

# T08 - Trabalho Prático de Redes Neurais e Sistemas Fuzzy:

## Opção 2: Aprendizado supervisionado do desvio de obstáculos para um robô móvel autônomo simulado no ambiente V-REP

Anderson Antonio Campanha  
Bruna de Paula Monarin  
Débora Sandi

Nesse arquivo será descrito como foi realizado o Trabalho a partir da biblioteca de Redes Neurais disponibilizada pelo Professor João Fabro. O robô escolhido pelo grupo é o K-Junior.

### 1 O Robô

O robô escolhido pela equipe foi o K-Junior. Este possui 10 sensores, sendo eles: 9 disposto na parte frontal, sendo 5 apontados para a frente e 4 para o solo; e 1 na parte traseira. A disposição dos sensores pode ser vista nas Figuras 1 e 2.

Sendo assim, foi trabalhado apenas com 6 sensores para o fornecimento de dados de entrada para a rede neural.

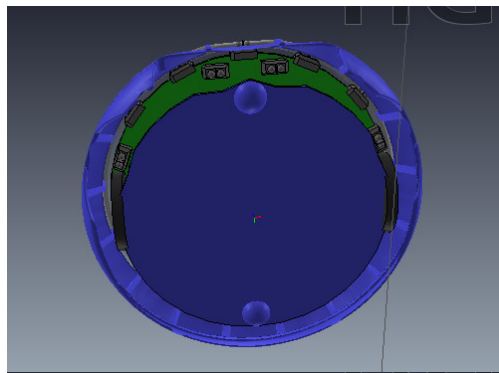


Figura 1: Sensores frontais

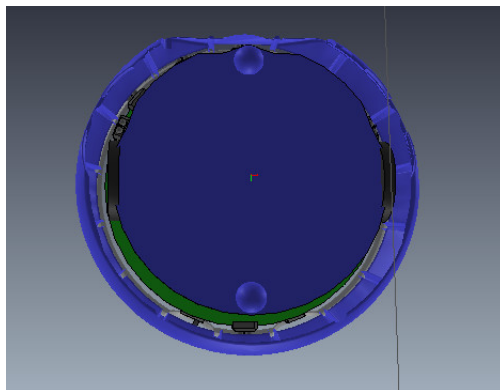


Figura 2: Sensor traseiro

## 2 Dados de Treinamento

Para o treinamento dos dados, primeiramente foi determinado a saída para cada um dos motores, com apenas um dos sensores ativado. Em seguida, os sensores foram ativados em pares adjacentes. Além de apenas os 3 centrais da frente, e todos os frontais.

O padrão de representação dos dados escolhido foi 1 para longe e 0.06 para muito perto, em relação à distância medida pelos sensores, e 1 e -1 para frente e para trás, respectivamente, em relação à direção que os motores devem girar. Os dados de treinamento escolhidos estão apresentados na Tabela 1.

Tabela 1: Dados de treinamento da rede neural

Dianteiro					Traseiro	Motor	
						Esquerdo	Direito
						1	1
						0	1
						0	1
						-1	1
						1	0
						1	0
						1	1
						-1	1
						-1	1
						1	-1
						1	-1
						-1	1
						-1	-1

Os dados foram inseridos no arquivo `training.junin`.

## 3 Treinamento da Rede

A implementação da rede neural se deu através da utilização de uma biblioteca fornecida pelo professor João Fabro, que utilizava a função ou-exclusivo (XOR).

Tendo como base o arquivo `config.comoehefeito`, os parâmetros foram alterados de acordo com a necessidade para o treinamento da rede neural, e foram inseridos no arquivo `config.junin`. O conteúdo dos arquivos citados acima estão apresentados a seguir, respectivamente.

```

1      - Number of Inputs of Input Pattern
4      - Number of Outputs of Output Pattern
10     - Number of Delayed Inputs
5      - Number of Delayed Outputs
2      - Number of Layers of the (Recurrent) Neural Net
51     - Number of neurons of 1st hidden layer
4      - Number of neurons of 2nd hidden layer ... etc!
0.5    - Parameter momentum
0.3    - Parameter learning-rate
0.0001 - Parameter max_err

6      - Number of Inputs of Input Pattern
2      - Number of Outputs of Output Pattern
2      - Number of Delayed Inputs
5      - Number of Delayed Outputs
2      - Number of Layers of the (Recurrent) Neural Net
0.9    - Parameter momentum
0.05   - Parameter learning-rate
0.01   - Parameter max_err

```

Os valores de *momentum*, *learning-rate* e *max\_err* foram escolhidos experimentalmente, testando diferentes valores e verificando que esses satisfaziam bem a rede esperada. O número de outputs é dois pois existe um motor para cada uma das duas rodas. O número de entradas é 6 pois existem cinco sensores frontais e um traseiro. Para a camada intermediária foram escolhidos cinco neurônios, e verificamos que essa escolha condiz com a rede que esperávamos obter.

## 4 Implementação da Rede ao Script do K-Junior

Para a implementação da rede neural ao script do robô no V-REP, primeiramente todos os arquivos pertencentes a pasta XOR foram transferidos para o seguinte diretório:

"diretorio do vrep"/programming/remoteApi

Alteração de diretórios:

- No arquivo `neuronet.cc`, na linha 242, na função `loadnet`, deve-se alterar o diretório para o que contém `neuralnet.junin`.
- Na linha 60 do script do K-Junior deve-se alterar o diretório para o diretório do arquivo `juninclient`.

Em seguida foram executados os arquivos `makefile0`, `neural.exe` e por último `makefile1`. No executável `neural` foi necessário passar a extensão `junin` como parâmetro. Para finalizar, o software V-REP foi executado e o arquivo `cena` foi aberto.

## 5 Checagem do Funcionamento na Simulação do V-REP

Com a execução dos arquivos mencionados, verificou-se o funcionamento da simulação: o robô lê as distâncias dos sensores, passa-as à rede, recebe como resposta as velocidades dos motores e, coerentemente, evita

paredes.