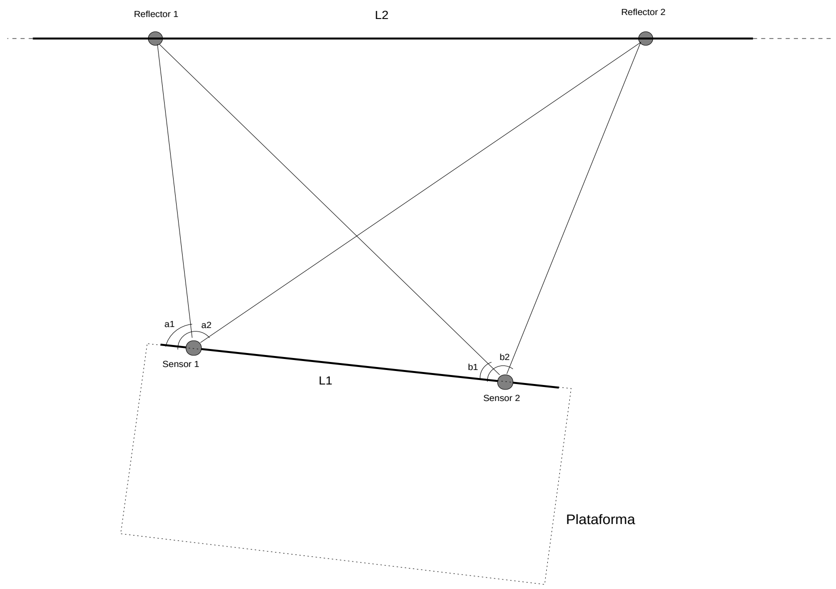


Figura 5: Triangulação

Para um teste comparativo de sistemas de navegação ver [5].

### 3.1 Descrição

O sistema sensorial projectado é constituído por 2 módulos de emissão/recepção de infra-vermelhos, montados sobre motores passo-a-passo. Inclui um microcontrolador (PIC 16C74), que faz a *interface* entre os módulos de emissão/recepção e respectivos motores, e o processo a correr no computador de bordo da plataforma móvel (ver Figuras 6 e 8). Todo o *software* desenvolvido para o microcontrolador prevê a utilização de até quatro módulos de emissão-recepção, juntamente com os respectivos motores.

Em fase de teste, foi utilizado um motor passo-a-passo adquirido num ferro-velho. Posteriormente, aquando da montagem da versão final, deparou-se com a inexistência de mais motores nos retalhistas da especialidade. Conseguiram-se, por oferta do Pólo de Coimbra do Instituto de Sistemas e Robótica, que aqui se agradece, 2 motores passo-a-passo, que no entanto tinham apenas 100 passos por volta (apesar de permitirem a utilização em meio-passo), quando os inicias tinham 200, e tensões de alimentação diferentes daquelas exigidas pelo motor inicial. Na motagem final, utilizaram-se, por isso, 2 motores adquiridos no ferro-velho.

Para a comunicação entre o computador de bordo e o microcontrolador utilizou-se a norma RS232, tendo sido definido um conjunto de comandos e respostas a enviar de um para outro.

Como reflectores utilizaram-se troços de tubo de PVC 40 mm, com cerca de 12 cm de comprimento, forrados exteriormente com uma fita reflectora de grande qualidade. O formato cilíndrico permite a detecção a partir de qualquer direcção (ver Figura 7).

### 3.2 Controlo

Por controlo do sistema sensorial, entendem-se as funções implementadas no microcontrolador, activadas através de comandos enviados pelo processo do computador de bordo (68020), destinadas a controlar os módulos de emissão/recepção de infra-vermelhos, juntamente com os respectivos motores (ver Figura 8).

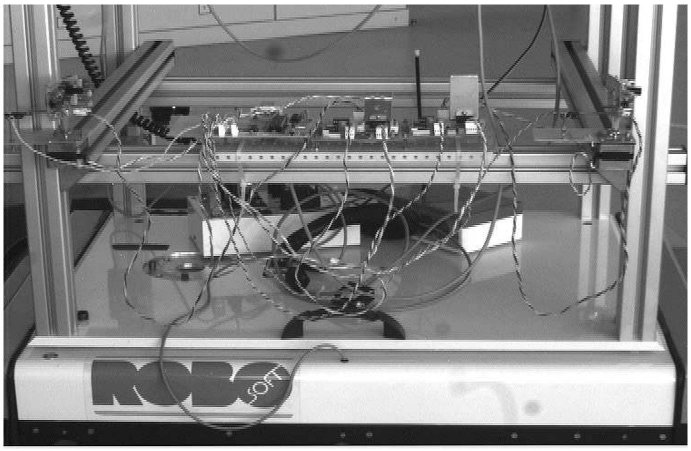


Figura 6: Aspecto geral do sistema sensorial

Nesta secção, começa por se descrever o microcontrolador. De seguida explica-se o modo como é feita a comunicação entre o microcontrolador e o computador de bordo da plataforma móvel e, por fim, apresentam-se os comandos implementados.

Na Figura 9 apresenta-se o aspecto geral da placa onde se implementa o sistema de controlo.

### 3.2.1 Descrição do microcontrolador PIC

O microcontrolador utilizado (PIC16C74) pertence à família PIC16/17 da Microchip. Trata-se de um microcontrolador de 8 *bits* do tipo RISC e arquitectura *Harvard* (memórias de programa e de dados separadas) com 4k x 14 *bits* de memória de programa e 192 *bytes* de memória de dados. Dos seus 33 pinos de I/O, 8 podem ser configurados como entradas analógicas, sendo o conversor A/D interno ao PIC e com resolução de 8 *bits*. Todos os pinos de I/O podem ser configurados individualmente como entradas ou saídas, podendo fornecer ou receber uma corrente de 25 mA. O PIC16C74 tem ainda dois temporizadores/contadores de 8 *bits* e um de 16 *bits*, tem dois geradores de PWM (*Pulse Width Modulation* — Modulação por Largura de Impulso) com factor de ciclo programável e resolução de 10 *bits*, uma *interface* de comunicação série (*USART*) e um temporizador *Watchdog*. A *USART* pode ser configurada para funcionar em modo síncrono ou assíncrono, paridade par ou ímpar ou sem paridade, e o seu gerador de *baud rate* (taxa de transmissão) pode chegar a 1250 k*bits* por segundo [9].

A arquitectura *Harvard*, que se caracteriza essencialmente pela separação entre as memórias de dados e de programa, aliada a um conjunto de instruções que ocupam apenas uma palavra de memória, permitem que o CPU execute uma instrução e ao mesmo tempo carregue a próxima. Assim, com excepção das instruções de salto, todas as instruções são executadas num único ciclo de relógio. As instruções que envolvem saltos no programa são executadas em dois ciclos pois a instrução que foi carregada automaticamente durante a instrução de salto não é a que vai ser executada de seguida. Esta característica permite, quando necessário, um conhecimento preciso do tempo de execução de cada rotina.

### 3.2.2 Comunicações

Como já foi referido, o computador de bordo da Robuter está equipado com 4 portas série, estando a ser utilizada a porta 2 para comunicações através da norma RS232. O *buffer* de entrada existente é de 128 *bytes*, o que implica que se tenha o cuidado de nunca pedir de uma só vez dados que excedam, em *bytes*, este número. No caso de isso se revelar necessário, fazem-se vários pedidos sucessivamente.

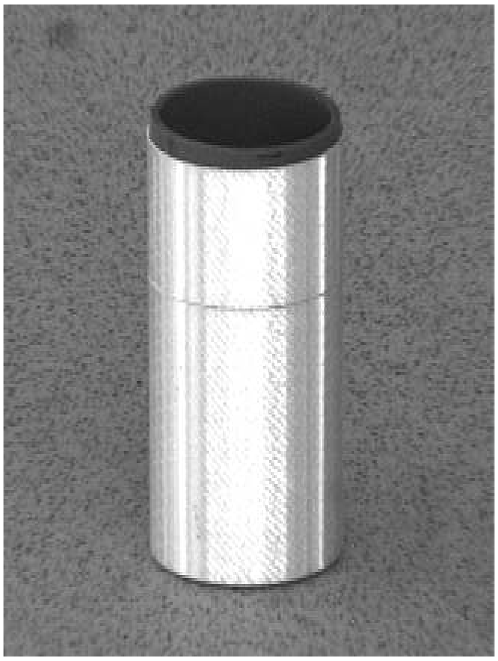

Figura 7: Reflector

Como também já foi dito, o PIC está equipado com uma *interface* de comunicações série, com um *buffer* de 2 *bytes*. O facto do *buffer* ser relativamente pequeno não representa, no entanto, um problema, dado que o programa implementado no PIC permite a leitura de dados do *buffer* a uma taxa maior do que a taxa de transferência utilizada entre o PIC e o computador de bordo.

Utilizou-se na comunicação série uma taxa de transmissão de 9600 *baud*, 8 *data bits*, sem *bit* de paridade, 1 *stop bit* e sem controlo de fluxo.

Não se encontraram erros de comunicação em nenhuma fase de desenvolvimento ou teste do sistema.

### 3.2.3 *Software* no PIC

Como já foi referido, utilizou-se o *kit* PICStart 16C [10] para o desenvolvimento do *software* implementado no microcontrolador.

Este *software* tem como objectivo dar resposta aos comandos enviados pelo processo a correr no 68020 da Robuter (descritos adiante), destinados ao controlo do sistema sensorial, e gerir a comunicação entre o 68020 e o PIC.

O PIC mantém internamente a posição actual de cada motor/sensor, validando qualquer pedido de movimento proveniente do processo da Robuter, garantindo que não se excedem os limites mecânicos do sistema.

O processo de leitura dos valores recebidos em cada sensor decorre em três fases consecutivas. Primeiramente, fazem-se 16 leituras do conversor A/D interno do PIC, calculando-se a média. As 16 leituras, todas correspondentes à mesma posição do motor, são extremamente importantes, dado que o desvio padrão das leituras, na mesma situação é considerável. Assim, a média das 16 leituras, em cada passo, faz uma filtragem preliminar dos valores recebidos. A segunda fase consiste em fazer o motor/sensor avançar um passo (ou meio passo, como se verá adiante) na direcção pretendida, havendo depois, na terceira fase, um período de espera, antes de se recomeçarem as leituras. Este período permite que o circuito de recepção e filtragem estabilize. As duas primeiras fases demoram

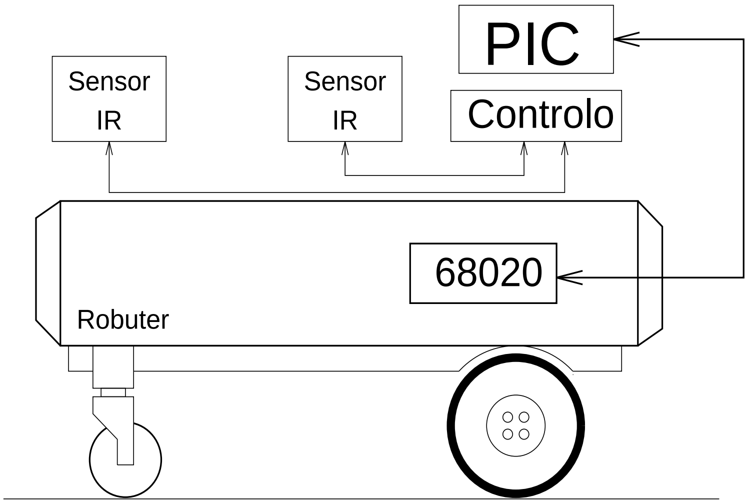


Figura 8: Diagrama do sistema sensorial

um tempo fixo, dependendo do comando que as activou (**scan** ou **scan2**), a que é então somado o tempo da terceira fase, determinado pelo processo que corre no processador da Robuter, e enviado como parâmtero. É com este parâmtero que se controla a velocidade de leitura nos comandos **scan** e **scan2**. Ver-se-á no estudo estatístico a sua importância.

Este período também existe nos comandos **move** e **seek** (ver Sub-secção 3.2.4), estando no entanto estabelecido no código do programa no PIC, determinando-se assim à partida a velocidade a que estes dois comandos são executados.

Pensou-se inicialmente em dotar a Robuter de 4 conjuntos motor/sensor, 2 de cada lado, para permitir a acostagem em qualquer das direcções. Assim, todo o *software* desenvolvido permite a utilização dos 4 conjuntos, apesar de, neste trabalho, se utilizarem apenas 2, de um só lado. Esta simplificação não retira generalidade ao resto do trabalho.

A listagem do programa do microcontrolador é apresentada em anexo.

### 3.2.4 Comandos

Para a comunicação entre o processo na Robuter e o PIC, definiram-se palavras de código - comandos e respostas de retorno -, fazendo corresponder um *byte* diferente a cada uma (ver Tabela 1).

O PIC aceita os seguintes comandos do processo que corre no processador 68020 da Robuter:

- **move** (num_sensor, num_passos, direc)
- **scan** (num_sensor, num_passos, período, direc, h/f)
- **seek** (num_sensor)
- **test** ()
- **en_ir** (num_sensor, ligar/desligar)
- **scan2** (num_sensor_1, num_sensor_2, direc_1, direc_2, h/f_1, h/f_2, período, num_passos)

Definiram-se ainda mais duas palavras utilizadas na comunicação, devolvidas pelo PIC ao processador da Robuter:

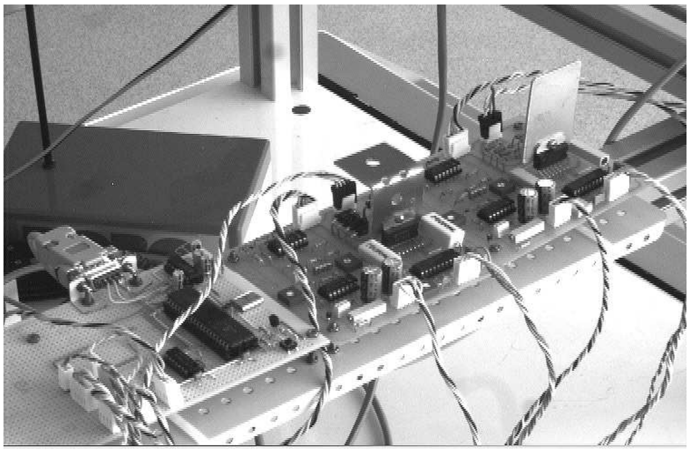

Figura 9: Aspecto geral do controlo do sistema sensorial

Tabela 1: Correspondência Comandos/Códigos

| **Comando** | **Código** |
|---|---|
| NPARERR | 1 |
| OK | 2 |
| **move** | 3 |
| **scan** | 4 |
| **seek** | 5 |
| **test** | 6 |
| **en_IR** | 7 |
| **scan2** | 8 |

- OK
- NPARERR

Os parâmetros **h/f**, **direc** e **ligar/desligar** podem tomar os valores 1 ou 2, com o significado apresentado na Tabela 2. Não se pôde utilizar o valor 0 e 1, como seria natural, pelo facto de o valor 0, nas rotinas de comunicação série da plataforma móvel, indicar o fim de transmissão.

**Move**

O comando **move** serve para movimentar o sensor numa dada direcção, o número de passos desejado. Os seus parâmetros são o número do sensor a que o comando se refere, o número de passos que o sensor deve rodar e o sentido desta rotação. O **move** utiliza sempre o motor passo-a-passo em passo completo. Este comando devolve OK, o número do sensor a que o comando se refere e a posição do sensor no final do movimento.

**Scan**

**Scan** é o comando que faz rodar o sensor numa determinada direcção, e que, em cada passo do movimento do sensor, lê a quantidade de energia reflectida e envia essa informação ao processo que corre no computador de bordo da Robuter. São esses valores que permitem que se detecte onde estão os reflectores. Os seus parâmetros são o número do sensor a que o comando se refere, o número de passos que o sensor deve rodar, o período de espera entre passos consecutivos, o sentido da rotação e

Tabela 2: Valores dos parâmetros

| Parâmetro | 1 | 2 |
|---|---|---|
| **h/f** | half step | full step |
| **direc** | sent. directo | sent. inverso |
| **ligar/desligar** | ligar | desligar |

indicação de meio passo ou passo completo para o motor. No final do movimento, o comando devolve OK, o número do sensor a que o comando se refere e a posição do sensor no final do movimento.

**Seek**

O comando **seek** serve para calibrar a posição do sensor. O seu objectivo é fazer rodar o sensor designado para a esquerda até que este active um microinterruptor, correspondente à posição zero. Começa por se rodar o sensor 10 passos para a direita para que, no caso em que o microinterruptor esteja activado à partida, o conjunto fique exactamente no primeiro passo em que o microinterruptor é ligado. Estes movimentos são sempre feitos no modo de passo completo. Não se põe o problema de não se poderem rodar os 10 passos iniciais para a direita, dado que o *software* no microcontrolador impede que se atinjam, em qualquer situação, posições excessivas.

Este comando tem como parâmetros o número do sensor a que se refere e devolve OK, o número do sensor a que o comando se refere e a posição final do mesmo. A posição é descrita por 2 *bytes*, dado que pode tomar valores superiores a 127. Assim, a resposta enviada pelo PIC contém 4 *bytes*.

É evidente que neste caso é redundante o envio da posição, dado que vai ser sempre zero. A única razão para a opção tomada é o facto de isto simplificar o programa no PIC, que retorna esta informação tanto para o **seek**, como para o **move**, **scan** e **scan2**.

A informação devolvida pode ser utilizada para verificar o correcto funcionamento do sistema, tornando-o simultaneamente mais flexível. A informação sobre a posição dos sensores em cada momento, que no presente caso se mantém internamente ao processo a correr na Robuter, pode ser útil em programas multi-processo.

**Test**

Este comando destina-se à verificação da comunicação. Existe apenas para dar apoio ao desenvolvimento e não é enviado pelo processo durante o funcionamento normal. Caso o microcontrolador receba este comando, devolve OK, permitindo ao processo que enviou o comando confirmar o bom funcionamento da comunicação.

**En_IR**

Comando que permite ligar ou desligar o emissor infra-vermelhos associado ao sensor referenciado no primeiro parâmetro do comando.

**Scan2**

Após a utilização do sistema sensorial durante algum tempo, e depois de um estudo estatístico do comportamento dos sensores em função da frequência de rotação, chegou-se à conclusão de que seria útil dotar o sistema de uma função que permitisse fazer leituras e rotações de dois sensores intercaladamente, aproveitando o tempo de movimentação de um dos motores para fazer a leitura do valor presente no outro sensor.

Dado que o tempo de processamento nesta função é significativamente maior do que o tempo de processamento no comando **scan**, um período de espera de 1 ms nesta função corresponde aproximadamente a um período de 20 ms num **scan**.

Chegou-se assim a esta função que recebe os mesmos parâmetros que o comando **scan** duplicados (para os dois sensores), à excepção do número de passos e do período que são sempre iguais para os dois sensores.

**NPARERR**

Sinaliza um erro nos parâmetros passados num comando (erro no número de parâmetros, parâmetros fora dos limites, ou parâmetros que levariam o sensor para uma posição inválida).

**OK**

A resposta "OK", enviada pelo microcontrolador para o processo na Robuter, sinaliza uma correcta recepção e interpretação do comando recebido, qualquer que este seja.

## 3.3 Electrónica

A electrónica do sistema de sensores pode ser dividida em 5 módulos funcionalmente distintos:

- Emissor de infra-vermelhos
- Receptor de infra-vermelhos
- *Driver* do motor
- Electrónica de comunicações
- Electrónica do PIC

Pelo facto de alguns destes módulos estarem distribuídos por placas de circuito impresso distintas, e também por haver placas de circuito impresso com diferentes módulos funcionais simultaneamente, optou-se por descrever os módulos funcionais, após o que se descrevem as placas de circuito impresso.

Note-se que as figuras com o desenho dos circuitos, retiradas directamente do Tango Schematic Editor, têm todas as ligações externas indicadas como saídas, dado que isso facilita o posterior desenho das placas nos restantes módulos do Tango. No entanto, respeitou-se a convenção que estabelece que as entradas se encontram do lado esquerdo do desenho do circuito, estando as saídas do lado direito. Note-se também que o valor das resistências tem como unidade o *ohm* ($\Omega$), não sendo este representado nos esquemáticos por o editor não ter a capacidade de representar símbolos especiais.

### 3.3.1 Emissor de infra-vermelhos

O emissor de infra-vermelhos, representado na Figura 10 é um circuito extremamente simples: seis LEDs emissores de infra-vermelhos ligados em série são atacados por um sinal rectangular, com uma relação de ciclo reduzida, para permitir a emissão de pulsos de grande potência, mantendo o valor médio da corrente que passa em cada LED abaixo de 100 mA. Os pulsos são gerados por um transistor montado em emissor comum, regulado por um sinal proveniente da saída PWM do PIC.

A frequência de emissão é regulada conjuntamente com o filtro passa-banda, de modo a que se consiga, na recepção, separar o sinal infra-vermelho emitido de ruído presente no LED receptor.

Vê-se na Figura 11 o aspecto geral do conjunto emissor/receptor, montados sobre o motor.

As duas principais fontes de ruído são o sol e a iluminação eléctrica. O sol tem uma componente contínua fortíssima quando comparada com a energia emitida pelos LEDs infra-vermelhos; as fontes de iluminação eléctrica, por seu turno, geram imenso ruído na frequência de 100 Hz, visto a frequência da potência emitida ser o dobro da frequência da tensão da rede. Por último, dado que o sensor se move de modo a fazer leituras em passos sucessivos, podem ocorrer degraus luminosos - janelas, por exemplo - que, na transição, introduzem picos com um espectro que se vai atenuando com o aumento da frequência.

Seria portanto vantajoso que a frequência de funcionamento do sistema sensorial projectado fosse a mais alta possível, dentro da gama permitida pelos componentes. Um valor aceitável estaria na ordem dos 40 kHz, pelo facto de permitir uma filtragem de grande qualidade, dado que esta frequência está mais afastada das frequências das principais fontes de ruído. No entanto, pelo facto de se utilizarem amplificadores operacionais de baixo custo e desempenho, reduziu-se a frequência de trabalho para cerca de 6 kHz, com a percepção de que o sistema fica mais sensível ao ruído [12].

A frequência exacta de trabalho (6172,8 Hz) foi ajustada em função do estudo feito sobre o filtro, como se apresenta mais à frente.

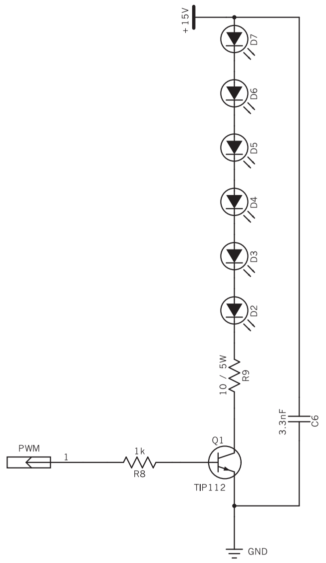


Figura 10: Circuito emissor de infra-vermelhos

### 3.3.2 Receptor de infra-vermelhos

O circuito receptor de infra-vermelhos pode ser subdividido em 3 módulos distintos: o LED receptor seguido de um amplificador de corrente, de um filtro, e de um amplificador condicionador de sinal. O amplificador de corrente e o filtro estão esquematicamente representados na Figura 12.

Os dois primeiros estão localizados numa mesma placa de circuito impresso, por duas razões: o LED receptor tem que estar fisicamente perto do amplificador de corrente, dado que o sinal é muito fraco e se atenua muito com a distância; como o amplificador de corrente utiliza um amplificador operacional, e o filtro passa-banda (*biquad*, activo) utiliza três, torna-se possível utilizar o circuito integrado LM324, que inclui 4 amplificadores operacionais, para, com poucos mais componentes discretos, implementar os dois módulos na mesma placa.

Os LEDs emissores também têm que estar na mesma placa, para que estejam sempre alinhados com o LED receptor. Assim, o circuito emissor é também implementado na mesma placa porque é necessário que o circuito de emissão esteja próximo do LED emissor, para que não tenha que existir cablagem a conduzir um sinal de grande potência (comparada com a de recepção) que introduz muito ruído, na frequência central do filtro. Ainda assim, essa influência faz-se notar fortemente.

A frequência central $f_{calc}$ do filtro é calculada, e chega-se a [13]:

$$f_{calc} = \frac{1}{RC} = 6290.7 \text{ Hz}$$

No entanto, dado que os cálculos feitos consideram o modelo ideal do amplificador operacional (corrente de entrada nula, ganho de modo comum nulo e ganho de modo diferencial infinito), e ainda que a frequência natural do filtro está deslocada, simulou-se o circuito no PSpice, verificando-se que a resposta máxima do filtro está centrada em:

$$f_{sim} = 6157,7 \text{ Hz}$$

O desvio da frequência natural do filtro em relação à frequência central (frequência dos pólos) deve-se ao facto de os pólos serem complexos conjugados [7].

A frequência do sinal de saída de PWM do PIC apenas pode tomar valores discretos, utilizando um *timer* (registo interno PR2), a cujo valor se chega pela expressão:

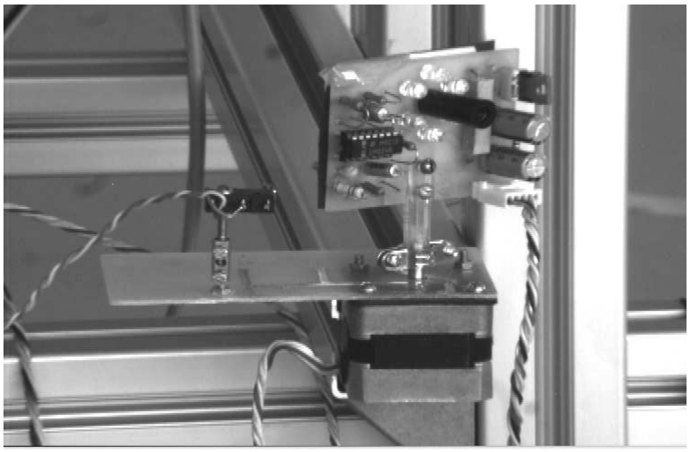

Figura 11: Aspecto geral do sensor

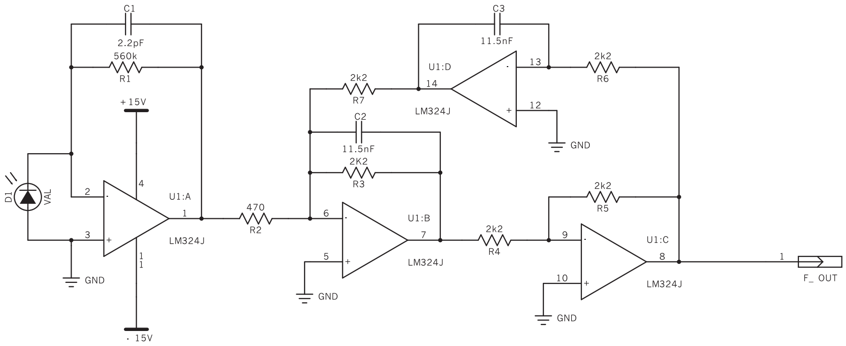


Figura 12: Circuito receptor de infra-vermelhos (amplificador de corrente e filtro)

$$PR2 = \frac{1}{f}10^6 - 1$$

ou seja:

$$PR2 = 161$$

ficando assim a frequência ajustada para:

$$f_{emiss} = \frac{10^6}{PR2+1} = 6172,8 \text{ Hz}$$

Observando o gráfico da Figura 13, verifica-se que $f_{emiss}$ está muito próxima da frequência natural do filtro, pelo que este cumpre a sua função de passa-banda.

O amplificador condicionador de sinal (ver Figura 14) tem como função rectificar o sinal recebido pelo filtro e amplificá-lo para que este se encontre na gama de valores aceite pelo conversor A/D interno do PIC.

O primeiro amplificador operacional é um ganho simples (inversor), a que se segue um rectificador de onda completa activo, que elimina a zona morta introduzida pelos díodos, o que é extremamente importante em face da baixa tensão dos sinais aqui encontrados. Segue-se um integrador, com controlo

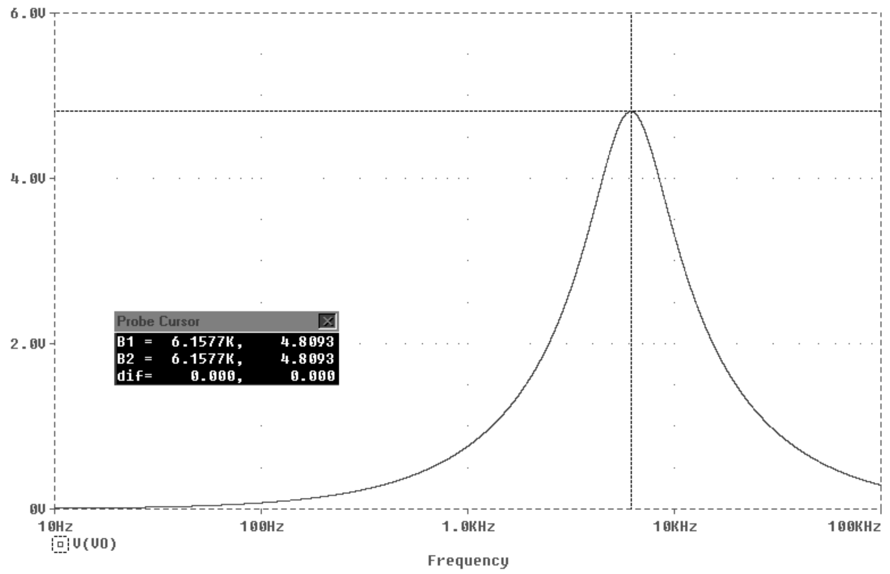


Figura 13: Resposta do filtro *biquad* (simulação)

de *offset*, destinado a condicionar o sinal que, depois do seguidor, ataca o amplificador logarítmico. O circuito seguidor foi introduzido pelo facto de se ter um amplificador operacional livre no integrado LM324 e se garantir assim um melhor isolamento entre os dois circuitos. O circuito amplificador logarítmico é extremamente importante para que se possam detectar correctamente os máximos de reflexão quando o sinal é fraco (aumenta o ganho para valores baixos do sinal de entrada). Este amplificador é muito sensível, pelo que é extremamente importante que o sinal à sua entrada esteja numa gama relativamente estreita de valores. É por essa razão que foi introduzido o potenciómetro de ajuste de *offset*.

À saída está um último amplificador inversor. É inversor porque o amplificador logarítmico também inverte o sinal. É importante o último controlo de ganho e *offset* para que o sinal recebido esteja a ocupar o melhor possível a gama de entrada do conversor A/D (0 a 5V).

### 3.3.3 *Driver* do motor

O *driver* do motor passo-a-passo é constituído por um controlador e por um amplificador de potência (ver Figura 15). O controlador é responsável por gerar os sinais de excitação dos enrolamentos do motor a partir dos sinais digitais que indicam direcção, modo (meio passo ou passo completo) e um sinal de relógio. É implementado pelo circuito integrado L297 da SGS-Thompson. Ao amplificador de potência cabe a tarefa de comutar a corrente em cada um dos dois enrolamentos do motor. É implementado pelo circuito integrado L298 também da SGS-Thompson que comuta as correntes através de duas pontes em H. Existem ainda díodos de roda livre para protecção das pontes e uma resistência em cada enrolamento para que o controlador tenha uma amostra das correntes no motor, cujos limites podem ser definidos por um potenciómetro de ajuste.

O *driver* do motor e o amplificador de sinal encontram-se montados na mesma placa de circuito impresso. A utilização de uma placa comum faz sentido dado que tem que existir um *driver* e um amplificador de sinal para cada sensor, sendo esta a solução mais económica.

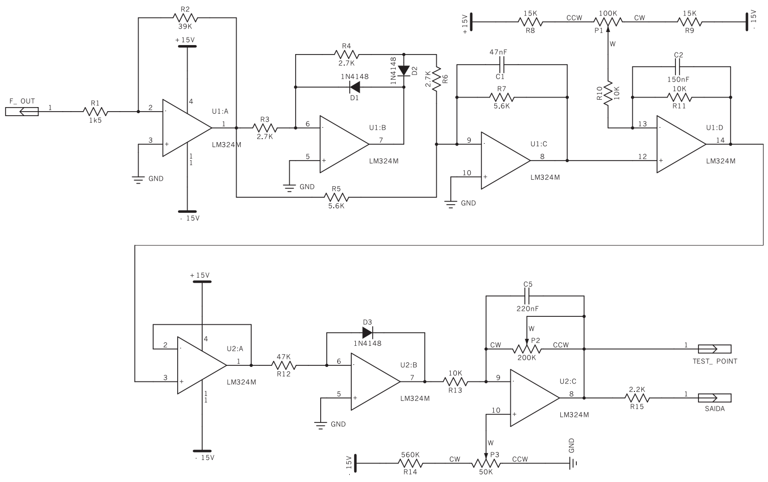


Figura 14: Circuito do amplificador condicionador de sinal

### 3.3.4 Electrónica de comunicações

A electrónica de comunicações é extremamente simples. Consiste numa ficha D9 fêmea, com o pino 2 para transmitir e o 3 para receber, sendo o pino 5 de massa. Utiliza-se um circuito integrado MAX232 (ver Figura 16) para converter os níveis de tensão da norma RS232 em níveis TTL (0 a 5 volt) utilizados pelo PIC. Esta electrónica encontra-se montada na mesma placa que a electrónica do PIC, visto existir apenas uma para todo o sistema de sensores.

### 3.3.5 Electrónica do PIC

Nesta sub-secção é descrita a electrónica que rodeia o microcontrolador PIC, que é constituída por sub-circuitos necessários ao funcionamento do microcontrolador ou à *interface* com os sensores (ver Figura 16).

O circuito oscilador e o circuito de *reset* são necessários ao funcionamento do PIC. O oscilador é composto pelo cristal X1 e pelos condensadores C3 e C4 e tem como função gerar o sinal de relógio para o microcontrolador. O circuito de *reset* está construído em torno do transístor BC558 e activa o pino MCLR (*master clear*) do PIC sempre que se pressiona a tecla S1 ou a tensão de alimentação está abaixo do nível mínimo (protecção *brown-out*). O circuito integrado 74LS08 (quatro portas lógicas *and*) permite ao microcontrolador controlar quais os emissores de infra-vermelhos que estão activos, fornecendo-lhes, neste caso, o sinal PWM.

Esta electrónica, em conjunto com o módulo de comunicações, está montada na mesma placa que contém o PIC. Foi utilizada uma placa de protótipo, visto não se justificar o projecto e fabrico de uma placa de circuito impresso para apenas uma unidade.

Como se pode ver na Figura 16, toda a electrónica, assim como o *software* do microcontrolador, estão preparados para controlar quatro conjuntos motor-sensor.

### 3.3.6 Placas

Apresentam-se, a título ilustrativo, as placas de circuito impresso desenhadas, vistas do lado dos componentes.

Como já foi referido, numa placa encontram-se o circuito emissor, o LED receptor e o filtro (Figura 17). Na outra placa encontram-se o amplificador de sinal, para atacar a entrada A/D do

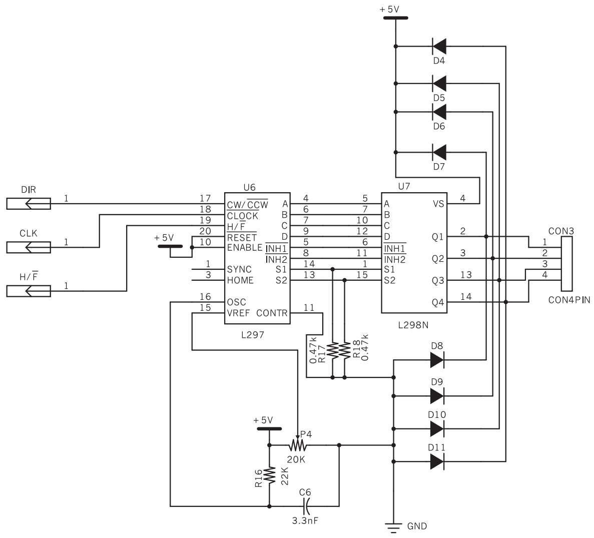


Figura 15: Circuito do *driver* do motor

PIC, e o *driver* do motor passo-a-passo (Figura 18). Há, portanto, duas placas de cada, visto haver dois sensores.

Não se apresenta a placa onde está montada a electrónica do PIC, visto ter sido utilizada uma placa de protótipo.

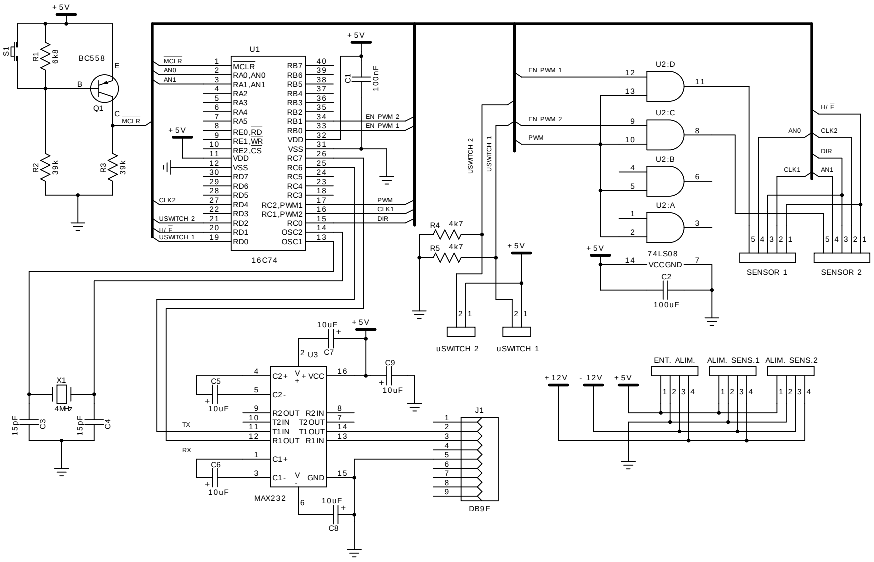

Figura 16: Circuito do microcontrolador PIC

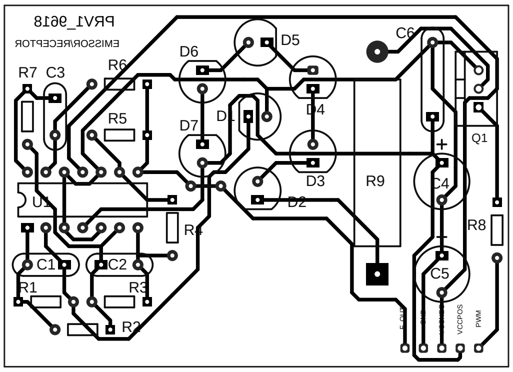

Figura 17: Placa de circuito impresso do emissor, receptor e filtro

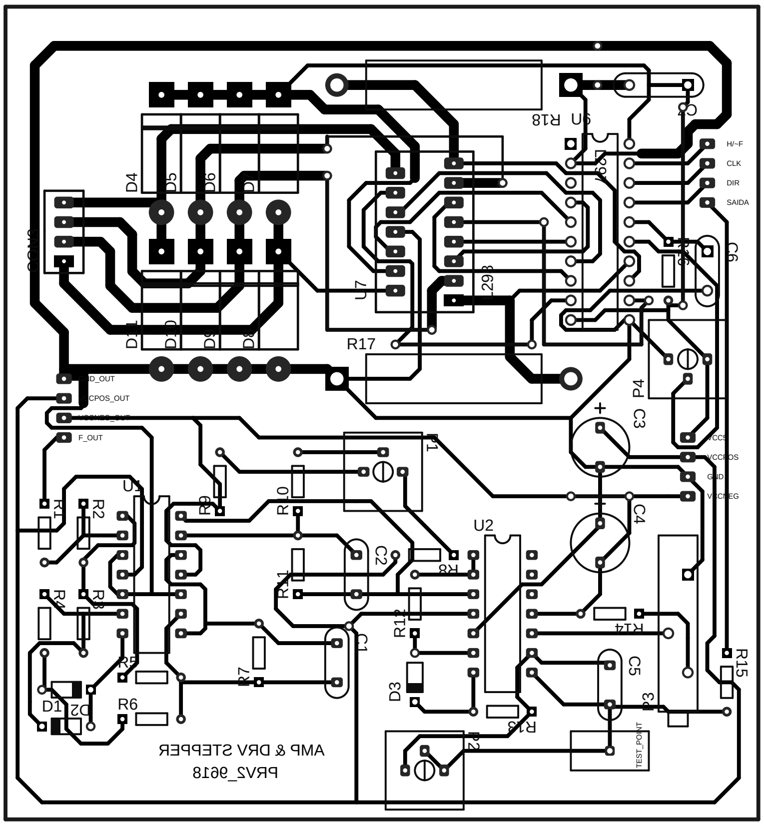


Figura 18: Placa de circuito impresso do amplificador e do *driver* do motor

## 3.4 Processamento de dados

Nesta secção apresentam-se os algoritmos utilizados no processamento dos dados obtidos a partir do sistema sensorial. Dos valores devolvidos pelo sistema sensorial, é necessário extrair a localização dos máximos de reflexão (candidatos à posição dos reflectores), determinar quais os ângulos a que correspondem e com essa informação determinar a posição e orientação da Robuter, relativamente a um referencial associado aos reflectores. É também importante a calibração do sistema sensorial, sujeito a erros de montagem.

### 3.4.1 Detecção de máximos

A cada comando **scan**, o sistema sensorial devolve um inteiro por cada passo percorrido, correspondente à amplitude do sinal recebido, constituindo-se assim um vector de dados para cada sensor, com todos os valores recebidos. De modo a determinar os ângulos a1, a2, b1 e b2 (ver Figura 5), é necessário encontrar os 2 máximos locais de reflexão no vector de valores recebidos que correspondem aos 2 reflectores de referência. Depois de encontrados os máximos, utiliza-se o procedimento explicado na Secção 3.4.3 para calcular os ângulos respectivos.

Os máximos locais correspondentes aos reflectores têm características próprias que os permitem distinguir de outros máximos locais devidos a ruído, superfícies reflectoras indesejáveis, janelas ou lâmpadas de grande intensidade luminosa, por exemplo. De facto, os máximos que interessa detectar são pouco largos, devido ao formato dos reflectores, e têm grande derivada em torno do máximo, devido à grande diferença de reflectividade entre os reflectores utilizados e o material de fundo. Na Figura 19 apresenta-se um exemplo de sinal obtido experimentalmente em que são claros os máximos locais devidos aos reflectores e outros devidos a ruído.

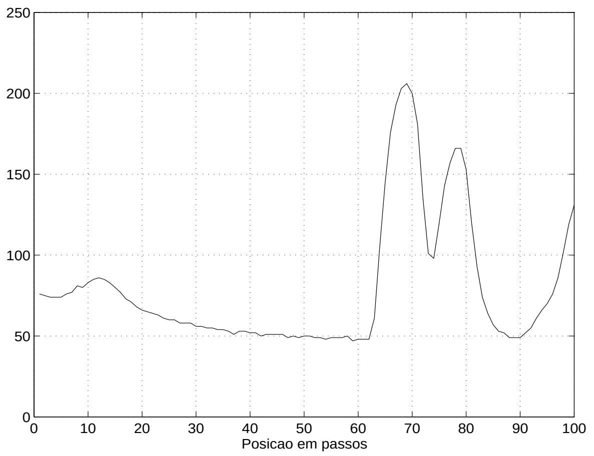


Figura 19: Sinal típico devolvido após um comando **scan**

Para a detecção deste tipo de máximos, passa-se o vector com os valores recebidos por um filtro derivativo, neste caso,

$$y = [1, 1, 0, -1, -1] * x$$

ou seja:

$$y[i] = -x[i-2] - x[i-1] + x[i+1] + x[i+2]$$

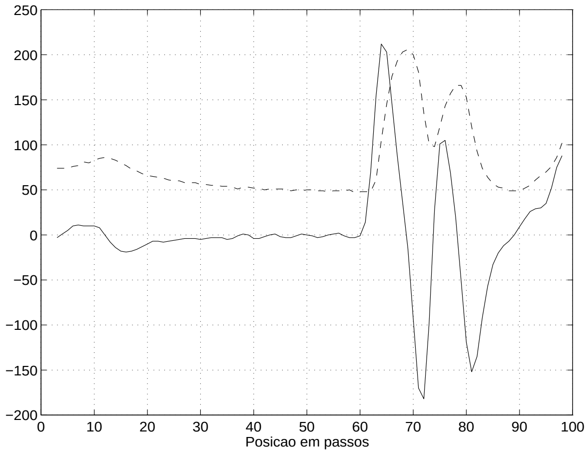


Figura 20: Sinal recebido (- -) e sinal filtrado (—)

Tabela 3: Posições, valores e pontuações dos máximos num sinal típico

| **Posição [passos]** | **Valor** | **Pontuação** |
|---|---|---|
| 69.7115 | 206 | 310 |
| 78.2754 | 166 | 304 |
| 12 | 86 | 102 |
| 39 | 53 | 58 |
| 44.3333 | 51 | 57 |
| 56.6666 | 49 | 55 |
| 50 | 50 | 52 |

Dado que este filtro introduz uma diferenciação, os máximos do vector **x**, são transformados em passagens por zero com derivada negativa em **y**. Para a detecção dos máximos em **x** interpola-se linearmente o vector **y** entre cada dois pontos, de modo a determinar o melhor possível a localização de cada uma das passagens por zero (ver Figura 20).

Concluiu-se, por inspecção, que às derivadas de maior valor absoluto de **y** em torno dos seus zeros correspondem os máximos mais conformes aos procurados.

A cada máximo local encontrado atribui-se uma pontuação, destinada a quantificar a qualidade do máximo, de acordo com o critério que se explica de seguida.

Chamando **p** à posição do zero no vector **y**, existem duas situações possíveis.

No caso de **p** estar entre dois valores discretos do vector **y**, ou seja:

$$\begin{aligned} i-1 &< p \\ p &< i \\ y[i-1] &> 0 \\ y[i] &< 0 \end{aligned}$$

a pontuação é calculada segundo:

$$pont = 2 \times (y[i-1] - y[i]) + x[i]$$

No caso de **p** coincidir exactamente com uma das posições discretas do vector **y**, ou seja:

$$p = i$$
$$y[i] = 0$$

calcula-se a pontuação utilizando a expressão:

$$pont = y[i-1] - y[i+1] + x[i]$$

Dado que se faz a interpolação do vector recebido, a localização de cada máximo toma valores reais.

Vêem-se na Tabela 3 os dados relativos às Figuras 19 e 20.

Os máximos locais encontrados são colocados num vector, ordenados por ordem decrescente da pontuação assim calculada, dado que a maior pontuação corresponde maior probabilidade de o máximo ter sido originado por um reflector. Estes máximos são utilizados para a o cálculo dos ângulos como se verá na Secção 3.4.3.

### 3.4.2 Calibração do sistema

Como foi introduzido na Secção 3.2.4, o comando **seek** leva o sensor até à posição zero, de referência, alinhada com o eixo **x** da Robuter (ver Figura 22).

No entanto, a montagem mecânica do microinterruptor e do parafuso que o deve activar está sujeita a erros, assim como os LEDs emissores e receptor podem não estar perfeitamente alinhados. Deste facto decorrem erros graves na determinação da posição e orientação da Robuter. Surgiu portanto a necessidade de se fazer uma calibração, ou correcção, do sistema. Os cálculos que seguem mostram como calcular os erros **a** e **b**, que afectam a medida dos ângulos (ver Figura 21).

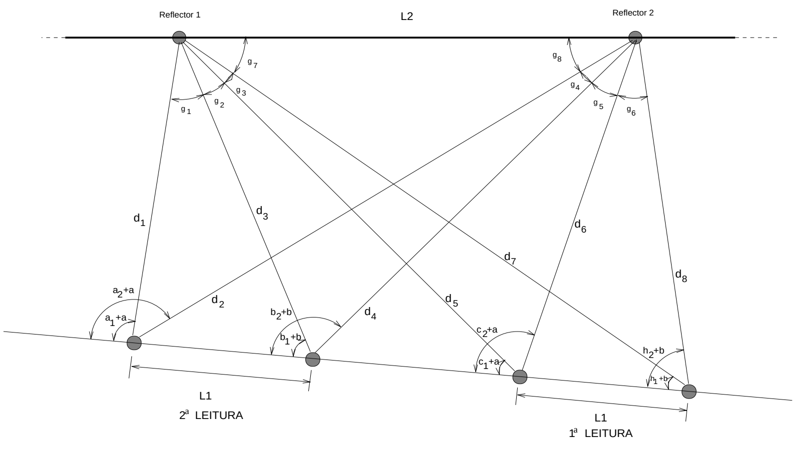


Figura 21: Denominação dos ângulos e distâncias para a calibração

Para o procedimento de calibração começa-se por fazer duas medidas de ângulos, fazendo com que, entre as duas medidas, a Robuter ande, em linha recta e não necessariamente paralela à estação de acostagem, uma distância maior do que a distância entre sensores, de tal modo que se possam medir os ângulos $a_1$, $a_2$, $b_1$, $b_2$, $c_1$, $c_2$, $h_1$ e $h_2$, em que $c_1$, $c_2$, $h_1$ e $h_2$ correspondem a $a_1$, $a_2$, $b_1$ e $b_2$, respectivamente. Para melhor precisão, cada uma das medidas corresponde à média de seis leituras na mesma posição.

Das equações da soma dos ângulos internos de alguns triângulos da Figura 21 obtém-se:

$$\begin{aligned}
g_1 + \pi - a_1 - a + b_1 + b &= \pi && (1)\\
g_2 + \pi - b_2 - b + c_1 + a &= \pi && (2)\\
g_3 + \pi - c_1 - a + h_1 + b &= \pi && (3)\\
g_4 + \pi - a_2 - a + b_2 + b &= \pi && (4)\\
g_5 + \pi - b_2 - b + c_2 + a &= \pi && (5)\\
g_6 + \pi - c_2 - a + h_2 + b &= \pi && (6)
\end{aligned}$$

de onde, por simplificação, vem:

$$\begin{aligned}
g_1 &= a_1 - b_1 + a - b && (7)\\
g_2 &= b_1 - c_1 + b - a && (8)\\
g_3 &= c_1 - h_1 + a - b && (9)\\
g_4 &= a_2 - b_2 + a - b && (10)\\
g_5 &= b_2 - c_2 + b - a && (11)\\
g_6 &= c_2 - h_2 + a - b && (12)
\end{aligned}$$

Da soma dos ângulos internos do triângulo cujos lados são $d_3$, $d_4$ e $L2$ vem:

$$g8 + g4 + b2 + b - b1 - b + g2 + g3 + g7 = \pi,$$

pelo que utilizando (8), (9) e (10) e simplificando, se obtém

$$g_7 = \pi + h_1 - a_2 + b - a - g_8 \tag{13}$$

Utilizando a regra dos senos para triângulos (ver Figura 21), obtêm-se as equações

$$\frac{\sin(g_1)}{L1} = \frac{\sin(\pi - a_1 - a)}{d_3} = \frac{\sin(b_1 + b)}{d_1} \tag{14}$$

$$\frac{\sin(\pi - b_1 - b)}{d_5} = \frac{\sin(c_1 + a)}{d_3} \tag{15}$$

$$\frac{\sin(g_3)}{L1} = \frac{\sin(\pi - c_1 - a)}{d_7} = \frac{\sin(h_1 + b)}{d_5} \tag{16}$$

$$\frac{\sin(g_4)}{L1} = \frac{\sin(\pi - a_2 - a)}{d_4} = \frac{\sin(b_2 + b)}{d_2} \tag{17}$$

$$\frac{\sin(\pi - b_2 - b)}{d_6} = \frac{\sin(c_2 + a)}{d_4} \tag{18}$$

$$\frac{\sin(g_6)}{L1} = \frac{\sin(\pi - c_2 - a)}{d_8} = \frac{\sin(h_2 + b)}{d_6} \tag{19}$$

$$\frac{\sin(a_2 - a_1)}{L2} = \frac{\sin(g_1 + g_2 + g_3 + g_7)}{d_2} = \frac{\sin(g_8)}{d_1} \tag{20}$$

$$\frac{\sin(b_2 - b_1)}{L2} = \frac{\sin(g_2 + g_3 + g_7)}{d_4} = \frac{\sin(g_4 + g_8)}{d_3} \tag{21}$$

$$\frac{\sin(c_2 - c_1)}{L2} = \frac{\sin(g_3 + g_7)}{d_6} = \frac{\sin(g_4 + g_5 + g_8)}{d_5} \tag{22}$$

$$\frac{\sin(h_2 - h_1)}{L2} = \frac{\sin(g_7)}{d_8} = \frac{\sin(g_4 + g_5 + g_6 + g_8)}{d_7} \tag{23}$$

De seguida, determinam-se as distâncias $d_i$, utilizando as equações (20) a (23):

$$d_1 = \frac{\sin(g_8)}{\sin(a_2 - a_1)} L2 \tag{24}$$

$$d_2 = \frac{\sin(g_1 + g_2 + g_3 + g_7)}{\sin(a_2 - a_1)} L2 \tag{25}$$

$$d_3 = \frac{\sin(g_8 + g_4)}{\sin(b_2 - b_1)} L2 \tag{26}$$

$$d_4 = \frac{\sin(g_2 + g_3 + g_7)}{\sin(b_2 - b_1)} L2 \tag{27}$$

$$d_5 = \frac{\sin(g_4 + g_5 + g_8)}{\sin(c_2 - c_1)} L2 \tag{28}$$

$$d_6 = \frac{\sin(g_3 + g_7)}{\sin(c_2 - c_1)} L2 \tag{29}$$

$$d_7 = \frac{\sin(g_4 + g_5 + g_6 + g_8)}{\sin(h_2 - h_1)} L2 \tag{30}$$

$$d_8 = \frac{\sin(g_7)}{\sin(h_2 - h_1)} L2 \tag{31}$$

Substituindo agora de $g_1$ a $g_7$ nas distâncias $d_i$, obtém-se

$$d_1 = \frac{\sin(g_8)}{\sin(a_2 - a_1)} L2 \tag{32}$$

$$d_2 = \frac{\sin(\pi + a_1 - a_2 - g_8)}{\sin(a_2 - a_1)} L2 \tag{33}$$

$$d_3 = \frac{\sin(a_2 - b_2 + a - b + g_8)}{\sin(b_2 - b_1)} L2 \tag{34}$$

$$d_4 = \frac{\sin(\pi - a_2 + b_1 - a + b - g_8)}{\sin(b_2 - b_1)} L2 \tag{35}$$

$$d_5 = \frac{\sin(a_2 - c_2 + g_8)}{\sin(c_2 - c_1)} L2 \tag{36}$$

$$d_6 = \frac{\sin(\pi - a_2 + c_1 - g_8)}{\sin(c_2 - c_1)} L2 \tag{37}$$

$$d_7 = \frac{\sin(a_2 - h_2 + a - b + g_8)}{\sin(h_2 - h_1)} L2 \tag{38}$$

$$d_8 = \frac{\sin(\pi + h_1 - a_2 - a + b - g_8)}{\sin(h_2 - h_1)} L2 \tag{39}$$

Das equações (14), (16) e (17), substituindo as distâncias $d_i$ pelas expressões dadas por (32) a (39) resultam

$$
\begin{aligned}
\sin(g_8) &= \frac{\sin(b_1+b)\sin(a_2-a_1)}{\sin(a_1-b_1+a-b)}\frac{L1}{L2} && (40)\\
\sin(c_1-h_1+a-b) &= \frac{\sin(h_1+b)\sin(c_2-c_1)}{\sin(a_2-c_2+g_8)}\frac{L1}{L2} && (41)\\
\sin(a_2-b_2+a-b) &= \frac{\sin(b_2+b)\sin(a_2-a_1)}{\sin(\pi+a_1-a_2-g_8)}\frac{L1}{L2} && (42)
\end{aligned}
$$

Resolvendo agora (40) em ordem a $g_8$, e substituindo em (41) e (42), chega-se ao seguinte sistema, em ordem a **a** e **b**:

$$
\begin{cases}
\sin(c_1-h_1+a-b) = \dfrac{\sin(d_1+b)\sin(c_2-c_1)}{\sin\left\{a_2-c_2+\arcsin\left[\frac{\sin(a_2-a_1)\sin(b+b_1)}{\sin(a_1-b_1+a-b)}\frac{L1}{L2}\right]\right\}}\dfrac{L1}{L2} \\[2ex]
\sin(a_2-b_2+a-b) = \dfrac{\sin(b_2+b)\sin(a_2-a_1)}{\sin\left\{\pi+a_1-a_2-\arcsin\left[\frac{\sin(a_2-a_1)\sin(b+b_1)}{\sin(a_1-b_1+a-b)}\frac{L1}{L2}\right]\right\}}\dfrac{L1}{L2}
\end{cases}
$$

Dado que este sistema é transcendente, optou-se por uma resolução numérica, com base no método de Newton-Raphson [6]. Para tal, escreve-se o sistema na forma:

$$
\begin{cases}
f_1(a,b) = 0 \\
f_2(a,b) = 0
\end{cases}
$$

e determina-se o Jacobiano

$$
J = \begin{bmatrix} \frac{\partial f_1(a,b)}{\partial a} & \frac{\partial f_1(a,b)}{\partial b} \\[1ex] \frac{\partial f_2(a,b)}{\partial a} & \frac{\partial f_2(a,b)}{\partial b} \end{bmatrix}
$$

após o que se inicia o processo iterativo que conduz à solução, em que as variáveis são actualizadas de acordo com:

$$
\begin{cases}
a_{i+1} = a - \left.\dfrac{f_1(a,b)\cdot\frac{\partial f_2(a,b)}{\partial b} - f_2(a,b)\cdot\frac{\partial f_1(a,b)}{\partial b}}{|J|}\right|_{\substack{a=a_i\\ b=b_i}} \\[3ex]
b_{i+1} = b + \left.\dfrac{f_1(a,b)\cdot\frac{\partial f_2(a,b)}{\partial a} - f_2(a,b)\cdot\frac{\partial f_1(a,b)}{\partial a}}{|J|}\right|_{\substack{a=a_i\\ b=b_i}}
\end{cases}
$$

Quando ambas as variáveis satisfazem a condição terminadora, ou seja, a variação de uma iteração para seguinte, em ambas as variáveis, é menor do que um determinado valor relativo, dá-se o procedimento por concluído.

$$
\begin{cases}
\left|\frac{a_i-a_{i-1}}{a_i}\right| < \epsilon \\[2ex]
\left|\frac{b_i-b_{i-1}}{b_i}\right| < \epsilon
\end{cases}
$$

Para este algoritmo é evidentemente necessário fornecer os valores iniciais $a_0$ e $b_0$, que neste caso são sempre inicializados a 0, dado que o erro é sempre baixo.

Para este problema, com estas condições iniciais, e com um limiar para a condição terminadora $\epsilon = 10^{-6}$, cerca de 20 iterações são suficientes para que o algoritmo estabilize.

O algoritmo aqui apresentado tem uma limitação grave. Considerou-se que todas as medidas são afectada de um erro igual, para cada sensor, o que não é exacto, dado que existe sempre alguma imprecisão. Assim, ao calcular **a** e **b** utilizando apenas as equações (14), (16) e (17), para a formulação do sistema final, só estas são satisfeitas, ficando as outras equações por resolver.

Não se resolve minimizando o erro quadrático definido pelas equações (14) a (23), incluindo as (14), (16) e (17), porque, de qualquer modo, os erros que podem existir nas duas medidas podem levar a uma solução pouco correcta.

Assim, se bem que o procedimento seja válido (desde que as medidas sejam muito precisas), considerou-se um outro método que permite uma calibração eficaz, como se passa a descrever.

Fazem-se **n** medidas dos ângulos $a1_{med}^{i}$, $a2_{med}^{i}$, $b1_{med}^{i}$, $b2_{med}^{i}$, e em várias posições diferentes $(i = 1..n)$. O facto de se fazerem diversas medidas permite que, em média, os erros sejam menores do que no algoritmo anterior (apenas duas medidas).

De seguida, para um determinado **a** e **b** de correcção dos ângulos, calcula-se $L2_{est}^{i}$ segundo

$$L2_{est}^{i}(a,b) = \frac{\frac{\sin\left(a_2^i - a_1^i\right) d_1}{\sin(g_3)} + \frac{\sin\left(b_2^i - b_1^i\right) d_4}{\sin(g_2)}}{2} \tag{43}$$

onde as variáveis $g_2$, $g_3$, $d_1$ e $d_4$ são dadas pelas equações (45) a (52), fazendo:

$$\left\{\begin{array}{lcl} a_1^i & = & a1_{med}^{i} + a \\ a_2^i & = & a2_{med}^{i} + a \\ b_1^i & = & b1_{med}^{i} + b \\ b_2^i & = & b2_{med}^{i} + b \end{array}\right.$$

Define-se então uma função de erro $f_{erro}$ dada por

$$f_{erro}(a,b) = \sum_{i=1}^{n} \left(L2_{est}^{i}(a,b) - L2\right)^2 \tag{44}$$

em que L2 é medido directamente, com fita métrica, na estação de acostagem (distância entre reflectores).

Esta função de erro, à falta de um sistema sensorial com maior exactidão do que o aqui descrito, fornece-nos uma estimativa do erro de localização e orientação do sistema, quando utilizado com diversas medidas independentes.

Procura-se de seguida minimizar esta função de erro, partindo das condições iniciais $a = 0°$ e $b = 0°$, pelo método do gradiente, utilizando passo adaptativo [4].

Dos resultados obtidos em várias ocasiões, conclui-se que os valores absolutos dos erros **a** e **b** são sempre menores do que 5°.

Os valores assim calculados são utilizados aquando de cada leitura para a correcção dos ângulos.

### 3.4.3 Determinação dos ângulos

A determinação dos ângulos $a_1$, $a_2$, $b_1$ e $b_2$ (ver Figura 5) poderia ser feita directamente a partir da localização dos máximos determinados como se explica na Secção 3.4.1, através de uma simples mudança de coordenadas.

No entanto, ocorrem dois problemas de ordem diferente:

i) o primeiro, é o facto de haver sempre algum atraso na electrónica ao fazer uma leitura (**scan**), pelo que é necessário fazer sempre duas leituras, em direcções opostas, para que a média da localização de dois máximos corresponda efectivamente à localização correcta;

ii) o segundo problema decorre do facto de acontecer, por vezes, que um sinal com ruído apresente máximos que superam em pontuação os máximos que de facto correspondem a reflectores, o que leva a erros na determinação dos ângulos.

Para tentar resolver este segundo problema, procuram-se mais do que dois máximos nos vectores recebidos, porque pode acontecer que um máximo que de facto corresponde a um reflector tenha apenas a terceira melhor pontuação. Verificou-se que procurar mais do que três máximos é quase sempre inútil, pois, para além do terceiro, os máximos já apresentam muito pouca definição.

Por esta razão, determina-se sempre a localização, nos dois vectores recebidos, cada um no seu sentido de leitura ($x_{dir}$ para o sentido directo e $x_{inv}$ para o sentido inverso), dos três máximos com melhor pontuação. Faz-se a inversão do vector $x_{inv}$, para que a posições iguais nos dois vectores correspondam direcções iguais no espaço percorrido pelo sensor, dado que as medidas são feitas com a Robuter imobilizada. O vector $x_{inv}$ denomina o vector já invertido.

**Definição (máximos emparelhados)**: diz-se que 2 máximos (um pertencente a $x_{dir}$ e outro pertencente a $x_{inv}$) estão "emparelhados" se e só se:

i) os valores respectivos distam menos de 10 passos.

ii) a posição do máximo no vector $x_{inv}$ é menor do que a posição do máximo correspondente no vector $x_{dir}$, dado haver sempre atraso na electrónica (nunca avanço), e não distam mais do que 10°, ou seja:

$$0^\circ \leq x_{dir} - x_{inv} < 10^\circ$$

Designe-se por $a_1$, $a_2$, $b_1$ e $b_2$ os ângulos a determinar (ver Figura 23). Sabe-se que para que um máximo corresponda a um destes ângulos é necessário que apareça emparelhado. A média dos dois máximos emparelhados conduz ao valor do ângulo:

$$a_1 = \frac{a_{1_{dir}} + a_{1_{inv}}}{2}$$

Assim, tentam-se emparelhar os três máximos mais pontuados recebidos numa das direcções com os três máximos mais pontuados recebidos na outra direcção, tanto para o sensor 1 (ângulos $a_1$, $a_2$) como para o sensor 2 (ângulos $b_1$, $b_2$), procurando as várias combinações possíveis. Evidentemente, o procedimento é separado para os vectores $x_{dir}$ e $x_{inv}$ referentes ao sensor 1 e para os referentes ao sensor 2. Os máximos assim emparelhados constituem os ângulos desejados ($a_1$, $a_2$, $b_1$ e $b_2$).

Explicam-se de seguida os procedimentos adoptados no caso de faltarem emparelhar mais de um ângulo, um ângulo ou nenhum ângulo dos quatro ângulos necessários para que se consiga, em algumas situações que se verão adiante, determinar a posição e localização da Robuter.

Como facilmente se verifica da geometria do sistema (ver Figura 23), podem-se estabelecer as seguintes relações entre os ângulos:

$$\begin{cases} a_1 < a_2 \\ b_1 < b_2 \\ a_1 > b_1 \\ a_2 > b_2 \end{cases}$$

que se revelarão de utilidade adiante.

**Quatro ângulos determinados**

É necessário de qualquer modo verificar a consistência dos valores determinados para os ângulos.

A primeira situação possível é a verificação, por parte dos ângulos, das relações estabelecidas. Não é necessário tomar nenhuma acção.

No caso de se ter $a_1 < b_1 < a_2 < b_2$, ou o $a_2$ determinado corresponde ao $a_1$ e o $a_1$ determinado não é válido, situação em se faz $a_1 = a_2$ e se considera o $a_2$ como indeterminado, ou então é o $b_2$ que não é válido, fazendo-se $b_2 = b_1$ e dando-se o $b_1$ como indeterminado.

Na eventualidade de $a_2 < b_1$, a situação é impossível, pelo que se considera a medida inválida.

Se $a_2 < b_2$, ou $a_2$ é incorrecto, ou $b_2$ é incorrecto, ou então $a_1$ é incorrecto, sendo o valor de $a_1$ substituído pelo valor de $a_2$.

Em qualquer dos casos precedentes, quando houver várias hipóteses, o ângulo que se considera inválido é aquele que apresenta menor média de pontuações nos vectores $x_{dir}$ e $x_{inv}$.

**Três ângulos determinados**

No caso de se terem emparelhado três máximos, ou seja, determinado três ângulos, é necessário identificar qual o ângulo que falta, uma vez que três ângulos permitem, em certas situações que se verão na secção seguinte, determinar a posição e orientação da Robuter.

Colocam-se então duas hipóteses:

H1: falta emparelhar um dos máximos correspondente a $a_1$ ou $a_2$

H2: falta emparelhar um dos máximos correspondente a $b_1$ ou $b_2$

**Hipótese H1**

Chame-se $a_i$ ao ângulo já determinado. É necessário saber se este corresponde a $a_1$, a $a_2$, ou ainda se é um valor inválido. Se se verificar $a_i \leq b_1$, $a_i$ não corresponde a nenhum dos ângulos, e considera-se a medida inválida. Caso contrário, considera-se $a_1 = a_i$, visto ser mais provável que se perca o $a_2$, dado que, próximo do ponto de acostagem, $d_2 > d_1$ (ver Figura 5), podendo portanto perder o reflector. A veracidade desta assumpção será verificada posteriormente, aquando do cálculo da posição e orientação, fazendo no entanto sentido, visto que o facto de não se ter perdido nem $b_1$ nem $b_2$, leva a supor que a Robuter está com a coordenada $x_0$ positiva, e logo $d_2 > d_1$.

**Hipótese H2**

Chamando-se $b_i$ ao ângulo determinado, uma dedução semelhante permite afirmar que a medida não é válida se $b_i \geq a_2$, e que $b_2 = b_i$ caso contrário.

**Dois ou menos ângulos determinados**

Se só se tiverem conseguido emparelhar dois ou menos pares de máximos dos quatro ângulos desejados, dá-se a medida como inválida.

### 3.4.4 Determinação da posição e orientação da Robuter

Começa-se por apresentar os cálculos simples para a determinação da posição e orientação (ver Figura 22), dadas por $x_0$, $y_0$ e $t_0$, conhecidos os ângulos $a_1$, $a_2$, $b_1$ e $b_2$, e ainda a distância **L1** entre os sensores (ver Figura 23). De seguida, apresentam-se cálculos análogos para as situações em que falta determinar um dos ângulos, quando tal for possível.

**Cálculo da localização com os quatro ângulos conhecidos**

Começam-se por calcular as grandezas $g_i$ e $d_i$, de acordo com as seguintes expressões:

$$g_1 = a_1 - b_1 \tag{45}$$

$$g_4 = a_2 - b_2 \tag{46}$$

$$d_1 = L1 \frac{\sin(b_1)}{\sin(g_1)} \tag{47}$$

$$d_2 = L1 \frac{\sin(b_2)}{\sin(g_4)} \tag{48}$$

$$d_3 = L1 \frac{\sin(a_1)}{\sin(g_1)} \tag{49}$$

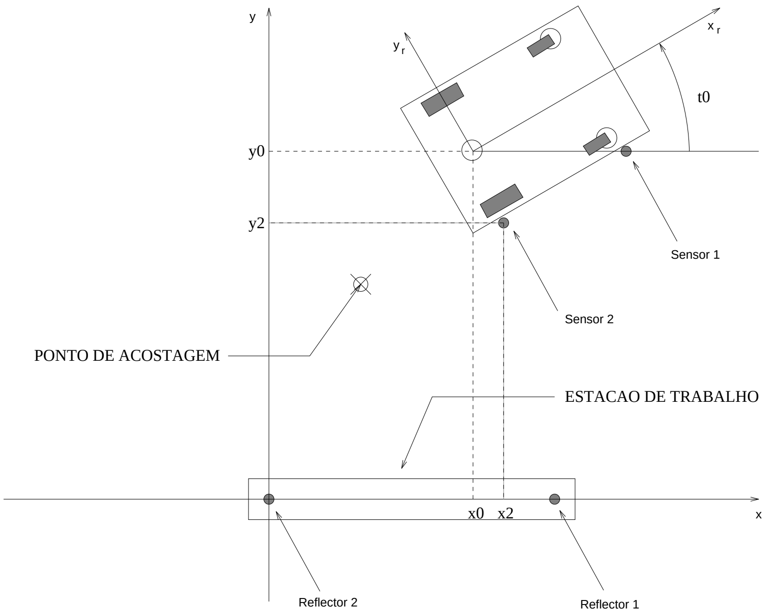


Figura 22: Robuter e estação de trabalho

$$\begin{aligned} d_4 &= L1\frac{\sin(a_2)}{\sin(g_4)} && (50)\\ g_2 &= \arctan\left(\frac{\sin(b_2-b_1)}{d_3/d_4-\cos(b_2-b_1)}\right) && (51)\\ g_3 &= \arctan\left(\frac{\sin(a_2-a_1)}{d_2/d_1-\cos(a_2-a_1)}\right) && (52)\end{aligned}$$

A partir destes valores chega-se à posição do sensor 2 em relação ao reflector 2, que se considera como a origem do referencial utilizado, através do seguinte conjunto de expressões:

$$\left\{\begin{array}{lcl} x_2 &=& d_4\cos(g_3+g_4)\\ y_2 &=& d_4\sin(g_3+g_4)\end{array}\right. \qquad (53)$$

que por sua vez permitem calcular a posição e orientação da Robuter em relação ao mesmo referencial, utilizando a seguinte transformação de coordenadas:

$$\left\{\begin{array}{lcl} t_0 &=& a_1-g_1-g_2\\ x_0 &=& -s_{2_x}\cos(t_0)+s_{2_y}\sin(t_0)+x_2\\ y_0 &=& -s_{2_x}\sin(t_0)-s_{2_y}\cos(t_0)+y_2\end{array}\right. \qquad (54)$$

em que $\left(s_{2_x}, s_{2_y}\right)$ é a posição do Sensor 2 no referencial da Robuter. Neste caso, $\left(s_{2_x}, s_{2_y}\right) = (-8.2, -34.0)$.

**Cálculo sem $a_1$**

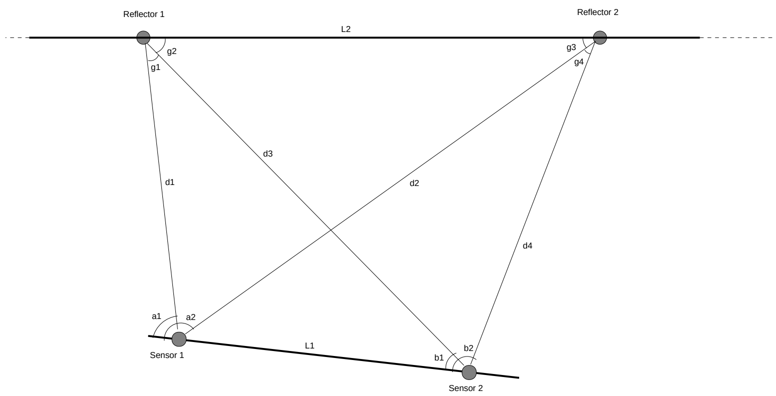


Figura 23: Triangulação

Para o cálculo da posição e orientação da Robuter sem o conhecimento de um dos ângulos, manteve-se a designação das variáveis adoptada para a situação em que se conhecem os quatro ângulos.

Apresentam-se as expressões para o cálculo de $g_i$ e de $d_i$, sendo os restantes cálculos em tudo semelhantes aos anteriormente apresentados. L1 representa ainda a distância entre sensores, enquanto que L2 representa a distância entre reflectores.

As expressões utilizadas para a localização, não sendo conhecido o ângulo $a_1$ são então:

$$\begin{aligned}
g_4 &= a_2 - b_2 \\
d_2 &= L1 \frac{\sin(b_2)}{\sin(g_4)} \\
d_4 &= L1 \frac{\sin(a_2)}{\sin(g_4)} \\
g_2 &= \arcsin\left(\frac{\sin(b_2 - b_1)\, d_4}{L2}\right) \\
g_3 &= \pi - g_2 - g_4 - b_2 + b_1 \\
d_3 &= d_4 \frac{\sin(g_3 + g_4)}{\sin(g_2)} \\
d_1 &= \sqrt{(d_2 \cos(g_3) - L2)^2 + d_2^2 \sin^2(g_3)} \\
a_1 &= \arcsin\left(\sin(b_1)\frac{d_3}{d_1}\right) \\
g_1 &= a_1 - b_1
\end{aligned}$$

A posição só é univocamente determinada se $d_4 \leq L2$, tal como se pode observar na Figura 24, sendo neste caso dada pelas expressões (53) e (54).

**Cálculo sem $a_2$**

Nos cálculos que se seguem, as variáveis $g_{ij}$ representam $g_i + g_j$. Calcula-se muitas vezes a soma, pois só esta faz sentido em face das variáveis conhecidas. Por exemplo, neste caso em que $a_2$ não é conhecido, só faz sentido falar de $g_{34} = g_3 + g_4$. No fim do cálculo, depois de determinar o ângulo $a_2$, já se podem calcular $g_3$ e $g_4$.

As expressões utilizadas são:

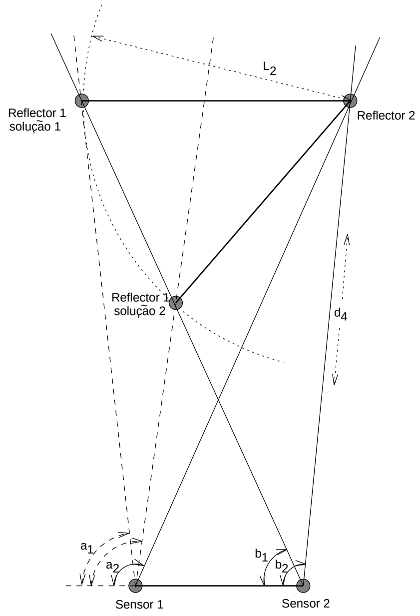


Figura 24: Situação em que $a_1$ não é univocamente determinável

$$
\begin{aligned}
g_1 &= a_1 - b_1 \\
d_1 &= L1 \frac{\sin(b_1)}{\sin(g_1)} \\
d_3 &= L1 \frac{\sin(a_1)}{\sin(g_1)} \\
g_{34} &= \arcsin\left(\frac{\sin(b_2 - b_1)\, d_3}{L2}\right) \\
g_2 &= \pi - g_{34} + b_1 - b_2 \\
d_4 &= L2 \frac{\sin(g_2)}{\sin(b_2 - b_1)} \\
d_2 &= \sqrt{(d_1 \cos(g_1 + g_2) - L2)^2 + d_1^2 \sin^2(g_1 + g_2)} \\
a_2 &= \arcsin\left(\sin(b_2) \frac{d_4}{d_2}\right) \\
g_4 &= a_2 - b_2 \\
g_3 &= g_{34} - g_4
\end{aligned}
$$

A posição só é univocamente determinada se $d_3 \leq L2$, sendo dada pelas expressões (53) e (54).
**Cálculo sem $b_1$**
Utilizam-se as seguintes expressões:

$$
g_4 = a_2 - b_2
$$

$$
\begin{aligned}
d_2 &= L1 \frac{\sin(b_2)}{\sin(g_4)} \\
d_4 &= L1 \frac{\sin(a_2)}{\sin(g_4)} \\
g_{12} &= \arcsin\left(\frac{\sin(a_2 - a_1)\, d_2}{L2}\right) \\
g_3 &= \pi + a_1 - a_2 - g_{12} \\
d_1 &= d_2 \frac{\sin(g_3)}{\sin(g_{12})} \\
d_3 &= \sqrt{(d_4 \cos(g_3 + g_4) - L2)^2 + d_4^2 \sin^2(g_3 + g_4)} \\
b_1 &= \arcsin\left(\sin(a_1) \frac{d_1}{d_3}\right) \\
g_1 &= a_1 - b_1 \\
g_2 &= g_{12} - g_1
\end{aligned}
$$

A posição só é univocamente determinada se $d_2 \leq L2$, sendo dada por (53) e (54).
**Cálculo sem $b_2$**
As expressões utilizadas são:

$$
\begin{aligned}
g_1 &= a_1 - b_1 \\
d_1 &= L1 \frac{\sin(b_1)}{\sin(g_1)} \\
d_3 &= L1 \frac{\sin(a_1)}{\sin(g_1)} \\
g_3 &= \arcsin\left(\frac{\sin(a_2 - a_1)\, d_1}{L2}\right) \\
g_2 &= \pi + a_1 - a_2 - g_1 - g_3 \\
d_2 &= L2 \frac{\sin(g_1 + g_2)}{\sin(a_2 - a_1)} \\
d_4 &= \sqrt{(d_3 \cos(g_2) - L2)^2 + d_3^2 \sin^2(g_2)} \\
b_2 &= \arcsin\left(\sin(a_2) \frac{d_2}{d_4}\right) \\
g_4 &= a_2 - b_2
\end{aligned}
$$

A posição só é univocamente determinada se $d_1 \leq L2$, sendo dada por (53) e (54).

## 3.5 Estudo estatístico

Para concluir o estudo feito sobre o sistema sensorial, apresenta-se nesta secção o estudo estatístico feito sobre um sensor isoladamente, seguido de um estudo sobre os resultados do sistema em funcionamento normal, ou seja, utilizando os dois sensores para se determinar a localização da Robuter.

### 3.5.1 Sensor isolado

O primeiro estudo feito procura caracterizar a amplitude do sinal recebido na zona dos máximos, estudando, para uma mesma localização do sensor, a sua variação em função do período de espera usado no **scan**.

Fizeram-se 30 medidas para cada período, mostrando-se a média dos resultados obtidos. Verificou-se que o desvio padrão não é significativo.

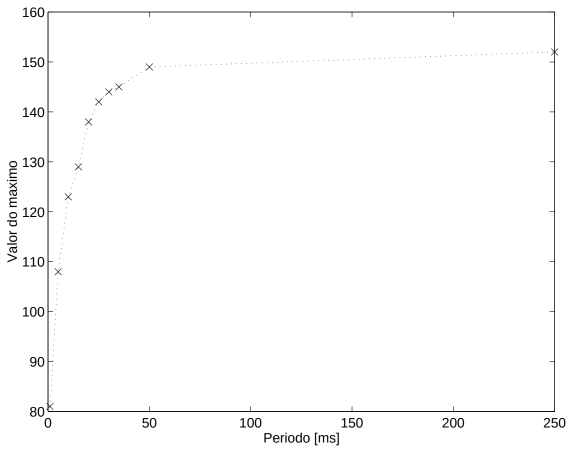


Figura 25: Amplitude do máximo em função do período

Como se pode verificar no gráfico da Figura 25, a amplitude do máximo, a partir de um período de 50 ms, já não melhora significativamente, notando-se também a queda acentuada para períodos de leitura abaixo dos 15 ms. Escolheu-se assim como período a utilizar em condições normais, o valor de 20 ms, que corresponde a um tempo de aquisição rápido, não implicando uma perda de alcance dos sensores significativa.

O teste seguinte, feito recolhendo 60 amostras para cada ponto (30 em cada direcção), estuda o atraso existente entre a posição detectada para o reflector e a posição real do mesmo (determinada fazendo a média das posições dos dois sentidos). Este atraso deve-se ao tempo de resposta da electrónica, nomeadamente do integrador. Depende por isso do período e também da distância (medida com fita métrica), pelo facto de se ter um sinal mais fraco na recepção (ver Figura 26).

Tabela 4: Valores em graus do atraso em função da distância e do período

| | **Distância [cm]** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Período [ms]** | **54** | **81** | **95** | **110** | **130** | **159** | **171** |
| **1** | 7.6849 | 7.3211 | 6.6901 | 6.3660 | 5.8595 | 5.1054 | 4.7781 |
| **10** | 4.7390 | 4.7544 | 4.3216 | 4.0517 | 3.7179 | 3.3490 | 3.1263 |
| **25** | 2.5280 | 2.6720 | 2.5035 | 2.3927 | 2.1552 | 1.7952 | 1.6304 |
| **50** | 1.0234 | 1.0989 | 1.1237 | 1.0964 | 1.0164 | 0.8096 | 0.7010 |

Pelo facto de este atraso ser dependente das duas variáveis (período e distância), e não se conhecer a distância quando se faz uma medida, não é possível corrigi-lo directamente.

Procurou-se então estudar o atraso sentido numa e noutra direcção. Assim, fizeram-se 30 pares de medidas, fazendo a média de cada par (uma medida num sentido e outra no outro). Apresentam-se os valores das médias na Tabela 5, podendo-se ver na Figura 27 e respectiva Tabela o desvio padrão para cada ponto distância-posição estudado. Note-se que a escala do eixo Z (atraso em °) é oito vezes menor do que a escala da Figura 26, verificando-se que o desvio padrão é insignificante.

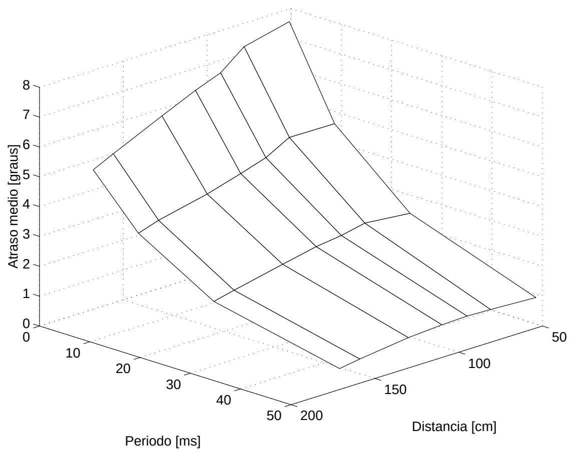


Figura 26: Atraso médio na leitura em função do período e da distância

Tabela 5: Valores das médias

| | Distância [cm] | | | | | | |
|---|---|---|---|---|---|---|---|
| **Período [ms]** | **54** | **81** | **95** | **110** | **130** | **159** | **171** |
| **1** | 16.1527 | 68.5472 | 38.2138 | 30.6908 | 57.2418 | 58.6378 | 50.5898 |
| **10** | 16.2838 | 68.4486 | 38.1995 | 30.7114 | 57.2695 | 58.6820 | 50.6163 |
| **25** | 16.2315 | 68.3020 | 38.1915 | 30.7305 | 57.2367 | 58.5932 | 50.5681 |
| **50** | 16.1844 | 68.1982 | 38.1720 | 30.6697 | 57.2671 | 58.5637 | 50.5679 |

Os resultados apresentados mostram a invariância da média das duas leituras, quer em relação ao período, quer em relação à distância, pelo que se considera uma boa estimativa para a localização efectiva do reflector.

### 3.5.2 Localização

Fizeram-se medidas dos ângulos em diversos locais (oito), escolhidos aleatoriamente, calculando-se as respectivas coordenadas da Robuter ($x_0$, $y_0$ e $t_0$). Fizeram-se 20 medidas em cada local, das quais se mostra a média e o desvio padrão das coordenadas calculadas (ver Tabela 7).

Os baixos valores de desvio padrão apresentados na Tabela 7 mostram a grande precisão de localização do sistema sensorial.

A grande variação do desvio padrão entre as diversas localizações deve-se ao facto de desvios iguais nos ângulos medidos $a_1$, $a_2$, $b_1$ e $b_2$ se reflectirem na determinação da localização da Robuter com gravidade que depende dessa mesma localização.

À falta de um sistema sensorial com exactidão conhecida, estudou-se a exactidão do sistema aqui

Tabela 6: Valores do gráfico do desvio padrão

| | Distância [cm] | | | | | | |
|---|---|---|---|---|---|---|---|
| **Período [ms]** | **54** | **81** | **95** | **110** | **130** | **159** | **171** |
| **1** | 0.0866 | 0.0845 | 0.0128 | 0.0145 | 0.0168 | 0.0221 | 0.0547 |
| **10** | 0.0216 | 0.1055 | 0.0090 | 0.0148 | 0.0110 | 0.0622 | 0.0302 |
| **25** | 0.0143 | 0.0733 | 0.0150 | 0.0154 | 0.0114 | 0.0132 | 0.0276 |
| **50** | 0.0123 | 0.0156 | 0.0159 | 0.0201 | 0.0135 | 0.0057 | 0.0165 |

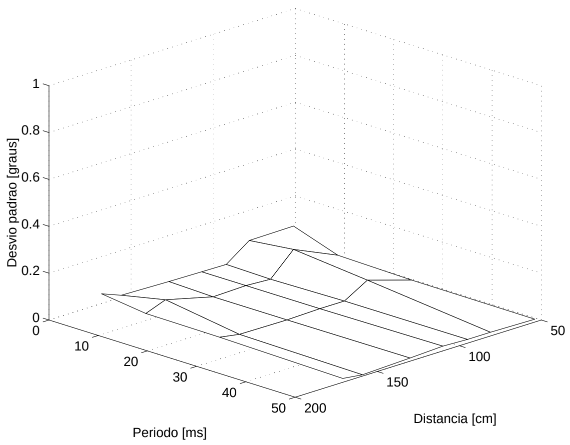


Figura 27: Desvio padrão

Tabela 7: Localização da Robuter

| | $t_0$ [°] | | $x_0$ [cm] | | $y_0$ [cm] | |
|---|---|---|---|---|---|---|
| **Local** | **Média** | **Desv. Pad.** | **Média** | **Desv. Pad.** | **Média** | **Desv. Pad.** |
| **1** | 2.1927 | 0.2038 | 58.0820 | 0.5205 | 120.2921 | 0.1613 |
| **2** | 5.4767 | 0.2252 | 0.9142 | 0.3843 | 97.9122 | 0.0692 |
| **3** | -5.6303 | 0.1774 | 48.4004 | 0.4435 | 108.6902 | 0.0354 |
| **4** | -17.3166 | 0.1084 | 39.4760 | 0.3395 | 166.6118 | 0.0534 |
| **5** | 24.7578 | 0.2210 | 9.6438 | 0.7686 | 171.6938 | 0.2851 |
| **6** | 15.6140 | 0.0883 | -22.1853 | 0.1850 | 118.0160 | 0.1027 |
| **7** | 17.6573 | 0.1153 | -29.3641 | 0.1298 | 84.8381 | 0.1558 |
| **8** | -0.3768 | 0.0678 | 30.4544 | 0.1397 | 83.7041 | 0.0998 |

apresentado recorrendo a um indicador indirecto ($L2_{calc}$), dado pela expressão (43), que se aproxima tanto mais de L2, medido com fita métrica, quanto mais exacto for o sistema.

Escolheram-se aleatoriamente 10 posições diferentes, fazendo-se 20 medidas dos ângulos para cada uma, calculando-se a partir desses dados o valor de L2 ($L2_{calc}$). O valor de L2, medido com fita métrica, era 54 cm.

Calculou-se então a média de L2 para cada posição, sobre as 20 medidas.

Apresentam-se na Tabela 8 as médias de $L2_{calc}$ em cada posição.

A média dos valores calculados de L2 é $L2_{med} = 53.7815$ e o desvio padrão 0.4245.

O facto de a média não ser exactamente L2 (54 cm) advém provavelmente do facto de o sistema não estar perfeitamente calibrado, e ainda, eventualmente, de erros de medição quer de L1 (distância entre sensores), quer de L2 (distância entre reflectores).

Tabela 8: Valores calculados de L2 ($L2_{calc}$)

| **Posição** | $L2_{calc}$ |
|---|---|
| **1** | 53.7270 |
| **2** | 53.4775 |
| **3** | 54.0689 |
| **4** | 52.9505 |
| **5** | 53.9476 |
| **6** | 54.3847 |
| **7** | 54.0923 |
| **8** | 53.9346 |
| **9** | 53.3100 |
| **10** | 53.9221 |

# 4 Geração de trajectórias

Neste capítulo são descritos os algoritmos de geração das trajectórias que levam a Robuter do seu ponto inicial até ao ponto de acostagem, (X_ACOST, Y_ACOST) com orientação 0° (ver Figura 28).

Foram desenvolvidos dois procedimentos distintos. O primeiro, que se designa por **aproximação**, destina-se a levar a plataforma móvel até uma posição em que seja possível utilizar o sistema sensorial, que, como se viu no capítulo anterior, tem um alcance limitado. O segundo, já de **acostagem**, utiliza a informação dos sensores para gerar as trajectórias que, iterativamente, levam a plataforma a posições sucessivamente mais perto da posição de acostagem, até um determinado limite de erro, como se verá adiante.

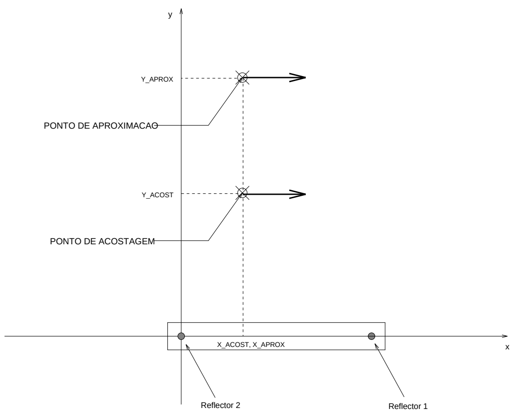


Figura 28: Pontos de aproximação e de acostagem

O procedimento de aproximação é utilizado quando não é possível obter uma medida de localização com os sensores, pelo que exige que se tenha informação sobre a posição e orientação da plataforma. Numa situação real essa informação será proveniente de outros sistemas de localização, por exemplo, odometria. Neste trabalho essas coordenadas são dadas pelo utilizador. Em ambos os casos, essas coordenadas são aproximadas, i.e., contêm erros, pelo que se usa como ponto final para o procedimento de aproximação um ponto mais afastado da estação de trabalho (X_APROX, Y_APROX), diminuindo a probabilidade de colisão.

Para a determinação de (X_APROX, Y_APROX), utilizou-se um critério simples:

$$\begin{cases} X\_APROX &= X\_ACOST \\ Y\_APROX &= Y\_ACOST + 40cm \end{cases}$$

O processo começa por fazer uma leitura do sistema sensorial. Se esta for válida, a posição e orientação do robot móvel são determinadas e o controlo é passado directamente para o procedimento de acostagem. Caso contrário, as coordenadas aproximadas da plataforma ($x_0$, $y_0$ e $t_0$) são pedidas ao utilizador. Se $y_0 \geq Y\_APROX$, é usado o procedimento de aproximação para levar a plataforma para a posição (X_APROX,Y_APROX) e em seguida o de acostagem para o estacionamento final (ver Figura 28).

Como a estação onde é feita a acostagem está colocada junto a uma parede, é necessário garantir que a plataforma não atinge posições demasiado próximas da parede. Isto não significa perda de generalidade dado que facilmente se alteram os procedimentos de aproximação e de acostagem por forma a que a acostagem possa ser feita de qualquer dos lados da estação de trabalho fixa.

Se a coordenada $y_0$ for menor do que Y_APROX, o procedimento de aproximação não pode ser utilizado pois, como se verá na Secção 4.1, este procedimento elimina todas as soluções de aproximação que passem por pontos em que $y_0 < Y\_APROX$. Assim, o programa começa por movimentar a plataforma de modo a aumentar a sua coordenada $y_0$ por uma das duas formas seguintes:

i) Se a orientação $t_0$ dada pelo utilizador é tal que a plataforma está orientada de forma a que $(-90° < t_0 < 90°)$, é usado o procedimento de acostagem partindo das coordenadas dadas pelo utilizador, uma vez que este só é válido para essa gama de valores de orientação. Como essas coordenadas estão sujeitas a erros, o primeiro movimento de acostagem é feito para o ponto (X_APROX, Y_APROX).

ii) Se ($t_0 < -90°$ ou $t_0 > 90°$, o procedimento de acostagem não pode ser utilizado. Assim, o programa escolhe um arco de circunferência que garanta que a plataforma atinja uma posição em que $y_0 > Y\_APROX$. Em seguida, usa o procedimento de aproximação, seguido do de acostagem.

## 4.1 Aproximação à estação de trabalho fixa

O objectivo deste algoritmo é gerar trajectórias suaves que aproximem a plataforma do ponto de acostagem tendo em conta que o veículo não é holonómico e só se permitem curvaturas de raio mínimo (não se permite que a plataforma gire sobre si própria). Assim, as trajectórias geradas são compostas por arcos de circunferência e segmentos de recta.

Sabendo que a trajectória mais curta entre duas localizações, (posição e orientação), para veículos não-holonómicos, é composta de dois arcos de circunferência unidos por uma recta, [8], (ver Figura 29), geraram-se as trajectórias que, a partir do ponto inicial, colocariam a Robuter no ponto (X_APROX, Y_APROX), com direcção 0°. Esta aproximação, com uma escolha apropriada do ponto de chegada, permite uma trajectória segura, ou seja, suficientemente afastada do ponto de acostagem para não colidir com a estação de trabalho ou com a parede, e simultaneamente uma colocação adequada à boa utilização do sistema sensorial.

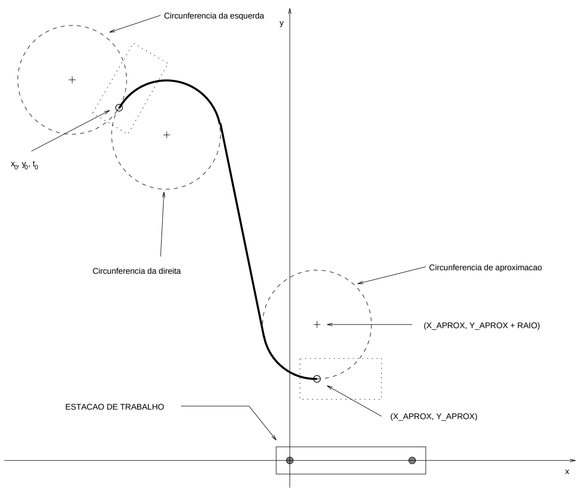


Figura 29: Trajectória de aproximação

Dado que na implementação feita só existem dois sensores, do lado direito da plataforma móvel, é necessário definir a trajectória de modo a que a orientação no fim da aproximação seja 0° e não 180°, independentemente de a velocidade no último arco de circunferência ser positiva ou negativa.

Definiram-se assim as duas circunferências possíveis para o primeiro arco, as quais, pelo facto de o veículo ser não-holonómico, são evidentemente tangentes à recta que define a orientação inicial da

Robuter. Denominaram-se por circunferência da esquerda e da direita, em função da sua posição em relação ao sentido da plataforma, como se pode ver na Figura 30.

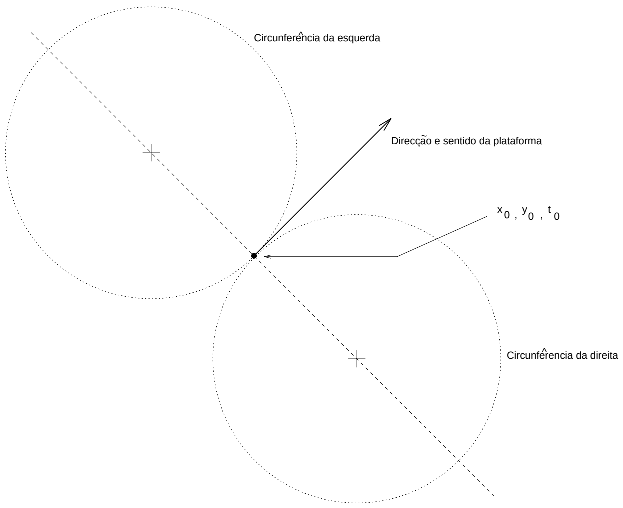


Figura 30: Denominação das circunferências

Dado que se quer uma orientação de 0° para a posição final da trajectória de aproximação, definiu-se a circunferência sobre a qual se fará o último arco pelo seu centro no ponto (X_APROX, Y_APROX + RAIO), e pelo raio (RAIO), escolhido de acordo com as especificações e dimensões da plataforma. Neste trabalho escolheu-se RAIO = 100 cm, igual para as três circunferências definidas. A circunferência assim definida designa-se por circunferência de aproximação.

As trajectórias possíveis estão sujeitas a diversas restrições. A primeira surge da necessidade de fazer o último arco, ao longo de uma circunferência de aproximação, com os sensores do lado exterior da circunferência, para que seja possível a utilização do sistema sensorial pelo procedimento de acostagem. Isso implica que, das quatro rectas simultaneamente tangentes à circunferência sobre a qual se faz o primeiro arco (esquerda ou direita) e à circunferência de aproximação, só duas sejam possíveis (ver Figura 31).

Dado que os sensores estão do lado direito da plataforma, pode-se concluir que se o primeiro arco for feito sobre a circunferência da esquerda (com os sensores no exterior da circunferência), o troço de recta tangente às duas circunferências é necessariamente paralelo à recta que une os centros das duas circunferências, ficando as outras duas trajectórias imediatamente eliminadas. Similarmente, para a circunferência da direita, importam as soluções que permitem a passagem dos sensores do interior da primeira circunferência para o exterior da segunda.

Resultam assim dezasseis trajectórias possíveis (ver Figura 32) para cada condição inicial ($x_0$, $y_0$, $t_0$): para cada um dos quatro troços de recta, existem quatro soluções, dado que sobre cada uma das duas circunferências envolvidas se pode fazer a trajectória no sentido directo ou inverso.

As restantes restrições advêm do facto de, por segurança, não se permitir que a plataforma passe por posições tais que $y < Y\_APROX$. Todas as trajectórias de aproximação em que tal aconteça são eliminadas.

No fim da geração das trajectórias, é escolhida aquela que tiver menor distância a percorrer.

Existem determinadas condições iniciais ($x$, $y$, $t$) = ($x_0$ - X_APROX, $y_0$ - Y_APROX, $t_0$) para as quais não existe nenhuma solução válida. Mostra-se na Figura 33 a curva em (x, y) acima da qual existem sempre soluções válidas, independentemente de $t$. Verifica-se que para $t = 180°$ esta curva corresponde exactamente à fronteira das restrições, ou seja, é um corte horizontal do volume apresentado na Figura 34, feito para $t = 180°$.

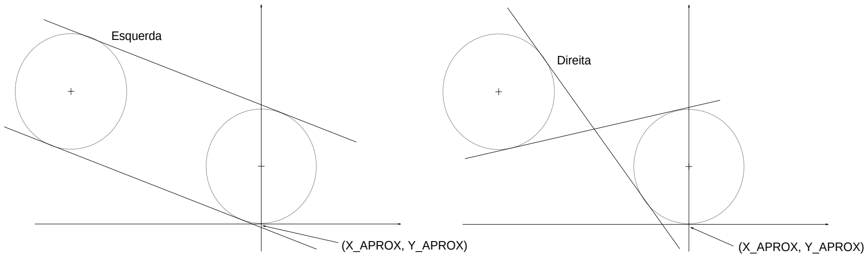


Figura 31: Soluções possíveis para a aproximação

Entre a linha representada na Figura 33 e o ponto (x, y) = (0, 0), ou seja, $(x_0, y_0)$ = (X_APROX, Y_APROX), existem determinadas orientações t = $t_0$ para as quais não existe nenhuma solução válida; a área em $(x, y)$, para um determinado $t = t_0$, dentro da qual não existem soluções válidas, constitui sempre um subconjunto da área onde não existem soluções válidas para $t = 180°$.

O volume (x, y, t) onde não existe nenhuma solução válida tem uma forma aproximadamente de sino, com a base assente sobre o plano definido pelos eixos **x** e **t**, como representado na Figura 34.

## 4.2 Acostagem à estação de trabalho fixa

Nesta secção é descrito o algoritmo que gera as trajectórias de acostagem. As trajectórias geradas são compostas por arcos de raio fixo $R = 100cm$ e segmentos de recta.

Dentro deste procedimento, o referencial usado tem a mesma orientação do referencial global (estação de trabalho fixa) e tem a origem no ponto de acostagem $(X\_ACOST, Y\_ACOST)$, sendo designado como referencial de acostagem. Assim, as coordenadas da plataforma móvel são transformadas do referencial da estação de trabalho para o referencial de acostagem pela seguinte transformação:

$$\begin{cases} x_0 &= x - X\_ACOST \\ y_0 &= y - Y\_ACOST \\ t_0 &= t \end{cases}$$

Cada iteração do procedimento de acostagem utiliza como sentido inicial, o sentido do último movimento da iteração anterior. Na iteração inicial, o sentido é o do último movimento da trajectória de aproximação, excepto no caso em que não foi necessária a aproximação sendo arbitrado, como sentido, andar para a frente $(v > 0)$.

O problema de acostagem foi dividido nas quatro situações seguintes:

- Andar para trás e $y_0 \geq 0$
- Andar para trás e $y_0 < 0$
- Andar para a frente e $y_0 \geq 0$
- Andar para a frente e $y_0 < 0$.

A divisão introduzida entre as situações de andar para trás e para a frente permite minimizar o número de inversões da velocidade. Esta minimização é relevante porque as rodas livres da plataforma móvel usada introduzem erros de direcção sempre que se inverte o sentido de marcha.

A separação em função do sinal de $y_0$ (ver Figura 22) permite tratar de modo diferente a situação em que $y_0 < 0$ pois esta apresenta o risco de colisão com a estação de trabalho à qual se está a acostar. Para eliminar este risco, o algoritmo escolhe o sentido de marcha que aumenta a coordenada $y_0$ sempre que esta for negativa.

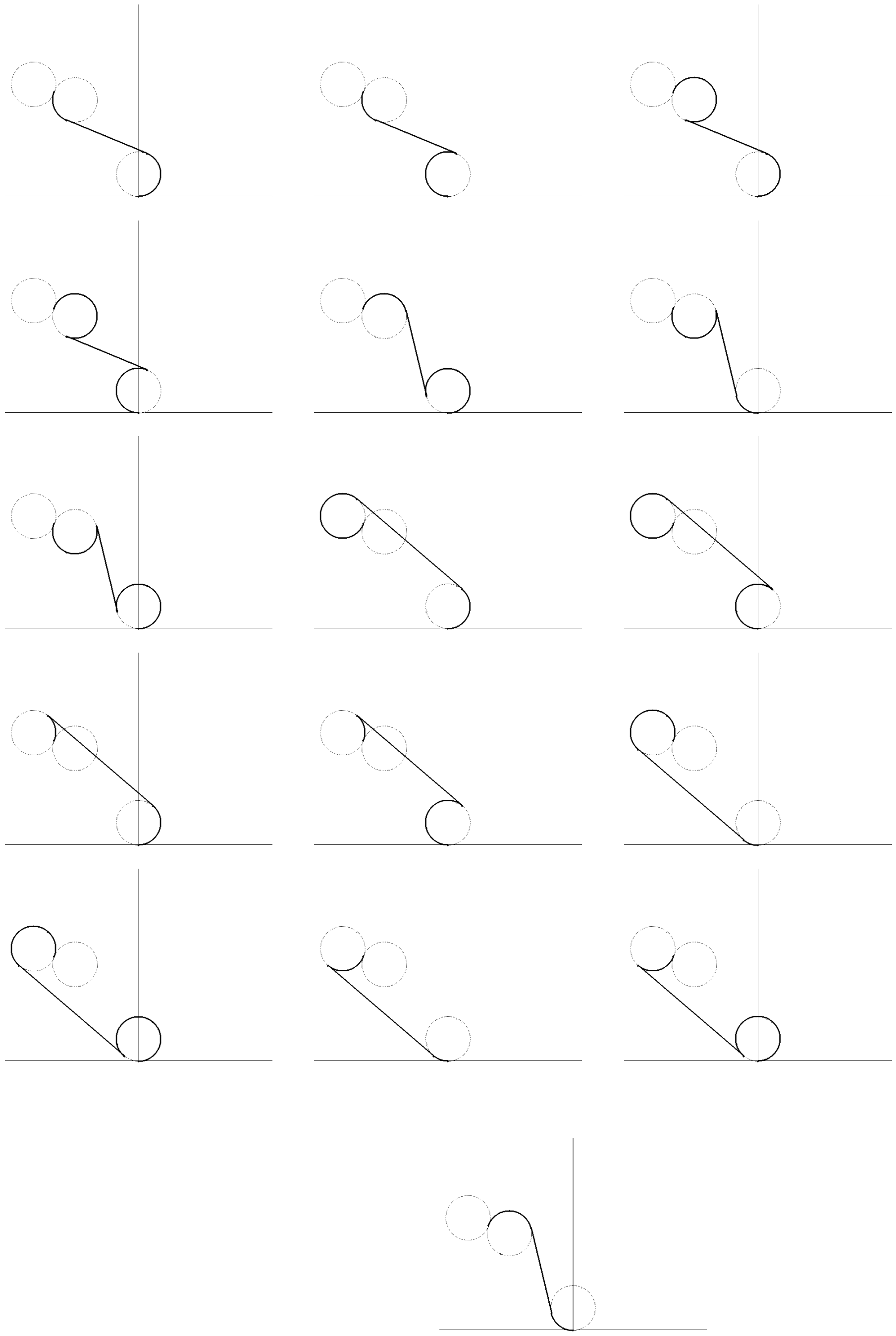

Figura 32: Soluções possíveis para uma dada condição inicial

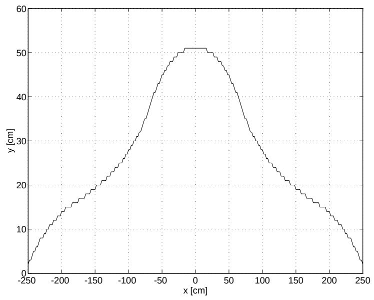


Figura 33: Área com soluções válidas

Note-se que, ao contrário do procedimento de aproximação, o último troço da trajectória de acostagem é um segmento de recta. Este segmento tem já a direcção de acostagem e, como esta é 0 graus, serve apenas para corrigir a coordenada $x_0$ da plataforma.

Na Figura 35 apresenta-se um exemplo de uma trajectória de acostagem.

Em cada situação, existem apenas três acções possíveis:

- A - mudar o sentido de marcha;
- B - descrever um arco, sobre a circunferência da esquerda ou da direita, que coloca a plataforma sobre a recta $y_0 = 0$ e de seguida um segmento sobre essa recta que leva a $x_0 = 0$;
- C - descrever um arco, sobre uma das duas circunferências, que coloca a plataforma em condições de realizar a acção B.

Após a realização de cada acção, o algoritmo volta a determinar a situação em que se encontra, i.e., a localização da plataforma relativamente ao ponto de acostagem, e a acção a tomar. O procedimento de acostagem é finalizado sempre pela acção B.

Como as medidas dos sensores feitas num ponto afastado do ponto de acostagem são menos exactas e como os movimentos da Robuter acumulam erros de posição, a sequência de procedimentos anteriores tem de ser repetida algumas vezes até a plataforma ter acostado à estação de trabalho com um erro menor ou igual ao erro máximo definido.

O algoritmo começa por determinar a localização da plataforma usando os sensores. De seguida, calcula os centros das circunferências de partida (esquerda e direita) e, consoante $y_0$ e o sentido em que vai andar, escolhe a situação apropriada.

Nas sub-secções seguintes, **cd** e **ce** são os centros das circunferências de partida da direita e da esquerda, respectivamente. Os índices **i** e **f** significam inicial e final.

### 4.2.1 Andar para trás e $y_0 \geq 0$

Se a circunferência da esquerda for tangente à recta horizontal que passa pelo ponto de acostagem ($y = 0$) o algoritmo executa a acção B descrevendo um arco dessa circunferência até a plataforma estar apontada para o ponto de acostagem e um segmento que a leva a esse ponto (ver Figura 36 - a)).

Se a circunferência da esquerda não intersectar a recta $y = 0$ ou se o ângulo $t_0$ da plataforma for negativo, a acção escolhida é a C. Assim, a plataforma descreve um arco sobre a circunferência da direita até estar em condições de executar a acção B (ver Figura 36 - b)).

Se não se está em nenhum dos casos anteriores, a circunferência da esquerda intersecta a recta $y = 0$ pelo que leva a $y_0 < 0$. Assim, a plataforma deve andar para a frente e não para trás, sendo

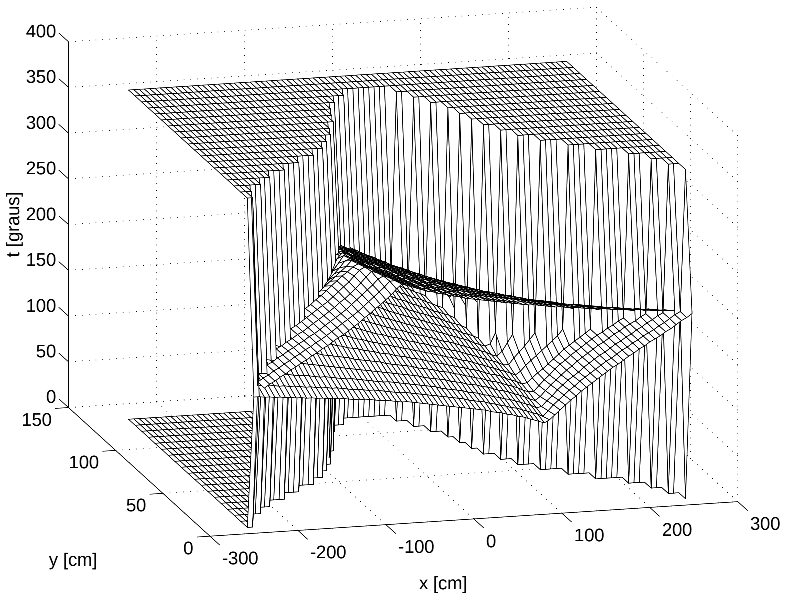


Figura 34: Volume em (x, y, t) sem soluções válidas

escolhida a acção A (ver Figura 36 - c)). Após a mudança de sentido, o procedimento escolhe uma das acções B ou C, tal como descrito na Subsecção 4.2.3.

### 4.2.2 Andar para trás e $y_0 < 0$

Se a plataforma tem uma orientação tal que andar para trás implique diminuir $y_0$ então deve andar para a frente, pelo que o algoritmo escolhe a acção A (ver Figura 37 - a)), trocando o sentido de movimentação, entrando depois na situação descrita na Subsecção 4.2.4.

Se a circunferência da direita for tangente à recta $y = 0$, a acção escolhida é a B. Assim, a plataforma descreve um arco dessa circunferência até estar apontada para o ponto de acostagem e um segmento que a leva a esse ponto (ver Figura  37 - b)).

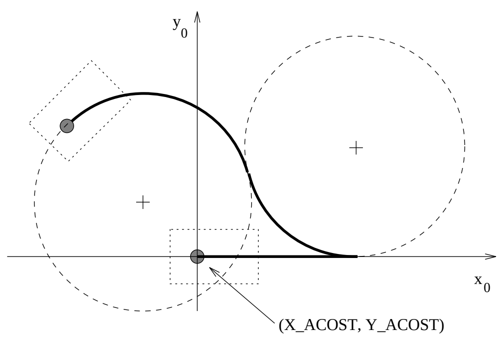


Figura 35: Exemplo de tajectória de acostagem

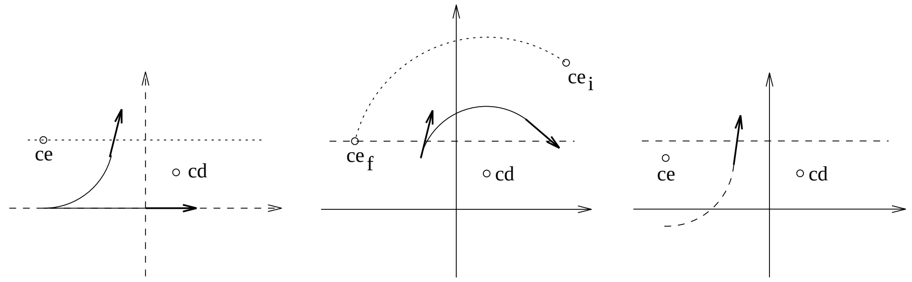


Figura 36: Andar para trás e $y_0 \geq 0$ - a), b) e c)

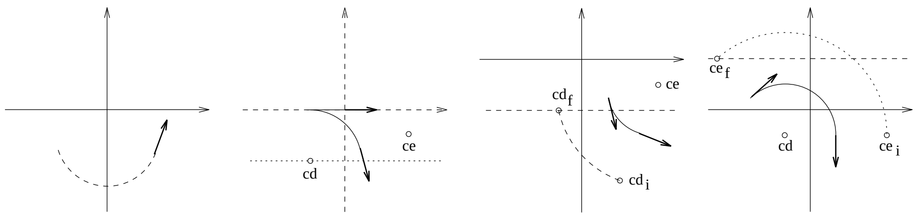


Figura 37: Andar para trás e $y_0 < 0$ - a), b), c) e d)

Se a circunferência da direita não intersectar a recta $y = 0$, escolhe-se a acção C. A plataforma descreve um arco sobre a circunferência da esquerda até poder executar a acção B (ver Figura 37 - c)).

Se a circunferência da direita intersectar a recta $y = 0$, escolhe-se também a acção C. Neste caso, a plataforma descreve um arco sobre a circunferência da direita até poder executar a acção B (ver Figura 37 - d)).

Estas duas situações estão esquematizadas no fluxograma da Figura 40.

### 4.2.3 Andar para a frente e $y_0 \geq 0$

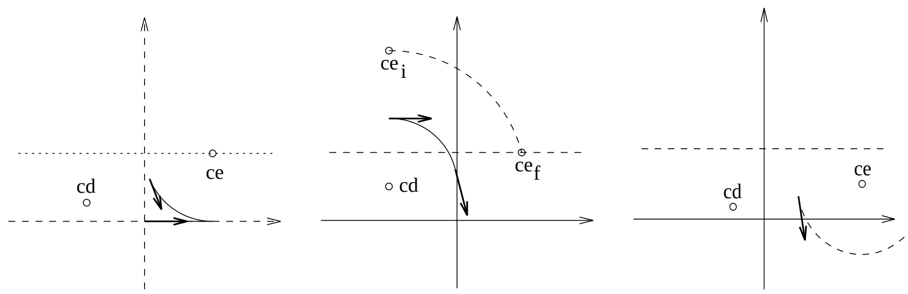


Figura 38: Andar para trás e $y_0 \geq 0$ - a), b) e c)

Se a circunferência da esquerda for tangente à recta $y = 0$, o algoritmo escolhe a acção B. Assim, a plataforma descreve um arco sobre essa circunferência até estar na direcção de acostagem e um segmento de recta até ficar no ponto de acostagem (ver Figura 38 - a)).

Se a circunferência da esquerda não intersectar a recta $y = 0$ ou se o ângulo $t_0$ da plataforma for positivo, a acção escolhida é a C pelo que a plataforma descreve um arco sobre a circunferência da direita até poder executar a acção B (ver Figura 38 - b)).

Se não se está em nenhum dos casos anteriores, a circunferência da esquerda intersecta a recta $y = 0$ pelo que leva a $y_0 < 0$. Assim, a plataforma deve andar para a trás e não para a frente, sendo escolhida a acção A (ver Figura 38 - c)). A acção escolhida de seguida é a B ou a C, tal como se explica na Subsecção 4.2.1.

### 4.2.4 Andar para a frente e $y_0 < 0$

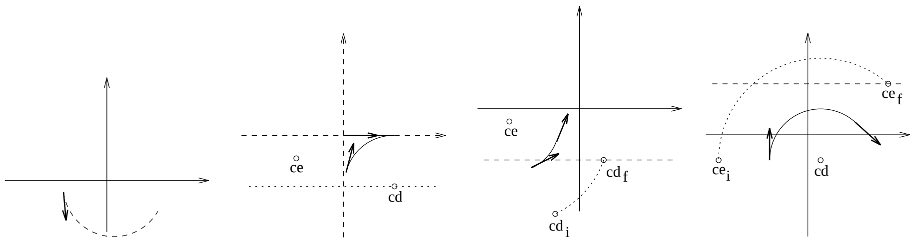


Figura 39: Andar para a frente e $y_0 < 0$ - a), b), c) e d)

Se a plataforma tem uma orientação tal que andar para a frente implique diminuir $y_0$ então deve andar para a trás pelo que algoritmo escolhe a acção A (ver Figura 39 - a)). De seguida é escolhida uma das acções descritas na Subsecção 4.2.2.

Se a circunferência da direita for tangente à recta $y = 0$, a acção escolhida é a B. Assim, a plataforma descreve um arco dessa circunferência até estar na direcção de acostagem e um segmento que a leva ao ponto de acostagem (ver Figura 39 - b)).

Se a circunferência da direita não intersectar a recta $y = 0$, escolhe-se a acção C. A plataforma descreve um arco sobre a circunferência da esquerda até poder executar a acção B (ver Figura 39 - c)).

Se a circunferência da direita intersectar a recta $y = 0$, escolhe-se também a acção C. Neste caso, a plataforma descreve um arco sobre a circunferência da direita até poder executar a acção B (ver Figura 39 - d)).

Estas duas situações estão esquematizadas no fluxograma da Figura 41.

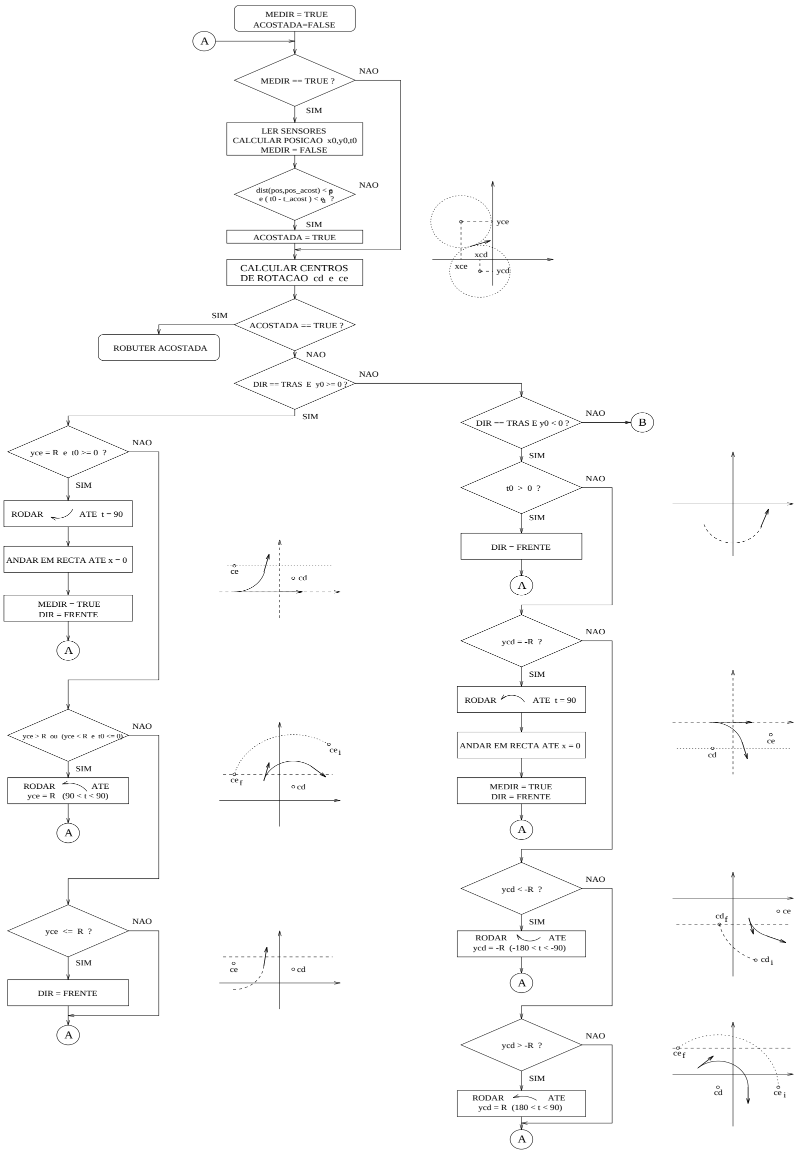


Figura 40: Fluxograma

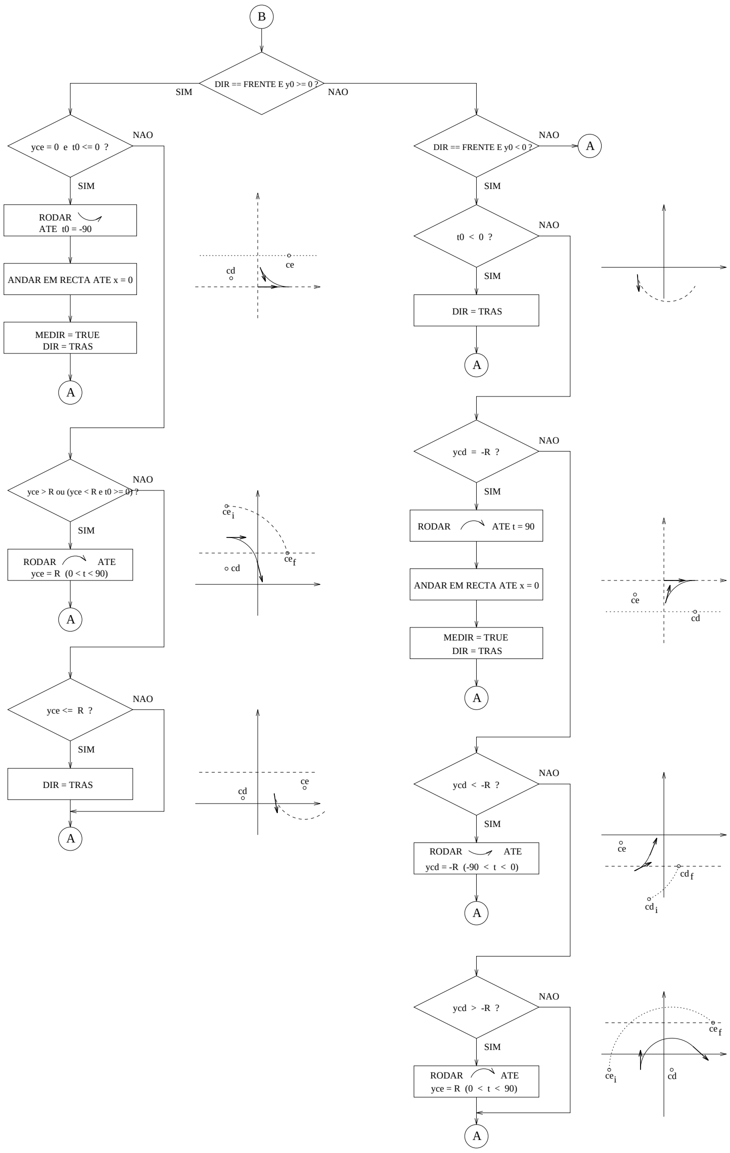


Figura 41: Fluxograma (continuação)

# 5 Resultados experimentais

Apresenta-se neste capítulo uma análise aos resultados experimentais obtidos para algumas leituras dos sensores, bem como para as duas fases do estacionamento - aproximação e acostagem.

Começam por se apresentar alguns exemplos de leituras dos sensores, mostrando-se quais os ângulos determinados.

Na secção seguinte são apresentadas algumas trajectórias de aproximação para diferentes condições iniciais.

Na última secçõ é apresentado um estudo do erro de posição e orientação para a fase de acostagem da plataforma.

Em anexo ao presente trabalho existe uma cassette de vídeo VHS, com uma demonstração do sistema em funcionamento pleno.

## 5.1 Dados sobre o sistema sensorial

Começou por se fazer uma medida de posição com o sistema sensorial. Os valores medidos pelos receptores de infravermelhos são apresentados nas Figuras 42 e 43 para ambos os sentidos de rotação do motor.

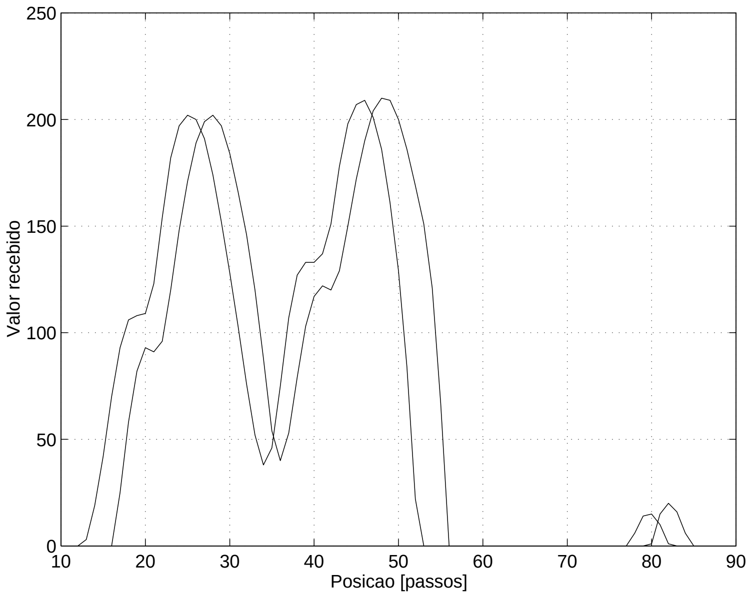


Figura 42: Valores devolvidos na leitura do sensor 1

Com os valores recebidos, aplicando os procedimentos descritos nas Secções 3.4.1, 3.4.3 e 3.4.4, calculou-se a posição e a orientação da Robuter em relação à estação de trabalho.

Para o sensor 1, a que correspondem os ângulos $b_1$ e $b_2$, determinaram-se os máximos $m_{1_{dir}}$, $m_{2_{dir}}$, $m_{3_{dir}}$, $m_{1_{inv}}$, $m_{2_{inv}}$ e $m_{3_{inv}}$, para os vectores $x_{dir}$ e $x_{inv}$, ordenados por ordem decrescente de pontuação:

$$m_{1_{dir}} = 48.34 \text{ passos} = 87.01°$$
$$m_{2_{dir}} = 27.85 \text{ passos} = 50.13°$$
$$m_{3_{dir}} = 82.17 \text{ passos} = 147.91°$$
$$m_{1_{inv}} = 45.65 \text{ passos} = 82.17°$$

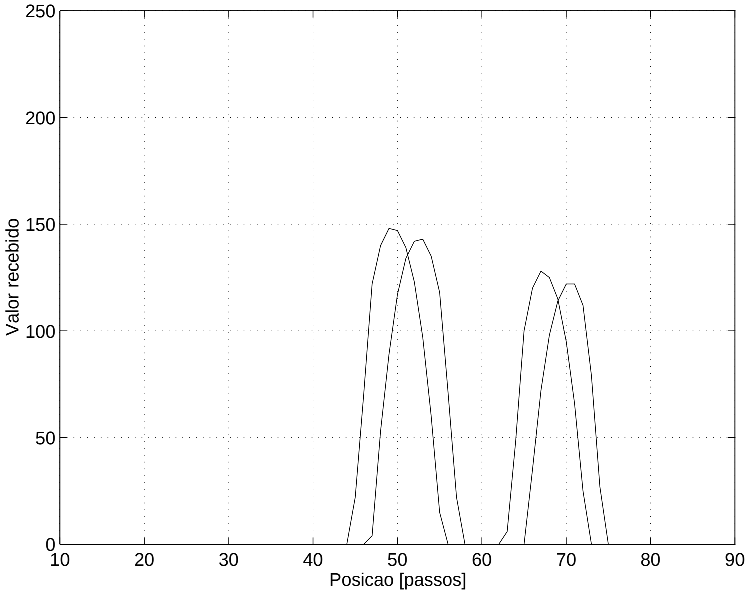


Figura 43: Valores devolvidos na leitura do sensor 2

$$m_{2_{inv}} = 25.26 \text{ passos} = 45.47^\circ$$
$$m_{3_{inv}} = 89.68 \text{ passos} = 143.42^\circ$$

O algoritmo **emparelha** os máximos $m_{2_{dir}}$ e $m_{2_{inv}}$, calculando

$$b_1 = \frac{50.13^\circ + 45.47^\circ}{2} + b_{cal} = 47.07^\circ$$

em que $b_{cal} = -0.732^\circ$ é o valor de calibração para os ângulos $b_i$, determinado previamente.

Do mesmo modo, $m_{1_{dir}}$ **emparelha** com $m_{1_{inv}}$, resultando $b_2 = 83.86^\circ$.

Para o sensor 2, a que correspondem os ângulos $a_1$ e $a_2$, procedeu-se de maneira semelhante, tendo sido encontrados os máximos

$$m_{1_{dir}} = 70.33 \text{ passos} = 126.59^\circ$$
$$m_{2_{dir}} = 52.54 \text{ passos} = 94.57^\circ$$
$$m_{3_{dir}} = 0 \text{ (inexistente)}$$
$$m_{1_{inv}} = 49.48 \text{ passos} = 89.06^\circ$$
$$m_{2_{inv}} = 67.34 \text{ passos} = 121.21^\circ$$
$$m_{3_{inv}} = 0 \text{ (inexistente)}$$

Assim, corrigindo $a_1$ e $a_2$ com $a_{cal} = -0.444$, obtém-se $a_1 = 91.37^\circ$ e $a_2 = 123.46^\circ$.

Com estes valores de ângulos, calcula-se:

$$L2_{calc} = 53.49 \text{ cm}$$
$$x_0 = 32.66 \text{ cm}$$
$$y_0 = 108.98 \text{ cm}$$
$$t_0 = -16.87^\circ$$

tal como se vê na Figura 44.

Fez-se uma segunda experiência, desta feita apenas com um sensor, com o intuito de mostrar o algoritmo de deteção de máximos (descrito na Secção 3.4.1).

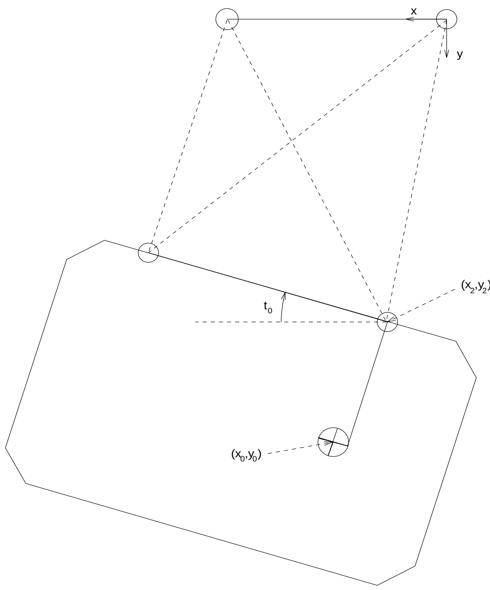


Figura 44: Posição e orientação determinadas experimentalmente

Tabela 9: Posições [passos], pontuações e valores para $x_{dir}$

| Máximo | Posição | Pontuação | Valor |
|---|---|---|---|
| $m1_{dir}$ | 72.50 | 292 | 204 |
| $m2_{dir}$ | 27.90 | 291 | 195 |
| $m3_{dir}$ | 48.36 | 286 | 203 |

Apresentam-se na Figura 45 os vectores recebidos ($x_{dir}$ e $x_{inv}$). As posições, as pontuações e os valores dos três melhores máximos, para cada vector, são apresentados nas Tabelas 9 e 10.

O algoritmo **emparelha** imediatamente os máximos $m2_{dir}$ e $m2_{inv}$. Os máximos $m1_{dir}$ e $m3_{inv}$ não são **emparelhados**, apesar da proximidade de posições, devido à grande diferença de valores. São também **emparelhados** os máximos restantes, ou seja, $m3_{dir}$ e $m1_{inv}$.

Mostra-se assim como se eliminou o ruído em torno da posição 70.

## 5.2 Aproximação

Apresentam-se três exemplos típicos de trajectórias de aproximação, para três condições iniciais distintas (ver Tabela 11), resultantes da simulação do algoritmo implementado.

As trajectórias são apresentadas nas Figuras 46, 47 e 48.

Tabela 10: Posições [passos], pontuações e valores para $x_{inv}$

| Máximo | Posição | Pontuação | Valor |
|---|---|---|---|
| $m1_{inv}$ | 45.69 | 300 | 204 |
| $m2_{inv}$ | 25.22 | 288 | 198 |
| $m3_{inv}$ | 68.17 | 229 | 113 |

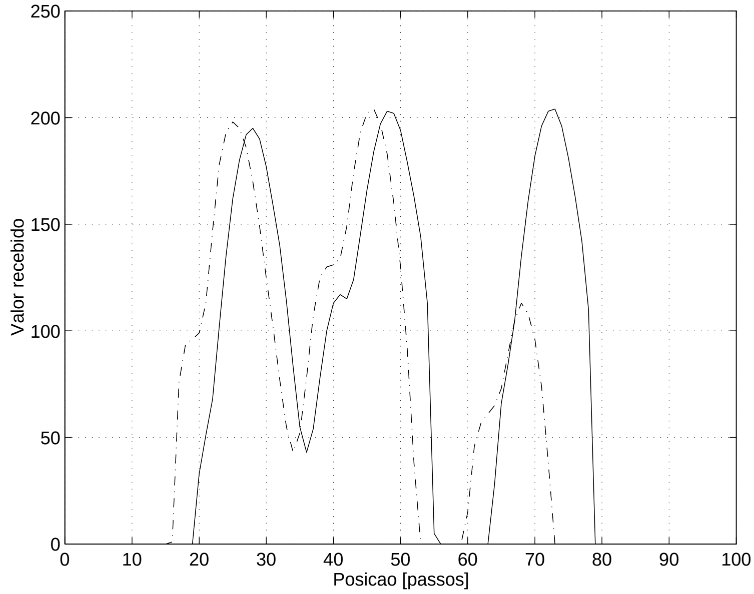


Figura 45: Valores devolvidos na leitura do sensor

Tabela 11: Posições iniciais para os exemplos de trajectórias de aproximação

| **Número** | **Posição inicial ($x_0$ [cm], $y_0$ [cm], $t_0$ [°])** |
|---|---|
| **1** | (-250, 400, 0) |
| **2** | (0, 300, 0) |
| **3** | (200, 500, -90) |

## 5.3 Acostagem

Nesta secção são mostrados os resultados obtidos com o procedimento de acostagem. Foram feitos dois estudos, sendo um relativo à variação do erro de posição e orientação da plataforma para iterações sucessivas deste procedimento e o outro relativo à precisão máxima obtida.

### 5.3.1 Variação do erro para iterações sucessivas

Foram feitas 10 acostagens, a partir de pontos iniciais distintos, tendo sido feita uma análise composta pela média, desvio padrão, valor mínimo e valor máximo para as três coordenadas de posição e orientação da plataforma móvel ($x_0$, $y_0$ e $t_0$), medidas pelo sistema sensorial projectado, ao longo da experiência.

As posições e orientações iniciais da plataforma e as zonas que contêm as posições nas primeiras e segundas iterações de cada acostagem estão representadas na Figura 49. Os resultados são apresentados nas Tabelas 12, 13 e 14. São apresentados os valores apenas para as primeiras três iterações dado que são suficientes para atingir a precisão máxima (estudada na sub-secção seguinte).

É nítido o rápido decréscimo dos desvios padrão das três variáveis (ver Figura 49). De facto, por inspecção das Tabelas 12, 13 e 14, conclui-se que os erros absolutos de cada uma das coordenadas $x_0$, $y_0$ e $t_0$ no fim de cada iteração são majorados pelos valores que se apresentam na Tabela 15.

Estes valores representam, portanto, o erro absoluto máximo esperado no fim de cada iteração, ou seja, a exactidão máxima conseguida.

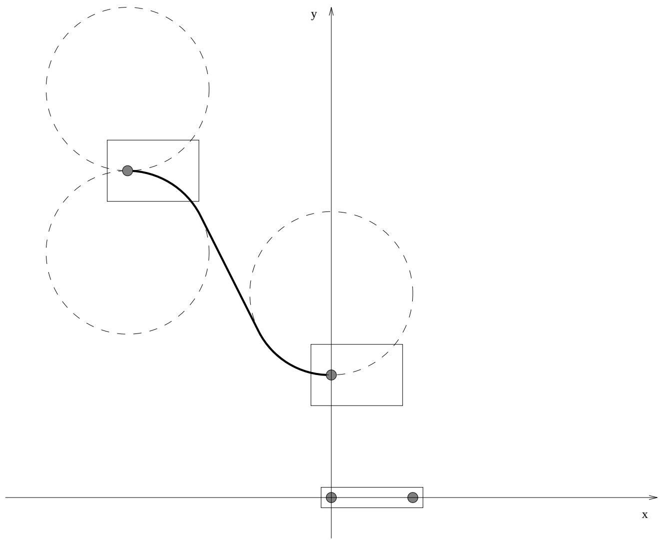


Figura 46: Trajectória de aproximação 1

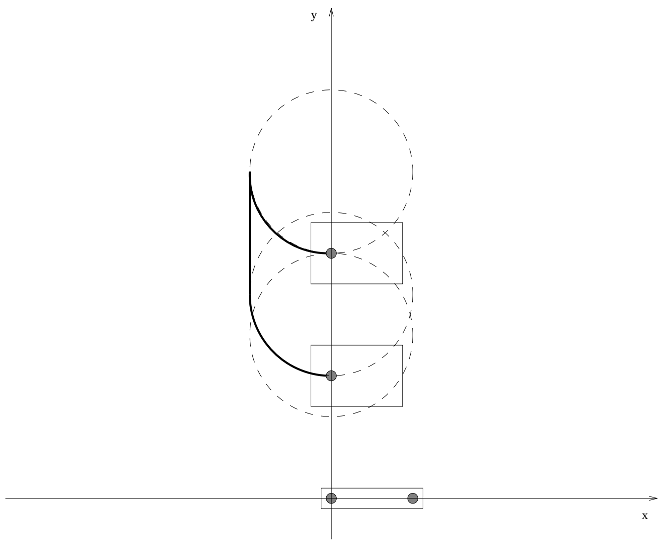


Figura 47: Trajectória de aproximação 2

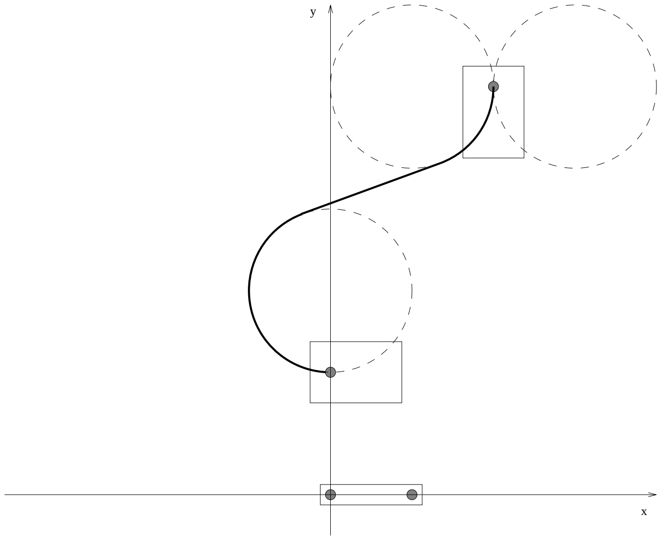


Figura 48: Trajectória de aproximação 3

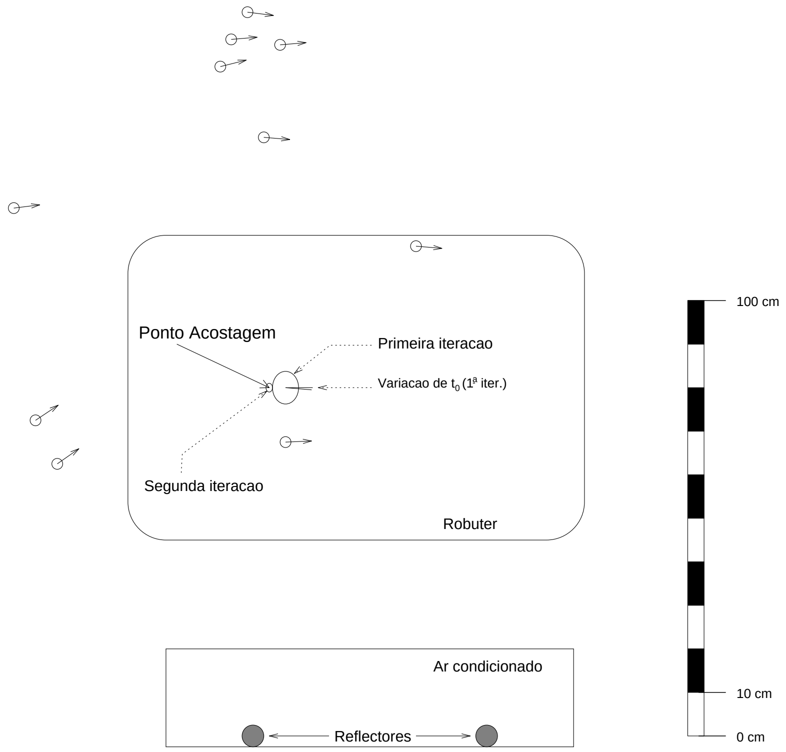


Figura 49: Posições iniciais de 10 acostagens (o—>) e zonas que contêm as posições das primeiras e segundas iterações

Tabela 12: Valores de $x_0$ em centímetros para 3 iterações sucessivas

| | $1^a$ iteração | $2^a$ iteração | $3^a$ iteração |
|---|---|---|---|
| **Média** | 3.8752 | -0.0898 | -0.0967 |
| **D. Pad.** | 2.5520 | 0.6483 | 0.2100 |
| **Mínimo** | -0.7606 | -0.9765 | -0.5783 |
| **Máximo** | 7.9596 | 1.2160 | 0.4326 |

Tabela 13: Valores de $y_0$ em centímetros para 3 iterações sucessivas

| | $1^a$ iteração | $2^a$ iteração | $3^a$ iteração |
|---|---|---|---|
| **Média** | 0.2824 | -0.2210 | 0.0089 |
| **D. Pad.** | 2.8036 | 0.6994 | 0.3468 |
| **Mínimo** | -4.6325 | -1.0571 | -0.5215 |
| **Máximo** | 3.4944 | 0.7895 | 0.5123 |

Como a posição inicial da plataforma foi sempre tal que permitiu a utilização dos sensores, nunca foi usado o procedimento de aproximação. Assim, a acostagem começou sempre com um movimento em que a plataforma andou para a frente (ver Secção 4.2), pelo que, após a primeira iteração, existe um erro sistemático em $x_0$ que se reflecte na Tabela 12 por uma média superior a zero.

Se a acostagem começar por um movimento em que a plataforma ande para trás, o erro de $x_0$ após a primeira iteração reflecte-se no sentido oposto correspondendo a uma média de $x_0$ menor que zero. Os resultados dessa experiência não são apresentados pois esta é a única diferença significativa em relação aos que se apresentaram.

### 5.3.2 Estudo da precisão máxima para uma acostagem

Após quatro iterações do procedimento de acostagem, foram feitas mais 64 iterações registando-se, após cada uma, as medidas das cordenadas $x_0$, $y_0$ e $t_0$ obtidas através do sistema sensorial. Estas medidas estão apresentadas graficamente na Figura 50.

Na Tabela 16 são apresentados os resultados da análise estatística feita sobre as medidas obtidas.

Observando a Figura 50 e a Tabela 16 tiram-se duas conclusões importantes:

i) a partir da quarta iteração não existe melhoria visível da posição e orientação da plataforma;

ii) o erro mínimo que se garante para uma iteração é de $\pm 0.5$ cm para $x_0$ e $y_0$ e $\pm 1°$ para $t_0$.

Note-se que a precisão dos sensores (ver Tabela 7) não permite garantir um erro menor, pelo que não vale a pena fazer especificar uma precisão menor do que esta.

Tabela 14: Valores de $t_0$ em graus para 3 iterações sucessivas

| | $1^a$ iteração | $2^a$ iteração | $3^a$ iteração |
|---|---|---|---|
| **Média** | -2.7277 | 0.8178 | -0.2284 |
| **D. Pad.** | 1.9817 | 0.8025 | 0.5111 |
| **Mínimo** | -6.0998 | -0.5597 | -1.1930 |
| **Máximo** | 0.1859 | 1.9344 | 0.9462 |

Tabela 15: Majorantes dos erros em 3 iterações sucessivas

| | **1ª iteração** | **2ª iteração** | **3ª iteração** |
|---|---|---|---|
| $x_0$ [cm] | 10 | 2 | 1 |
| $y_0$ [cm] | 6 | 2 | 1 |
| $t_0$ [°] | 8 | 2.5 | 1.5 |

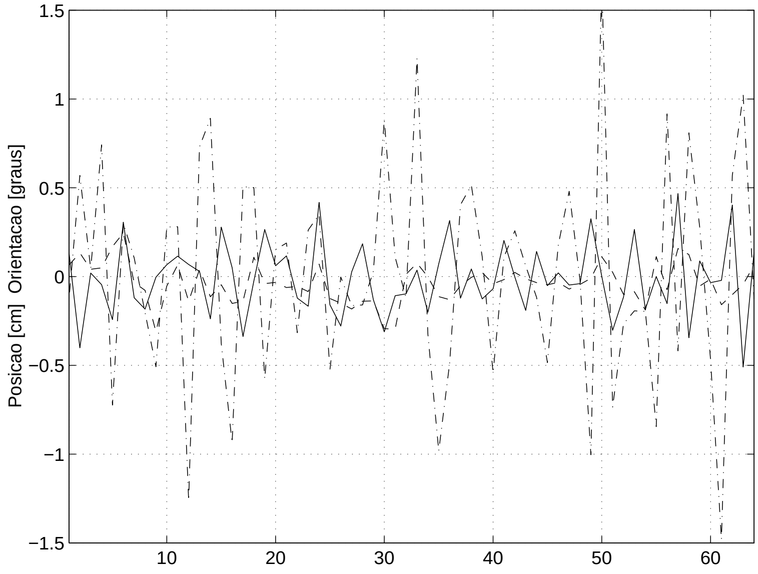


Figura 50: Medidas de $x_0$ (—), $y_0$ (- -) e $t_0$ (-.-) em 64 iterações de acostagem

Tabela 16: Análise estatística das medidas obtidas nas 64 iterações

| | $x_0$ [cm] | $y_0$ [cm] | $t_0$ [°] |
|---|---|---|---|
| **Média** | -0.0128 | -0.0370 | 0.0136 |
| **D. Pad.** | 0.2045 | 0.1094 | 0.6066 |
| **Mínimo** | -0.5090 | -0.3100 | -1.4802 |
| **Máximo** | 0.4675 | 0.2456 | 1.6904 |

# 6 Conclusões

Considera-se que, de um modo geral, os objectivos propostos no início do trabalho foram alcançados.

Conseguiu-se uma acostagem de precisão, com um erro menor do que ($\pm$ 0.5 cm, $\pm$ 0.5 cm, $\pm$ 1°) em ($x$, $y$, $t$).

Conseguiu-se um sistema sensorial suficientemente preciso e exacto para a correcta localização do robot móvel em relação à estação de trabalho à qual se quer acostar. O custo dos componentes e placas do protótipo foi menor do que 20 000$00.

Determinaram-se trajectórias suaves que conduzem sempre o robot móvel à estação de trabalho, respeitando a limitação das curvaturas com raio mínimo.

O *software* desenvolvido para a Robuter permite que, em algumas situações, se perca a medida de um dos ângulos (devido a um obstáculo de permeio, por exemplo), ou ainda que se obtenha um máximo na leitura, devido a ruído não filtrado presente na recepção (e.g. uma faixa estreita de luz), conseguindo ainda assim determinar correctamente a posição e a orientação da plataforma móvel.

A electrónica projectada para a filtragem e condicionamento do sinal recebido poderia ter sido mais desenvolvida, com vista a conseguir um maior alcance e melhor redução de ruído. Um possível desenvolvimento poderia ser no sentido de melhorar o isolamento entre os circuitos de emissão e recepção de infra-vermelhos. Seria também útil um aumento da velocidade de resposta da electrónica de recepção e filtragem.

A pouca robustez do suporte mecânico dos sensores foi um factor que criou alguma instabilidade no sistema sensorial, tanto na fase de desenvolvimento, como durante a sua utilização. Foi esta limitação que levou à necessidade de calibração, descrita no presente relatório.

Para o desenvolvimento de todo o sistema sensorial teria sido útil que existisse um outro sistema com maior exactidão.

A triangulação com apenas dois sensores e dois reflectores é muito sensível a erros de medição, podendo-se optar por dotar a Robuter de mais sensores (inclusivamente dos dois lados, com vista a melhorar as situações de leitura), ou então utilizar mais do que dois reflectores. Uma sugestão é a utilização de reflectores de largura diferente para a correcta detecção a diferentes distâncias

No caso da utilização de mais sensores, seria útil implementar um procedimento de calibração automatizado.

A geração de trajectórias está relacionada com o ambiente de trabalho em que foi desenvolvido o projecto, nomeadamente pelo facto de se considerar que o espaço livre está sempre limitado pelo eixo **x** do referencial colocado sobre a estação de trabalho. O algoritmo poderá ser extendido facilmente, por forma a considerar outro tipo de ambiente.

Outras restrições, como por exemplo uma iluminação diferente, o aparecimento de obstáculos inesperados ou restrições diferentes ao espaço navegável, que poderá ser limitado (por exemplo, acostagem num corredor estreito), limitam as possibilidades de utilização destes procedimentos, tais como estão.

Não se usaram os sonares disponíveis na plataforma móvel, sendo, no entanto, uma hipótese a considerar, principalmente junta à estação de trabalho, como complemento à informação obtida pelo sistema sensorial desenvolvido. Poderia também ser útil para o contorno de obstáculos.

Para a movimentação da plataforma móvel Robuter, foi utilizado o comando **motv**, já implementado no computador de bordo da mesma. No entanto, um estudo mais detalhado da dinâmica da Robuter, juntamente com o desenvolvimento de um comando específico poderia minimizar os erros de seguimento de trajectória.

O correcto controlo de velocidade e posição da Robuter, uma electrónica mais rápida e/ou diminuindo o espaço percorrido em cada leitura, poderão permitir a utilização do sistema sensorial durante a movimentação da Robuter, acelerando o processo de aquisição de dados.

## Referências


[1] "Albatros$^{TM}$ 4.0 Reference Manual"

[2] "Albatros$^{TM}$ User's Manual V 5.2"

[3] "Albatros$^{TM}$ C Interface Reference Manual"

[4] M. S. Bazaraa, C. M. Shetty, "Nonlinear Programming - Theory and Algorithms," John Wiley & Sons, 1979.

[5] P. Boegli, "A comparative evaluation of AGV navigation techniques," 3rd International Conference on Automated Guided Vehicle Systems, Outubro de 1985, pp. 169-180.

[6] S. C. Chapra, R. P. Canale, "Numerical Methods for Engineers," McGraw-Hill International Editions, 2nd edition, 1989.

[7] G. F. Franklin, J. D. Powell, A. Emami-Naeini, "Feedback Control of Dynamic Systems," 2nd edition, Addison Wesley, 1991.

[8] J. P. Laumond, M. Taix, P. Jacobs, "A motion planner for car-like robots based on a mixed global/local approach," IEEE International Workshop on Intelligent Robots and Systems, Vol. 2, Julho de 1990, pp. 765-773.

[9] "PIC 16/17 Microcontroller Data Book," Microchip Technology, 1995.

[10] "PICStart$^{TM}$ - 16C Development System User's Guide," Microchip Technology, 1994.

[11] "Robuter$^{TM}$ User's Manual 3.1"

[12] A. S. Sedra, K. C. Smith, "Microelectronic Circuits," 3rd edition, Saunders College Publishing, 1991.

[13] C. F. Wojslaw, E. A. Moustakas, "Operational Amplifiers," John Wiley & Sons, 1986.