

Sistemas Robóticos e Robótica Móvel – 2015/2016

Navegação de Robôs Móveis Utilização de Campos de
Potencial para Navegação de Robôs Móveis.

Labwork - 5

João Toscano Ferreira

2010144652

José Pedro Medeiros

2010129934

Introdução

Com este trabalho pretende-se dar à plataforma móvel a capacidade de navegar de um ponto inicial a um ponto objectivo, percorrendo um mapa que poderá incluir obstáculos. Para se atingir este objectivo foram utilizados campos de pontencial.

Os campos de potencial são utilizados em robótica móvel para evitar que as plataformas móveis colidam com obstáculos. O princípio de funcionamento destes campos é simples, no local do ponto objectivo é colocada uma carga atractiva que gera um campo de potencial que vai atrair a plataforma móvel, por outro lado no locais do mapa em que são detectados obstáculos (durante o percurso) é colocada uma carga repulsiva que gera um campo de potencial que irá repelir a plataforma.

Finalmente, somando os dois campos de pontencial (repulsivo e atractivo) que ajudará a plataforma a navegar afastando-a dos obstáculos e aproximando-a para o ponto objectivo.

1 Execução do Simulador - GUI

Para correr o simulador, o utilizador deve correr o comando ***Simulador*** no Matlab, este comando vai correr a interface gráfica por nós desenvolvida da qual o código correspondente se encontra no ficheiro *Simulador.m*.

1.1 Simulação da plataforma

A interface gráfica sofreu alterações relativamente ao trabalho anterior dado que foi introduzida navegação ponto a ponto e ainda que foram introduzidos os campos de potencial. É apresentada a figura 1 que ilustra o novo ambiente para este trabalho, incluindo o novo

mapa de campos de potencial.

O utilizador deverá escolher entre navegação reactiva ou navegação ponto a ponto. No caso de escolha de navegação ponto a ponto, o utilizador deverá inicialmente clicar em - **iniciar navegação**, posteriormente escolher o ponto objectivo (**pressionando ENTER**) e de seguida deve clicar no mapa na posição desejada para o início da navegação, clicando uma segunda vez para a orientação pretendida e a navegação irá começar automaticamente.

No caso de o utilizador escolher navegação reactiva para que este possa escolher a posição e orientação do robot basta escolher no simulador - **Iniciar Simulação -> Escolher ponto e orientação**. e de seguida clicar no mapa na posição desejada, clicando uma segunda vez para a orientação pretendida, o utilizador pode também optar por uma posição predefinida. É dada a oportunidade ao utilizador de definir o tempo de simulação desejado.

Para iniciar a simulação basta ao utilizador clicar no botão - **iniciar comportamento reactivo**.

Isto pode ser visualizado na figura 1.

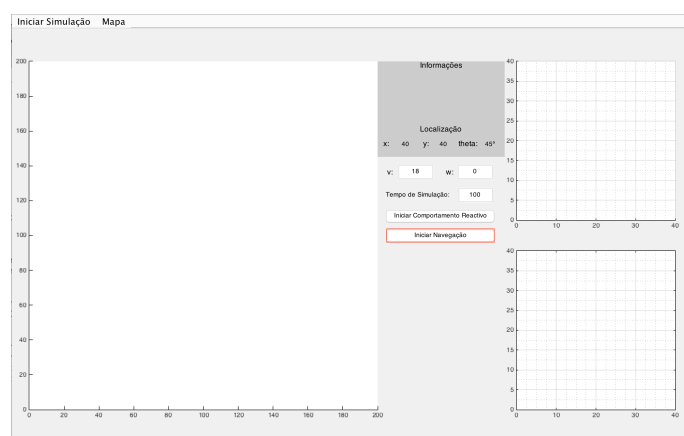


Figura 1: Captura do Simulador no momento de escolha da posição e orientação inicial.

1.2 Definição do mapa de obstáculos

O simulador oferece ao utilizador possibilidade de desenhar obstáculos, criar e guardar mapas assim como permite a possibilidade de os carregar no futuro.

Para que o utilizador possa criar obstáculos este deve escolher - **Mapa -> Criar obstáculo**. e de seguida deve clicar no mapa com o botão esquerdo do rato por forma a definir os diversos vértices do polígono e de seguida deverá clicar com o botão direito do rato definindo o último vértice e finalizando assim o preenchimento do obstáculo. Para carregar um mapa o utilizador selecciona - **Mapa -> Carregar mapa**. , o mesmo é válido para guardar o mapa devendo o utilizador seleccionar - **Mapa -> Guardar mapa**. e ainda para quando o utilizador pretende limpar o mapa - **Mapa -> Limpar mapa**.

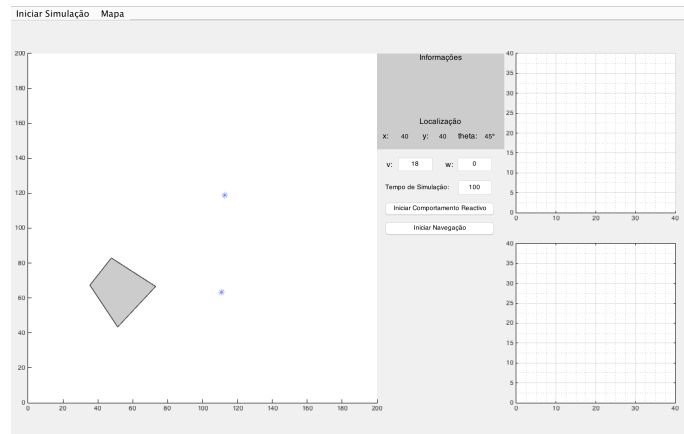


Figura 2: Captura do simulador no momento de criar um mapa.

1.3 Implementação prática:

Para este labwork é pretendido que a plataforma (que mantém as características dos labworks anteriores) seja capaz de se deslocar de um ponto inicial para um ponto final sem conhecimento prévio do mapa (independentemente da existência de obstáculos), a plataforma apenas não conseguirá atingir o seu objectivo caso exista um mínimo local. Para que a plataforma consiga chegar ao objectivo é criado um mapa de pontencial atractivo em função do ponto destino e um mapa de pontencial repulsivo em função da presença de obstáculos, o mapa de ocupação que utiliza o algoritmo desenvolvido por **Sebastian Thrun** continua a ser utilizado na detecção de obstáculos.

Posteriormente ao atingir o ponto objectivo a plataforma deverá regressar ao ponto inicial, agora com conhecimento do mapa já percorrido.

A plataforma deve como objectivo final navegar entre diversos pontos.

1.3.1 Funções:

Com a introdução dos mapas de atracção e repulsão e com a introdução de um novo tipo de navegação houve necessidade de criar e alterar algumas funções novas.

- **actualizaGrelhaPotencialObstaculos.m:** – Esta função tem como parâmetros de entrada (**matrizOcupado**, **grelhaPotencialObstaculosAnterior**).
 - **matrizOcupado** - matriz de ocupação com o mapa de ocupação determinado com o algoritmo de **Sebastian Thrun**.
 - **grelhaPotencialObstaculosAnterior** - Grelha que inclui o mapa com o pontencial reactivo anterior.

Esta função, actualizada, vai calcular o pontencial repulsivo com base nos valores de anteriores e nos valores da **matriz ocupado** como é descrito nas fórmulas apresentadas em 3.

- **criaGrelhaPotencial.m:** – Esta função cria as grelhas de pontencial repulsivo.

- **criaGrelhaPotencialDestino.m:** – Esta função cria as grelhas de potencial atractivo.
- **desenhaGrelhaPotencial.m:** – Esta função desenha a grelha de potencial final (soma da grelha de potencial atractivo com a grelha de potencial repulsivo).
- **calcParamMovimento.m:** – Esta função tem como parâmetros de entrada (**matrizPotencial,x,y,theta**) .
 - **matrizPotencial** - matriz com a soma das grelhas de potencial atractivo e repulsivo.
 - **x,y** - Coordenadas x,y da plataforma.
 - **theta** - Orientação da plataforma.

Esta função vai utilizar a informação da posição da plataforma para determinar o ângulo de rotação em relação ao destino. Após este cálculo esta função desloca a plataforma em direcção ao ponto objectivo.

(**matrizOcupadoAnterior,theta,rho,m,n,R**)

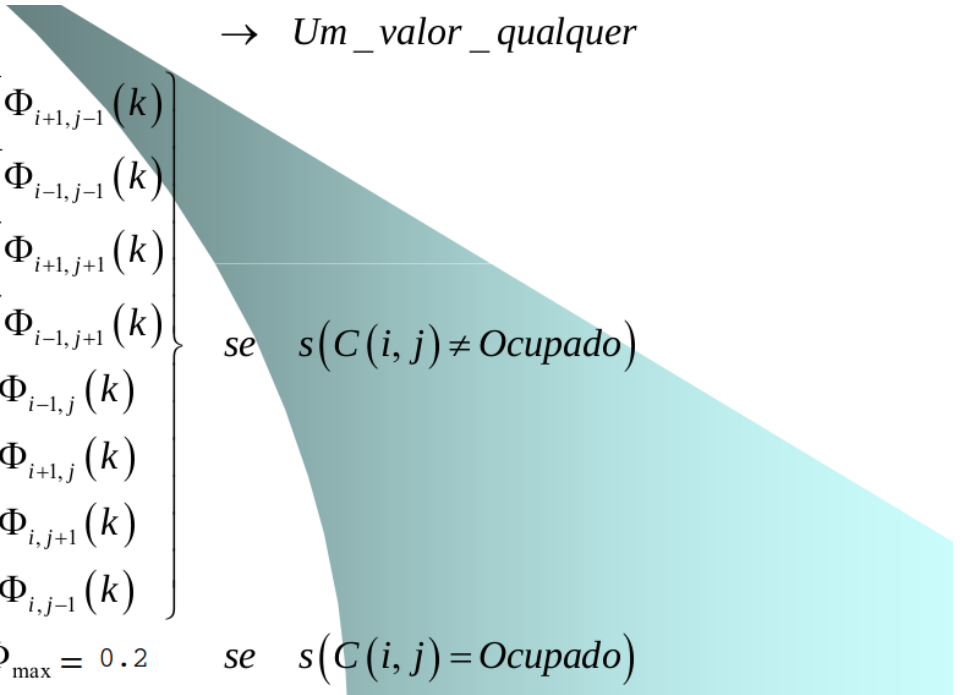
$$\begin{aligned}
 & \Phi_{ij}(t=0) \rightarrow \text{Um_valor_qualquer} \\
 & \Phi_{ij}(k+1) = \max \left\{ \begin{array}{l} \alpha^{\sqrt{2}} \Phi_{i+1,j-1}(k) \\ \alpha^{\sqrt{2}} \Phi_{i-1,j-1}(k) \\ \alpha^{\sqrt{2}} \Phi_{i+1,j+1}(k) \\ \alpha^{\sqrt{2}} \Phi_{i-1,j+1}(k) \\ \alpha \Phi_{i-1,j}(k) \\ \alpha \Phi_{i+1,j}(k) \\ \alpha \Phi_{i,j+1}(k) \\ \alpha \Phi_{i,j-1}(k) \end{array} \right\} \quad \text{se } s(C(i,j) \neq \text{Ocupado}) \\
 & \Phi_{ij}(k+1) = \Phi_{\max} = 0.2 \quad \text{se } s(C(i,j) = \text{Ocupado})
 \end{aligned}$$


Figura 3: Cálculo da grelha de potencial repulsivo .

1.4 Análise de Resultados:

O simulador foi testado em diferentes mapas.

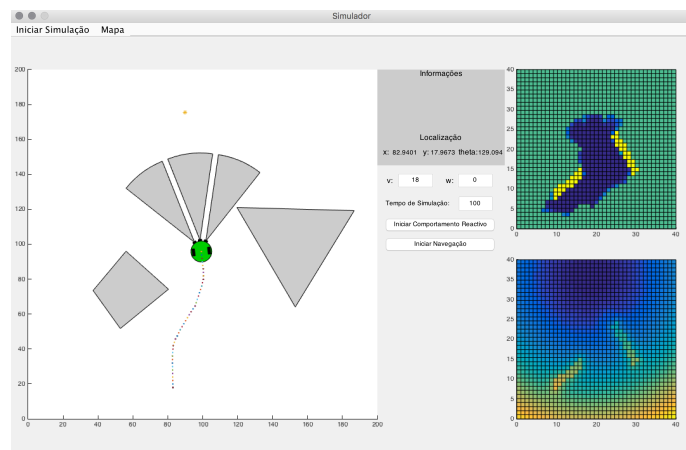


Figura 4: Captura do simulador no durante o percurso . Mapa de ocupação e de potencial à direita.

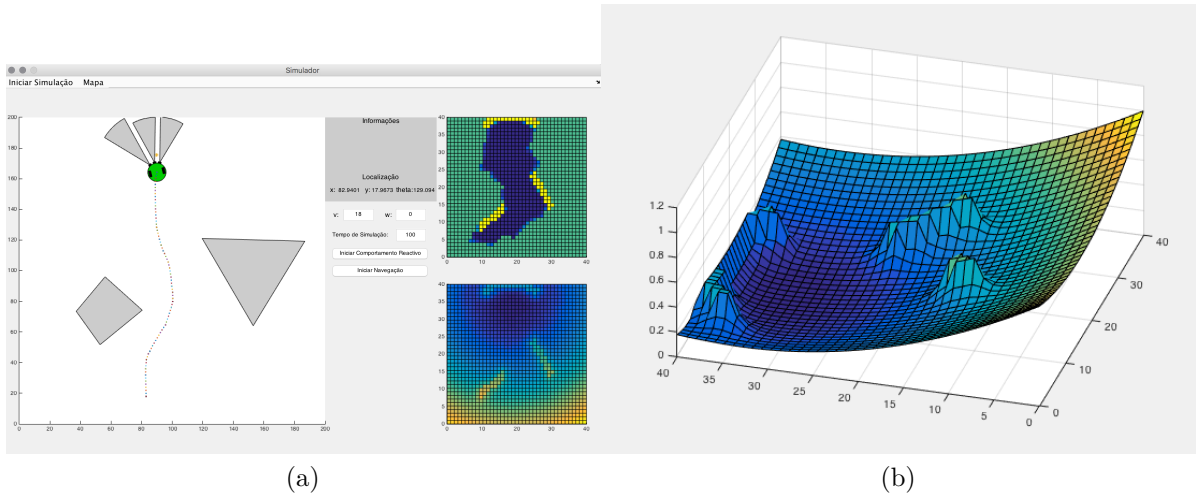


Figura 5: Captura do final da simulação (esquerda) e do mapa de potencial final .

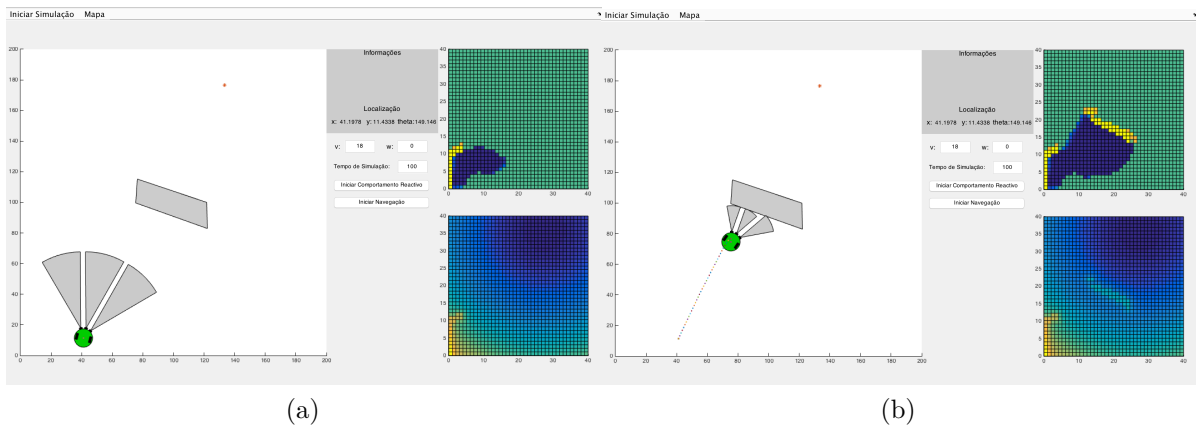


Figura 6: Simulação com um mapa diferente.

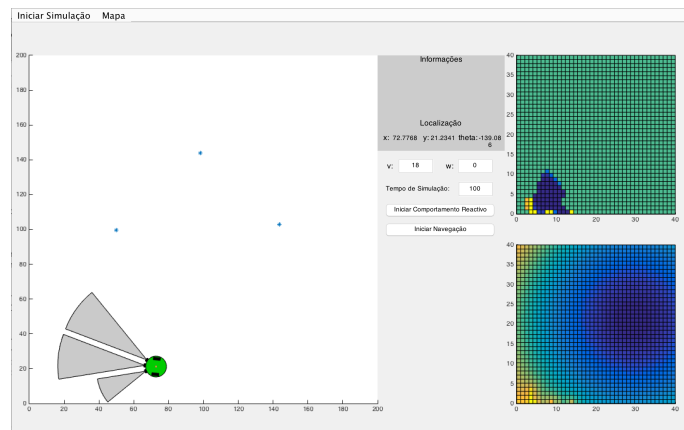


Figura 7: Navegação ponto a ponto.

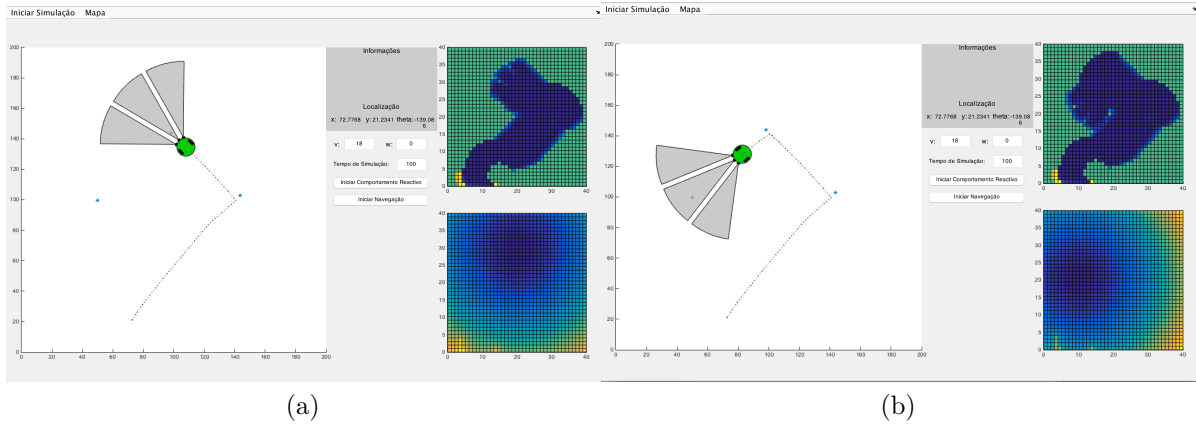


Figura 8: Navegação ponto a ponto .

Verifica-se consegue convergir para diferentes pontos independentemente da existência de obstáculos desde que não existam mínimos locais.