



TÉCNICO LISBOA

Técnicas básicas de navegação para robôs móveis

Cadeira de Robótica, 2º Semestre 2018/2019

Engenharia Eletrotécnica e de Computadores, Instituto Superior Técnico

José Silva, 84109

Manuel Carvalho, 84127

Miguel D'Ajuda, 84144

24 de Maio, 2019

Resumo

O presente documento visa explicitar a estratégia adotada para a navegação do robô móvel Pioneer3 em torno do piso 5 da torre Norte. A referida teve como base o controlo do robô utilizando pontos de referência, e foi implementada utilizando os dados da odometria e dos sonares do robô. Paralela à navegação, foi implementada a funcionalidade adicional da deteção de portas utilizando *Laser Range Finders*.

Conteúdo

1	Introdução	3
2	Navegação	4
2.1	Controlo do Robô	4
2.1.1	Lei de Controlo	4
2.1.2	Convergência	5
2.2	Trajectoria	5
2.3	Ajuste da trajetória	6
2.4	Arquitetura	7
2.5	Simulação	8
2.6	Resultados	10
3	Deteção de portas	12
3.1	Método	12
3.2	Deteção da proximidade de uma porta	12
3.3	Deteção do centro da porta	13
3.3.1	Transformação para coordenadas cartesianas	13
3.3.2	Parametrização de paredes	13
3.3.3	Deteção de portas	14
3.3.4	Determinar o centro da porta	16
3.4	Determinação do estado da porta	16
3.5	Resultados	17
4	Solução Geral	18
4.1	Arquitetura	18
5	Discussão	19
5.1	Navegação do Robô	19
5.2	Deteção das portas	19
5.3	Geral	19

1 Introdução

Este documento apresenta a solução desenvolvida para um problema de navegação de um robô móvel num ambiente conhecido a priori. Este, possui duas componentes, sendo estas

1. Movimentação do Robô: Fazer o robô movimentar-se no 5º piso da Torre Norte, com a seguinte trajetória: Começar dentro do laboratório de Robótica, completar uma volta pelos corredores do piso e terminar no mesmo local onde começou. Para tal, o robô deve evitar embater contra os obstáculos presentes neste percurso, controlando-se através dos sensores à disposição dos alunos - Odometria e Sonares do robô e o *Laser Range Finder*.

2. Detecção de portas: Fazer com que o robô detete uma porta sempre que, durante o seu percurso, passa por elas, virando-se de frente. Este deve, também, conseguir diferenciar uma porta fechada, aberta e semi-aberta, reagindo de forma diferente aos 3: produzindo um som ou frase que se aplique ao respetivo caso.

O projeto encontra-se inserido no contexto do segundo laboratório de robótica que visa desenvolver a capacidade dos alunos de lidar com um robô móvel, apelando a abordagens criativas. Adicionalmente, através do contacto com os diversos sensores e com o próprio robô, é também desenvolvida alguma capacidade de saber lidar com os elementos físicos de um problema de Robótica.

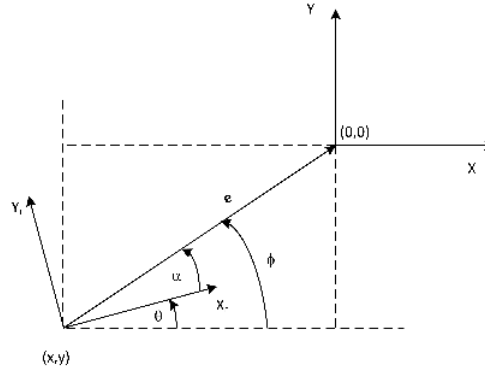


Figura 1: Representação geométrica do problema de controle

2 Navegação

A navegação do robô pode ser dividida em três partes, o controle do robô, a definição da trajetória do mesmo e os ajustes à trajetória, necessários devido a fatores como os erros na leitura da odometria ou a variações no estado inicial do robô. Irão ser apresentadas nesta secção as abordagens tomadas para cada um destes pontos.

2.1 Controle do Robô

O controle do robô será feito com base em pontos de referência. Isto é, tem-se o ponto para onde se deseja que o robô se desloque e são-lhe fornecidas as velocidades lineares e angulares, de forma a deslocá-lo da sua posição atual para o ponto desejado.

2.1.1 Lei de Controle

O problema exposto anteriormente pode ser resumido pela Figura 1.

Da figura, podem-se retirar as seguintes relações

$$\begin{cases} e = \sqrt{x^2 + y^2} \\ \phi = \tan^{-1}\left(\frac{-y}{-x}\right) \\ \alpha = \phi - \theta \\ \dot{e} = \frac{x}{e}\dot{x} + \frac{y}{e}\dot{y} = -\cos(\phi)\dot{x} - \sin(\phi)\dot{y} \\ \dot{\phi} = \frac{1}{e}\left(-\frac{y}{e}\dot{x} + \frac{x}{e}\dot{y}\right) = \frac{1}{e}\left(-\sin(\phi)\dot{x} - \cos(\phi)\dot{y}\right) \\ \dot{\alpha} = \dot{\phi} - \dot{\theta} \end{cases} \quad (1)$$

$$\begin{cases} \dot{e} = -\cos(\alpha)v \\ \dot{\phi} = \frac{\sin(\alpha)}{e}v \\ \dot{\alpha} = -\omega + \frac{\sin(\alpha)}{e}v \end{cases} \quad (2)$$

Uma possível lei de controle será

$$v = v_{max} \tanh(K_1 e) \quad (3)$$

$$\omega = v_{max} \left[\left(1 + K_2 \frac{\phi}{\alpha}\right) \frac{\tanh(K_1 e)}{e} \sin(\alpha) + K_3 \tanh(\alpha) \right] \quad K_1, K_3 > 0 \quad (4)$$

2.1.2 Convergência

Pretende-se comprovar que o sistema, robô + controlador, converge para o ponto $(0,0,0)$. Este verifica-se ser um ponto de equilíbrio visto que $\dot{e} = \dot{\alpha} = \dot{\phi} = 0$.

De forma a testar a convergência da lei de controlo proposta, será utilizada a seguinte função

$$V(e, \phi, \alpha) = \frac{1}{2}e^2 + \frac{1}{2}(K_2\phi^2 + \alpha^2) \quad (5)$$

a qual corresponde a uma função de Lyapunov visto que

- $V(e, \phi, \alpha)$ é continua
- $V(e, \phi, \alpha)$ possui derivadas parciais contínuas
- $V(e, \phi, \alpha) > 0$ para qualquer $(e, \phi, \alpha) \neq (0, 0, 0)$

Ora, como $\dot{V}(e, \phi, \alpha) \leq 0$ a convergência do sistema para $(0,0,0)$ é garantida, pelo menos no sentido de Lyapunov (o sistema fica preso numa vizinhança do ponto de equilíbrio). Adicionalmente, como $V(e, \phi, \alpha) \rightarrow +\infty$ quando $|x| \rightarrow \infty$, conclui-se que a origem é no mínimo globalmente estável, no sentido de Lyapunov.

2.2 Trajetória

Com a lei de controlo acima descrita, o robô é capaz de se deslocar até um dado ponto no espaço. Ora isto para o problema em questão não é suficiente, pretendendo-se que o mesmo siga uma trajetória no espaço, saído do laboratório e voltando a entrar no mesmo após completar uma volta em torno do corredor da torre.

Para resolver esta questão, é definida no espaço uma trajetória discreta, representada por um conjunto de pontos distribuídos de forma relativamente uniforme como a que se encontra representada na Figura 2.

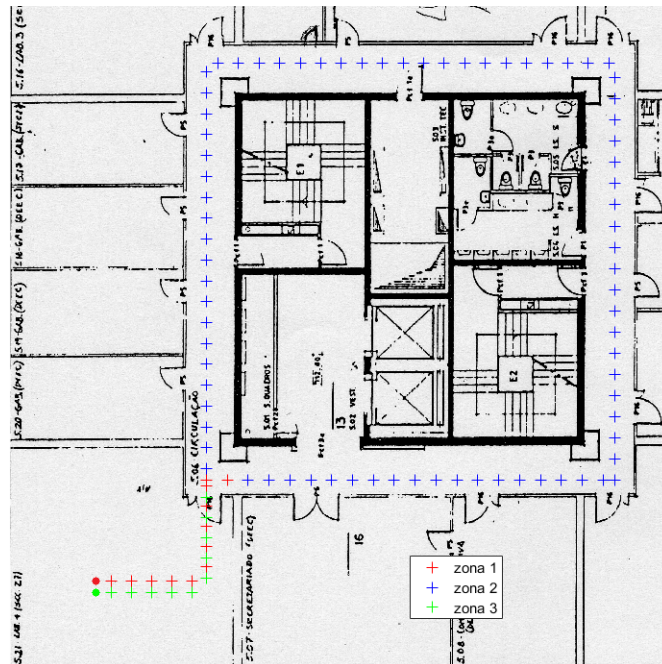


Figura 2: Mapa da trajetória desejada.

O robô começa por se deslocar até ao primeiro ponto e, depois de o "capturar", deve mudar a sua referência para o imediatamente a seguir e assim sucessivamente. O ponto considera-se "capturado" a partir do momento em que o robô se encontra numa vizinhança de raio 40cm dele.

2.3 Ajuste da trajetória

Para o controlo do robô ao longo da trajetória referida anteriormente, será necessário conhecer a posição do robô relativamente ao ponto de referência. A localização do robô com base na sua Odometria, de forma "cega", seria impossível de implementar no mundo real. Pequenos erros na leitura da mesma iriam resultar em desvios enormes entre a posição do robô e a estimativa da mesma, o que resultaria no embate inevitável contra as paredes do corredor.

A abordagem tomada para combater este problema toma como princípio que as leituras da odometria são de alguma confiança. Com isto, a localização do robô poderá ser feita somente com a sua odometria, no entanto o trajeto a percorrer estará sujeito a atualizações. Assim que o robô estiver próximo de uma parede, a trajetória é deslocada no sentido oposto desta, dando ao robô a ideia que está fora do percurso, forçando uma correção. Dado que se está perante uma trajetória quadrada no sentido contra-horário, o ajuste da trajetória pode ser feito como evidenciado pela Figura 3.

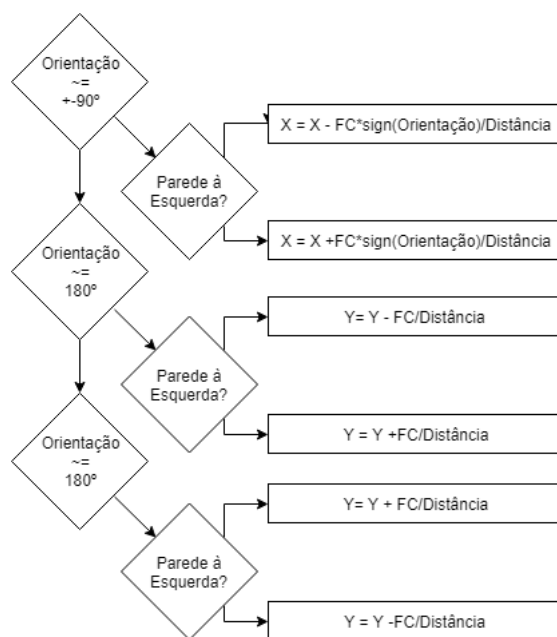


Figura 3: Esquema para a correção da trajetória do robô.

Como é referido na Figura 3, a trajetória é deslocada $\frac{FC}{D}$ mm onde FC é o Fator de Correção, uma constante a impor, e D corresponde à distância do robô à parede. Com isto quanto mais perto da parede o robô estiver maior será a correção. Com FC pode ser ajustado a intensidade da correção consoante a velocidade do robô. Quanto maior esta for, maior terá de ser o valor de FC para serem realizadas correções mais bruscas e evitar a colisão.

As distâncias às paredes serão obtidas através da menor leitura dos sonares nas extremidades do robô como indicado na Figura 4, a negrito.

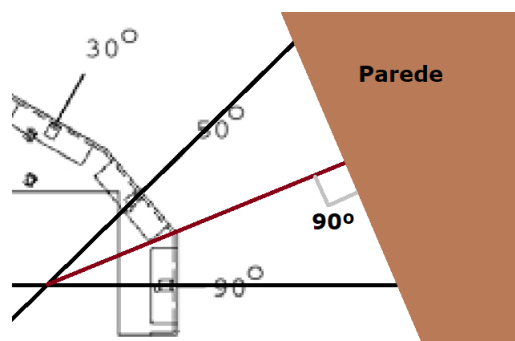


Figura 4: Representação gráfica da leitura dos sonares

No entanto, como os sonares são muito voláteis ao ruído, convém fazer alguma tipo de filtragem. Uma abordagem simples será fazer uma mediana das ultimas X medições dos sonares, assumindo que o robô não altera de forma brusca a sua distância com a parede. Foram escolhidas as 3 últimas medições de forma a conciliar a robustez da estimativa com a rapidez de leitura, visto que os sonares demoram cerca de 0.6 s a realizar uma leitura, como irá ser visto mais à frente.

2.4 Arquitetura

Dos pontos anteriores conclui-se que a navegação do robô se encontra segmentada em 3 partes, controlo do robô, seguido da atualização da sua posição atual (através da odometria) e, por fim, o ajuste da sua trajetória com base na localização do robô em relação às paredes do corredor (através dos sonares).

Será crucial ter em conta o intervalo mínimo de tempo entre execuções de cada tarefa. Com base em testes realizados, chegou-se aos valores presentes na Tabela 1.

Controlo [ms]	Odometria [ms]	Sonares [ms]
400	0	600

Tabela 1: Período mínimo de cada tarefa

De notar que apesar de a odometria poder ser acedida instantaneamente, esta apenas é atualizada periodicamente. Com isto, o funcionamento da arquitetura pode ser descrito no diagrama de PETRI da Figura 5. Os períodos de ambos os *timers* foram definidos de acordo com os valores da Tabela 1.

Como descrito na figura, os temporizadores encontram-se inicialmente em oposição de fase, de forma a evitar conflitos na comunicação com o robô, verificados ao longo da implementação da arquitetura desenvolvida.

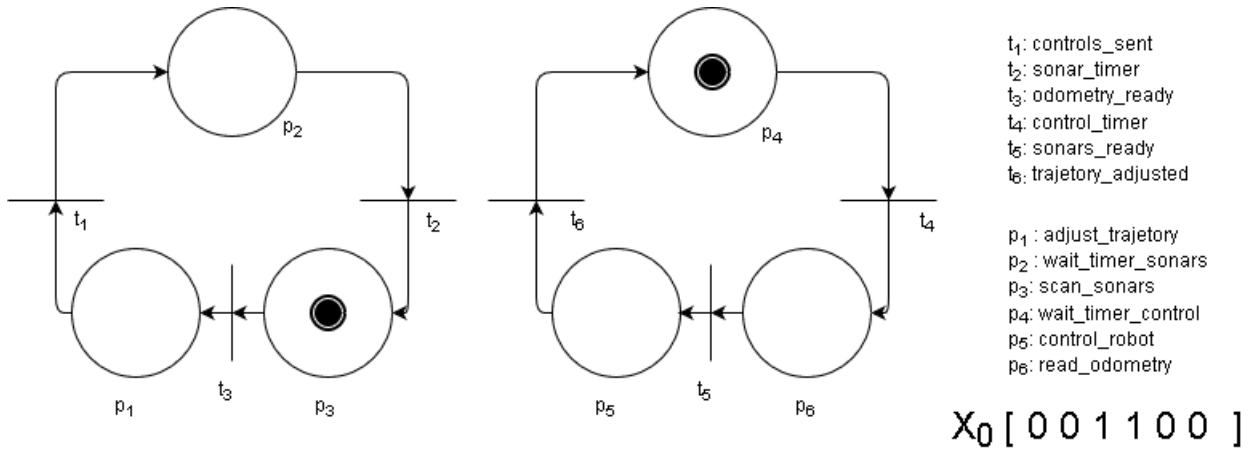


Figura 5: Diagrama PETRI do funcionamento da arquitetura.

2.5 Simulação

Ao longo do programa existem inúmeros parâmetros, aos quais devem ser atribuídos os valores adequados para uma implementação sólida da arquitetura desenhada.

Para acelerar o processo da determinação de valores aceitáveis para os diferentes parâmetros, foi elaborado um ambiente de simulação simples presente na Figura 6.

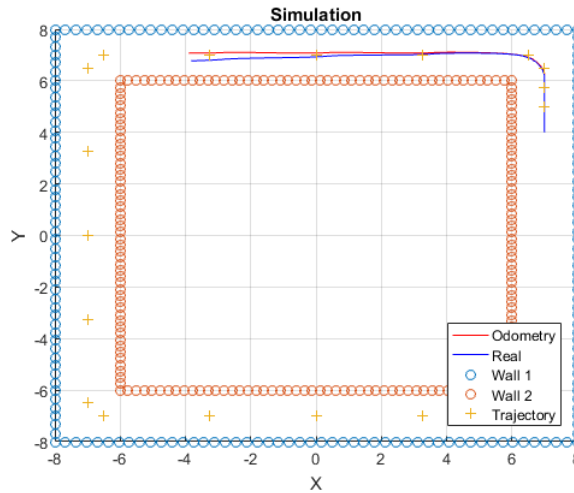


Figura 6: Ambiente de simulação.

Nesta, existem dois robôs em movimento, ambos regidos pelas seguintes expressões

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_k \end{bmatrix} + \Delta t \left(\begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{k-1} + \epsilon \\ \omega_{k-1} + \xi \end{bmatrix} \right), \quad (6)$$

onde Δt corresponde ao intervalo de tempo desde a ultima iteração da simulação e ξ e ϵ representam um erro aditivo Gaussiano com uma determinada média μ e variância σ . Estas permitem simular os desvios entre os valores lidos pela odometria ($\mu = 0$ e $\sigma = 0$, robô a vermelho na Figura 6) e o trajeto real do robô ($\mu \neq 0$ e $\sigma \neq 0$, robô a azul na Figura 6). Tendo em conta ensaios realizados, os valores de $\mu = \frac{\pi}{1500}$ e $\sigma = \frac{\pi}{3000}$, são representativos da situação real.

Para a determinação dos valores dos ganhos K_1 , K_2 e K_3 , foi feita uma abordagem de tentativa e erro ajustando os valores de cada um deles de forma a que o robô seja capaz de seguir o conjunto

de pontos de forma suave, sem oscilações evidentes e com sinais de controlo (v, ω) simples, i.e., sem variações bruscas e amplitudes relativamente baixas.

Com base nestes critérios foram encontrados os valores $K_1 = 2$, $K_2 = 0.1$ e $K_3 = 2$, que produzem para a trajetória da Figura 6 e os sinais de controlo da Figura 7.

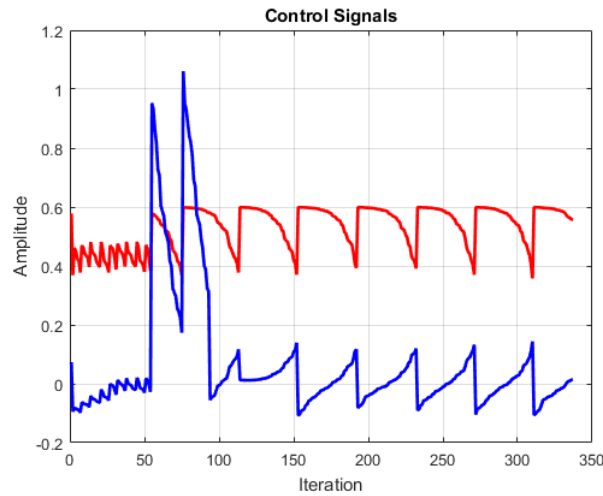


Figura 7: Velocidade do robô a vermelho e a velocidade angular a azul .

Apesar do fator de correção da trajetória ser de certa forma proporcional ao desvio da trajetória real do robô em relação à sua odometria, pode ser determinado em simulação um valor base para este parâmetro. Para tal o cenário foi ligeiramente alterado para uma situação em que o trajeto do robô, ao contrário do caso anterior, não se encontra paralelo às paredes do corredor, simulando assim a situação em que o robô não iniciou o seu movimento na orientação certa.

Uma vez mais, o valor do F.C. foi determinado por uma abordagem de tentativa e erro, pretendendo-se que o trajeto real do robô nunca intersecte uma parede e mantendo uma trajetória suave. Com base nestes critérios foi encontrado o valor de $10mm \text{ correction}^{-1}$ que levou o robô a realizar a trajetória em simulação da Figura 8.

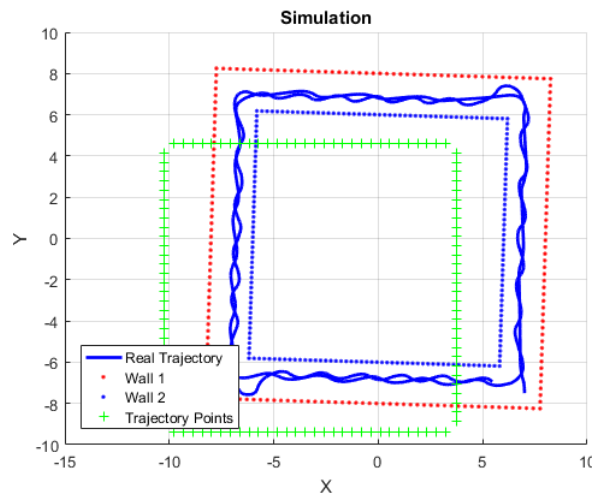


Figura 8: Duas voltas ao cenário de simulação com $F.C. = 10mm \text{ correction}^{-1}$.

Foi também testada no mesmo cenário, um método desenvolvido para a correção da orientação da trajetória através da parametrização das paredes por mínimos quadrados a plicar a inclinação ao trajeto desejado, tendo sido obtidos os resultados da Figura 9.

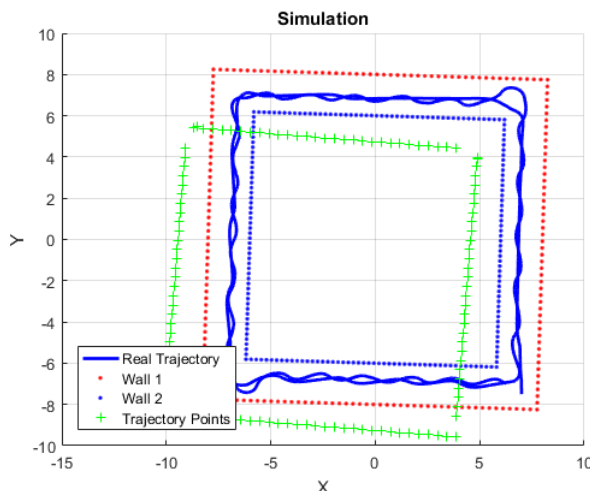


Figura 9: Simulação de 2 voltas ao cenário, com correção da orientação do conjunto dos pontos de referência

Como pode ser observado, os resultados das Figuras 9 e 8 não diferem muito entre si. Visto que os sonares reais são muito mais suscetíveis a erros e o ganho em desempenho é praticamente nulo, a trajetória será apenas corrigida através do deslocamento dos pontos de referência e não será perdido mais tempo neste método.

2.6 Resultados

A arquitetura referida nos pontos anteriores foi implementada de forma a movimentar um robô Pioneer em torno do 5º piso da torre de Eletricidade.

O robô foi colocado num dos cantos do corredor com o objetivo de realizar o maior número de voltas possível no sentido anti-horário.

Na Figura 10 encontra-se o resultado do sucedido, em que o robô foi capaz de dar 2 voltas e $\frac{3}{4}$, tendo o computador que o controlava ficado sem bateria no final do trajeto.

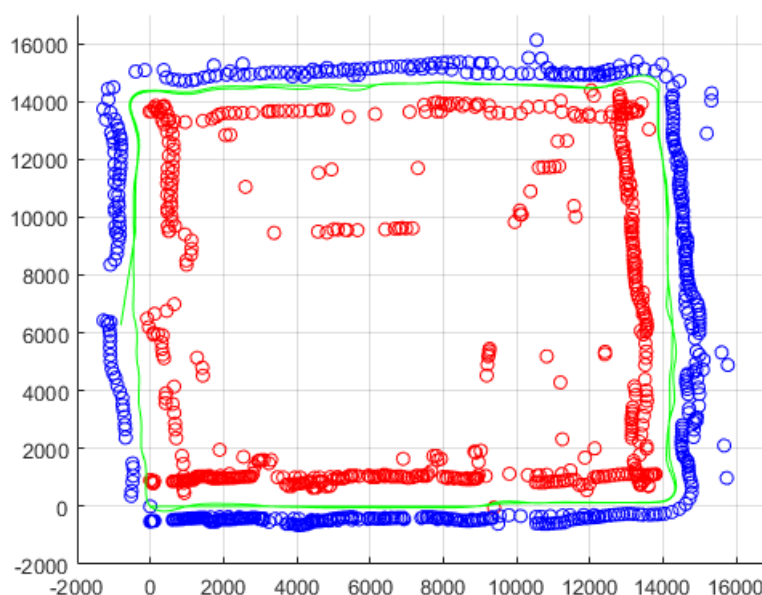


Figura 10: Trajeto efetuado pelo robô a verde, leituras do sonares direitos a azul e sonares esquerdos a vermelho

Como é evidente na figura, o robô executou um movimento consistente e suave tal como foi verificado em simulação, sem colidir com os bancos presentes no corredor ou com as paredes do mesmo. É também visível a disparidade dos valores lidos pelos sonares.

Com base neste resultado, nas simulações efetuadas e em inúmeros ensaios não registados, pode-se concluir que a navegação do robô é à partida robusta e segura.

3 Detecção de portas

3.1 Método

A detecção de portas será feita comparando a posição do robô com a posição conhecida das diferentes portas do corredor. Adicionalmente, através das leituras do LRF será possível localizar o centro da porta e, se necessário, corrigir a posição do robô de forma a que este fique no meio do corredor e centrado com a porta. Com isto o robô roda, encarando a porta de frente e determinando o estado da mesma com a leitura central do LRF.

A detecção e determinação do estado das portas do corredor pode ser resumida no fluxograma da Figura 11.

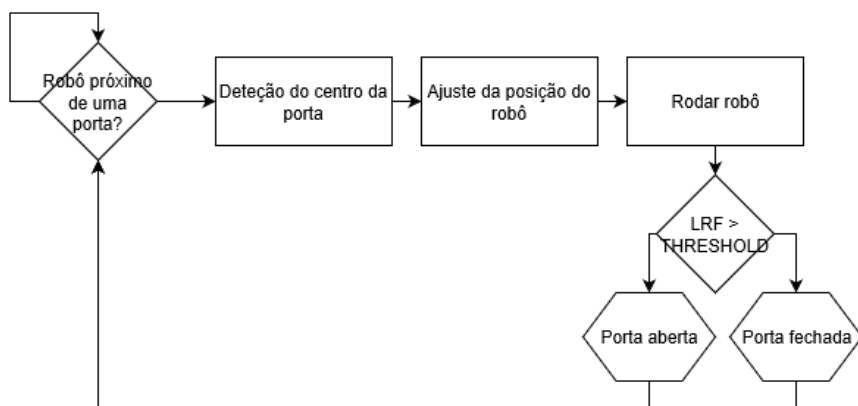


Figura 11: Fluxograma da detecção e determinação do estado das portas em alto nível

3.2 Detecção da proximidade de uma porta

A localização das portas é feita relativamente ao início do corredor onde a mesma se encontra. Esta é então dada sob a forma de distância a que se encontra do ponto comum ao centro do corredor em que a porta se encontra, e o centro do corredor anterior.

Visto que em todos os cantos da torre existe uma porta, a posição de um canto é definida pelo ponto em que o robô encontrou a última porta do corredor, sendo a posição do canto a posição do robô no instante referido. Assim, se a distância já percorrida pelo robô no corredor estiver a 50cm da posição da porta no corredor, esta é detetada.

Ao atualizar o referencial das portas em cada canto, reduz-se o impacto que os desvios na posição do robô vai acumulando na detecção de portas.

A localização de cada porta foi retirada da planta do local, Figura 2, estando estas presentes na Tabela 2.

Tabela 2: Posição de cada porta relativamente ao início do respetivo corredor.

Corredor 1				
Porta	1	2	3	
Distância na planta [cm]	2.3	4.92	7.15	
Distância no corredor [m]	4.6	9.84	14.3	

Corredor 2							
Porta	1	2	3	4	5	6	7
Distância na planta [cm]	1.35	3.4	3.4	4.2	4.87	5.56	7.05
Distância no corredor [m]	2.7	6.8	6.8	8.4	9.74	11.12	14.30

Corredor 3					
Porta	1	2	3	4	5
Distância na planta [cm]	1.56	3.65	4.7	6.13	7.1
Distância no corredor [m]	3.12	7.3	9.4	12.26	14.2

Corredor 4						
Porta	1	2	3	4	5	6
Distância na planta [cm]	1.33	2.73	3.48	4.19	5.64	7.15
Distância no corredor [m]	2.66	5.46	6.96	8.38	11.28	14.3

3.3 Detecção do centro da porta

Para a detecção do centro da porta serão realizadas 3 etapas

1. Transformação da leitura do LRF (coordenadas polares) para coordenadas cartesianas,
2. Parametrização das paredes,
3. Detecção de uma porta e determinar o seu centro

3.3.1 Transformação para coordenadas cartesianas

O LRF é capaz de obter medições para ângulos entre 120° e -120° com uma resolução de aproximadamente 0.36° . Com isto, podem ser reconstruídas as coordenadas dos objetos em torno do robô no seu referencial através de

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = LRF(k) \begin{bmatrix} \cos(\theta_k) \\ \sin(\theta_k) \end{bmatrix} \text{ para } \theta_k = 0.36k, \quad (7)$$

onde $LRF(k)$ corresponde ao valor da k -ésima medida do LRF.

3.3.2 Parametrização de paredes

Das medidas do LRF, será oportuno distinguir as medidas do lado direito do robô das do lado esquerdo de forma a facilitar a diferenciação do lado em que a porta se encontra. Para tal basta seleccionar os dados cujos valores de y sejam superiores a 100mm (corredor esquerdo), ou inferiores a -100mm (corredor direito).

A parede será parametrizada por

$$y_{parede}(x) = \beta_0 + \beta_1 x, \quad (8)$$

através de mínimos quadrados. Pretende-se então descobrir os valores de β_0 e β_1 que melhor descrevam a parede em questão. Este é um problema cuja solução é conhecida e dada por

$$\beta = (X^T X)^{-1} Y \quad (9)$$

$$\beta = [\beta_0 \ \beta_1], \ X_k = [1 \ x_k], \ Y_k = [y_k], \quad (10)$$

A parametrização poderia ser feita com base no conjunto de pontos original. No entanto, nem todos pertencem a uma parede, como tal terá de ser feita uma seleção de pontos em que estes pertençam efetivamente a uma parede.

Para tal será feito um RANSAC (Random Sample Consensus). Este corresponde a um processo iterativo, em que em cada iteração são realizadas as seguintes ações,

1. Escolhem-se 4 pontos (bastariam 2) de forma aleatória do conjunto inicial,
2. Realiza-se a parametrização com estes quatro pontos
3. É calculado o desvio entre cada ponto do conjunto inicial e a reta parametrizada
4. São considerados como pontos da parede aqueles cujo desvio seja menor que **20cm**

São realizadas 200 iterações e no final, é seleccionado o maior conjunto de pontos considerados como pertencentes a uma parede no processo. Com este, é feita uma última parametrização correspondendo à parametrização efetiva da parede, se esta existir.

Na Figura 12 encontram-se a azul os pontos lidos à direita do robô, a verde encontram-se os pontos considerados como pertencentes a uma parede e a vermelho, a reta parametrizada por estes pontos.

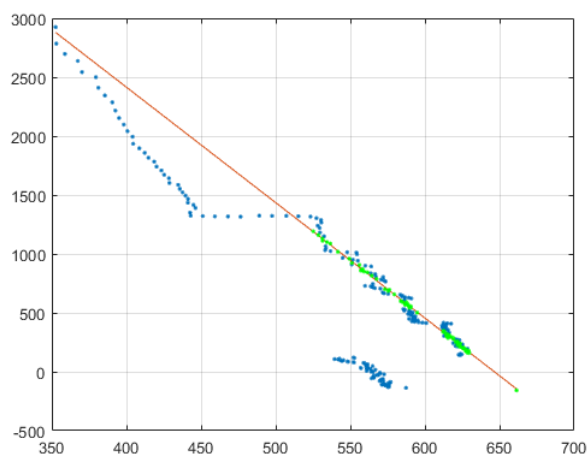


Figura 12: Representação gráfica do procedimento para a parametrização da parede.

De notar que a reta encontrada não corresponde à parametrização da parede em si, mas da porta que nela está contida. Contudo, isto não apresenta problemas para o resultado final visto que o parâmetro que realmente interessa é o β_1 , correspondente à inclinação da parede relativamente ao robô. Adiante irá ser mais evidente o porquê.

3.3.3 Detecção de portas

À reta parametrizada, subtrai-se o conjunto inicial de pontos. O resultado apresenta o desvio entre os pontos da leitura e a reta parametrizada. Na Figura 13 encontra-se o referido desvio para o caso da Figura 12, em que uma porta se encontra nas proximidades do lado direito do robô.

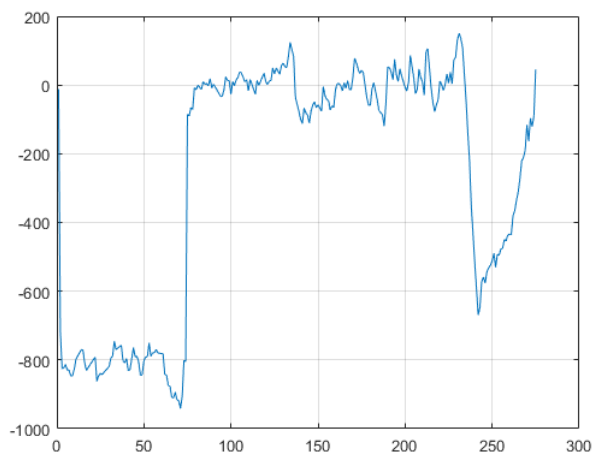


Figura 13: Desvio do da leitura do LRF em relação à reta parametrizada

Ora como foi verificado na Figura 12, a porta irá corresponder ao conjunto de pontos cujo desvio será próximo de 0. De forma a vincar a presença de uma porta na Figura anterior, o resultado é suavizado por um filtro da mediana espacial de raio 50, resultando no gráfico da Figura 14

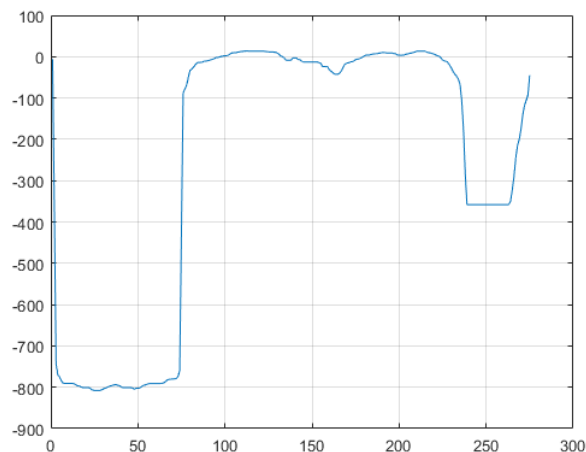


Figura 14: Desvio da leitura do LRF, suavizado com um filtro da mediana.

Para a detecção da porta, podem ser descobertas as suas extremidades, evidenciadas no gráfico anterior pelas duas discontinuidades observadas. Para as captar, o resultado é então diferenciado obtendo-se o resultado da Figura 15

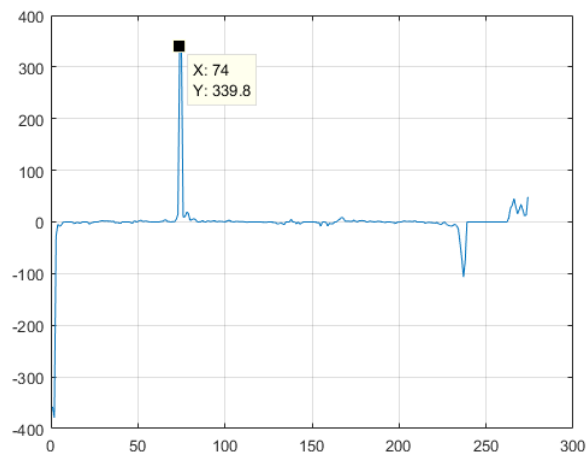


Figura 15: Desvio da leitura do LRF, suavizado e diferenciado.

Com isto pode concluir-se que, no gráfico anterior, uma porta terá um pico com amplitude próxima dos 350, seguido de um vale com amplitude próxima de -100.

Para a detecção de uma porta, pretende-se então detetar este padrão. Ora uma forma simples de o fazer será detetar o pico mais próximo de 350 e o vale mais próximo de -100. O pico corresponderá ao início da porta e o vale ao final da mesma, caso a distância entre estes dois

pontos for $100 \pm 40\text{cm}$ (comprimento médio das portas do corredor mais uma tolerância) então considera-se que os pontos do pico e vale encontrados pertencem a uma porta.

Para portas do lado esquerdo, o procedimento é o mesmo, no entanto, o padrão a detetar será um vale de amplitude -350 seguido por um pico de 100.

3.3.4 Determinar o centro da porta

Do ponto anterior são retirados os extremos da porta, nas suas coordenadas cartesianas. O centro da porta será então dado, no referencial do robô por

$$x_{centro} = \frac{x_1 + x_2}{2}, \quad y_{centro} = \frac{y_1 + y_2}{2} \quad (11)$$

Como pode ser verificado na Figura 16.



Figura 16: Leitura do LRF a azul e o centro da porta detetada assinalado com um circulo verde.

Assim, a distância que o robô deverá deslocar-se de forma a se centrar com uma porta detetada à direita será dada por

$$d = \|(x_{centro}, y_{centro})\|_1 - d_{direita}, \quad (12)$$

onde $d_{direita}$ corresponde à mediana das 3 últimas medições do sonar direito do robô. Idem, para portas do lado esquerdo

$$d = \|(x_{centro}, y_{centro})\|_1 - d_{esquerda}. \quad (13)$$

3.4 Determinação do estado da porta

Para determinar o estado da porta será necessário primeiro colocar o robô centrado com ela e de seguida rodá-lo de forma a que a encare de frente.

Tanto para o deslocamento do robô tal como a sua rotação, é assumido que a orientação do robô é paralela às paredes do corredor. Com isto, para deslocar o robô, impõe-se uma velocidade constante de 20 cm por segundo até que a distância percorrida pelo robô obtida através da odometria do mesmo, exceda a distância referida no ponto anterior.

Quanto à rotação é aplicado um procedimento semelhante, rodando o robô com velocidade angular constante de $\pm 15^\circ \text{ seg}^{-1}$ até que a rotação exceda um ângulo de $\pm 90^\circ$ dependendo do lado onde a porta foi detetada

Visto que o robô se encontra já centrado com a porta e a encará-la de frente, basta agora testar a leitura do LRF a 0° e compará-la com um determinado *threshold*. Se a leitura for superior a 1500mm ou inferior a 10mm (pontos fora do alcance máximo do LRF), a porta é classificada como aberta. Caso contrário é classificada como fechada.

3.5 Resultados

Foi testado o algoritmo para a deteção de portas apenas com base nas leituras do LRF de forma offline. Para tal, o robô foi colocado no corredor, deslocando-se através do método referido anteriormente enquanto as leituras do LRF estavam a ser guardadas de 1 em 1 segundo. Às leituras foi aplicado o procedimento descrito anteriormente podendo o resultado ser conferido no seguinte **video** onde foi testado o método para o lado direito do corredor.

O algoritmo construído foi capaz de detetar o centro de cada porta, exceto em casos complicados como cantos do corredor, situações com alunos a passar ao lado do robô ou em casos em que o robô se encontra a ajustar a sua trajetória como é visível no vídeo. Nestes casos são observados um elevado número de falsos positivos.

A robustez do procedimento poderia ser altamente melhorada caso se abandonasse a ideia da deteção através dos *thresholds* referidos, e se adotasse por exemplo um método de classificação automática, como SVM. Devido ao tempo e recursos necessários, não foi possível trabalhar tal abordagem.

A deteção do estado da porta será discutida no ponto 5.

4 Solução Geral

4.1 Arquitetura

Os diferentes módulos expostos anteriormente (controlo, ajuste e deteção de portas) de certa forma trabalham independentemente uns dos outros. Como tal, basta agregar os diferentes módulos numa arquitetura conjunta.

Adicionalmente, terá de ser feita uma distinção entre a maneira como o robô se movimenta dentro do laboratório e a maneira como o faz no corredor. No corredor o robô pode movimentar-se de forma mais rápida, aumentando o valor de v_{max} . Com o aumento da velocidade, o valor do fator de correção para o ajuste da trajetória terá de ser aumentado.

Por fim, o caminho a percorrer dentro do laboratório é bastante mais apertado que os corredores da torre. Como tal os *thresholds* esquerdo e direito que accionam a correção da trajetória terão de ser ajustados. A integração dos 3 módulos encontra-se explicitada na Figura 17.

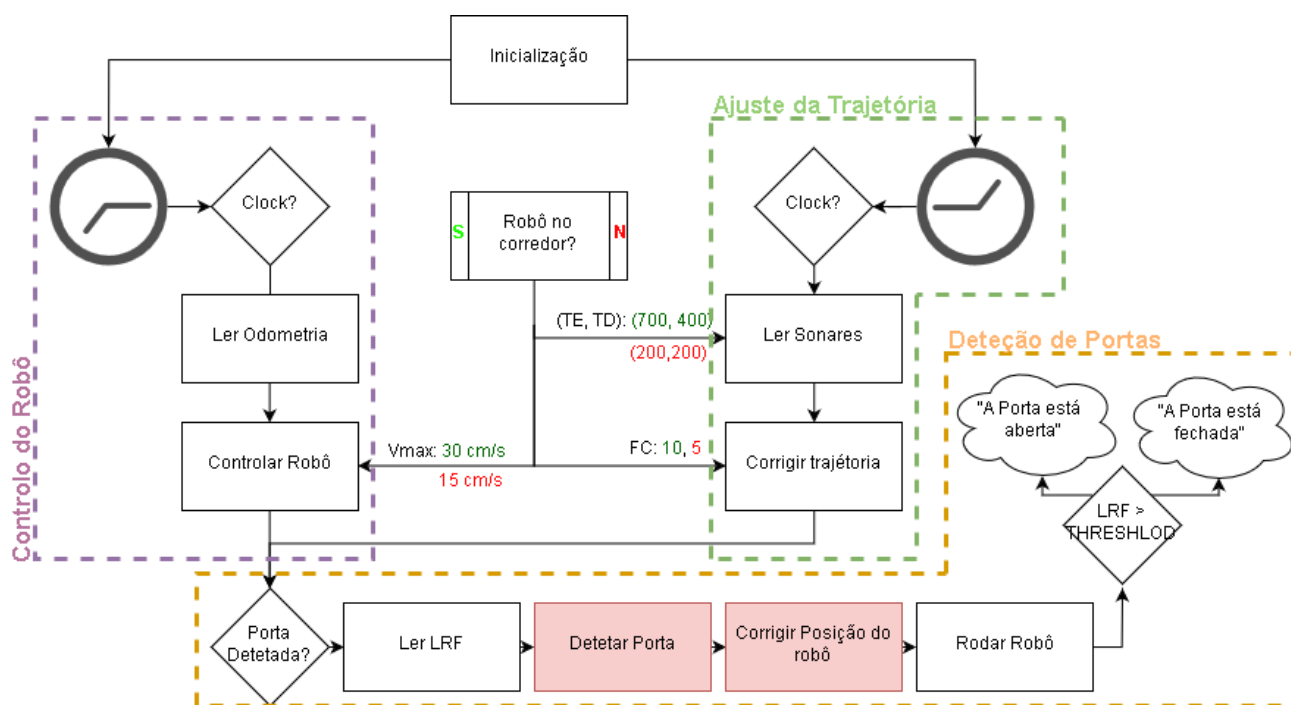


Figura 17: Esquema com a arquitetura da solução global.

5 Discussão

5.1 Navegação do Robô

No que toca a esta secção, o principal objetivo foi alcançado, que era evitar que o robô embatesse em algum obstáculo durante o percurso, mais propriamente nas paredes e nos bancos do corredor. Contudo, a navegação não foi perfeita, por não ter sido totalmente retilínea e estável.

Isto deve-se maioritariamente à maneira como são feitas as correções na trajetória do robô como também foi verificado em simulação. Contudo, as oscilações verificadas não são violentas ao ponto de serem perspetíveis a olho em grande parte do trajeto mantendo o robô uma orientação paralela às paredes do robô no decorrer do seu percurso.

Em geral, a navegação do robô revelou resultados positivos. O robô foi capaz de sair do laboratório, realizar uma volta ao corredor e voltar ao interior do laboratório inúmeras vezes de forma segura sem indícios de possíveis falhas. Estes incluem os 3 ensaios nos quais o robô pára e roda a cada porta encontrada. Com isto pode ser afirmado que a navegação do robô revela ser robusta como também versátil, visto que através de pequenas correções no valor do fator de correção, foi possível cumprir o objetivo da navegação para diferentes robôs.

5.2 Detecção das portas

Como pode ser visto na Figura 17, os módulos referentes ao mecanismo responsável pelo enquadramento do robô com a porta encontram-se a vermelho de forma a explicitar que estes não foram usados na implementação final. Como foi referido, estes dependem de valores de thresholds que devem ser dimensionados à priori através de tentativa e erro. Este processo iria ser demorado e face ao tempo em que o sensor LRF esteve disponível durante o decorrer do projeto foi tomada a decisão de excluir o referido módulo da restante implementação.

Em relação ao segundo objetivo deste trabalho, a deteção de portas, pode-se dizer que o resultado foi bastante satisfatório com o uso da solução apresentada anteriormente. De facto, o robô em 3 voltas completas descritas foi capaz de detetar todas as portas do corredor e acertar se estava fechada ou aberta em todos os casos. No entanto, para o caso de portas entreabertas a classificação do robô encontra-se indefinida. Como tal neste caso o robô errou 100% das vezes, classificando a porta como aberta ou fechada e não como entreaberta.

Mesmo não tendo sido implementado o mecanismo capaz do enquadramento com a porta com base nas leituras do LRF, o robô foi capaz de parar sempre muito próximo do centro das portas devido não só a robustez da sua navegação mantendo-se sempre quase paralelo às paredes, mas também devido à forma de como é detetada a porta com base na posição atual do robô.

5.3 Geral

Com base nos pontos anteriores, pode ser afirmado que a solução cumpriu todas as expectativas impostas de forma robusta e segura, salvo a exceção da deteção de portas entreabertas, para a qual não houve tempo de implementar.

Referências

- [1] João Silva Sequeira. "Rob3645111326 Course viewgraphs", Spring 2019
- [2] Jorge S. Marques. "Machine Learning Slides", October 18, 2018
- [3] José Santos-Victor. "Image Processing and Computer Vision Slides", October 2018.
- [4] Yoichi Okubo, Cang Y, Johann Borenstein. "Characterization of the Hokuyo URG-04LX Laser Range Finder for Mobile Robot Obstacle Navigation", 2009