

5 Testes e Resultados

Neste capítulo serão apresentados experimentos que ilustram as implementações desenvolvidas e propostas no capítulo anterior. São mostrados experimentos que investigam o algoritmo de exploração proposto mas feito inicialmente para um ambiente fechado pequeno em relação ao robô.

5.1 Testes e Resultados no Player-Stage

No Player-Stage se fez até agora três tipos de simulações, o primeiro é um algoritmo de seguimento da parede utilizando o conceito da percepção geral e um sistema de controle com lógica fuzzy, esta simulação foi feita para avaliar a capacidade da percepção geral usando só 6 sensores ultra-sônicos o resultado se ilustra nas Figuras 5.1-5.10:

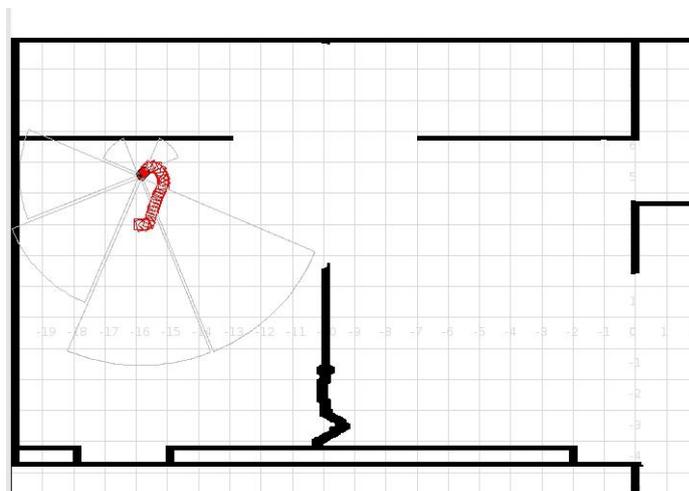


Figura 5.1: Exploração WallFollow (etapa 1/10) - robô se move na direção de vetor percepção geral até chegar próximo da parede

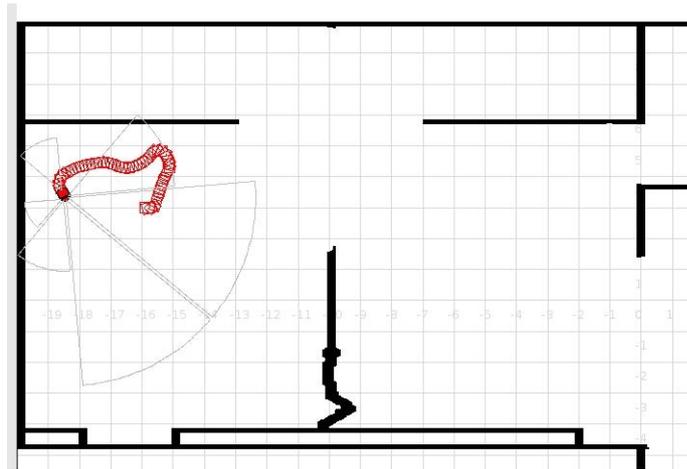


Figura 5.2: Exploração WallFollow (etapa 2/10) - robô se move seguindo a parede

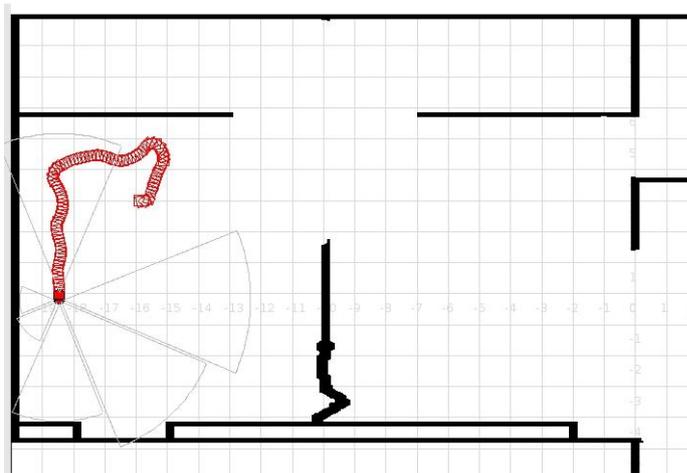


Figura 5.3: Exploração WallFollow (etapa 3/10) - robô trata de seguir a parede com certa oscilação

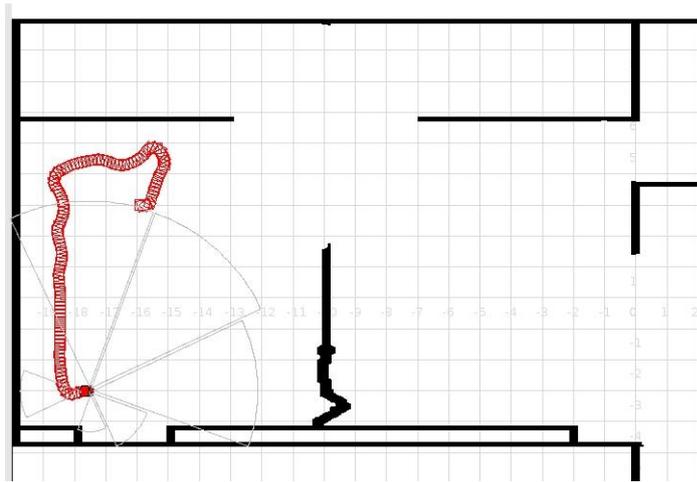


Figura 5.4: Exploração WallFollow (etapa 4/10) - robô se move paralelo à parede e vira à esquerda

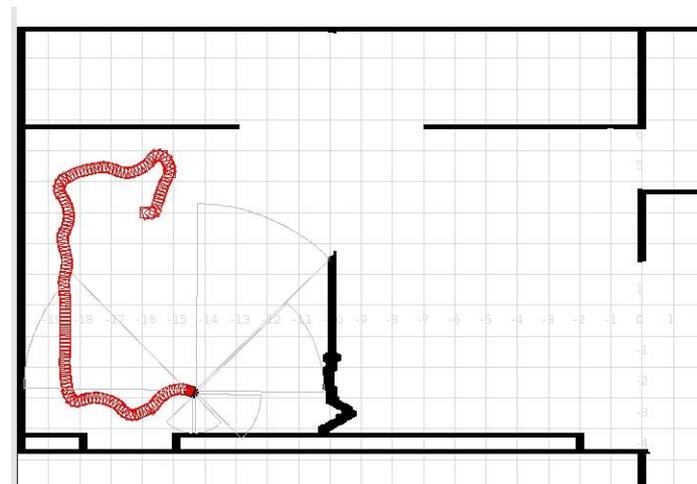


Figura 5.5: Exploração WallFollow (etapa 5/10) - robô se move perto da parede com oscilações devido à parede irregular

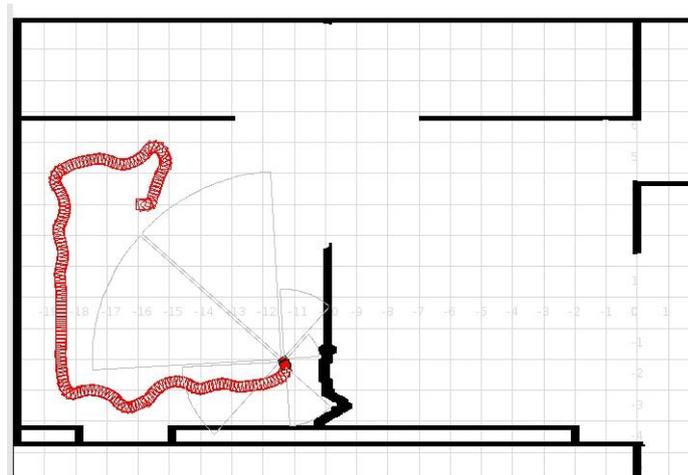


Figura 5.6: Exploração WallFollow (etapa 6/10) - robô se move paralelo à parede

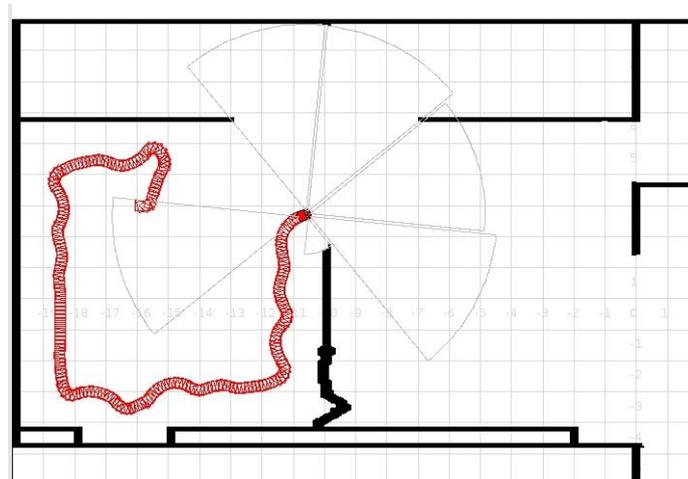


Figura 5.7: Exploração WallFollow (etapa 7/10) - robô vira à esquerda e segue à parede

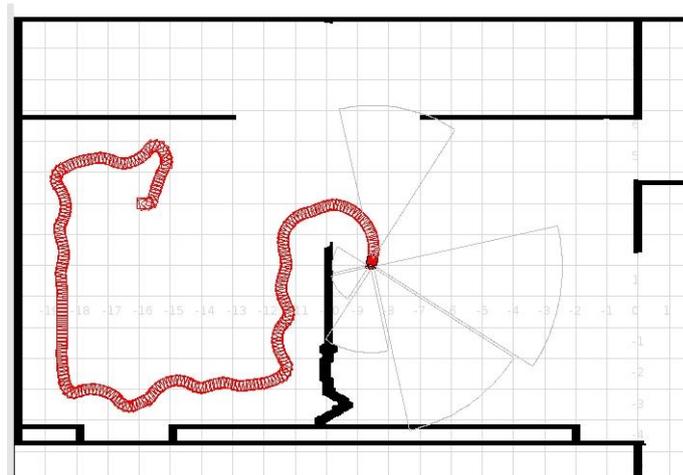


Figura 5.8: Exploração WallFollow (etapa 8/10) - robô vira com aceleração para conseguir virar 180°

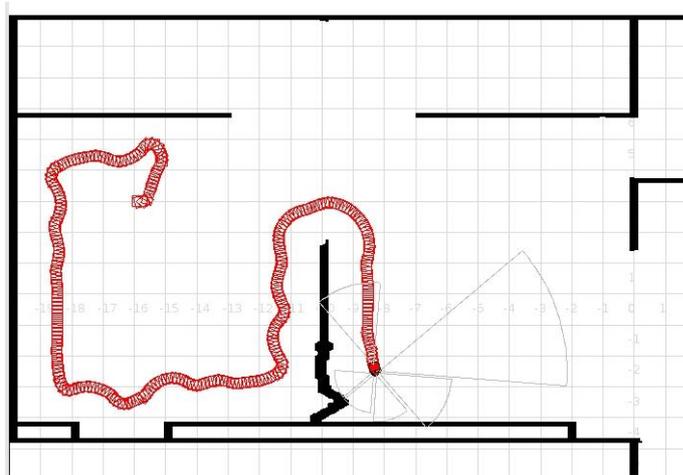


Figura 5.9: Exploração WallFollow (etapa 9/10) - robô se move paralelo à parede

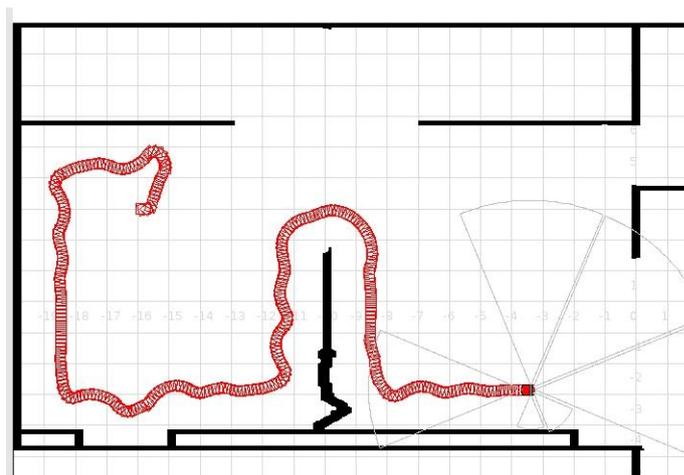


Figura 5.10: Exploração WallFollow (etapa 10/10) - robô segue paralelo à parede da parte inferior do mapa

O segundo tipo de simulação é a navegação do robô seguindo os espaços livres no ambiente. Isto também foi feito usando o conceito da percepção geral, neste caso a idéia é que o robô navegue pelo espaço onde não tem obstáculos, é dizer ao contrario do caso anterior o vetor de percepção geral tem que ser o menor possível (espaço livre), isto se ilustra nas Figuras 5.11-??:

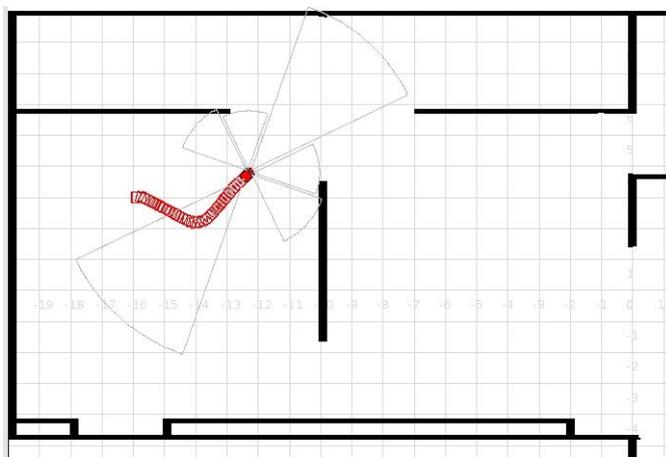


Figura 5.11: Exploração do Espaço Livre (etapa 1/6) - robô procura o espaço aberto

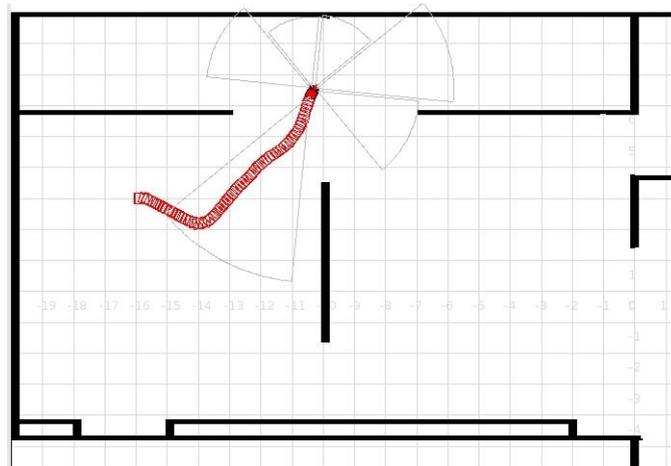


Figura 5.12: Exploração do Espaço Livre (etapa 2/6) - robô procura achar o caminho livre de obstáculos

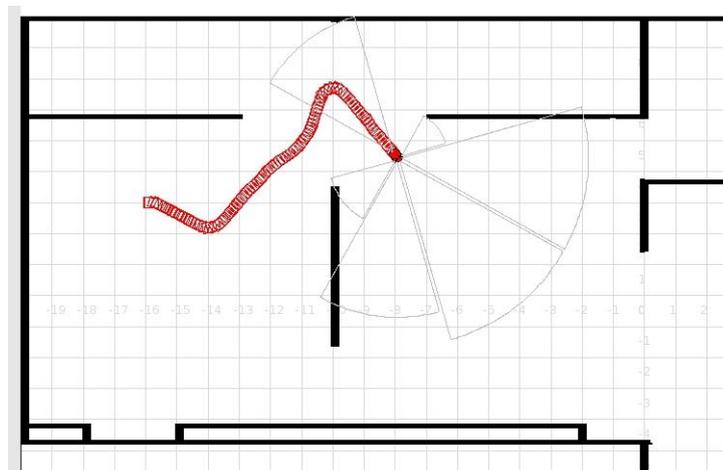


Figura 5.13: Exploração do Espaço Livre (etapa 3/6) - robô segue na direção do menor valor do vetor percepção geral

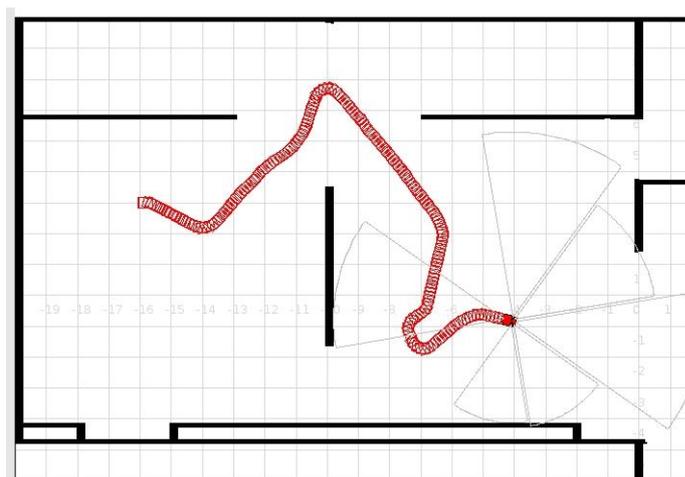


Figura 5.14: Exploração do Espaço Livre (etapa 4/6) - robô segue procurando o espaço livre

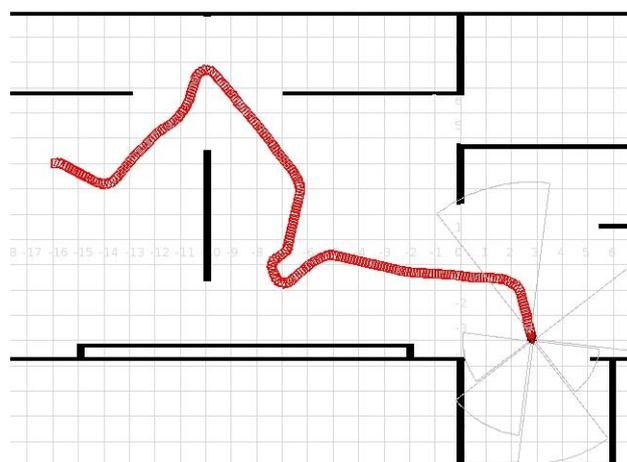


Figura 5.15: Exploração do Espaço Livre (etapa 5/6) - robô encontra um obstáculo e procura virar para sair em outra direção

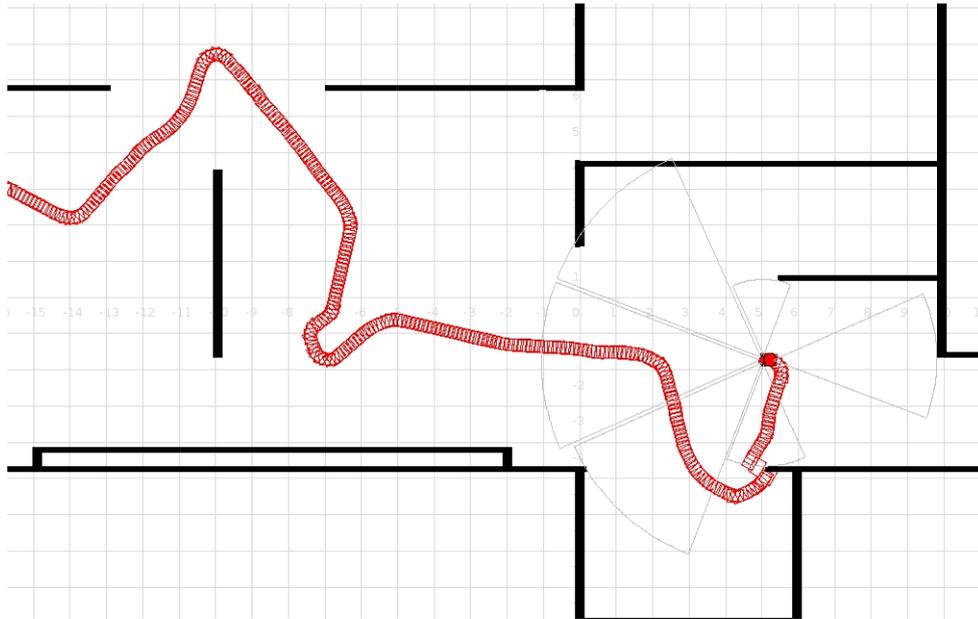


Figura 5.16: Exploração do Espaço Livre (etapa 6/6) - robô consegue sair do obstáculo e novamente procura o espaço livre

A terceira Simulação foi feita usando o algoritmo proposto no capítulo anterior mas sem o algoritmo do *SIFT* porque não é possível simulá-lo no Player-Stage. Embora foi feito só para um lugar pequeno, a simulação tem bom resultado ilustrando que o algoritmo pode cumprir a exploração sem problemas num ambiente pequeno, a navegação se ilustra nas Figuras 5.17-5.32:

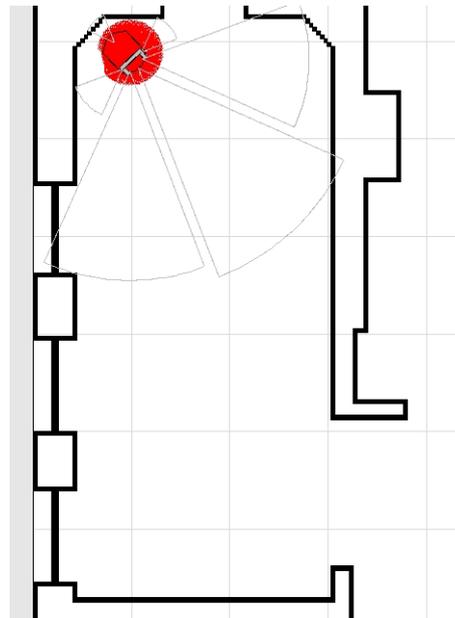


Figura 5.17: Navegação num ambiente desconhecido(etapa 1/16) - robô na posição inicial procura as possíveis direções de navegação

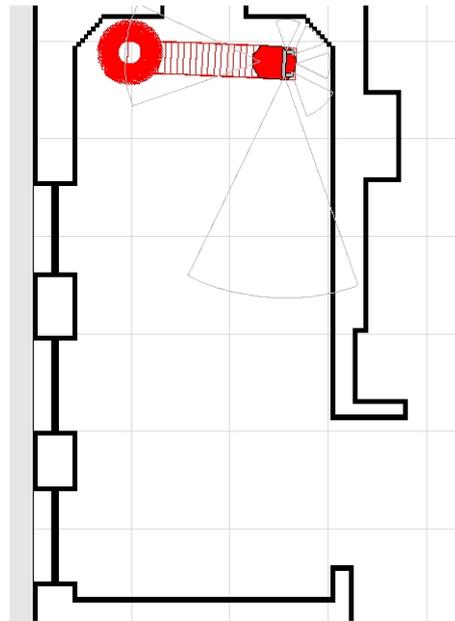


Figura 5.18: Navegação num ambiente desconhecido (etapa 2/16) - robô começa navegar na primeira direção

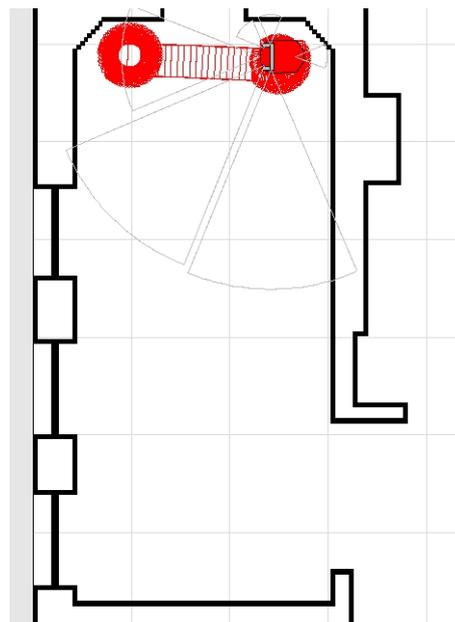


Figura 5.19: Navegação num ambiente desconhecido (etapa 3/16) - robô procura voltar novamente ao nó pai

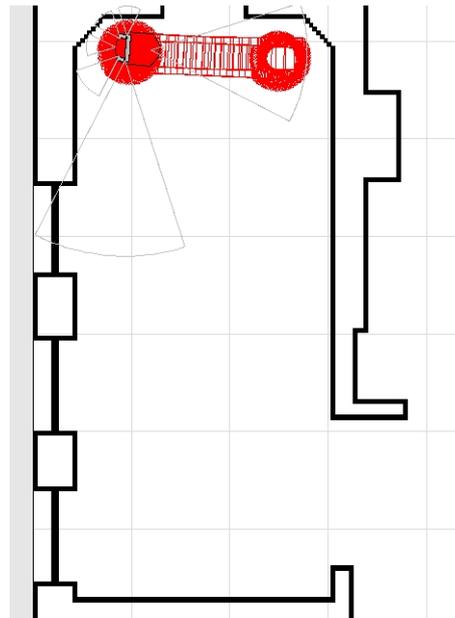


Figura 5.20: Navegação num ambiente desconhecido (etapa 4/16) - robô chega ao nó pai

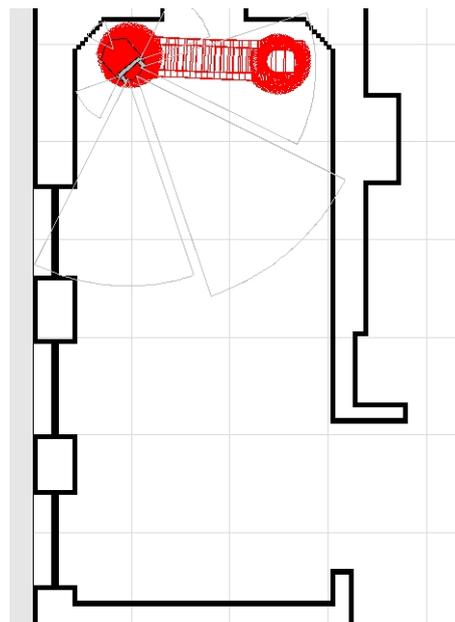


Figura 5.21: Navegação num ambiente desconhecido (etapa 5/16) - robô procura outra direção

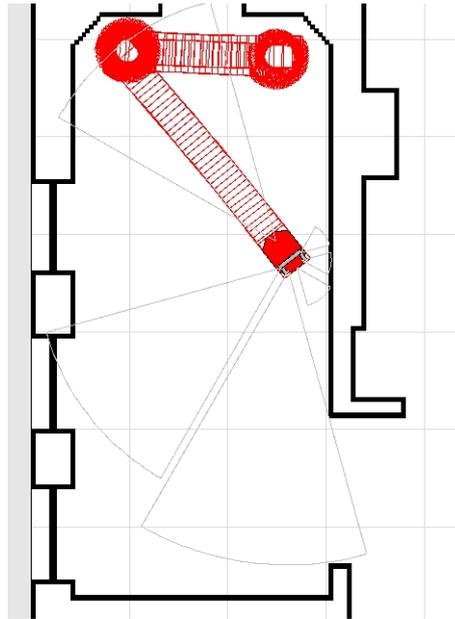


Figura 5.22: Navegação num ambiente desconhecido (etapa 6/16) - robô chega até estar próximo à parede

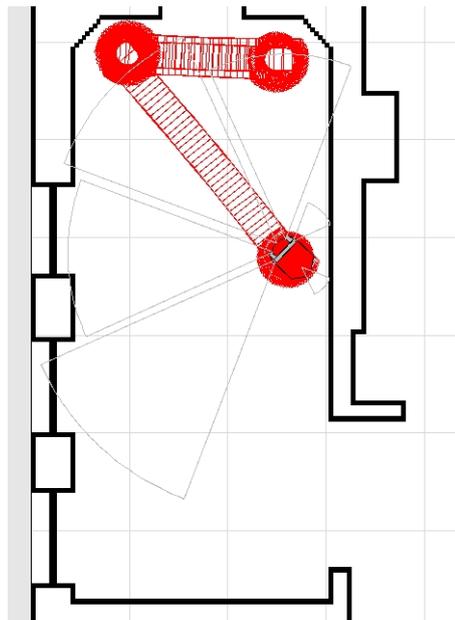


Figura 5.23: Navegação num ambiente desconhecido (etapa 7/16) - robô procura a direção do nó pai

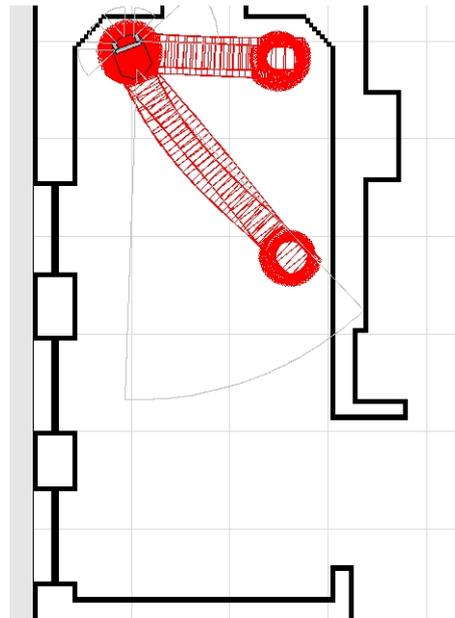


Figura 5.24: Navegação num ambiente desconhecido (etapa 8/16) - robô segue uma trajetória diferente até alcançar o nó pai

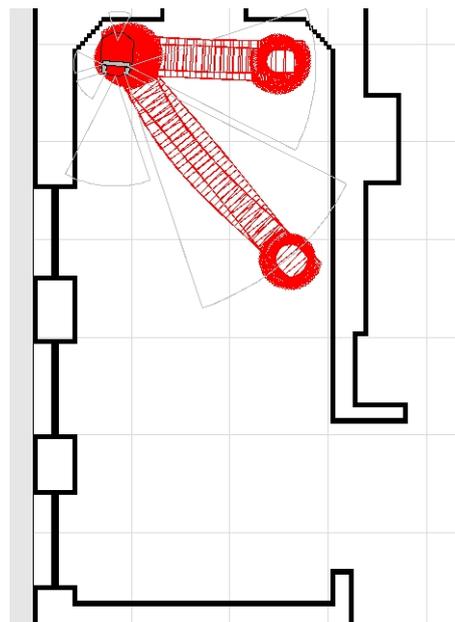


Figura 5.25: Navegação num ambiente desconhecido (etapa 9/16) - robô procura uma nova direção

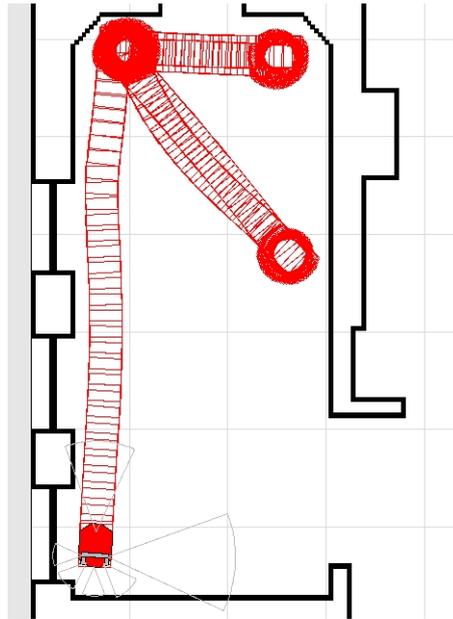


Figura 5.26: Navegação num ambiente desconhecido (etapa 10/16) - robô faz um wallfollow nesta trajetória

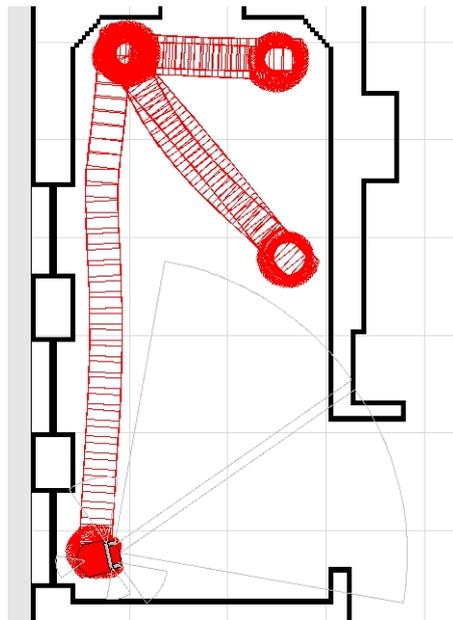


Figura 5.27: Navegação num ambiente desconhecido (etapa 11/16) - robô encontra uma possível trajetória

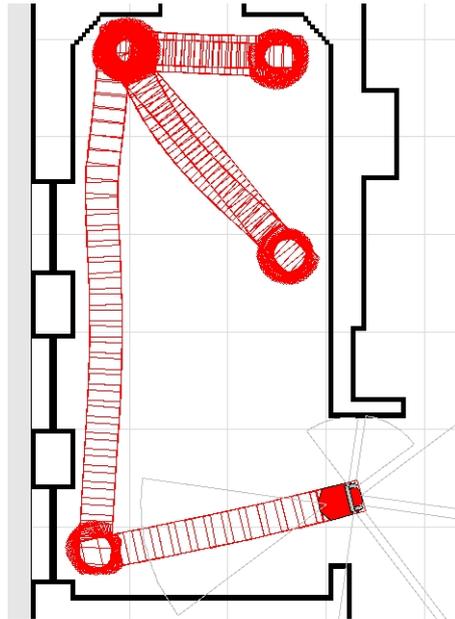


Figura 5.28: Navegação num ambiente desconhecido (etapa 12/16) - robô navega para outro nó

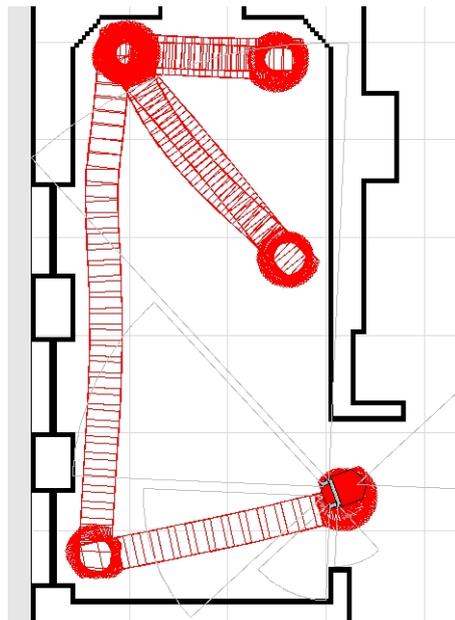


Figura 5.29: Navegação num ambiente desconhecido (etapa 13/16) - robô tem uma condição de retorno

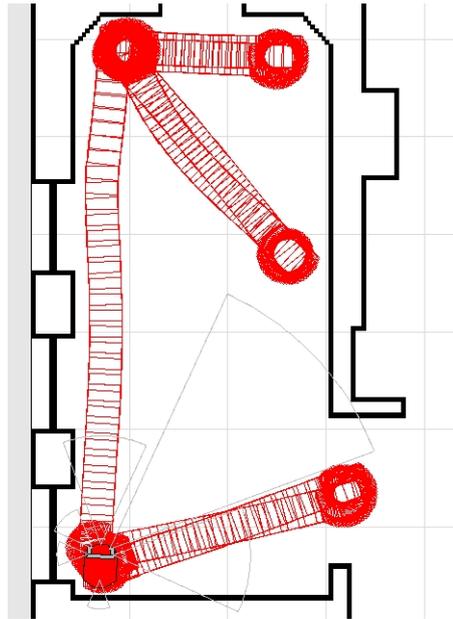


Figura 5.30: Navegação num ambiente desconhecido (etapa 14/16) - robô volta para o nó pai e gira até encontrar a direção do nó inicial

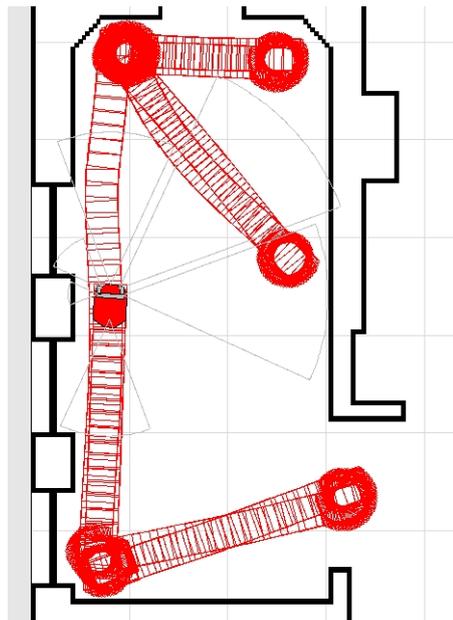


Figura 5.31: Navegação num ambiente desconhecido (etapa 15/16) - robô novamente faz um wallfollow para chegar ao nó pai

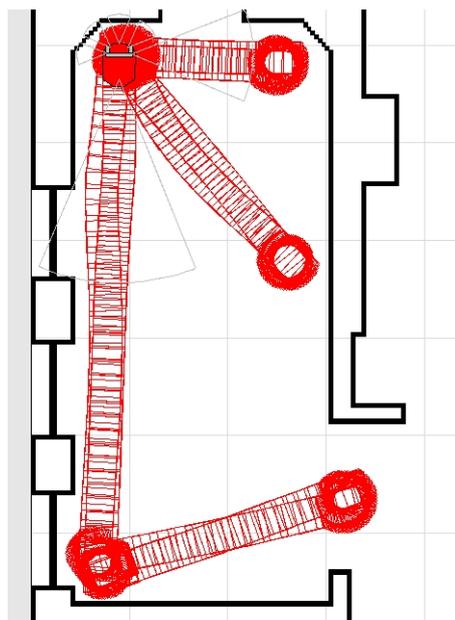


Figura 5.32: Navegação num ambiente desconhecido (etapa 16/16) - robô consegue chegar ao nó pai e consegue a condição de parada

5.2

Experimentos de identificação dos nós

Se realizou testes com o robô, o primeiro foi o reconhecimento de padrões no nó pai (Figura 5.33), se tirou fotos do nó 24 vezes cada 15° até completar uma volta completa; se achou os descritores *SIFT* para cada direção e se fez a leitura dos sensores, para um posterior treinamento da rede neuronal proposto no capítulo anterior. É dizer com a leitura dos sensores em cada um das 24 direções e com os descritores *SIFT* como entradas à rede neural pode-se generalizar e aprender o nó atual. Os resultados dos descritores se ilustram nas Figuras :



Figura 5.33: Nó pai no inicio da navegação



Figura 5.34: Descritores SIFT do nó depois do processamento no matlab num range total de visão

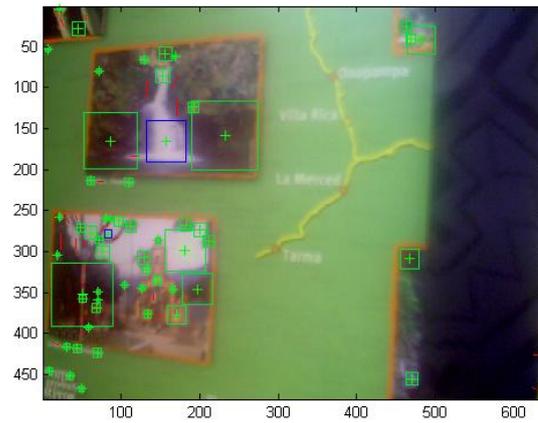


Figura 5.35: Descritores SIFT do nó pai num ângulo de 15°

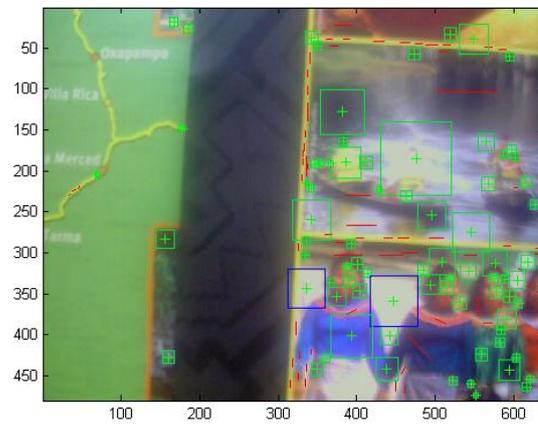


Figura 5.36: Descritores SIFT do nó pai num ângulo de 30°

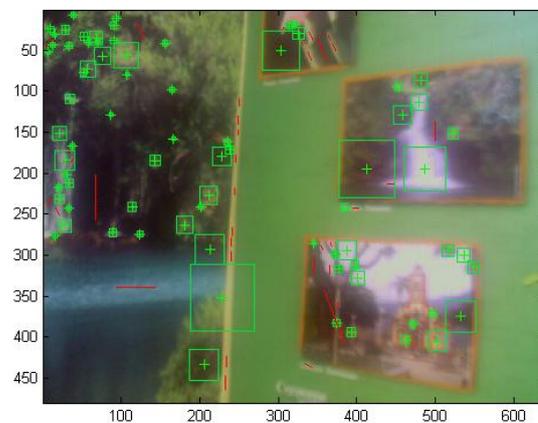


Figura 5.37: Descritores SIFT do nó pai num ângulo de 45°

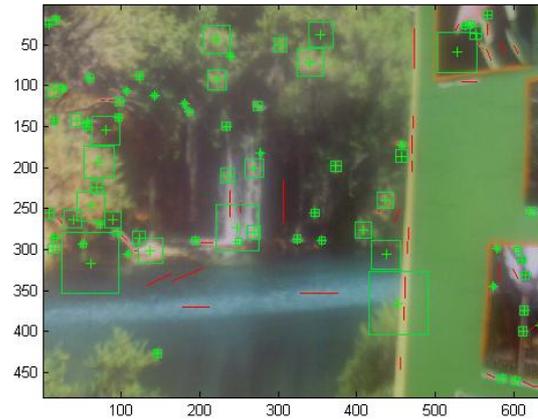


Figura 5.38: Descritores SIFT do nó pai num ângulo de 60°

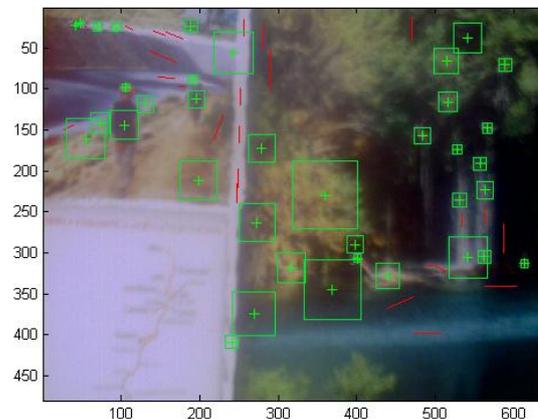


Figura 5.39: Descritores SIFT do nó pai num ângulo de 75°

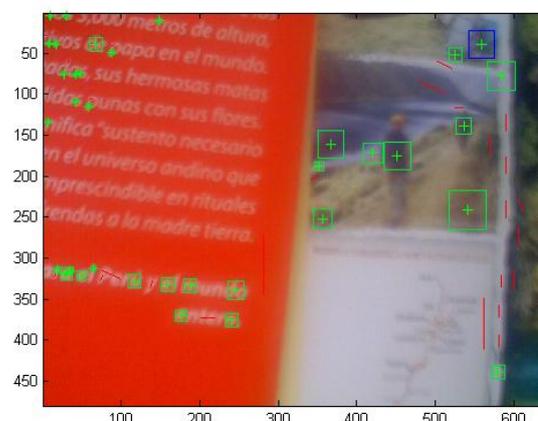


Figura 5.40: Descritores SIFT do nó pai num ângulo de 90°

Os eixos X e Y são as dimensões de cada foto e que é uma variável importante para o tempo de processamento para achar os descritores SIFT, se a foto é muito grande então o custo do processamento é maior.

Depois de fazer a leitura dos sensores em cada direção e tendo os descritores *SIFT* se procedeu ao tratamento dos dados e logo ao treinamento da rede neuronal proposto tendo os seguintes resultados para diferentes números de neurônios da camada escondida como se ilustra nas Figuras 5.41 ao 5.45.

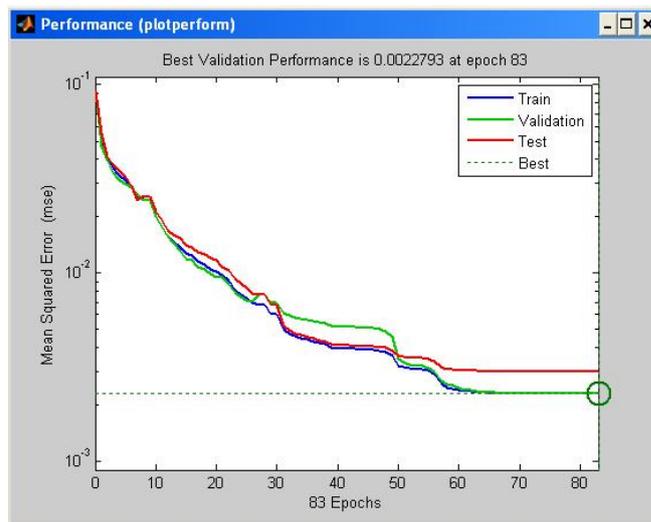


Figura 5.41: Desempenho da rede neuronal para uma camada escondida de 20 neurônios

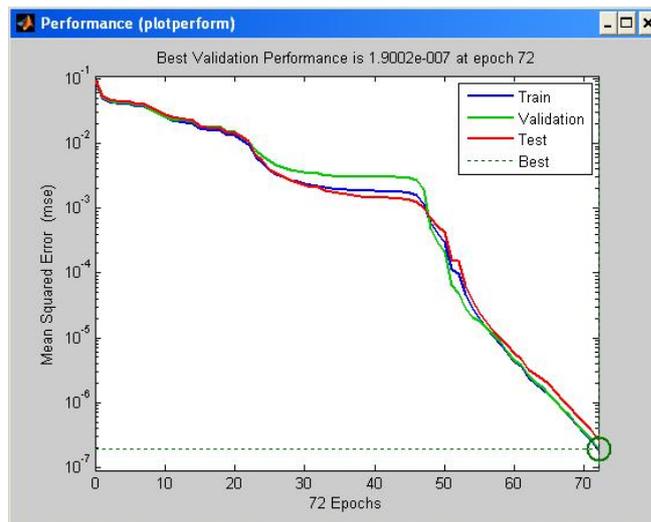


Figura 5.42: Desempenho da rede neuronal para uma camada escondida de 25 neurônios

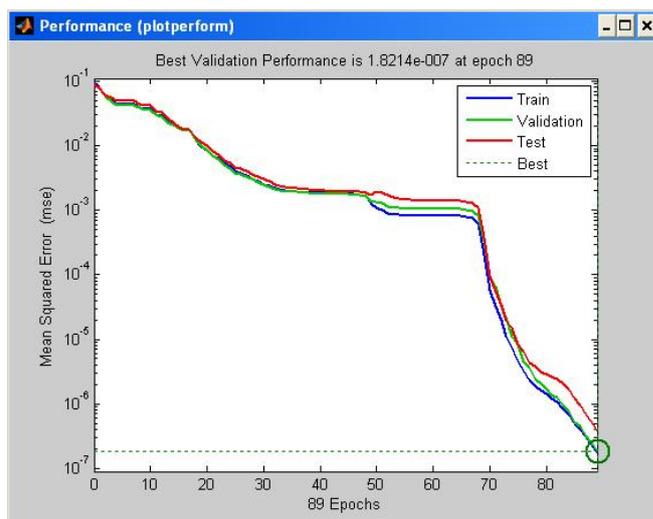


Figura 5.43: Desempenho da rede neuronal para uma camada escondida de 26 neurônios

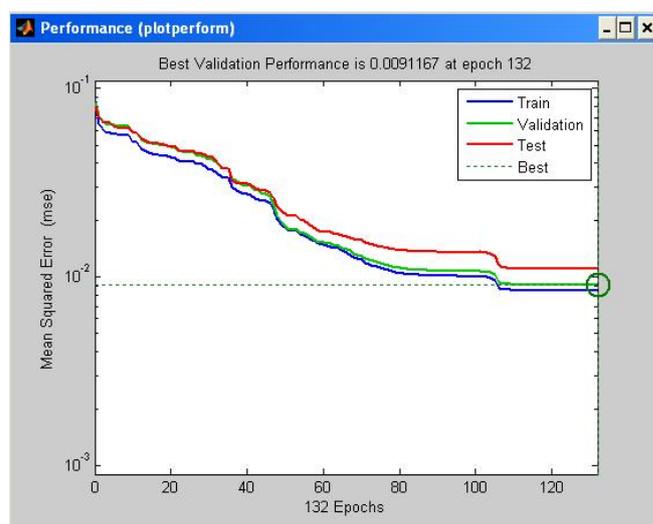


Figura 5.44: Desempenho da rede neuronal para uma camada escondida de 27 neurônios

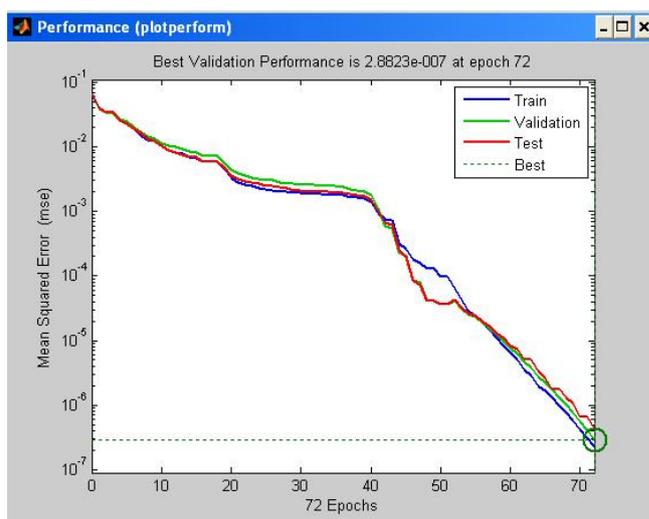


Figura 5.45: Desempenho da rede neuronal para uma camada escondida de 30 neurônios

Das gráficas podem se observar que o melhor desempenho que tem a rede neuronal é com uma camada escondida de 26 neurônios, isto quer dizer que a rede com as entradas que tem pode aprender o nó atual da exploração. Se realizou testes só com os valores dos sensores e se teve também bons resultados só que os descritores SIFT são dados importantes para fazer a diferença com outros nós.

A dificuldade de usar redes neurais é o custo computacional que é demasiado, é por isso que foi usado o método de *SVM*, melhorando o tempo de treinamento e a precisão da classificação.

O *SVM* é ao igual que a rede neural uma caixa preta que consegue fazer o aprendizagem mas com poucas épocas ou iterações.

5.3

Experimentos de exploração completa

Neste experimento de prova de este sistema de controle de navegação, o robô tem que ser capaz de sortear um obstáculo, tomando em conta tudo o problema de navegação, o sorteio de obstáculos poderia reduzir a dois aspectos importantes:

1. Quando começa o robô a sortear o obstáculo.
2. Como sorteia o obstáculo.

5.3.1

Começo de navegação

Nesta parte da navegação, o robô para evitar um obstáculo inesperado começa a realizar cálculos das leituras dos sensores e obtém a mínima distância entre o robô e o obstáculo em cada movimento realizado no ambiente desconhecido, esta mínima distância é programada pelo controlador fuzzy, o qual ante uma presença de um obstáculo muito próximo o robô vai tratar de evadir o obstáculo para não colidir com o obstáculo, na Figura 5.46 se apresenta o momento quando o robô começa a interatuar ante a presença do obstáculo.

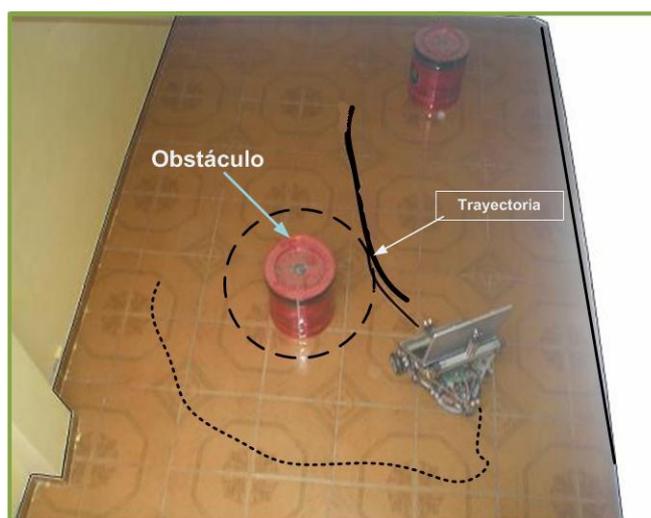


Figura 5.46: O robô ER1 evita um obstáculo

5.3.2

Sorteio dos Obstáculos

O processo de sortear obstáculos que realiza o robô consiste em seguir uma trajetória que siga o contorno do obstáculo no caso que exista só um obstáculo, mas se existisse muitos obstáculos percebidos no mesmo instante, o robô segue uma trajetória evitando colidir com qualquer obstáculo pelo tanto o robô também pode seguir uma trajetória paralela à parede como se fosse um controle *wall-following*¹ (ver Figura 5.47).

¹controle de seguimento de parede

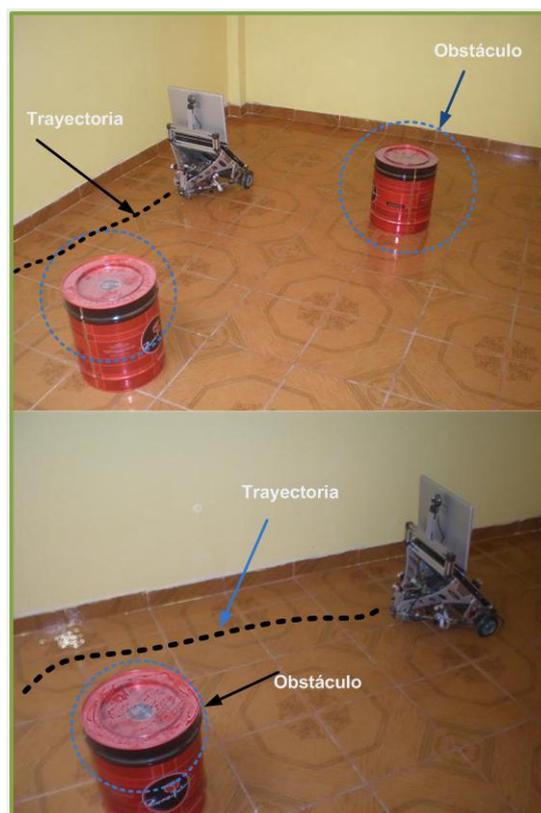


Figura 5.47: O robô faz um controle de seguimento de parede