

Fundamentos de Inteligencia Artificial

Relatório projeto TP1

Estudante	João Moreira	joaomoreira@student.dei.uc.pt	2020230563	TP1
Estudante	Tomás Pinto	tomas Pinto@student.dei.uc.pt	2020224069	TP1
Professor Dr.	Nuno Lourenço			

1 Introdução

Os veículos de Braitenberg são conhecidos pela sua simplicidade: um robô com um sensor¹ ligado ao motor de uma roda. Com esta mecânica simples, facilmente obtemos robôs com complexidades incríveis.

Braitenberg, no seu livro propôs quatorze veículos, dos quais quatro mais famosos:

1. *Fear* - Muda de direção e aumenta a sua velocidade, quando deteta a luz. Transmite a sensação de medo da luz.
2. *Aggressor* - Vai em direção à luz e aumenta consoante a sua proximidade. Transmite a sensação de agressividade sobre a luz.
3. *Explorer* - Quando deteta a luz, diminui a velocidade e muda a sua direção.
4. *Lover* - Quando deteta a luz, diminui a velocidade e mantém a sua direção.

2 Meta 1 – Sense It

Foi-nos disponibilizado "prefabs" que continham um objeto (fonte de luz) e os seus sensores já implementados. A nossa tarefa consistiu em explorar e melhorar as capacidades dos veículos com objetivo de permitir a deteção de outros veículos e obstáculos (blocos), para além das luzes.

2.1 Luz

Começamos por modificar o veículo inicial, do tipo *Fear*, que se afastava da luz para um comportamento do tipo *Aggressor*. Para isto trocamos os sensores lineares de deteção de luz do veículo, ou seja, ligamos o sensor esquerdo à roda direita e vice versa.

2.2 Veículos

Seguidamente passamos a fase de interação entre veículos, em que o objetivo era fazer com que um veículo seguisse outro, sendo assim do tipo *Lover*. Para isto tivemos de desenvolver o script *CarDetectorScript.cs*, semelhante ao script de deteção de luz, que permite a deteção de veículos que tenham a tag "CarToFollow", interagindo com o que se encontrar mais próximo. Para o cálculo da energia usamos:

$$Energy = \frac{1}{distance(closestCar, car) + 1}$$

No *CarDetectorLinearScript* invertemos a energia com:

$$Energy_{final} = 1 - Energy_{initial}$$

¹Normalmente sensores de luz são os mais citados. No entanto, trabalhos com narizes eletrónicos têm sido realizados.

para que o carro tenha mais energia quando mais distante do veículo que segue.

2.3 Blocos

Para expandir as capacidades do nosso veículo, implementamos um *explorer*. Desta forma, o carro irá explorar todo o espaço, quando detetar um bloco a sua velocidade baixará e desvia-se do objeto.

A sua implementação está nas classes *BlockDetectorScript* e *BlockDetectorLinearScript*. Para o calculo da energia usamos:

$$Energy = \frac{1}{distance(closestBlock, car)}$$

Vale ter em atenção que a energia é calculada com o bloco mais próximo ao carro.

No *BlockDetectorLinearScript* invertemos novamente a energia com:

$$Energy_{final} = 1 - Energy_{initial}$$

para que o carro tenha mais energia quanto mais distante do bloco.

3 Meta 2 - Tune it & Test it

Nesta fase do projeto tivemos como objetivo fazer com que os veículos descrevessem algumas trajetórias definidas, sendo estas um círculo, uma elipse e um infinito. Para realizar o objetivo tivemos de desenvolver duas funções de ativação, Linear e Gaussiana.

3.1 Funções de ativação

Tivemos fornecida, no código disponibilizado, a função de ativação linear pronta a utilizar. Devido ao facto desta função nos impor algumas restrições e impossibilitar a implementação de vários comportamentos, fomos motivados a implementar a função de ativação gaussiana e para ambas implementar a possibilidade de possuir limiar (threshold) de ativação mínimo e máximo, bem como limite superior e inferior.

3.1.1 Linear

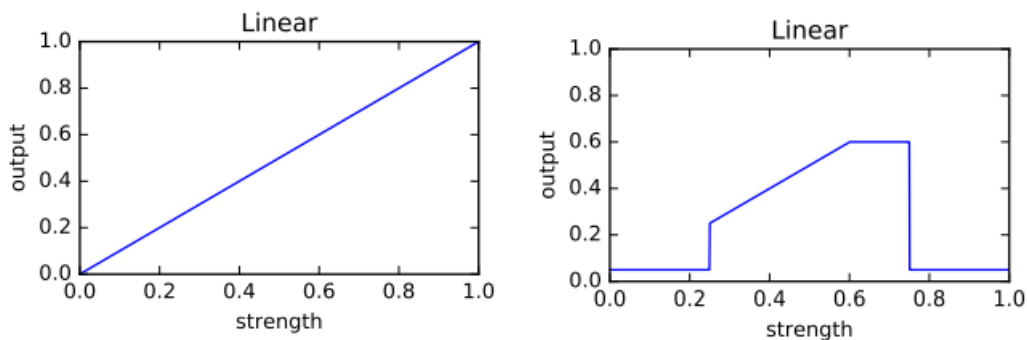


Figura 1: Função de ativação linear

3.1.2 Gaussiana

A função de ativação gaussiana é dada pela seguinte formula:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

São recebidos como parâmetros a energia x , a média σ e também o desvio padrão μ . Quanto maior for o valor do desvio padrão mais larga a curva da função irá ser, a média localiza o pico da função.

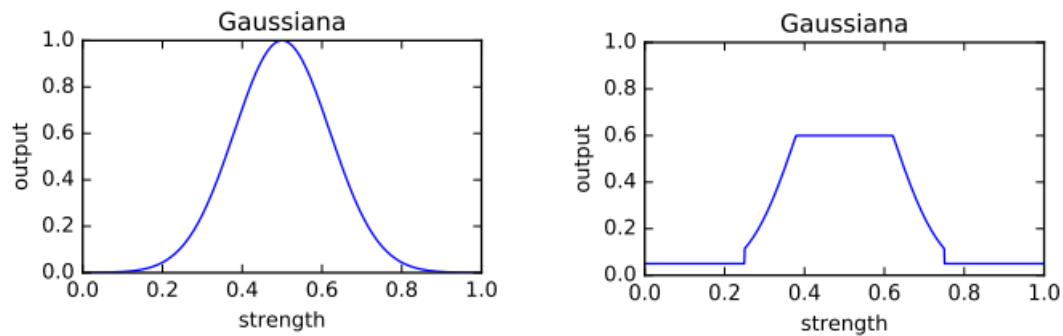


Figura 2: Função de ativação gaussiana

3.2 Círculo

Para que o carro descreva o movimento de um círculo, a roda exterior face à luz tem de ter mais velocidade do que a roda interior. Visto que a distância dos sensores ao centro da luz será sempre a mesma, os valores de *output* vão manter-se constantes, o sensor do lado interno está ligado à roda externa e vice-versa, comportamento *lover*. Isto porque o sensor interno terá maior valor de *output* e por sua vez a roda externa terá mais energia.

Os nossos valores de partida, foram os valores do enunciado: $\sigma = 0.12$ e $\mu = 0.5$.

Exp	Sensor	Média	Desvio Padrão	Velocidade	Funcionou
1	Direita	0.5	0.12	5	Sim
	Esquerda	0.5	0.12	5	
2	Direita	0.7	0.12	5	Sim
	Esquerda	0.7	0.12	5	
3	Direita	0.8	0.12	5	Não
	Esquerda	0.8	0.12	5	
4	Direita	0.7	0.2	5	Sim
	Esquerda	0.7	0.2	5	
5	Direita	0.7	0.3	5	Não
	Esquerda	0.7	0.3	5	

Tabela 1: Testes realizados para o círculo.

Comportamentos:

1. Descreve um círculo no sentido contrário ao que estávamos à espera;
2. Vai em direção à luz. Quando foge dá a volta lentamente e descreve o círculo;
3. Não sai do sitio;
4. Tem o mesmo comportamento do que na exp 2;
5. Sai da rota logo ao iniciar;

Logo no primeiro teste, confirmamos que a nossa intuição sobre a lógica dos sensores estava correta. Para além disso, conseguimos ficar perto daquilo que se pretendia.

3.3 Infinito

Agora que temos o círculo a funcionar, para fazer o infinito o raciocínio é semelhante. No entanto, desta vez queremos que quando o carro acabe de contornar a luz ele sinta atração pela outra luz e siga na sua direção.

Como no círculo, começamos os testes a partir dos valores do enunciado.

Exp	Sensor	MinY	MaxY	Média	Desvio Padrão	Velocidade	Funcionou
1	Direita	None	None	0.7	0.3	15	Não
	Esquerda	None	None	0.7	0.3	15	
2	Direita	None	None	0.5	0.12	15	Não
	Esquerda	None	None	0.5	0.12	15	
3	Direita	0	0.5	0.5	0.12	15	Não
	Esquerda	0	0.5	0.5	0.12	15	
4	Direita	0	0.7	0.5	0.12	15	Não
	Esquerda	0	0.7	0.5	0.12	15	
5	Direita	0	1	0.5	0.12	15	Sim
	Esquerda	0	1	0.5	0.12	15	
6	Direita	0	1.2	0.5	0.12	15	Sim
	Esquerda	0	1.2	0.5	0.12	15	
7	Direita	0	1	0.09	0.47	15	Sim
	Esquerda	0	1	0.09	0.47	15	
8	Direita	0	1.2	0.7	0.3	15	Não
	Esquerda	0	1.2	0.7	0.3	15	
9	Direita	0	1	0.7	0.3	15	Não
	Esquerda	0	1	0.7	0.3	15	
10	Direita	0	1	0.5	0.3	15	Não
	Esquerda	0	1	0.5	0.3	15	

Tabela 2: Testes realizados para o infinito.

Comportamentos:

1. Contorna bem a primeira luz, dá uma volta na segunda e volta na diagonal contrária.
2. Anda em zig-zag entre as luzes.
3. Contorna a primeira luz bem e depois vai diretamente para a outra luz. Depois de fazer um zig-zag na segunda luz, contorna perfeitamente e tem o mesmo comportamento para a primeira luz.
4. Contorna a primeira luz perfeitamente e passa muito perto da segunda luz. Aqui, como está muito perto contorna a segunda luz com alguma distorção e este comportamento começa a repetir-se.
5. Faz o trajeto correto. No entanto, realiza-o com algum ruído.
6. Faz o trajeto corretamente.
7. Faz o trajeto corretamente, o primeiro momento da curva não faz com a velocidade suave.
8. Contorna bem a primeira luz, contorna a segunda luz com alguma turbulência e volta na diagonal contrária.

9. Contorna bem a primeira luz, contorna a segunda luz com muita turbulência e volta na diagonal contrária.

10. Fica preso na primeira luz em círculos.

Neste comportamento, ao fim de alguns testes conseguimos ver que o carro não saía da primeira luz, porque estava com demasiada velocidade na roda externa. Sendo assim, tentamos limitar a velocidade máxima, *MaxY*. De forma progressiva conseguimos ajustar facilmente a energia das rodas até o resultado final.

3.4 Elipse

Ao contrário dos comportamentos anteriores, neste caso a roda que tem de ter mais velocidade está do lado da luz, ou seja, o sensor externo terá maior *output* e por sua vez, irá contribuir para que a roda externa tenha maior velocidade (comportamento medroso), isto é, não fizemos a troca dos sensores. Queremos que a roda direita tenha sempre mais velocidade do que a esquerda para poder fazer a curvatura da elipse, mesmo que o sensor direito esteja mais afastado da luz, para isso foi importante aplicarmos um threshold mais alto a este sensor relativamente ao esquerdo.

Mais uma vez, os nossos valores de partida foram os valores do enunciado.

Exp	Sensor	MinX	MaxX	MinY	MaxY	Média	Desvio Padrão	Velocidade	Funcionou
1	Direita	None	None	None	None	0.5	0.12	15	Não
	Esquerda	None	None	None	None	0.5	0.12		
2	Direita	None	None	None	None	0.5	0.12	30	Não
	Esquerda	None	None	None	None	0.5	0.12		
3	Direita	0.25	0.75	None	None	0.5	0.12	15	Não
	Esquerda	0.25	0.75	None	None	0.5	0.12		
4	Direita	None	None	0.05	0.6	0.5	0.12	15	Não
	Esquerda	None	None	0.05	0.6	0.5	0.12		
5	Direita	0.25	0.75	0.05	0.6	0.5	0.12	15	Não
	Esquerda	0.25	0.75	0.05	0.6	0.5	0.12		
6	Direita	None	None	0.05	0.6	0.5	0.12	15	Não
	Esquerda	None	None	0.05	0.6	0.5	0.8		
7	Direita	None	None	0.05	0.6	0.5	0.5	15	Não
	Esquerda	None	None	0.05	0.6	0.5	0.8		
8	Direita	None	None	0.05	0.6	0.5	0.5	15	Não
	Esquerda	None	None	0.05	0.4	0.5	0.8		
9	Direita	None	None	0.05	0.6	0.5	0.5	15	Não
	Esquerda	None	None	0.05	0.4	0.2	0.8		
10	Direita	None	None	0.05	0.6	0.6	0.5	15	Não
	Esquerda	None	None	0.05	0.41	0.2	0.8		
11	Direita	None	None	0.05	0.6	0.6	0.5	15	Sim
	Esquerda	None	None	0.05	0.41598	0.2	0.8		
12	Direita	None	None	0.05	0.6	0.6	0.5	15	Sim
	Esquerda	None	None	0.05	0.4135	0.2	0.8		

Tabela 3: Testes realizados para a elipse.

Comportamentos:

1. Praticamente não se move.
2. Praticamente não se move.
3. Não se move.
4. Segue em frente até à luz de cima e desvia-se desta.

5. Segue em frente até à luz de cima e desvia-se desta.
6. Fica as voltas no mesmo sitio no sentido horário.
7. Realiza 1 elipse no sentido anti-horário e depois desvia a trajetória.
8. Descreve uma trajetória circular entre as duas luzes no sentido anti-horário.
9. Descreve uma trajetória circular entre as duas luzes no sentido anti-horário.
10. Realiza 1 elipse no sentido anti-horário e depois desvia a trajetória.
11. Realiza 6 elipses no sentido anti-horário e depois desvia a trajetória.
12. Realiza 7 elipses no sentido anti-horário e depois desvia a trajetória.

4 Mundo

Nesta etapa, nós criamos uma pista de formula 1. Usamos os blocos fornecidos para simular os contornos do percurso.

O objetivo do jogo é que os carros realizem o percurso sem que fiquem presos e evitem colisões com os blocos.

Para que isso aconteça, no carro azul usamos a função de ativação linear invertida que deteta blocos com os sensores invertidos em relação as rodas, ou seja, o sensor da direita controla a roda esquerda (comportamento explorador).

Já para o carro amarelo, ele deteta as luzes, faz uso da função de ativação linear e tem a mesma mecânica dos sensores do carro azul (comportamento agressor).

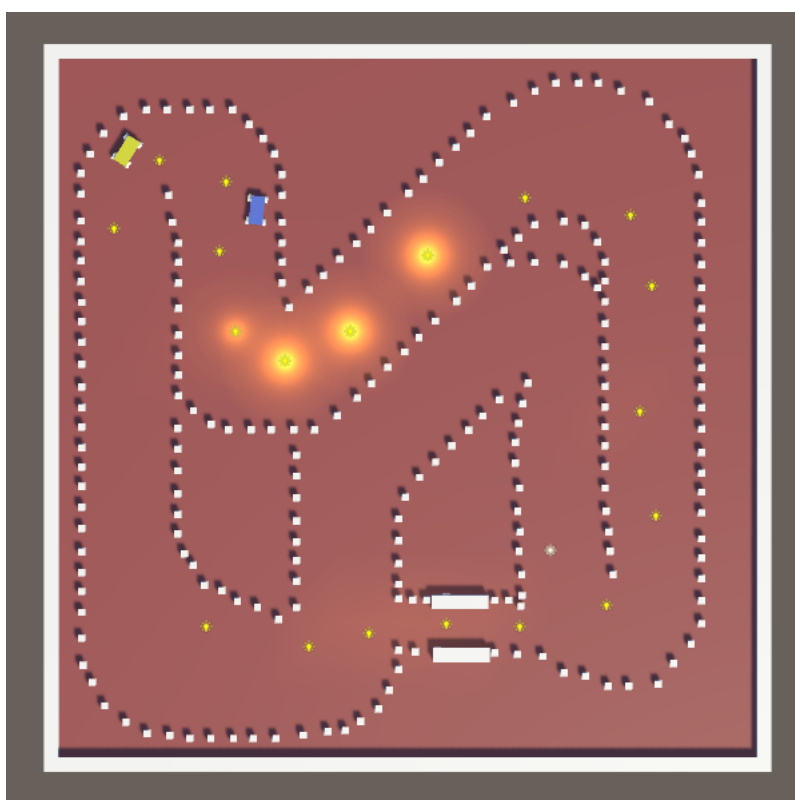


Figura 3: Mundo: Pista de corrida.

Forças do carro azul (deteta blocos):

- Navega por qualquer percurso;
- Deteta os limites do percurso e evita o embate.

Fraquezas do carro azul (deteta blocos):

- Eventualmente, em curvas pontiagudas pode ter dificuldades em manter o percurso e até mesmo inverter o sentido.

Forças do carro amarelo (deteta luzes):

- A possibilidade de controlar o carro com as luzes e prevenir zig-zags.

Fraquezas do carro amarelo (deteta luzes):

- Necessidade de colocar luzes para todos os percursos novos;
- As luzes de caminhos adjacentes interferem com o caminho atual;
- Facilidade em desviar do percurso pretendido e embater nos blocos.

Na generalidade dos testes, o carro azul tem melhor performance. Faz o percurso com maior velocidade e evita os embates. Salvas exceções que nas curvas invertia o sentido da rota. O carro amarelo, anda com menor velocidade e por vezes fica preso nos blocos.