

4

Desenvolvimento de Algoritmos de controle

4.1

Introdução

Os estudos das características sensoriais e equações de movimento realizados com o robô ER1, junto com as ferramentas de simulação, permitem o estudo e elaboração de algoritmos de controle de robôs móveis. Este capítulo inicialmente define o problema a ser tratado. São definidas as tarefas que o robô deve realizar e os motivos de tal escolha. O ambiente no qual está inserido também é determinado. Depois disso, são abordados os algoritmos e as técnicas são divididas em três etapas: etapa de exploração do ambiente desconhecido, etapa de navegação num ambiente já conhecido e a etapa de otimização da trajetória.

4.2

Definição do problema

O objetivo dos sistemas de controle desenvolvidos é fazer que o robô navegue num ambiente desconhecido, aprenda certos pontos do ambiente que ele considere importante (*landmarks*, que saiba como navegar através desses pontos já conhecidos e que possa depois navegar de um ponto a outro ponto otimizando a trajetória percorrida desviando obstáculos; para isso foram desenvolvidos algumas técnicas que a continuação se explicam.

4.3

Percepção Geral

As seguintes considerações são baseadas no robô móvel com três graus de liberdade num movimento planar como se amostra na Figura 4.2. O robô é equipado com um anilha de 6 sensores ultra-sônicos, e outro anilha de 11 sensores infravermelhos (Figura 4.1), o anilha de sensores infravermelhos são usados como detectores digitais a uma distância de 25cm para cobrir as zonas onde os sensores ultra-sônicos não podem perceber nenhuma informação perto do robô; O anilha de sensores ultra-sônicos vai se considerar como se ilustra na figura 4.2.

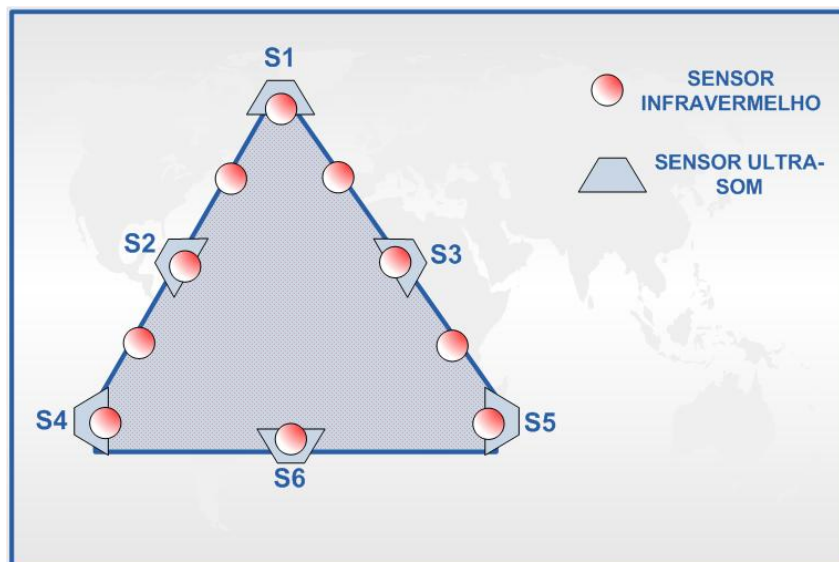


Figura 4.1: Posição dos sensores no robô ER1

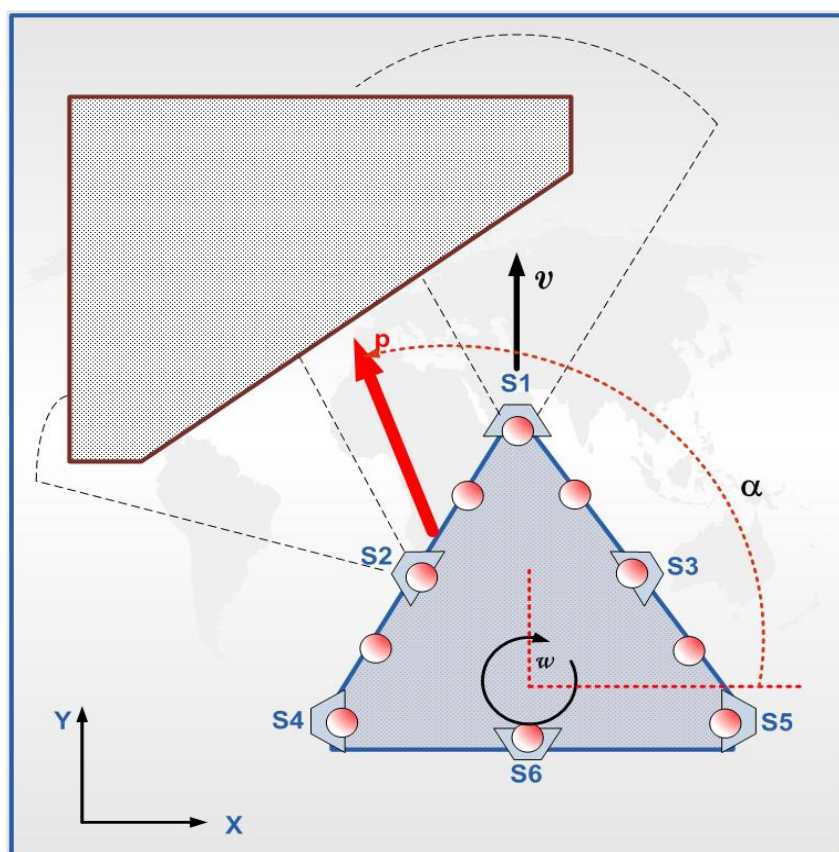


Figura 4.2: Rob móvel com o vetor de percepção

A idéia desta técnica que cada sensor i do robô móvel seja referido a um vetor p_i , esta direção vai ser a orientação dos eixos do sensor e a função de longitude vai ser em função da distancia d_i medido pelo sensor.

$$p_i = \frac{d_{max} - d_i}{d_{max} - d_{min}} \quad (4-1)$$

onde d_{min} e d_{max} refere-se distancia mínima e máxima respectivamente onde pode estar posicionado um objeto detectado. p_i limitado no range $[0 - 1]$

A percepção geral do vetor p composto da suma individual das percepções p_i , seu direção a suma das percepções de todos os sensores e seu módulo o maior módulo de todas as percepções individuais.

$$\mathbf{p} = p_{i,max} \times \frac{\sum(p_i)}{|\sum(p_i)|} \quad (4-2)$$

A razão de cambio da percepção geral no tempo representado por $\dot{p}$ e expressado por um escalar.

$$\dot{p}_i = \frac{dp_i}{dt} = \frac{\Delta d_i}{\Delta t \times (d_{max} - d_{min})} \quad (4-3)$$

Com estas equações do conceito da percepção geral podemos fazer uso das técnicas da lógica *fuzzy* para fazer um controle da navegação inicial do robô móvel.

4.4

Algoritmo de Exploração

Para a primeira etapa da exploração do ambiente desconhecido se fez algumas suposições:

- As paredes do ambiente desconhecido não é do mesmo cor e não é homogêneo.
- A posição inicial do robô vai ser sempre escolhido de modo que o robô tenha espaço para navegar, isto para segurar que o robô vai ter uma direção para começar a navegar.
- A exploração tem que ser num ambiente que tenha iluminação adequada.

Feitas estas suposições na Figura 4.3 se ilustra o diagrama de fluxo do algoritmo de navegação e a continuação se descreve passo a passo o processo.

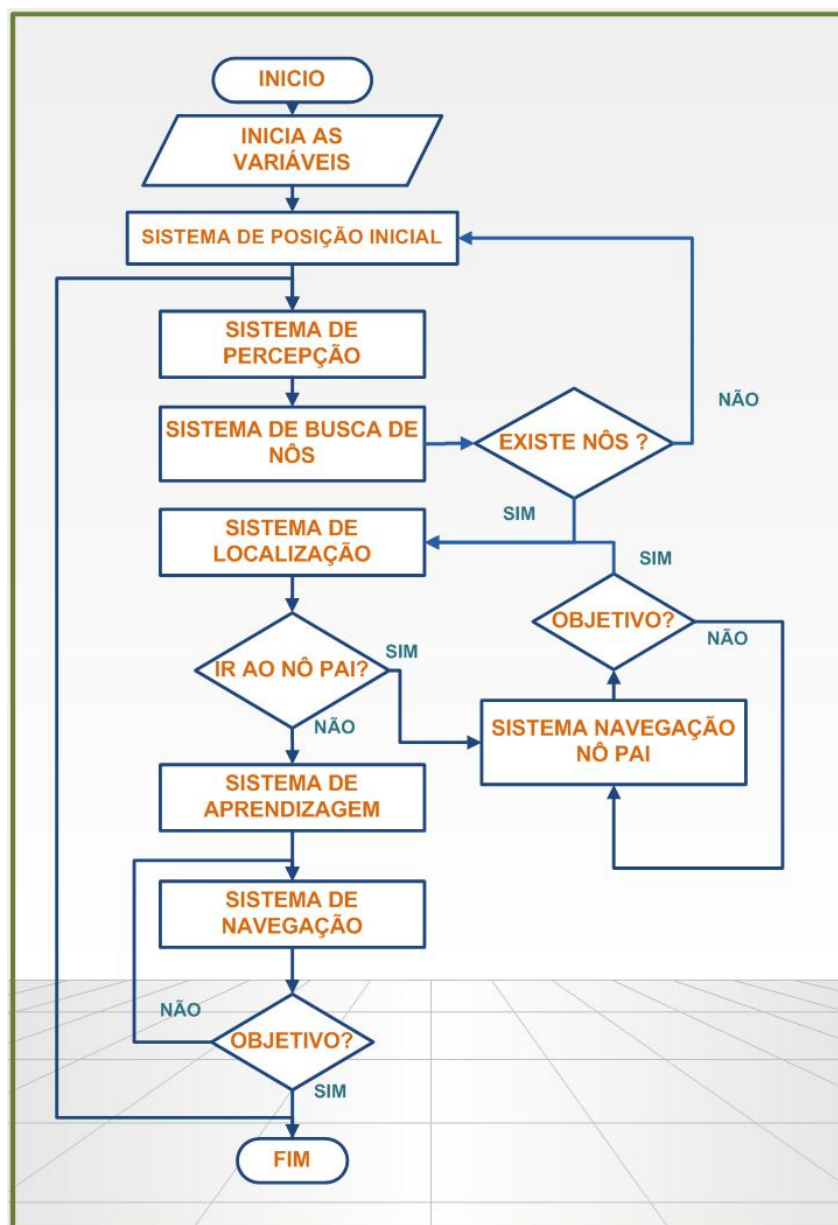


Figura 4.3: Diagrama de fluxo da etapa de exploração

1. O robô procura uma posição inicial para começar a exploração.
2. O robô ativa seu sistema de percepção para guardar a informação do nó.
3. Se procura os possíveis nós a explorar.
4. Se existe nós para explorar se executa o sistema de localização se não se procura outra posição inicial.
5. Se verifica se existe nós filhos a explorar se não existe se executa o sistema de retorno ao nó pai de outro modo se executa o sistema de navegação ao nó filho.

6. Se executa o sistema de aprendizagem.
7. Se executa o sistema de navegação ao nó filho.
8. Se o robô ainda não chega ao objetivo então continua com a navegação fazendo as leituras dos sensores.
9. Se chegou ao possível nó filho então faz novamente a etapa de percepção e continua o processo novamente desde 2.

Este algoritmo se assemelha ao proposto por (Fel06), pelo arvore de nós, mas a principal diferença está no uso de redes neurais com informações obtidas pelos sensores de ultra-som e infra-vermelho, além do uso da câmera. O trabalho de (Fel06), se limitou ao uso da câmera e de algoritmos determinísticos para a identificação dos nós, sem o uso de técnicas de inteligência computacional. Os sensores infra-vermelhos foram usados em (Fel06), apenas para cumprir o papel de sensores de contato (*bump sensores*).

4.4.1

Sistema de posição inicial

Nesta etapa o robô começa com uma leitura dos sensores e procura uma posição inicial segundo a percepção geral num ângulo definido como 0° , isto significa que a posição inicial é na direção do menor vetor de percepção geral. Se encontra esta direção o robô já tem um nó inicial mas se não encontra o robô procura a direção do objeto mais próximo ao robô, este algoritmo se ilustra na Figura 4.4.

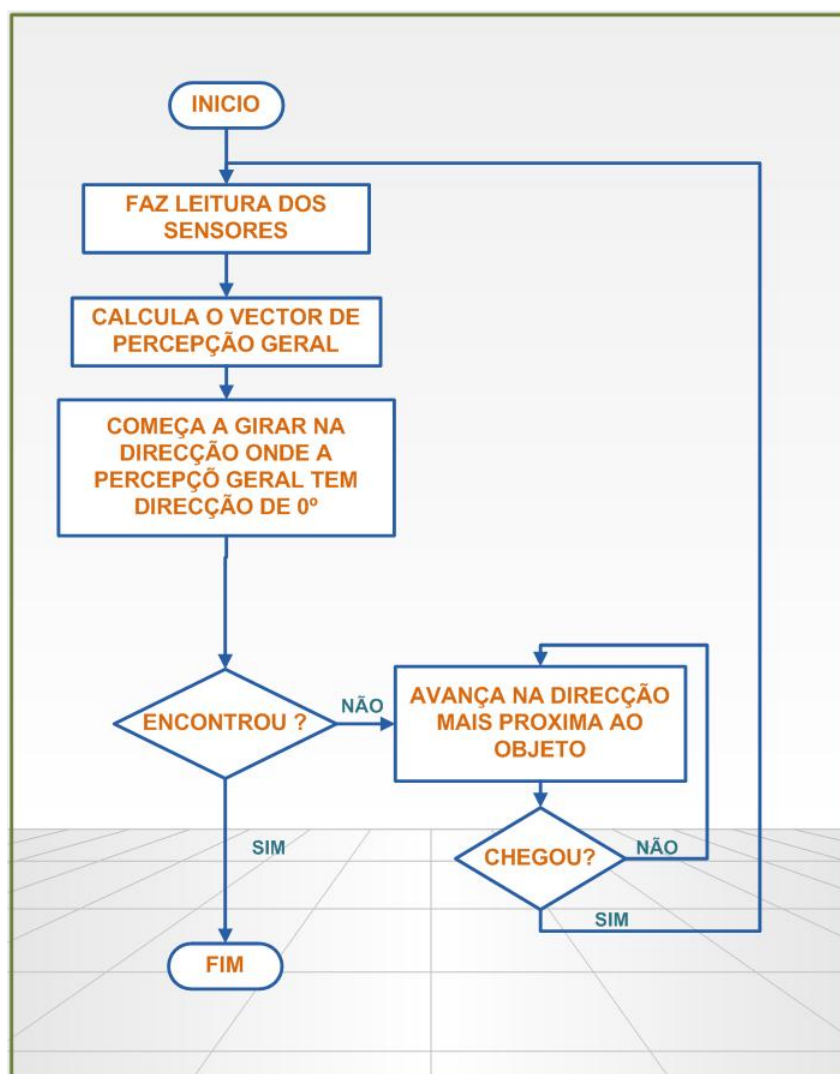


Figura 4.4: Diagrama de fluxo do sistema de posição inicial

4.4.2

Sistema de percepção

Nesta etapa o robô faz a leitura dos sensores ultra-sônicos e infravermelhos na direção inicial ($\text{ângulo} = 0^\circ$), e também tira uma foto e calcula os descritores *SIFT* e os guarda num vetor; depois disso o robô gira um ângulo de 15° e repete a leitura dos sensores e o cálculo dos descritores e assim por diante até completar o giro de 360° , logo todos os dados são guardados. A leitura dos sensores é feita pelo *DSP* e o cálculo dos descritores é feito pelo *notebook*, o diagrama se ilustra na Figura 4.5.

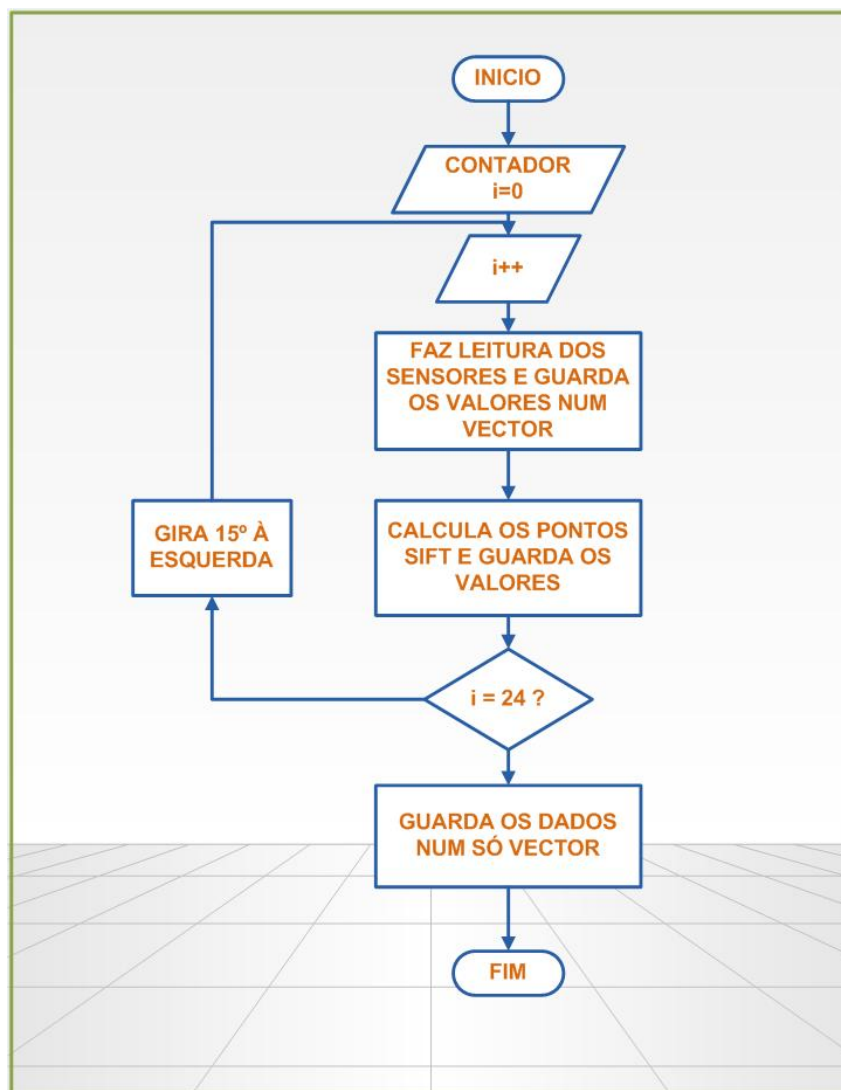


Figura 4.5: Diagrama de fluxo do sistema de Percepção

4.4.3

Sistema de busca de nós

Nesta parte do algoritmo procura se a informação do sensor ultra-sônico $S1$ para cada direção cada 45° e ordenados em forma crescente até 360° , se existe distâncias ao objeto maiores a $720mm$ (escolhido em relação ao diâmetro do robô), então essas direções se consideram possíveis nós filhos para explorar depois é verificado se existe dois possíveis nós filhos adjacentes, se isso é certo então a direção que tem menor distância ao objeto se considera que vai ser um nó folha, se faz isto para evitar que dois direções adjacentes tenham os mesmos nós filhos; se não existisse direções para explorar então se ativa um *flag* que indique isso; o diagrama de fluxo se ilustra na Figura 4.6.

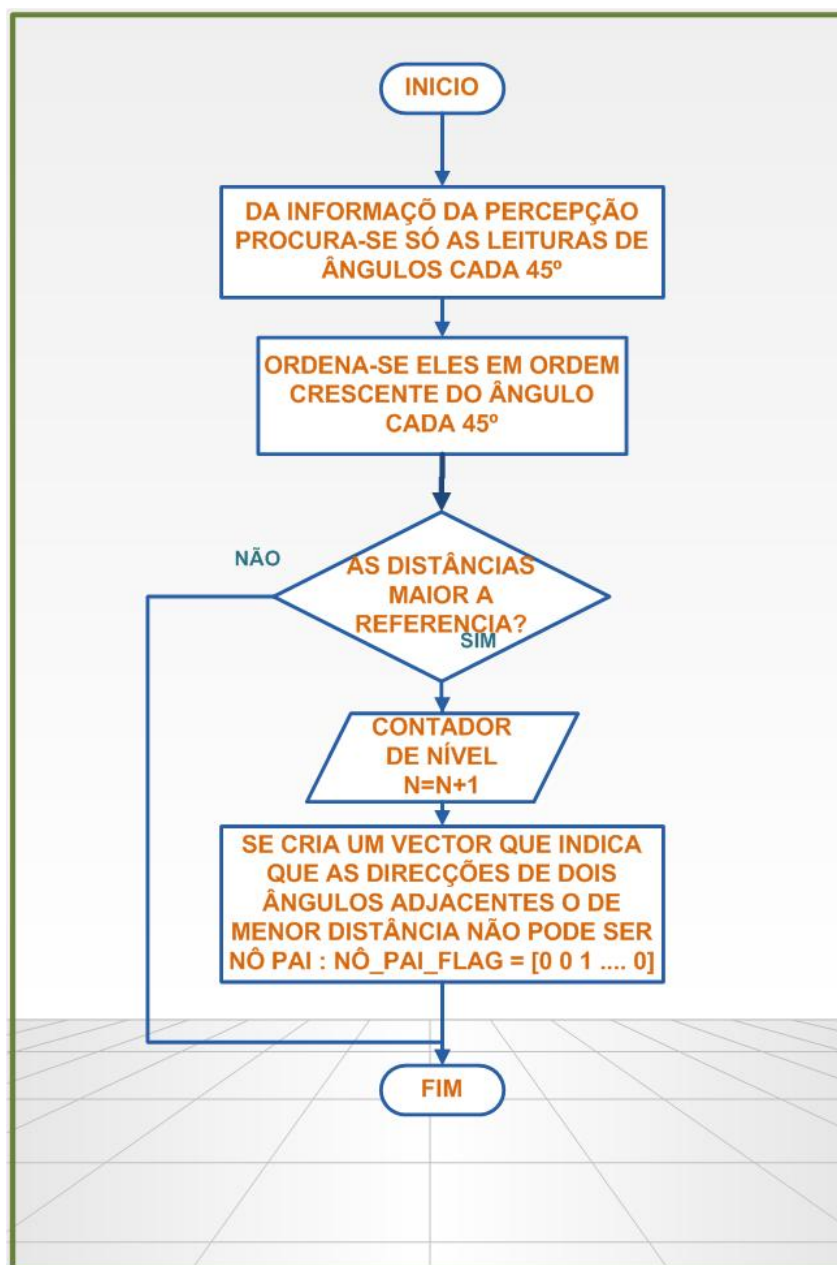


Figura 4.6: Diagrama de fluxo do sistema de Busca dos nós

4.4.4

Sistema de localização

A idéia da localização é que o robô tenha presente em que nó ele se encontra, com isto o robô sabe se vai explorar a um nó filho ou voltar a um determinado nó pai, isto se ilustra num grafo de árvore (Figura 4.7).

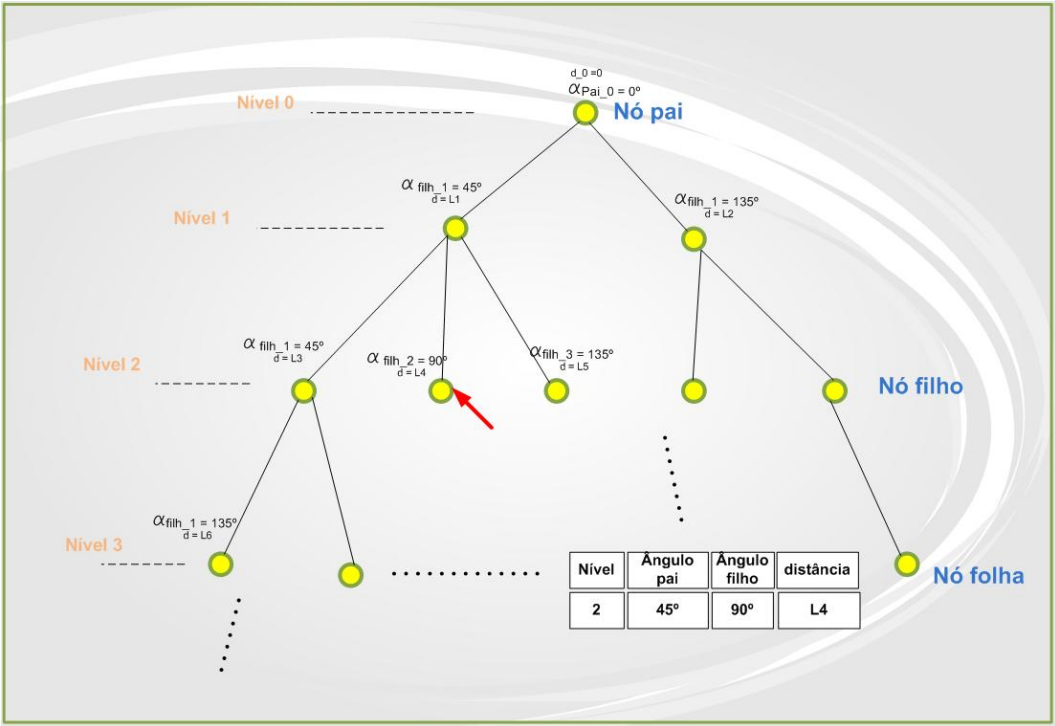


Figura 4.7: Grafo de árvore de nós da exploração

Para poder identificar o nó temos um vetor de identificação que se amostra na Figura 4.7; Por exemplo se queremos saber qual é a identificação do nó que está apontado com a flecha, teremos 4 parâmetros para isso, o primeiro indica o nível de profundidade da exploração, o segundo indica o ângulo do nó pai que saiu, o terceiro indica o nó filho atual que se encontra o robô, e o quarto indica a distância percorrida do nó pai até o nó filho.

O algoritmo desta etapa de localização se ilustra na Figura 4.8, onde pode-se observar as atualizações de todas as variáveis, contadores e *flags* para saber onde se encontra o robô; depois disso se aumenta em 45° a direção do nó filho a explorar mas se ele tem uma distância menor ao de referência então essa direção não é explorada pelo tanto aumenta novamente em 45° até chegar a 360°; se já chegamos então tem que voltar ao nó pai.

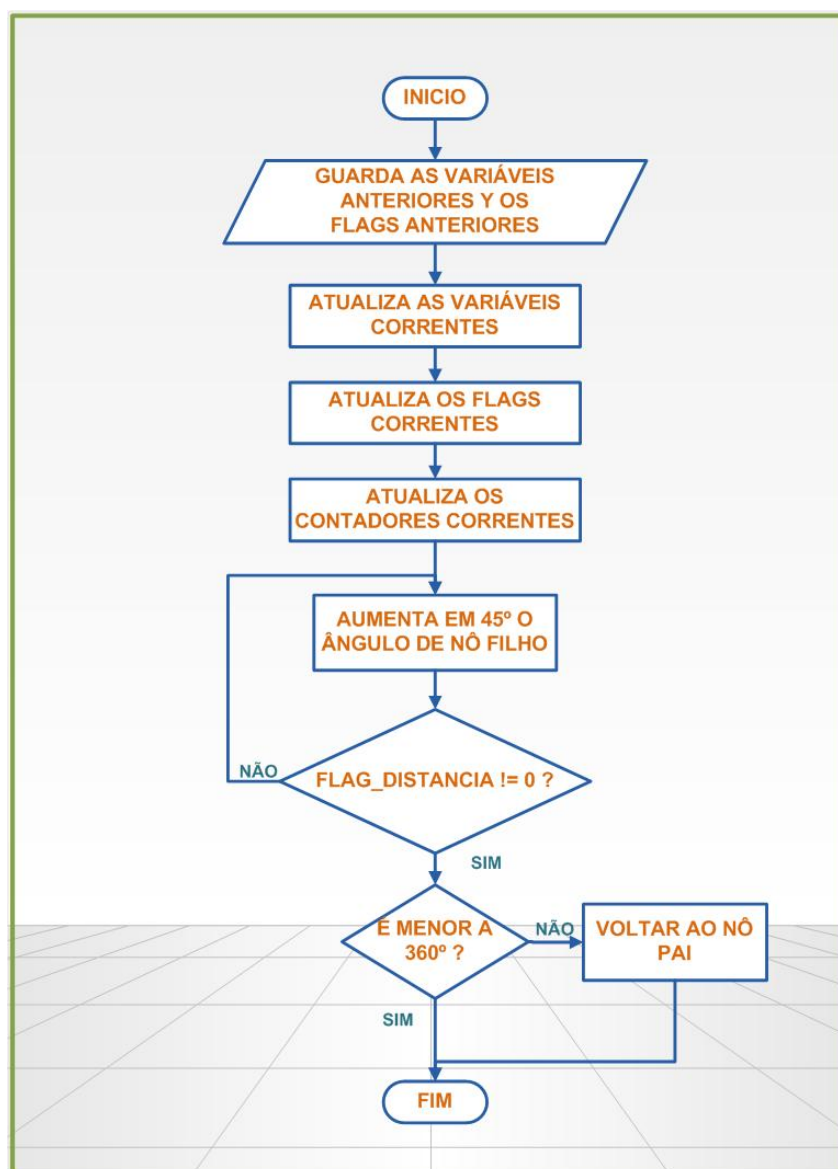


Figura 4.8: Diagrama de fluxo do sistema de Localização

4.4.5

Sistema de navegação para o nó pai

O algoritmo para esta etapa se ilustra na Figura 4.9, o primeiro que se faz é saber o ângulo do nó pai, o ângulo do nó filho atual, a distância percorrida e o nível a retornar; o robô gira o ângulo para retornar ao nó pai, logo executa o sistema *fuzzy* de velocidade até chegar ao nó pai, quando o robô está perto ao nó pai, se faz a leitura dos sensores infravermelhos e a aquisição dos descritores *SIFT* com uma velocidade do robô é mínima para comprovar a chegada do robô ao nó pai.

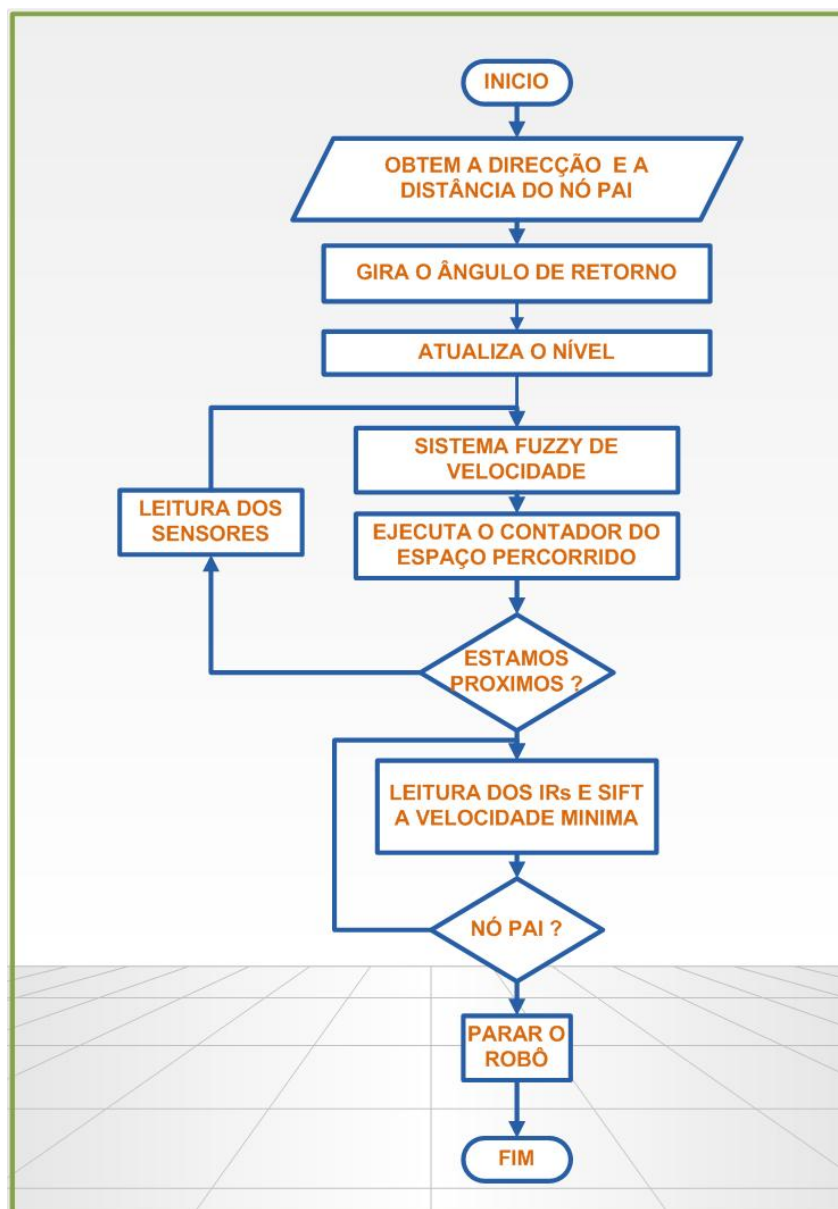


Figura 4.9: Diagrama de fluxo do sistema de Navegação ao nó pai

4.4.6 Sistema de Aprendizagem

Na parte do aprendizagem dos nós é feito pela rede neuronal que se ilustra na Figura 4.10, onde as entradas são as leituras dos sensores codificados em sistema binário e os descritores *SIFT* normalizados, o número de entradas à rede neuronal é 42; as variáveis de saída são toda a informação que pode obter-se do nó (o nível onde se encontra, o ângulo pai de onde se deriva o nó atual, o ângulo atual, a distância desde o nó pai, e o número de nós filhos que tem o nó atual); estas variáveis de saída estão codificadas em sistema binário e o número das variáveis de saída são 23.

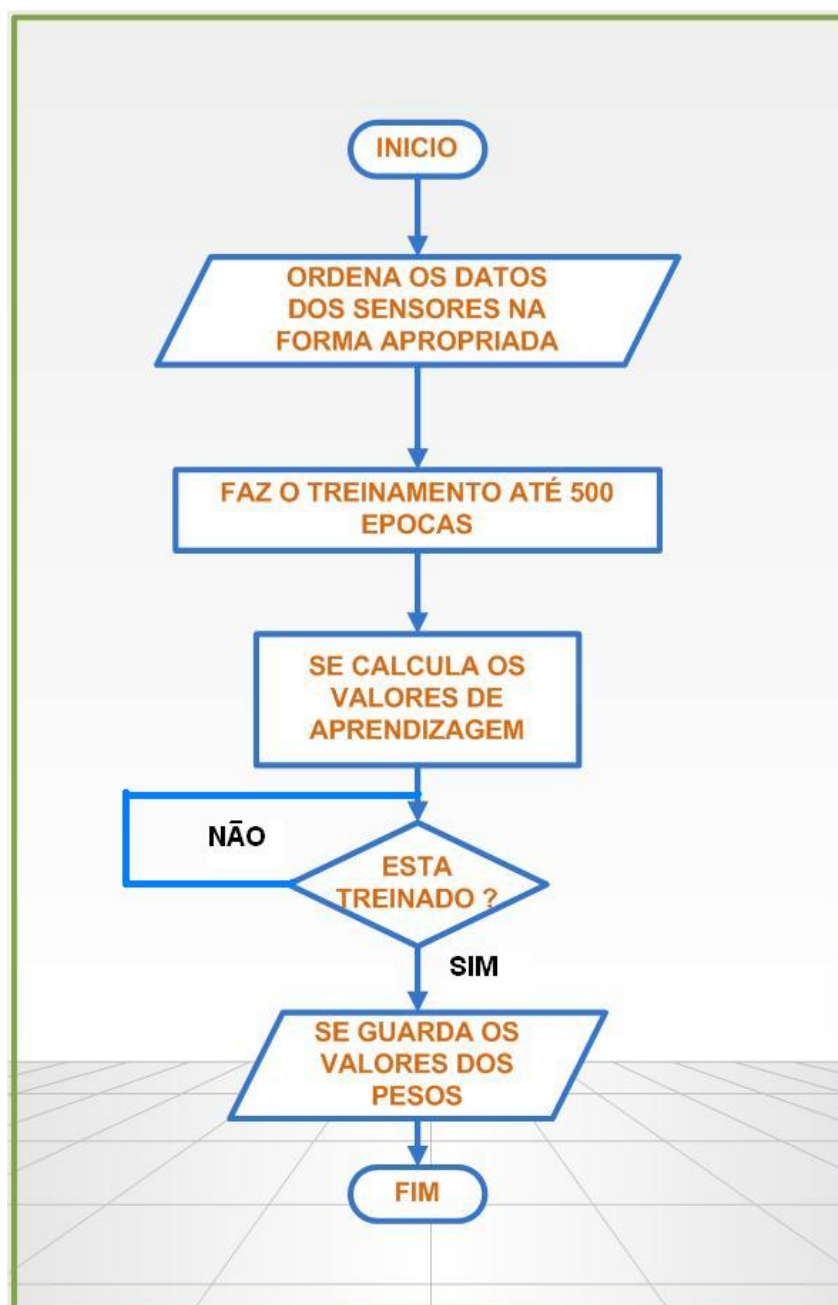


Figura 4.11: Diagrama de fluxo do sistema de Aprendizagem

4.5

Controlador Fuzzy

O controlador fuzzy foi desenhado baixo a suposição de um ambiente estático, é dizer não existe obstáculos que estão no movimento; existem dois controladores independentes fuzzy no robô *ER1*, um controlador para o movimento angular do robô e outro controlador para a rapidez do robô, cada um de estes controladores executam seu control diretamente sobre os motores *brushless*, permitindo assim controlar o movimento e a navegação do robô *ER1*.

4.5.1

Entradas fuzzy

O controlador fuzzy consiste de uma regra total de 27 regras, os quais representam instruções para estimar o comportamento do robô em diferentes situações. Os valores de as entradas são o ângulo α , quem esta entre o vector de percepção geral e a velocidade do robô, a intensidade do vector percepção geral p , e seu mudança no tempo $\dot{p}$.

O robô classifica as situações usando a percepção descrita no item anterior. Para este propósito α , p e $\dot{p}$ são chamados "*angulopercepo*", "*percepo*" e "*mudapercepo*" respectivamente cada um deles com seus respectivos conjuntos fuzzy, tal como se ilustra nas Figuras (4.12),(4.13) y (4.14).

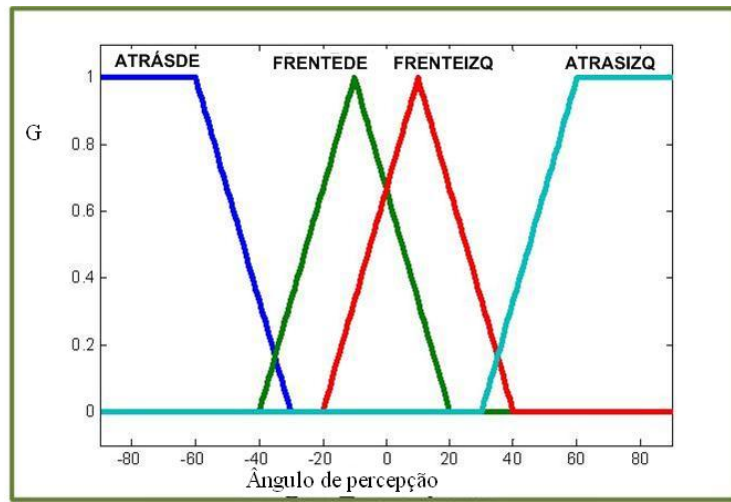


Figura 4.12: Conjuntos fuzzy da variável ângulo de percepção

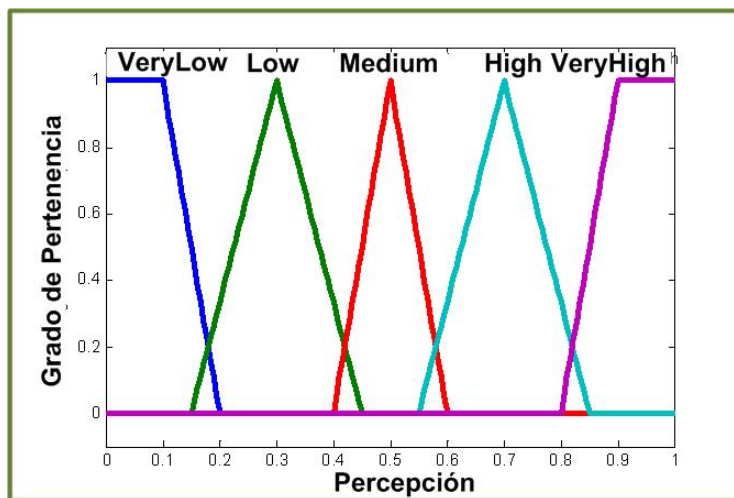


Figura 4.13: Conjuntos fuzzy da variável percepção

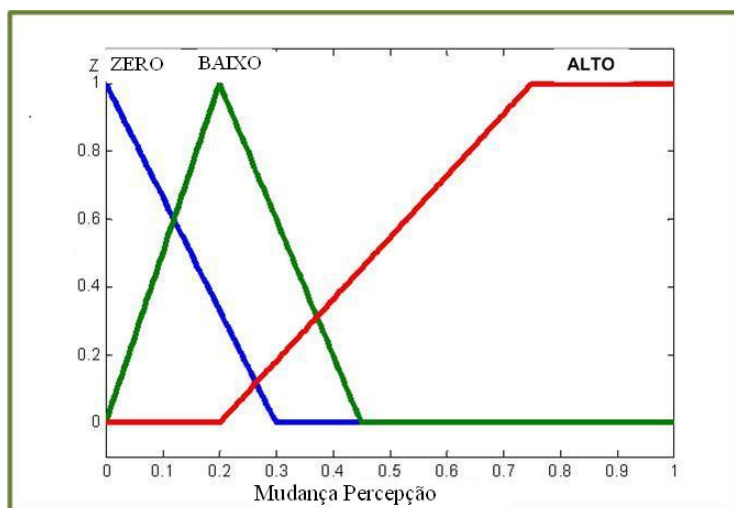


Figura 4.14: Conjuntos fuzzy da variável mudança de percepção

4.5.2

Saídas fuzzy

Las variables de saída fuzzy são a *direction* e a aceleração; a variável de saída fuzzy "direction" executa o controle de velocidade angular do robô desde -60° a 60° respeito ao eixo de velocidade linear do robô. A variável de saída fuzzy "aceleração" executa o controle da aceleração linear do robô, os conjuntos fuzzy das variáveis de saída fuzzy se ilustram nas Figuras (4.15) y (4.16).

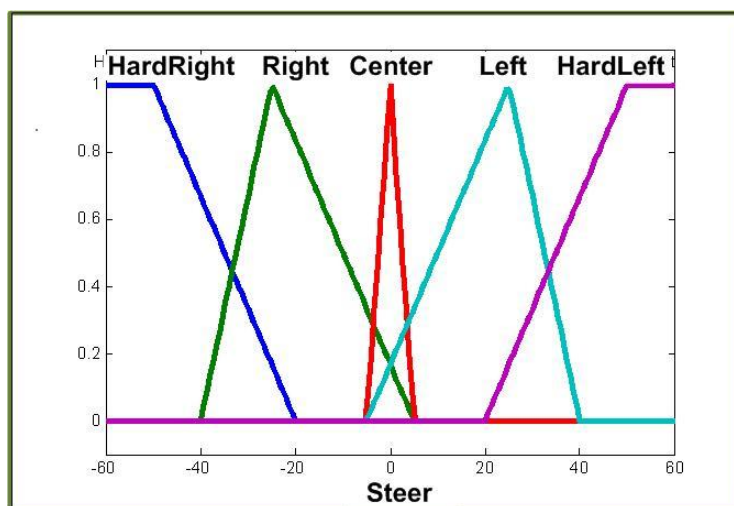


Figura 4.15: Variável de saída fuzzy *Direction*

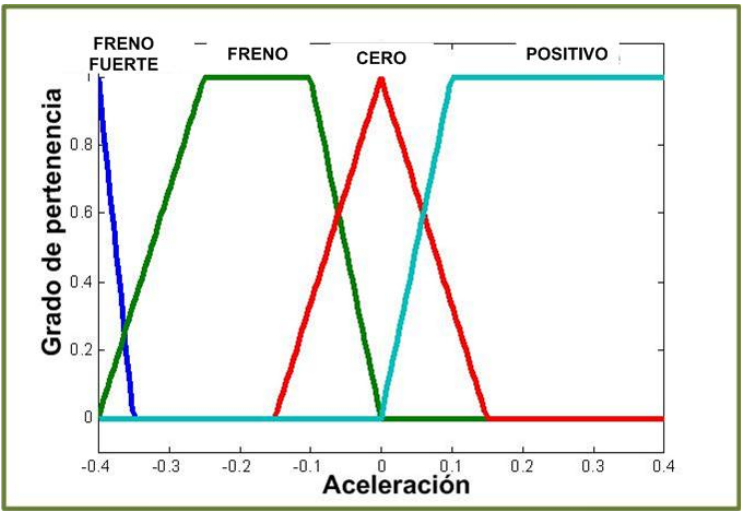


Figura 4.16: Variável de saída fuzzy aceleração

4.5.3
Controle de direção

Neste controle fuzzy se usam como variáveis de entrada o "ângulo da percepção" e a "percepção" e como variável de saída a "direction" tal como se ilustra na Figura 4.17.

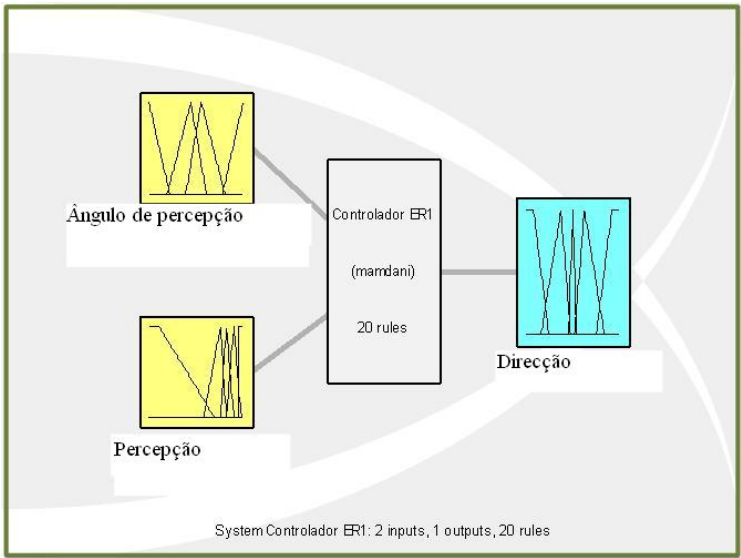


Figura 4.17: Controle de direção do robô ER1

As regras de controle fuzzy para o controle de direção tratam de manter o robô separado dos obstáculos e navegue até espaços livres do ambiente desconhecido, a tabela 4.1 amostra todas as regras usadas neste controle.

Tabela 4.1: Regras do controle de direção do robô ER1

α/p	VL	L	M	H	VH
RB	HR	R	C	R	C
RF	R	R	L	L	HL
LF	HL	L	R	R	HR
LB	HL	L	C	L	C

Tabela 4.2: Regras do controle de velocidade do robô ER1

$\dot{p}/p$	VL o VH	Lo H	ME	—
ZE	ZE	P	P	—
L	EB	B	Z	—
H	—	—	—	EB

4.5.4
Controle de velocidade

Este controle tem 7 regras, que se ilustram na tabela 4.2 , este control tem como entradas às variáveis fuzzy "percepção"e "mudança de percepção"as quais controlam a variável de saída "aceleração". Este controle faz acelerar o robô o diminuir a aceleração do robô; a gráfica do controle de velocidade se ilustra na Figura 4.18

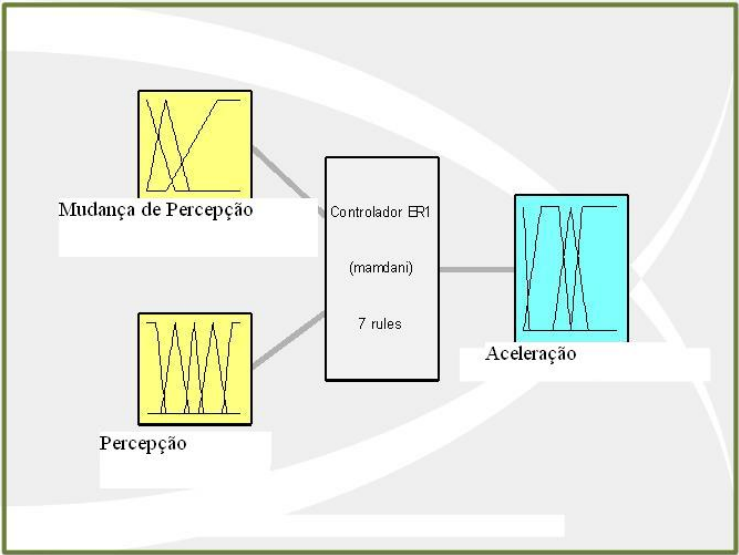


Figura 4.18: Controle de velocidade do robô ER1