

Universidade Federal de Itajubá

Sistema Inteligente Paraconsistente para Controle de Robôs Móveis Autônomos

Cláudio Rodrigo Torres

Orientador: Prof. Dr. Germano Lambert Torres

Co-orientador: Prof. Dr. Jair Minoro Abe

Dissertação apresentada à
Universidade Federal de Itajubá,
para obtenção do título de Mestre
em Engenharia Elétrica.

Novembro/04

SUMÁRIO

Agradecimentos.....	iv
Resumo.....	v
Abstract.....	vi
Capítulo 1 – Apresentação	
1.1 – Considerações iniciais.....	1
Capítulo 2 – Lógica Paraconsistente Anotada	
2.1 – Considerações iniciais.....	4
2.2 – Lógica Paraconsistente, Paracompleta e Não-alética	6
2.3 – A Lógica Proposicional Paraconsistente Anotada <i>Et</i>	7
Capítulo 3 – Robô Móvel Autônomo <i>Emmy</i>	
3.1 – Considerações iniciais	21
3.2 – Robô <i>Emmy</i>	22
3.3 – Sistema de controle do robô móvel <i>Emmy</i>	23
3.4 – Funcionamento básico.....	25
3.5 – Programação do comportamento do robô.....	27
3.6 – Conclusões.....	32
Capítulo 4 – Sistema de Controle do Robô Móvel Autônomo <i>Emmy II</i>	
4.1 – Considerações iniciais.....	35
4.2 – Controle de direção.....	35
4.3 – Controle de velocidade.....	40

Capítulo 5 – Construção Física do Robô Móvel Autônomo <i>Emmy II</i>	
5.1 – Considerações iniciais.....	44
5.2 – Descrição do robô móvel autônomo <i>Emmy II</i>	45
5.2.1 – Circuito de alimentação.....	45
5.2.2 – Circuito dos sensores.....	47
5.2.3 – Circuito de controle.....	48
5.2.4 – Circuito de acionamento.....	50
5.3 – Circuito completo do robô <i>Emmy II</i>	54
 Capítulo 6 – Programação do Robô <i>Emmy II</i>	
6.1 – Considerações iniciais.....	57
6.2 – Determinação do grau de evidência favorável	57
6.3 – Determinação do grau de evidência contrária	58
6.4 – Algoritmo <i>Para-analisador</i>	59
6.5 – Testes	61
 Capítulo 7 – Conclusões.....	65
 Referências.....	67
Anexo I.....	71

AGRADECIMENTOS

Nesta oportunidade, gostaria de expressar agradecimento às seguintes pessoas:

Ao Prof. Dr. Germano Lambert Torres pela oportunidade de estudar na UNIFEI e pelo apoio e ajuda na realização deste trabalho.

Ao Prof. Dr. Jair Minoro Abe pelas correções e sugestões feitas nas muitas reuniões realizadas durante a confecção deste trabalho.

Ao Prof. Dr. João Inácio da Silva Filho que me proporcionou as primeiras oportunidades, ainda na graduação, de estudar os assuntos que deram origem a este trabalho.

Ao Jarbas Roberto Pecoraro que me ajudou na construção do robô *Emmy II*.

Aos amigos Marcos Roberto Bombacini e Alexandre Scalzitti cujos conselhos contribuíram diretamente para a realização deste trabalho.

A muitas outras pessoas que durante minha vida me ajudaram, às vezes sem consciência, com suas palavras e atitudes.

A minha família que sempre me proporcionou as condições necessárias para eu seguir o caminho escolhido.

RESUMO

Nesta dissertação apresenta-se um controlador lógico baseado na lógica paraconsistente anotada evidencial $E\tau$ - *Paracontrol* que constitui uma variação do analisador lógico estudado em [Da Silva Filho 99]. Também se apresenta um robô móvel autônomo, denominado *Emmy II*, utilizado para demonstrar as novas propriedades do *Paracontrol*.

Como característica inovadora o *Paracontrol* apresenta, além das características do controlador anterior (manipulação de informações incertas, contraditórias ou paracompletas), controle de velocidade nas diversas ações do robô.

PALAVRAS CHAVE

Lógica Paraconsistente Anotada, Sistema de Controle, Sistema Inteligente, Robô

Móvel Autônomo.

ABSTRACT

This thesis intends to show a controller based on the evidential annotated paraconsistent logic $E\tau$ – *Paracontrol*. The *Paracontrol* is a variation of the logic analyzer studied in [Da Silva Filho 99]. This work also shows an autonomous mobile robot, which is named *Emmy II*, in order to demonstrate the *Paracontrol*'s new properties.

As an innovation, the *Paracontrol* presents besides the characteristics of the previous controller (manipulation of uncertainties, contradiction and paracompleteness informations), the speed control in the various robot's actions.

Key Words

Paraconsistent Annotated Logic, Control System, Intelligent System, Autonomous Mobile
Robot

Capítulo 1

Apresentação

1.1 – Considerações iniciais

Os Sistemas Inteligentes tradicionais são usualmente baseados na lógica clássica ou em algumas de suas extensões. Assim, em certas situações, quando defrontamos com incertezas, contradições (inconsistência) ou paracompletezas, muitas vezes algumas simplificações ou adaptações são necessárias para que o sistema continue funcionando, podendo torná-lo complexo ou, paradoxalmente, ineficiente.

Por outro lado, nos sistemas de controle de robôs móveis autônomos são comuns o aparecimento de incerteza, inconsistência e paracompleteza. No tocante ao tratamento da incerteza, grandes avanços foram alcançados graças à descoberta da lógica fuzzy e outros sistemas lógicos não-clássicos [Kubota, Hisajima, Kojima & Fukuda 03], [Mills, Walker & Himebaugh 03], [Sossai & Chemello 03]. Conflitos aparecem por diferentes motivos. Eles podem surgir de diversos modos: dados contraditórios, limites de recursos, limitação dos sensores, etc. Conflitos podem ser expressos como meras diferenças ou contradição lógica. Em [Da Silva Filho 99] propôs-se a construção de um robô móvel autônomo, chamado Emmy, cujo funcionamento baseia-se na Lógica Paraconsistente Anotada. O sistema de controle utilizado na Emmy é capaz de lidar com as situações de incerteza, inconsistência e paracompleteza de modo não-trivial. O sistema proposto utiliza como base o controlador lógico paraconsistente *Paracontrol* [Da Silva Filho 99].

Este trabalho possui como objetivo propor um sistema de controle para um robô móvel autônomo semelhante ao descrito em [Da Silva Filho 99], porém utilizando uma variação do *Paracontrol*.

Devido à estrutura física do robô utilizado, o sistema de controle estudado utiliza seis estados lógicos ao invés dos 12 estados lógicos utilizados no *Paracontrol* da *Emmy*. Além disso, ele possui controle de velocidade, o que não ocorre no robô *Emmy*.

O *Paracontrol* é um controlador lógico que constitui a materialização em circuito eletro-eletrônico do algoritmo *Para-analisador* [Abe & Da Silva Filho 01a], [Abe & Da Silva Filho 01b], [Da Silva Filho & Abe 99a], [Da Silva Filho & Abe 99b], [Da Silva Filho & Abe 99c], [Da Silva Filho & Abe 01a], [Da Silva Filho & Abe 01b], [Da Silva Filho & Abe 01c], [Da Silva Filho & Abe 01e]. O *Para-analisador* possui sua estrutura baseada na Lógica Paraconsistente Anotada Evidencial, da qual falaremos mais adiante com detalhe, e possui basicamente duas entradas (uma, a evidência favorável e a outra, a evidência contrária) e possui como saída 12 estados (incluindo entre eles, os estados verdadeiro, falso, inconsistente e paracompleto, além de estados “intermediários”) que constituem, simplificadaamente, os estados de tomada de decisão do robô *Emmy* [Abe & Da Silva Filho 03].

A *Emmy* possui dois sensores ultra-sônicos: um para determinar o grau de evidência favorável e o outro para determinar o grau de evidência contrária. Com o *Paracontrol*, a *Emmy* pode agir adequadamente a certas situações “especiais”, tais como frente a dados contraditórios: um sensor pode detectar um obstáculo à frente (por exemplo, uma parede) enquanto o outro pode não detectar obstáculo algum (por exemplo, pode estar em direção a uma porta aberta).

Havendo essa situação, a Emmy pára e gira 45° na direção que está mais livre. Assim, se na nova medição, não houver inconsistência, ela poderá tomar outra decisão, por exemplo, de se locomover à frente, contornando-se, assim, o obstáculo.

O presente trabalho apresenta algumas características adicionais ao controlador comentado, podendo ser classificado como um refinamento e melhoria no que tange à locomoção do robô Emmy.

O sistema de controle proposto, o qual manteremos o nome *Paracontrol*, utiliza 6 (seis) estados lógicos e apresenta-se como novidade alguns comandos não existentes na Emmy:

- 1) Controle de velocidade nas diversas ações: por exemplo, detectando-se um obstáculo à frente, o novo *Paracontrol* permite que o robô freie escalonadamente, permitindo uma parada “mais suave”. Também, frente a dados contraditórios como aludidos no início, o robô gira “mais suavemente”.
- 2) O novo controlador permite movimentos retrógrados. Em certas situações o robô pode se movimentar para trás ou girar com uma roda fixa e outra girando para trás, permitindo ao robô realizar manobras mais fluentes que as da Emmy.
- 3) A combinação das duas características acima, mais as demais apresentadas no protótipo original, faz do novo protótipo um robô com movimentos mais “sofisticados” que o anterior, dando-se assim, mais um passo importante no que se espera de um robô móvel autônomo.

O robô móvel autônomo construído com o novo *Paracontrol* denomina-se Emmy II.

Como este trabalho baseia-se fortemente na Lógica Paraconsistente Anotada, pensamos ser importante fazer uma breve apresentação da mesma, o que será feito no capítulo seguinte.

Capítulo 2

Lógica Paraconsistente Anotada

2.1 – Considerações iniciais

A lógica clássica surgiu, ao que tudo indica, por volta de 384-322 a.C. com Aristóteles e possui os seguintes princípios básicos, entre outros:

- Princípio da identidade: todo objeto é idêntico a si mesmo.
- Princípio da contradição¹: de duas proposições contraditórias (i.e., uma é a negação da outra) uma delas deve ser falsa.
- Princípio do meio (ou do terceiro) excluído: de duas proposições contraditórias uma delas deve ser verdadeira.

Em 1910, o lógico russo Nicolai A. Vasil'ev (1880 – 1940) e o lógico polonês Jan Lukasiewicz (1878 – 1956), publicaram independentemente, trabalhos que trataram de lógicas que admitem contradições, porém, em nível Aristotélico.

Em 1948, o lógico polonês Stanislaw Jaskowski (1906 – 1965) formalizou com base na lógica discursiva (ou discussiva) um cálculo proposicional paraconsistente denominado cálculo proposicional discursivo. Independentemente, o lógico brasileiro Newton C. A. da Costa (1929 -) em 1954 desenvolveu vários sistemas paraconsistentes contendo todos os níveis lógicos usuais: cálculo proposicional, cálculo de predicados, cálculo de predicados com igualdade, cálculo de descrições e lógica de ordem superior (na forma de teoria dos conjuntos).

¹ Alguns autores o denominam de Princípio da não-contradição.

Com a proliferação de sistemas lógicos alternativos à lógica clássica, hodiernamente podemos conceber a ciência Lógica como a ciência que engloba os diversos sistemas lógicos (clássica, paraconsistente, fuzzy², etc.). Assim, sem muito rigor, a ciência Lógica pode ser dividida em duas classes: a lógica dedutiva e a lógica indutiva.

A lógica dedutiva estuda as inferências logicamente necessárias (ou válidas), de modo que, se as premissas forem verdadeiras, a conclusão necessariamente também é verdadeira. Esta categoria de lógica, por sua vez, pode ser dividida em dois grandes grupos: a lógica dedutiva clássica e a lógica dedutiva não-clássica.

A parte nuclear da lógica dedutiva clássica constitui-se no estudo do cálculo de predicados de primeira ordem e alguns de seus importantes sub-sistemas tais como o cálculo proposicional clássico e o cálculo implicativo clássico.

A lógica dedutiva não-clássica pode ser dividida em dois tipos de estudos:

- a) aquele que complementa o escopo da lógica clássica. Inclui-se nesta categoria os diversos sistemas modais, tais como, lógica do conhecimento, lógica deôntica, lógica temporal e outras;
- b) aquele que substitui a lógica clássica em alguns de seus pontos ou na maioria de seu domínio. Este último ramo chama-se lógicas rivais ou heterodoxas. Inclui-se nesta categoria: as diversas lógicas polivalentes (ou multivaloradas), lógicas fuzzy, lógicas paraconsistentes, lógicas anotadas, etc.

Como nesta dissertação a lógica paraconsistente possui papel importante, tecemos as considerações a seguir.

² Outras denominações correntes na literatura são: teoria dos conjuntos Fuzzy, lógica nebulosa, lógica difusa, etc.

2.2 – Lógica Paraconsistente, Paracompleta e Não-alética

A lógica paraconsistente pode ser definida como se segue. Seja T uma teoria fundada sobre uma lógica L , e suponha-se que a linguagem de L e T contenha um símbolo para a negação (se houver mais de uma negação, uma delas deve ser escolhida pelas suas características lógico-matemáticas). A teoria T diz-se *inconsistente* se ela possuir teoremas contraditórios, i.e., tais que um é a negação do outro; caso contrário, T diz-se *consistente*. A teoria T diz-se *trivial* se todas as fórmulas de L (ou todas as fórmulas fechadas de L) forem teoremas de T ; em hipótese contrária, T chama-se *não-trivial*.

Uma lógica L chama-se *paraconsistente* se puder servir de base para teorias inconsistentes, mas não-triviais.

Uma lógica L denomina-se *paracompleta* se puder ser a lógica subjacente de teorias nas quais se infringe o princípio da contradição, i. e., existem fórmulas tais que elas e suas negações são ambas falsas. De modo preciso, uma lógica se diz *paracompleta* se nela existirem teorias não-triviais maximais às quais não pertencem uma dada fórmula e sua negação.

Finalmente, uma lógica L denomina-se *não-alética* se L for *paraconsistente* e *paracompleta*. A não adequação da lógica clássica no desenvolvimento de teorias T inconsistentes deve-se ao fato de que, se T é uma axiomatização de uma teoria inconsistente, então T é trivial e vice-versa.

No escopo das aplicações, uma questão indesejável da lógica clássica é sua fragilidade. Com efeito, pode-se demonstrar que a presença de uma contradição na lógica clássica, trivializa qualquer teoria baseada na mesma. Conseqüentemente, a lógica clássica é impotente para manipular diretamente o conceito de inconsistência, ficando impossibilitada de se fazer um tratamento não-trivial na presença de contradições. Em

consequência, quando necessitamos tratar inconsistências diretamente, precisamos lançar mão das lógicas paraconsistentes.

2.3 – A Lógica Proposicional Paraconsistente Anotada *Et*

As lógicas anotadas são uma família de lógicas não-clássicas inicialmente utilizadas em programação lógica [Subrahmanian 87].

Subseqüentemente, tais lógicas foram estudadas do ponto de vista de seus fundamentos por N. C. A. da Costa, J. M. Abe, S. Akama e outros [Da Costa, Abe & Subrahmanian 91], [Abe 92], [Sylvan & Abe 98].

Por volta de 1993, Abe iniciou um estudo de aplicabilidade dos sistemas anotados e juntamente com Prado e Ávila implementou uma Programação Lógica Paraconsistente – Paralog [Ávila 96], [Da Costa, Prado, Abe, Ávila & Rillo 95], [Prado 96], [Ávila, Abe & Prado 97] independentemente de [Subrahmanian 87]. Tais idéias foram aplicadas na construção e especificação de um protótipo de arquitetura em Inteligência Artificial que integra vários sistemas computacionais – planejadores, base de conhecimentos, sistema de visão, etc., de uma célula de manufatura [Prado 96] e em representação de conhecimento por Frames, permitindo o tratamento de exceções e inconsistências [Ávila 96].

Em [Abe & Da Silva Filho 98], [Da Silva Filho 97] foram implementados circuitos digitais (portas lógicas NOT, AND, OR) inspiradas nas lógicas anotadas $P\tau$. Em sua estrutura, tais circuitos admitem sinais “inconsistentes” implementados de modo não-trivial. Também, uma variação da programação lógica anotada foi empregada em um protótipo de verificação de segurança de tráfego de trens, semáforos inteligentes e controladores de pouso de aeronaves [Nakamatsu, Abe & Suzuki 99c].

As lógicas anotadas também abarcam muito dos conceitos envolvidos em raciocínio não-monotônico, *defeasible*, *default* e deôntico [Abe, Ávila & Nakamatsu 99], [Nakamatsu, Abe & Suzuki 01]. Axiomatizações de versões da teoria dos conjuntos Fuzzy baseadas nas lógicas anotadas também foram desenvolvidas [Akama & Abe 00b] e a teoria anotada de conjuntos envolve totalmente a teoria de conjuntos Fuzzy [Da Costa, Abe & Subrahmanian 91], [Abe & Akama 99], [Abe 92].

Desenvolveu-se, também, a teoria das Redes Neurais Artificiais Paraconsistentes em [Da Silva Filho & Abe 01d] que vem sendo aplicada em reconhecimento de padrões e se mostra uma promissora teoria [Abe 97].

O analisador lógico (Para-analisador), controlador lógico (Paracontrol), o simulador lógico (Parasim) e outros conceitos em *hardware*, baseado na lógica paraconsistente anotada, permitiram a construção de um protótipo de robô móvel autônomo – Emmy [Da Silva Filho 99], [Da Silva Filho & Abe 01a], da qual falaremos com mais detalhes no capítulo seguinte, pois esta dissertação é um aprimoramento das idéias desenvolvidas do citado robô.

Passamos a apresentar um resumo da lógica proposicional anotada com base na referência [Abe 92], [Abe 01].

Inicialmente, fixamos um reticulado finito denominado de reticulado de valores-verdade, $\tau = \langle |\tau|, \leq \rangle$. Como se sabe, τ é um reticulado se:

1. $\forall x, x \leq x$ (reflexividade)
2. Se $x \leq y$ e $y \leq x \Rightarrow x = y$ (anti-simetria)
3. Se $x \leq y$ e $y \leq z \Rightarrow x \leq z$ (transitividade)
4. $\forall x, y \in |\tau|$, existe o supremo de x e y que denotamos por $x \vee y$.

5. $\forall x, y \in |\tau|$, existe o ínfimo de x e y que denotamos por $x \wedge y$.

Usamos os seguintes símbolos:

\perp : indica o mínimo de τ

\top : indica o máximo de τ

\vee : indica a operação de supremo³

\wedge : indica a operação de ínfimo⁴

Fixamos, também, um operador $\sim : |\tau| \rightarrow |\tau|$ que possui, intuitivamente o “significado” da negação da lógica $P\tau$.

A linguagem de $P\tau$ é composta do seguinte vocabulário:

1) Variáveis proposicionais: um conjunto infinito enumerável.

2) Conectivos lógicos:

\neg (negação)

\wedge (conjunção)

\vee (disjunção)

\rightarrow (implicação)

3) Constantes de anotação: $\theta, \lambda, \mu, \dots$ (elementos do reticulado τ).

4) Símbolos auxiliares: $(,)$.

As fórmulas de $P\tau$ definem-se pela seguinte definição indutiva generalizada:

- a. Se p é uma variável proposicional e λ é uma constante de anotação, então p_λ é uma fórmula atômica anotada.

³ Com respeito a subconjuntos de $|\tau|$.

⁴ Com respeito a subconjuntos de $|\tau|$.

- b. Se A é uma fórmula então $(\neg A)$ é uma fórmula.
- c. Se A e B são fórmulas, então $(A \wedge B)$, $(A \vee B)$, e $(A \rightarrow B)$ são fórmulas.
- d. Uma expressão é uma fórmula se e somente se for obtida pela aplicação de uma das cláusulas 1, 2, ou 3 acima.

Lê-se a fórmula $(\neg A)$ como “A negação de A ”.

Lê-se a fórmula $(A \wedge B)$ como “A conjunção de A e B ”.

Lê-se a fórmula $(A \vee B)$ como “A disjunção de A e B ”.

Lê-se a fórmula $(A \rightarrow B)$ como “A implicação de B por A ”.

Intuitivamente uma fórmula atômica p_μ é lida como: “creio na proposição p com grau de crença⁵ de no máximo m ou até m ”.

Se p é uma letra proposicional e $\mu \in |\tau|$, então uma fórmula do tipo $\underbrace{p}_k p_m$ onde,

$k \geq 0$, denomina-se *hiper-literal* (ou simplesmente *literal*). As demais fórmulas denominam-se *fórmulas complexas*.

A semântica das lógicas $P\tau$ apresenta-se resumidamente assim:

Uma interpretação relativa a $P\tau$ é uma função $I : P \rightarrow |\tau|$ (onde P é o conjunto das variáveis proposicionais).

A cada interpretação I , associamos uma *valoração*, $V_1 : F \rightarrow \{0, 1\}$, onde F é o conjunto de todas as fórmulas.

A valoração V_1 é definida indutivamente por:

- 1) Se p é uma letra proposicional e μ uma constante de anotação, então:

$$V_1(p_\mu) = 1 \Leftrightarrow I(p) \geq \mu$$

⁵ Pode haver outras leituras tais como grau de evidência. Consultar também [Dempster 68] para generalização de inferência Bayesiana.

$$V_1(P_\mu) = 0 \Leftrightarrow \text{N\~{a}o \acute{e} que } I(p) \geq \mu$$

$$V_1(\neg^k p_\mu) = V_1(\neg^{k-1} p_{\sim\mu}) \text{ onde, } k \geq 1$$

2) Se A e B s\~{a}o f\~{o}rmulas quaisquer, ent\~{a}o:

$$V_1(A \rightarrow B) = 1 \text{ se e somente se } V_1(A) = 0 \text{ ou } V_1(B) = 1$$

$$V_1(A \wedge B) = 1 \text{ se e somente se } V_1(A) = 1 \text{ e } V_1(B) = 1$$

$$V_1(A \vee B) = 1 \text{ se e somente se } V_1(A) = 1 \text{ ou } V_1(B) = 1$$

Defini\~{c}o\~{e}s:

- Uma interpreta\~{c}\~{a}o relativa a $P\tau, I : P \rightarrow |\tau|$, se diz *inconsistente* se existir $p \in P$ e $\mu \in |\tau|$ tal que: $V_1(p_\mu) = 1 = (\neg p_\mu)$.
- Uma interpreta\~{c}\~{a}o relativa a $P\tau, I : P \rightarrow |\tau|$, se diz *n\~{a}o-trivial* se existir $p \in P$ e $\mu \in |\tau|$ tal que: $V_1(p_\mu) = 0$.
- Uma interpreta\~{c}\~{a}o relativa a $P\tau, I : P \rightarrow |\tau|$, se diz *paraconsistente* se for inconsistente e n\~{a}o-trivial.
- Uma interpreta\~{c}\~{a}o relativa a $P\tau, I : P \rightarrow |\tau|$, se diz *paracompleta* se existir uma f\~{o}rmula at\~{o}mica p_μ tal que $V_1(p_\mu) = 0 = V_1(\neg p_\mu)$.
- A l\~{o}gica $P\tau$ se diz *paraconsistente* se ela admitir uma interpreta\~{c}\~{a}o paraconsistente.
- A l\~{o}gica $P\tau$ se diz *paracompleta* se ela admitir uma interpreta\~{c}\~{a}o paracompleta.
- A l\~{o}gica $P\tau$ se diz *n\~{a}o-al\~{e}tica* se ela admitir uma interpreta\~{c}\~{a}o paraconsistente e uma interpreta\~{c}\~{a}o paracompleta.

Nesta disserta\~{c}\~{a}o fixamos um reticulado particular $\tau = \langle |\tau|, \leq \rangle$ que \acute{e} o seguinte:

a) $|\tau| = [0, 1] \times [0, 1]$ (onde $[0, 1]$ é o intervalo unitário real).

b) A relação de ordem sobre $|\tau|$ define-se como $(\mu_1, \lambda_1) \leq (\mu_2, \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ e $\lambda_1 \leq \lambda_2$, onde \leq é a relação de ordem usual dos números reais.

c) O operador $\sim: |\tau| \rightarrow |\tau|$ é definida como $\sim(\mu, \lambda) = (\lambda, \mu)$

A constante de anotação (μ, λ) recebe uma leitura como se segue: μ é o grau de evidência favorável ou grau de crença favorável da proposição p e λ o grau de evidência contrária ou grau de crença contrária da proposição p .

Assim, se $p \equiv$ “Há obstáculo à frente” e tivermos a anotação $(0.8, 0.3)$, temos a proposição “Há obstáculo à frente” com evidência favorável 0.8 e evidência contrária de 0.3.

Com estas considerações, cada constante anotacional do reticulado representa-se pelo par (μ, λ) , onde:

μ = Grau de evidência favorável.

λ = Grau de evidência contrária.

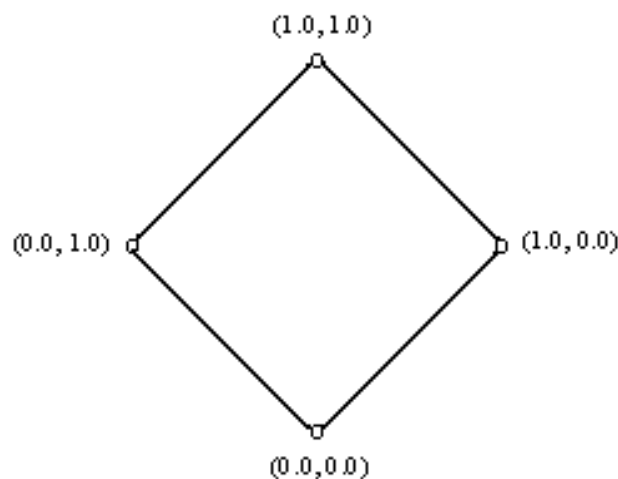


Figura 2.1 – Reticulado

Pontos extremos do reticulado:

Ponto A $\equiv (0.0, 0.0) \Rightarrow$ Paracompleto

Ponto B $\equiv (1.0, 0.0) \Rightarrow$ Verdadeiro

Ponto C $\equiv (0.0, 1.0) \Rightarrow$ Falso

Ponto D $\equiv (1.0, 1.0) \Rightarrow$ Inconsistente

A Lógica Paraconsistente Anotada Evidencial $E\tau$ se apresenta resumidamente assim: as proposições desta lógica são do tipo $p_{(\mu, \lambda)}$ onde p é uma proposição e $\mu, \lambda \in [0, 1]$ (intervalo real unitário fechado). Intuitivamente, μ indica o grau de evidência⁶ favorável de p e λ o grau de evidência contrária de p . A leitura dos valores μ, λ dependem das aplicações consideradas e podem sofrer mudanças: com efeito, μ pode ser o grau de crença⁷ favorável e λ poder ser o grau de crença contrária da proposição p . Como vimos anteriormente, as proposições atômicas $p_{(\mu, \lambda)}$ da lógica $E\tau$ podem ser intuitivamente ser lidas como: creio em p com o grau de crença favorável μ e o grau de crença contrária λ , ou o grau de evidência favorável de p é μ e o grau de evidência contrária de p é λ .

Desse modo, temos algumas leituras interessantes:

- $p_{(1.0, 0.0)}$ pode ser lida como uma proposição verdadeira (evidência favorável total e evidência contrária nula).

- $p_{(0.0, 1.0)}$ pode ser lida como uma proposição falsa (evidência favorável nula e evidência contrária total).

⁶ O termo evidência se encontra empregado num sentido não rigoroso, podendo intuitivamente ser “certeza” manifesta ou dados e informações que suportam opiniões. O termo “grau de evidência” significa o que está explanado no curso do trabalho.

⁷ O termo crença também se encontra empregado em um sentido não rigoroso. Convém ressaltar que usualmente possui uma subjetividade.

- $p_{(1.0, 1.0)}$ pode ser lida como uma proposição inconsistente (evidência favorável total e evidência contrária total).
- $p_{(0.0, 0.0)}$ pode ser lida como uma proposição paracompleta (evidência favorável nula e evidência contrária nula).
- $p_{(0.5, 0.5)}$ pode ser lida como uma proposição indefinida (evidência favorável igual à evidência contrária de 0.5).

Note que o conceito de paracompleteza é o dual da de inconsistência.

Exemplos: Seja a proposição $p \equiv$ “Há obstáculo à frente”. Temos então:

- $p_{(1.0, 0.0)}$ pode ser lida como “Há obstáculo à frente” com evidência favorável total e evidência contrária nula. Intuitivamente, trata-se de uma proposição verdadeira.
- $p_{(0.0, 1.0)}$ pode ser lida como “Há obstáculo à frente” com evidência favorável nula e evidência contrária total. Intuitivamente, trata-se de uma proposição falsa.
- $p_{(1.0, 1.0)}$ pode ser lida como “Há obstáculo à frente” com evidência favorável total e evidência contrária também total. Intuitivamente, trata-se de uma proposição contraditória.
- $p_{(0.0, 0.0)}$ pode ser lida como “Há obstáculo à frente” com evidência favorável nula e evidência contrária também nula. Intuitivamente, trata-se de uma proposição paracompleta.
- $p_{(0.5, 0.5)}$ pode ser lida como “Há obstáculo à frente” com evidência favorável idêntica á evidência contrária e é 0.5. Intuitivamente, temos aí uma indefinição.

Uma propriedade interessante que a lógica $E\tau$ apresenta é quando analisamos negações de proposições. Qual é a negação de $p_{(0.5, 0.5)}$? Intuitivamente temos que é a própria proposição $p_{(0.5, 0.5)}$, ou seja, $\neg p_{(0.5, 0.5)} \leftrightarrow p_{(0.5, 0.5)}$. Agora suponhamos que $p_{(0.5, 0.5)}$ seja verdadeira. Logo temos a situação: $p_{(0.5, 0.5)}$ verdadeira e $\neg p_{(0.5, 0.5)}$ também verdadeira. Ora, a lógica em questão admite intuitivamente contradições verdadeiras. Coisa semelhante se passa se $p_{(0.5, 0.5)}$ for falsa. Temos $p_{(0.5, 0.5)}$ falsa e $\neg p_{(0.5, 0.5)}$ também falsa, ou seja, $E\tau$ é também paracompleta. Daí, $E\tau$ é não-alética. De modo geral temos $\neg p_{(\mu, \lambda)} \leftrightarrow p_{(\lambda, \mu)}$. O fato de a negação lógica ser “absorvido” na anotação, faz com que a lógica $E\tau$ tenha propriedades de fundamental importância na hora de implementações físicas, bem como propriedades de extrema fecundidade em programação lógica paraconsistente, como se pode ver em [Ávila 96].

Voltemos a algumas terminologias. O par (μ, λ) denomina-se constante de anotação. Tal par é um elemento de $[0, 1] \times [0, 1]$ (onde $[0, 1]$ é o intervalo unitário fechado real) que algumas vezes indicamos por $[0, 1]^2$. O conjunto $[0, 1]^2$ está munido de uma relação de ordem assim definida: $(\mu_1, \lambda_1) \leq (\mu_2, \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2 \text{ e } \lambda_1 \leq \lambda_2$. O quadrado unitário em questão com a relação de ordem constitui um reticulado que simbolizamos por τ .

No reticulado τ notamos quatro pontos que nortearão nossas perquirições. Denominemos de pontos ou estados cardeais. Tais são elas:

$A \equiv (1.0, 0.0) \equiv$ estado verdadeiro

$B \equiv (0.0, 1.0) \equiv$ estado falso

$C \equiv (1.0, 1.0) \equiv$ estado inconsistente

$D \equiv (0.0, 0.0) \equiv$ estado paracompleto

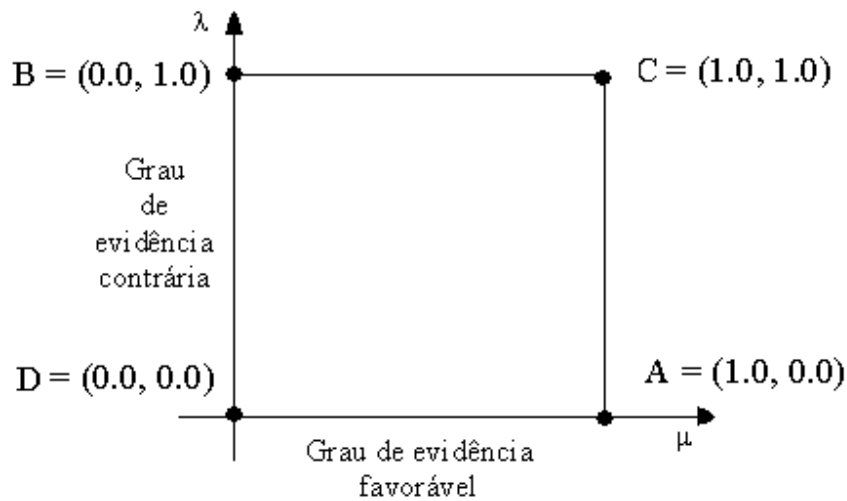


Figura 2.2 – Reticulado representado pelo quadrado unitário no plano cartesiano – QUPC.

Embasados nos estados cardeais, e pelo uso das propriedades dos números reais, vamos cuidadosamente erigir uma estrutura matemática com o fito de materializar nossas idéias de como queremos manipular mecanicamente o conceito de incerteza, contradição e de paracompleteza, entre outros. Tal mecanismo embarcará, naturalmente, de algum modo os estados verdadeiro e falso tratados dentro do escopo da lógica clássica, com todas as suas conseqüências.

Para tanto, iremos introduzir diversos conceitos que julgamos “intuitivos” para a finalidade acima bosquejada.

$$\text{Segmento perfeitamente definido AB: } \mu + \lambda - 1 = 0; 0 \leq \mu, \lambda \leq 1$$

$$\text{Segmento perfeitamente indefinido DC: } \mu - \lambda = 0; 0 \leq \mu, \lambda \leq 1$$

Introduzimos as aplicações $G_{ic}: [0, 1] \times [0, 1] \rightarrow [0, 1]$, $G_{pa}: [0, 1] \times [0, 1] \rightarrow [-1, 0]$, $G_{ve}: [0, 1] \times [0, 1] \rightarrow [0, 1]$, $G_{fa}: [0, 1] \times [0, 1] \rightarrow [-1, 0]$ definidas por:

Grau de Inconsistência:	$G_{ic}(\mu, \lambda) = \mu + \lambda - 1$, desde que $\mu + \lambda - 1 \geq 0$
Grau de Paracompleteza:	$G_{pa}(\mu, \lambda) = \mu + \lambda - 1$, desde que $\mu + \lambda - 1 \leq 0$
Grau de Veracidade:	$G_{ve}(\mu, \lambda) = \mu - \lambda$, desde que $\mu - \lambda \geq 0$
Grau de Falsidade:	$G_{fa}(\mu, \lambda) = \mu - \lambda$, desde que $\mu - \lambda \leq 0$

Vê-se que o Grau de Inconsistência “mede” o quão uma anotação (μ, λ) “distância”⁸ do segmento perfeitamente indefinido e quão se “aproxima” do estado inconsistente e o Grau de Paracompleteza “mede” o quão uma anotação (μ, λ) “distância” do segmento perfeitamente indefinido e quão se “aproxima” do estado paracompleto.

Chama-se Grau de Incerteza $G_{in}(\mu, \lambda)$ de uma anotação (μ, λ) a qualquer um dos graus de inconsistência ou de paracompleteza. Por exemplo, o grau de Incerteza é máximo no estado inconsistente, ou seja, $G_{ic}(1, 1) = 1$.

De modo similar, o Grau de Veracidade “mede” o quão uma anotação (μ, λ) “distancia” do segmento perfeitamente definido e quão se “aproxima” do estado verdade e o Grau de Falsidade “mede” o quão uma anotação (μ, λ) “distancia” do segmento perfeitamente definido e quão se “aproxima” do estado falso.

Chama-se Grau de Certeza $G_{ce}(\mu, \lambda)$ de uma anotação (μ, λ) a qualquer um dos graus de verdade ou de falsidade. Por exemplo, o grau de verdade da anotação $(\frac{1}{2}, \frac{1}{4})$ é $\frac{1}{4}$, ou seja, $G_{ve}(\frac{1}{2}, \frac{1}{4}) = \frac{1}{4}$.

⁸ Usualmente considera-se “distância” uma função não-negativa. Neste caso, é claro, considera-se em módulo. No entanto vamos manter tal termo no correr da exposição e o significado intuitivo não afetará a compreensão das idéias aqui expostas.

Observação (problema em aberto). Altamente instigador é estudar outros modos de considerar os graus de incerteza e de indefinição. Aliás, a observação precedente é mais profunda e anterior: com efeito, a escolha do reticulado é fundamental para as considerações de todos estes estudos no tocante à aplicação prática.

Com os conceitos acima ventilados, podemos trabalhar com “faixas” de verdade ao invés de a “verdade” ser uma coisa hirta, inflexível⁹. Para determinarmos tais faixas, introduzimos os seguintes conceitos:

Quatro valores limites externos:

$$V_{cve} = C_1 = \text{Valor de controle de veracidade; } 0 \leq V_{cve} \leq 1$$

$$V_{cfa} = C_2 = \text{Valor de controle de falsidade; } -1 \leq V_{cfa} \leq 0$$

$$V_{cic} = C_3 = \text{Valor de controle de inconsistência; } 0 \leq V_{cic} \leq 1$$

$$V_{cpa} = C_4 = \text{Valor de controle de paracompleteza; } -1 \leq V_{cpa} \leq 0$$

Tais valores nortearão quando uma proposição é considerada, por exemplo, “verdadeira” no sentido de tomarmos uma decisão positivamente, e assim por diante. A seguinte figura nos ajudará a introduzir conceitos suplementares.

⁹ Aqui também, obviamente, estamos usando o termo “verdade” não no sentido clássico, mas segundo o que se está explanado no texto. O mesmo se aplica aos termos falso, inconsistente e paracompleto. Outros abusos de linguagem estão empregados, porém conscientemente.

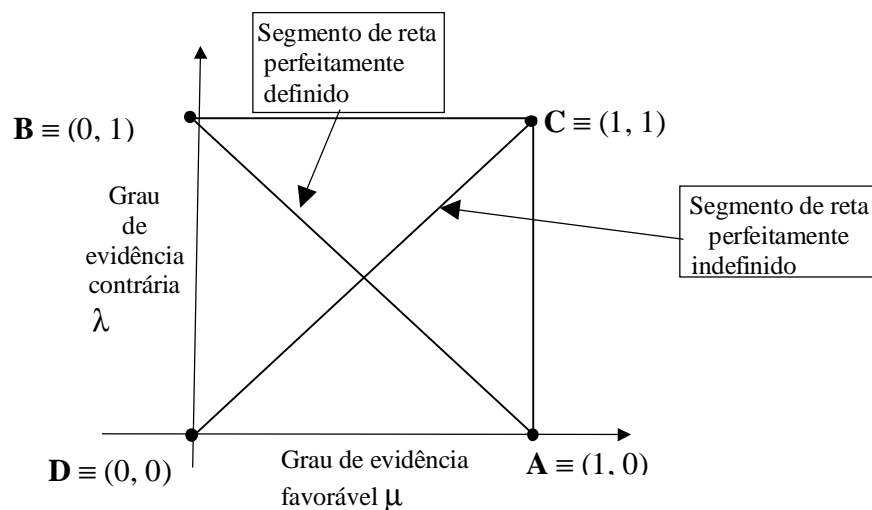


Figura 2.3 – Reticulado representando os segmentos perfeitamente definido e indefinido.

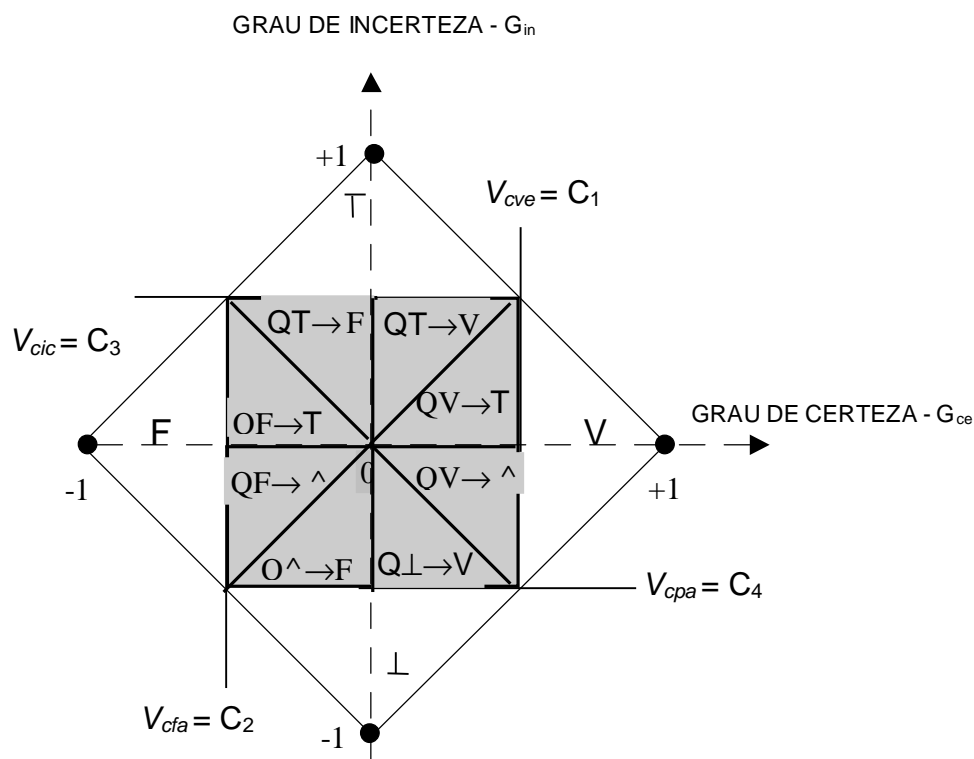


Figura 2.4 - Diagrama com os graus de incerteza e de certeza, com valores ajustáveis de controle limites indicados nos eixos.

Observe também as regiões consideradas na tabela abaixo.

Estados Extremos	Símbolo
Verdadeiro	V
Falso	F
Inconsistente	T
Paracompleto	\perp

Estados Não-Extremos	Símbolo
Quase-verdadeiro tendendo ao Inconsistente	$QV \rightarrow T$
Quase-verdadeiro tendendo ao Paracompleto	$QV \rightarrow \perp$
Quase-falso tendendo ao Inconsistente	$QF \rightarrow T$
Quase-falso tendendo ao Paracompleto	$QF \rightarrow \perp$
Quase-inconsistente tendendo ao Verdadeiro	$QT \rightarrow V$
Quase-inconsistente tendendo ao Falso	$QT \rightarrow F$
Quase-paracompleto tendendo ao Verdadeiro	$Q\perp \rightarrow V$
Quase-paracompleto tendendo ao Falso	$Q\perp \rightarrow F$

Tabela 1 – Símbolos dos Estados Extremos e Não-Extremos

Tais regiões são facilmente caracterizáveis através dos graus de incerteza, de certeza, o que deixaremos de fazer neste texto. Detalhes completos se encontram, por exemplo, em [Da Costa, Abe, Da Silva Filho, Murolo e Leite 99]. Novamente, a subdivisão apresentada é modificável e ajustável de acordo com as características e peculiaridades de cada aplicação, como já observamos anteriormente.

Capítulo 3

Robô Móvel Autônomo *Emmy*

3.1 – Considerações iniciais

As contradições ou inconsistências são comuns quando descrevemos partes do mundo real. Os sistemas de controle utilizados em Automação e Robótica funcionam em geral com base na lógica clássica, onde a descrição do mundo é considerada por apenas dois estados. Estes sistemas binários não conseguem tratar adequadamente as situações contraditórias. Em alguns casos os sistemas de controle clássico são projetados para ignorar estas situações que evidenciam conflito, perdendo informações que poderiam ser importantes para o aumento da eficiência de controle.

Na aplicação do controle clássico em Robótica o tratamento de situações que fogem àquelas convencionais consome um tempo muito longo para serem processadas provocando lentidão nas respostas do robô, diminuindo sua capacidade de reação.

Quando se trata de sistemas de controle para a navegação de robôs móveis autônomos as dificuldades para se obter eficiência são muitas. Isto ocorre porque no projeto de um controle de robô móvel autônomo, envolvem-se várias situações prováveis causadoras de inconsistências, especialmente em uma movimentação por ambiente desconhecido, por exemplo, ao passar perto de paredes, em caminhos estreitos, curvos e em terrenos acidentados ou ainda na presença de tráfego [Shoval, Ulrich & Borenstein 03].

3.2 – Robô Emmy

O robô móvel autônomo *Emmy* consiste de uma plataforma móvel de alumínio de formato circular de 30 cm de diâmetro e 60 cm de altura. O robô foi projetado em módulos sobrepostos separados por função de cada um deles no sistema de controle, facilitando a visualização da ação de cada módulo no controle de movimentação do robô.

Na movimentação do robô *Emmy* em ambiente não-estruturado as informações sobre a existência ou não de obstáculo na sua trajetória são obtidas por intermédio do dispositivo denominado *Parasônico*. O *Parasônico* é capaz de captar obstáculos na trajetória transformando-os proporcionalmente as medidas de distância entre o robô e o obstáculo em sinais elétricos, na forma de uma tensão elétrica contínua que pode variar de 0 a 5 volts.

O *Parasônico* é basicamente composto por dois sensores de ultra-som tipo POLAROID 500 [POL 96] e um Microcontrolador 8051. A função do Microcontrolador 8051 é fazer a sincronização entre as medições dos dois sensores de ultra-som e a transformação da grandeza distância em tensão elétrica.

O sistema de controle de navegação do robô é feito por um Controlador Lógico Paraconsistente – *Paracontrol* construído em *Hardware* e que recebe e faz o tratamento dos sinais elétricos.

O *Paracontrol* recebe informações na forma de graus de evidência favorável e contrária, faz uma análise paraconsistente que resulta em sinais representativos de estados lógicos e graus de certeza e de incerteza. As duas formas de saída podem ser utilizadas no controle, dependendo do projeto. O *Paracontrol* ainda oferece alternativa de ajustes de controle externamente através de potenciômetros.

Com a informação do estado resultante, o microcontrolador decide a ação a ser tomada pelo robô.

A figura 3.1 mostra as partes principais do robô *Emmy*.

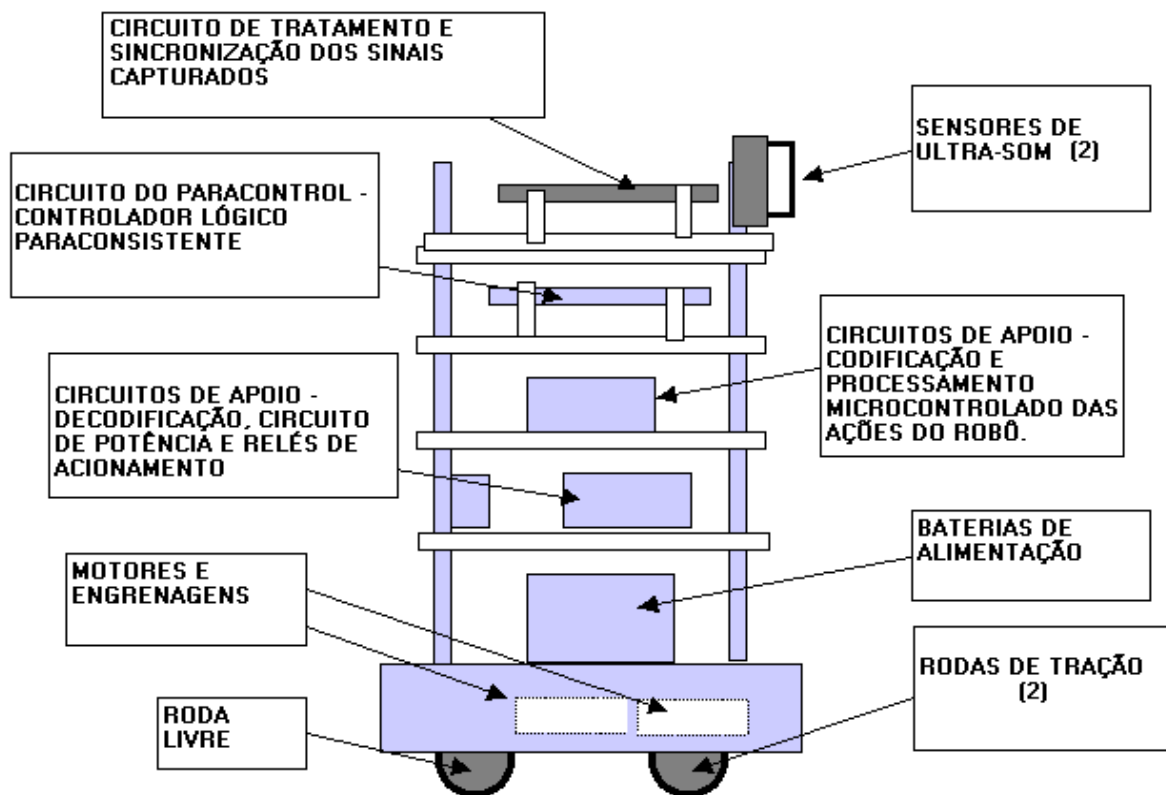


Figura 3.1 – Destaque das partes principais do robô *Emmy*.

3.3 – Sistema de controle do robô móvel *Emmy*

O sistema de controle do robô móvel autônomo *Emmy* utiliza um controlador lógico paraconsistente, denominado *Paracontrol*. Este controlador é inspirado no algoritmo *Para-analisador*, que baseia-se na Lógica Proposicional Paraconsistente Anotada *Et*.

O circuito do *Paracontrol* possui a finalidade de tratar sinais elétricos de acordo com a estrutura do *Para-analisador*. Assim, sinais elétricos são interpretados como grau de evidência favorável (μ) e como grau de evidência contrária (λ).

O algoritmo *Para-analisador* pode ser resumido da seguinte forma:

- Se $G_{ve} \geq V_{cve}$ então V.
- Se $G_{fa} \leq V_{cfa}$ então F.
- Se $G_{in} \geq V_{cic}$ então T.
- Se $G_{pa} \leq V_{cpa}$ então \perp .
- Para $0 \leq G_{ve} < V_{cve}$ e $0 \leq G_{in} < V_{cic}$
 - se $G_{ve} \geq G_{in}$ então $QV \rightarrow T$.
 - se $G_{ve} < G_{in}$ então $QT \rightarrow V$.
- Para $0 \leq G_{ve} < V_{cve}$ e $V_{cpa} < G_{pa} \leq 0$
 - se $G_{ve} \geq |G_{pa}|$ então $QV \rightarrow \perp$.
 - se $G_{ve} < |G_{pa}|$ então $Q\perp \rightarrow V$.
- Para $V_{cfa} < G_{fa} \leq 0$ e $V_{cpa} < G_{pa} \leq 0$
 - se $|G_{fa}| \geq |G_{pa}|$ então $QF \rightarrow \perp$.
 - se $|G_{fa}| < |G_{pa}|$ então $Q\perp \rightarrow F$.
- Para $V_{cfa} < G_{fa} \leq 0$ e $0 \leq G_{in} < V_{cic}$
 - se $|G_{fa}| \geq G_{in}$ então $QF \rightarrow T$.
 - se $|G_{fa}| < G_{in}$ então $QT \rightarrow F$.

3.4 – Funcionamento básico

Apresenta-se na figura 3.2 o sistema de controle paraconsistente do robô *Emmy* composto pelo *Parasônico*, *Paracontrol* e circuitos de apoio.

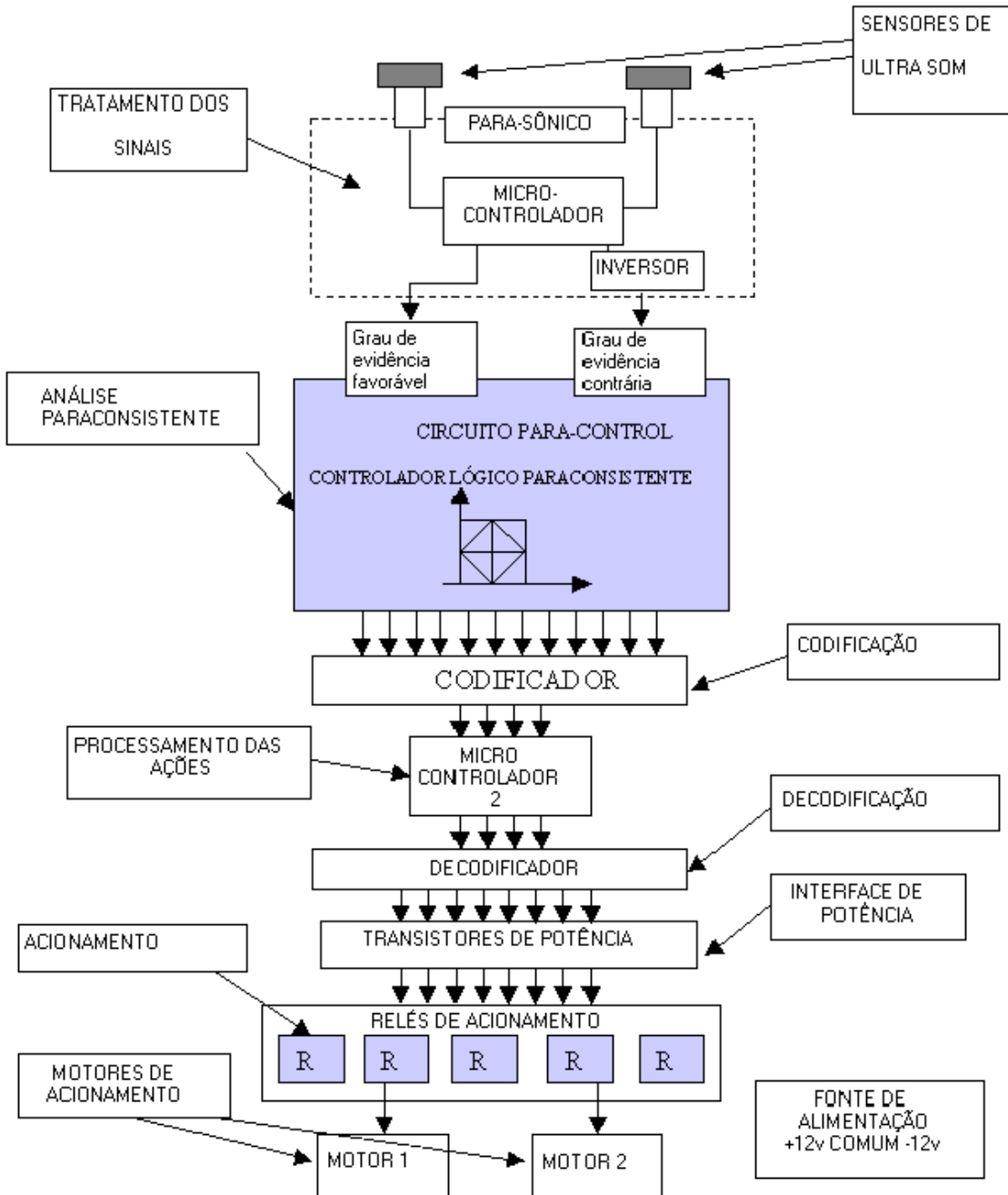


Figura 3.2 – Esquematização do sistema de controle do robô *Emmy*.

A descrição dos componentes do robô *Emmy* é a seguinte:

- Sensores de ultra-som: os dois sensores de ultra-som realizam a detecção da distância entre o robô e o objeto através da emissão de um trem de pulso em frequência de ultra-som e captação do retorno do sinal (eco).

- Tratamento dos sinais: o tratamento dos sinais captados é feito através do *Para-sônico*. O microcontrolador é programado para transformar o tempo decorrido entre a emissão do trem de pulso ultra-sônico e a captação do eco em um sinal elétrico de 0 a 5 Volts para o grau de evidência favorável e de 5 a 0 Volts para o grau de evidência de evidência contrária. A amplitude de cada sinal é proporcional ao tempo decorrido entre a emissão do trem de pulso ultra-sônico e o retorno recebido pelos sensores.

- Análise paraconsistente: o circuito *Paracontrol* faz a análise lógica dos sinais conforme o *Para-analisador*.

- Codificação: o circuito codificador transforma a palavra binária de 12 dígitos em um código de 4 dígitos para ser processada por outro microcontrolador.

- Processamento das ações: o microcontrolador é convenientemente programado para acionar os relés em seqüências que estabelecem ações para o robô.

- Decodificação: o circuito decodificador transforma a palavra binária de 4 dígitos em sinais para acionar os relés nas seqüências programadas.

- Interface de potência: o circuito de interface potência é composto por transistores que amplificam os sinais possibilitando o acionamento dos relés por sinais digitais.

- Acionamento: os relés acionam os motores M_1 e M_2 conforme a palavra binária decodificada.

- Motores de acionamento: os motores M_1 e M_2 movimentam o robô conforme a seqüência de acionamento dos relés.

- Fontes de alimentação: o robô *Emmy* é alimentado por duas baterias formando uma fonte simétrica de ± 12 volts.

Além dos elementos descritos, componentes para circuito de apoio são necessários para permitir que os sinais resultantes da análise paraconsistente sejam direcionados e transformados em ação.

A existência do codificador e do decodificador deve-se ao fato da análise paraconsistente ser realizada por um microcontrolador de quatro entradas e quatro saídas.

3.5 – Programação das ações do robô

O objetivo do robô *Emmy* é se deslocar de um ponto a outro desviando de obstáculos que porventura esteja na sua trajetória. Para cada estado lógico resultante da análise existe uma rotina pré-definida.

A proposição analisada é a afirmativa “Existe obstáculo à frente”. Logo, para cada estado lógico resultante referente a esta proposição, têm-se uma rotina programada no microcontrolador que possibilita uma ação ao robô.

Neste sistema de controle existem 12 estados lógicos possíveis. Descreve-se abaixo as ações do robô para cada estado lógico detectado.

- Estado lógico verdadeiro: o robô detecta este estado quando existe um obstáculo à sua frente numa distância próxima. Nesta situação uma atitude possível de um ser humano é o desvio rápido do obstáculo. Neste estado lógico, o robô *Emmy* está programado para desviar 90° à esquerda, avançar por um tempo T_1 , desviar 90° à direita e avançar por um tempo T_2 . A figura 3.3 mostra a ação do robô *Emmy* no estado lógico verdadeiro.

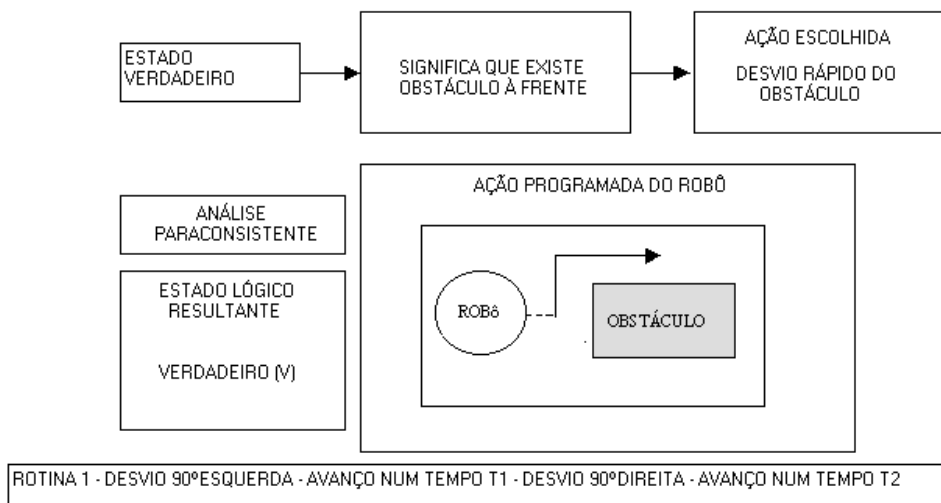


Figura 3.3 – Ação do robô *Emmy* no estado lógico verdadeiro.

- Estado lógico falso: o robô detecta este estado quando não existe um obstáculo à sua frente. Nesta situação, uma atitude possível de um ser humano é o avanço rápido. Neste estado lógico, o robô *Emmy* está programado para avançar por um tempo T_2 , avançar por um tempo T_2 , e avançar por um tempo T_1 . A figura 3.4 mostra a ação do robô *Emmy* no estado lógico falso.

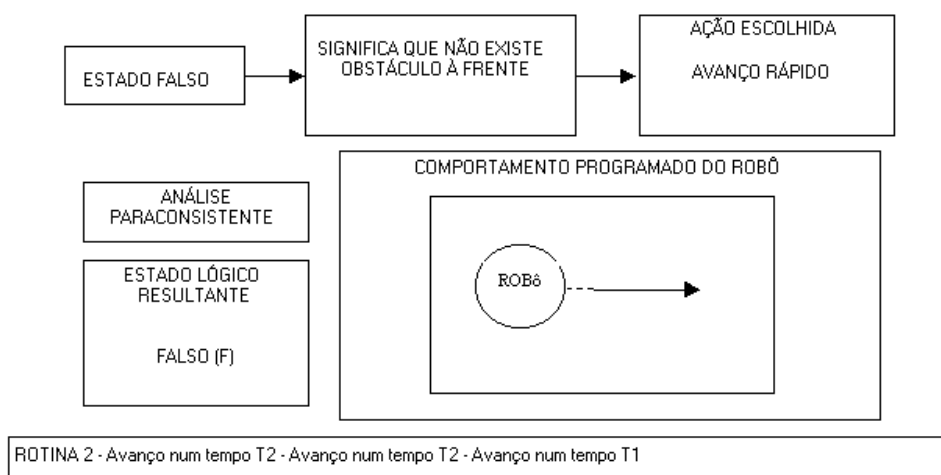


Figura 3.4 – Ação do robô *Emmy* no estado lógico falso.

- Estado lógico inconsistente: neste estado lógico há uma informação conflitante acerca da existência de um obstáculo à frente do robô. Um ser humano nessa situação poderia parar e procurar um novo ângulo para captação de informação. Neste estado lógico o robô *Emmy* pode realizar a rotina 3 ou a rotina 4. Na rotina 3 o robô desvia 90° para a esquerda, avança num tempo T_1 e desvia 90° para a direita. Na rotina 4 o robô desvia 90° para a direita, avança num tempo T_1 e desvia 90° para a esquerda. A figura 3.5 mostra o comportamento do robô *Emmy* no estado lógico inconsistente.

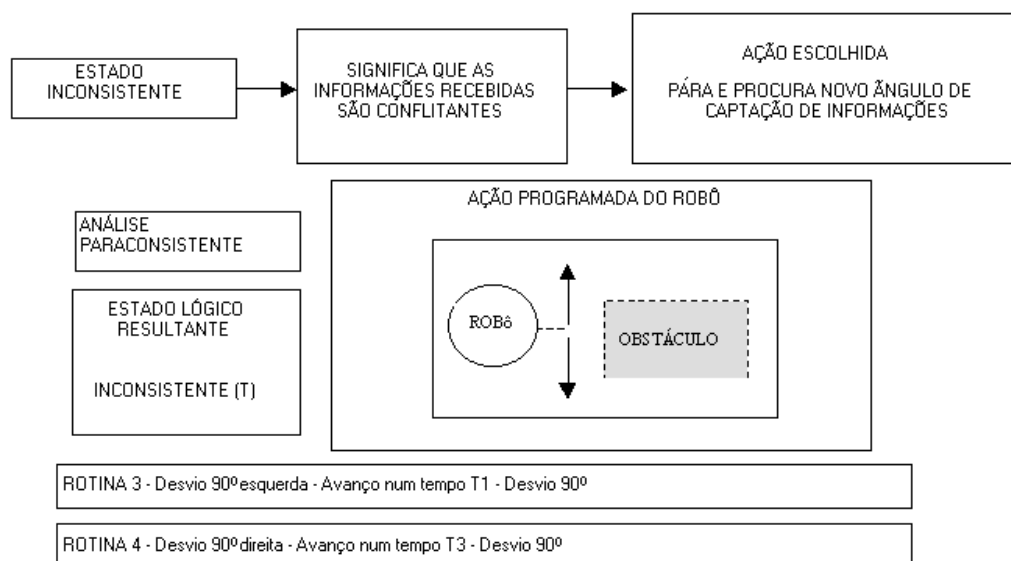


Figura 3.5 – Comportamento do robô *Emmy* no estado lógico inconsistente.

Estado lógico paracompleto: neste estado lógico há uma informação conflitante acerca da existência de um obstáculo à frente do robô. Um ser humano nessa situação poderia parar e procurar um novo ângulo para captação de informação. Neste estado lógico o robô *Emmy* pode realizar a rotina 3 ou a rotina 4. A figura 3.6 mostra o comportamento do robô *Emmy* no estado lógico paracompleto.

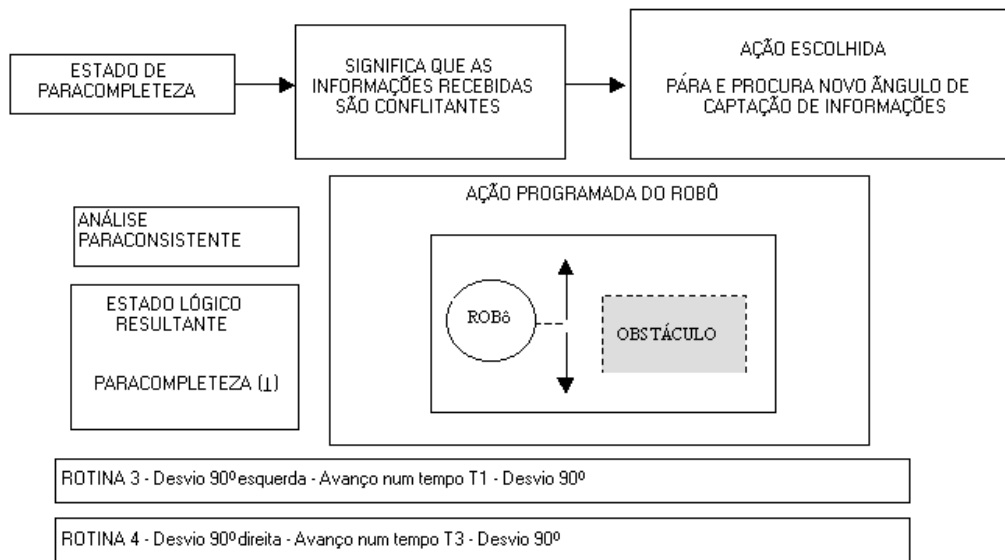
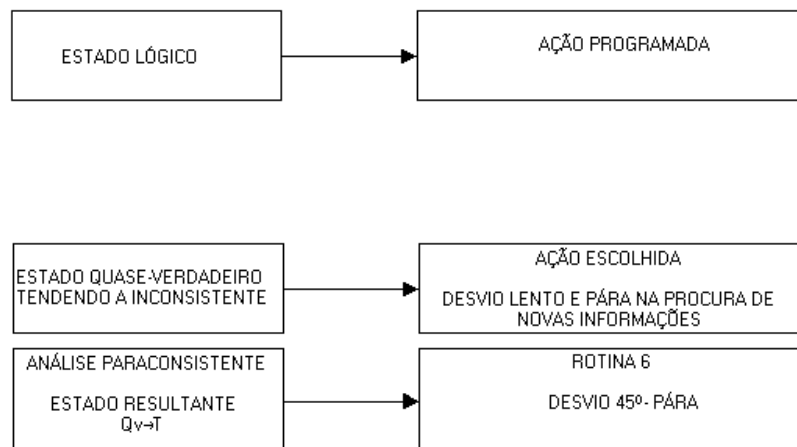
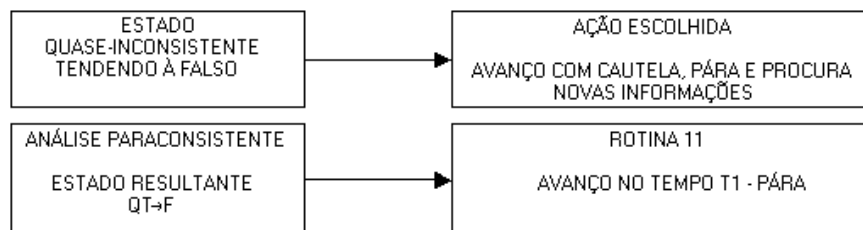
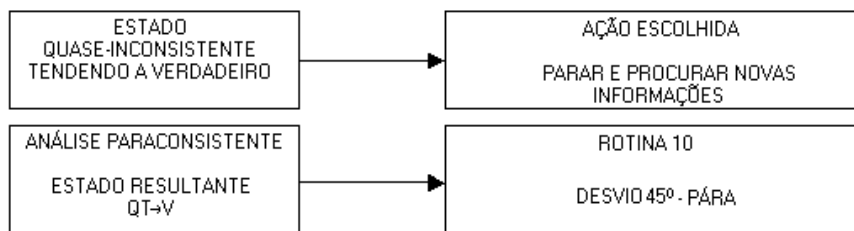
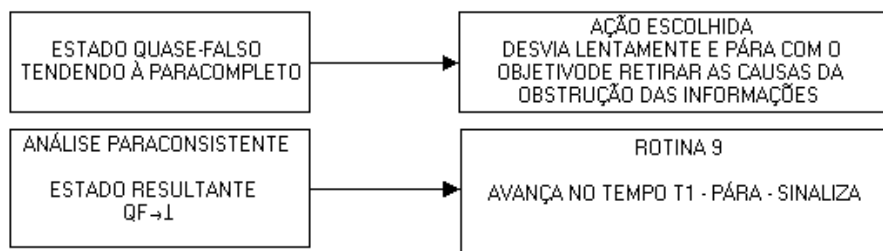
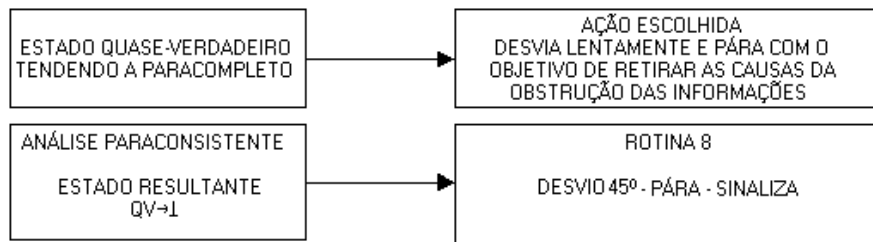
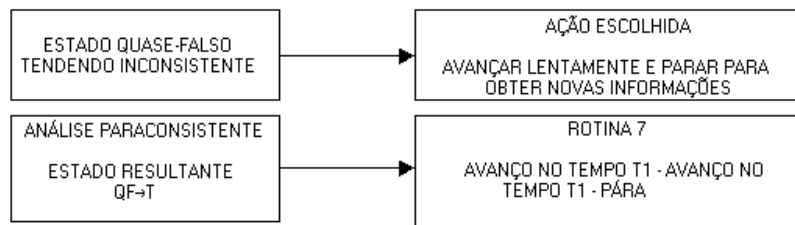


Figura 3.6 – Ação do robô *Emmy* no estado lógico paracompleto.

Para as situações referentes aos estados lógicos não extremos as seqüências de rotina são relacionadas, procurando fazer uma “analogia” com a atitude humana:





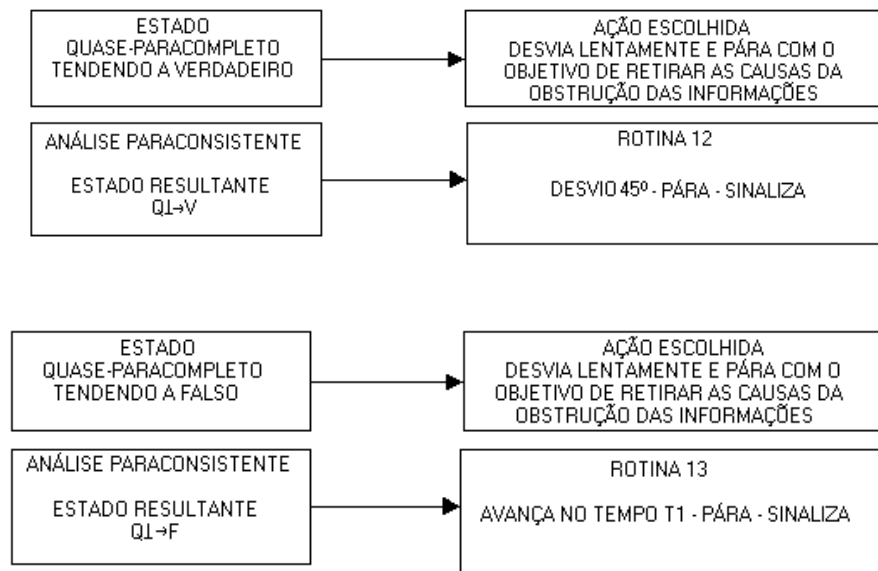


Figura 3.7 – Relação entre as seqüências de movimento do robô *Emmy* nos estados lógicos não-extremos com o comportamento humano.

Os tempos T_1 , T_2 e T_3 que aparecem nas rotinas são determinados a partir do comportamento apresentado pelo robô nos testes efetuados. Os tempos utilizados foram:

$$T_1 = 0,75s, T_2 = 1,5s \text{ e } T_3 = 2s$$

Para obter o desvio dos obstáculos num ângulo determinado pela rotina um código de 4 dígitos aplicado aos circuitos de potência aciona apenas um dos motores num tempo suficiente para o robô girar 90^0 ou 45^0 .

3.6 – Conclusões

Nos trabalhos relacionados com controle clássico de robôs móveis autônomos descrevem-se técnicas tradicionais de planejamento e de navegação em roteiros conhecidos e desconhecidos. Muito dos resultados apresentam tempo computacional muito alto ou um *hardware* muito complexo o que impossibilita a efetiva aplicação do controle.

Porém, a maior dificuldade é o reconhecimento em situações reais de mudanças de ambiente, pois muitas indefinições e inconsistências aparecem nas análises dos sinais captados.

A lógica paraconsistente anotada possibilita considerar incertezas, inconsistências e paracompletezas de um modo não-trivial e por isso, se mostra mais propícia no enquadramento de problemas ocasionados por situações de incertezas, contradições e paracompletezas.

É difícil se obter um bom resultado no comportamento do robô móvel autônomo utilizando uma metodologia baseada na lógica clássica, portanto o robô *Emmy* vem contribuir com a aplicação da lógica paraconsistente trazendo uma nova forma de controle de um robô móvel autônomo que se mostra mais propícia em alguns aspectos relevantes.

O robô *Emmy* obteve bons resultados apesar das limitações técnicas como: ausência de múltiplas velocidades, diferentes tipos de sensores, acionamentos de braços mecânicos, sincronização de velocidade entre os motores, amortecedores, freios, etc. Os testes efetuados demonstram que em Robótica o *Paracontrol* pode ser aplicado para ajudar a solucionar problemas ligados a navegação e tratamento de sinais representativos de informações sobre o ambiente.

O sistema de controle utilizando o *Paracontrol* apresenta boa capacidade de modificar o comportamento do robô quando nas modificações inesperadas das condições ambientais.

Neste primeiro protótipo, devido à baixa capacidade de memória do Microcontrolador que elabora as ações, programou-se algumas seqüências de rotinas que apresentam poucas diferenças de ações entre alguns estados lógicos.

Na figura 3.8 vê-se os aspecto físico do robô *Emmy*.

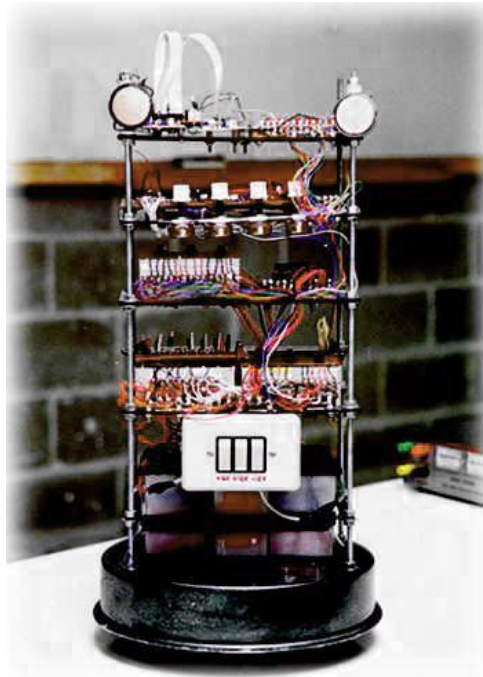


Figura 3.8 – Robô *Emmy*.

Capítulo 4

Sistema de Controle do Robô Móvel

Autônomo *Emmy II*

4.1 – Considerações iniciais

Investigando-se o controlador *Paracontrol* do robô *Emmy*, percebeu-se que programando convenientemente as saídas determinadas pelos estados lógicos não-extremos, pode-se sofisticar os movimentos do robô. Nesta dissertação, investiga-se então uma modificação do *Paracontrol* do robô *Emmy* de tal sorte que combinando as saídas dos estados não-extremos convenientemente escolhidas obtém-se uma melhor performance do *Paracontrol*. A seguir passamos a descrever tal projeto.

4.2 – Controle de direção

O sistema de controle proposto deve ser implementado em um robô móvel autônomo com as características apresentadas na figura 4.1. Tal robô móvel recebeu o nome de *Emmy II*. Essa figura mostra o esquema do robô móvel autônomo *Emmy II*.

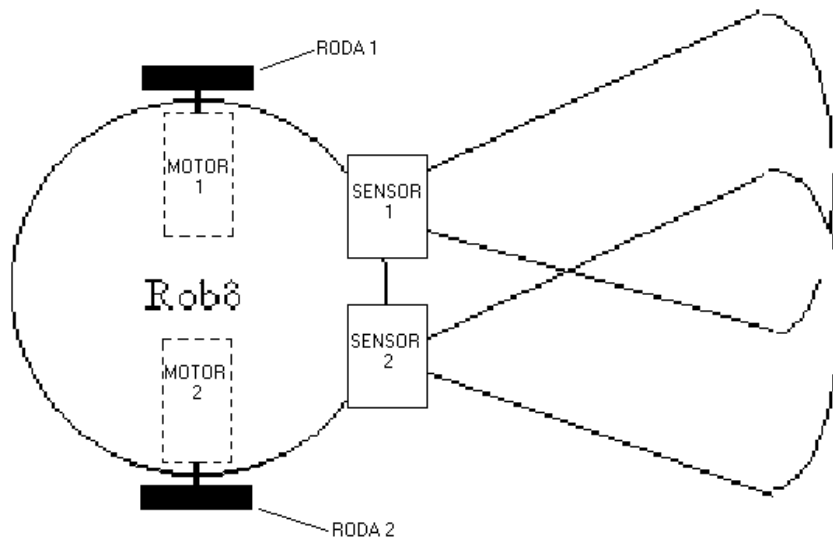


Figura 4.1 – Representação do robô *Emmy II*.

O robô *Emmy II* constitui-se, basicamente, de dois sensores (S_1 e S_2) e dois motores (motor 1 e motor 2) com rodas presas a seus eixos. Estes elementos estão acoplados a uma plataforma circular de 25cm de diâmetro. Um microcontrolador é responsável por receber os sinais gerados pelos gerados pelos sensores de ultra-som, manipulando-os segundo o algoritmo *Para-analisador* e acionar os motores.

Os movimentos possíveis neste robô são os seguintes:

1. Seguir em frente em linha reta. Os motores 1 e 2 são acionados para frente ao mesmo tempo com a mesma velocidade.
2. Ir para trás em linha reta. Os motores 1 e 2 são acionados para trás ao mesmo tempo com a mesma velocidade.
3. Girar para a direita. Apenas o motor 1 é acionado para frente com o motor 2 parado.
4. Girar para a esquerda. Apenas o motor 2 é acionado para frente com o motor 1 parado.

5. Girar para a direita. Apenas o motor 2 é acionado para trás com o motor 1 parado.

6. Girar para a esquerda. Apenas o motor 1 é acionado para trás com o motor 2 parado.

O sinal gerado pelo sensor S_1 considera-se como grau de evidência favorável e o sinal originado pelo sensor S_2 considera-se como grau de evidência contrária na proposição “A frente do robô está livre”. Quando existe um obstáculo próximo ao sensor S_1 o grau de evidência favorável é baixo e quando o obstáculo está distante do sensor S_1 o grau de evidência favorável é alto. Por outro lado, quando existe um obstáculo próximo ao sensor S_2 o grau de evidência contrária é alto e quando o obstáculo está distante do sensor S_2 o grau de evidência contrária é baixo. Nos capítulos posteriores descreve-se com detalhes como isso é feito.

O robô decide qual movimento escolher baseado nos valores do grau de evidência favorável, no grau de evidência contrária e no sistema de controle proposto de acordo com o reticulado com os respectivos estados lógicos extremos e não-extremos da figura 4.2.

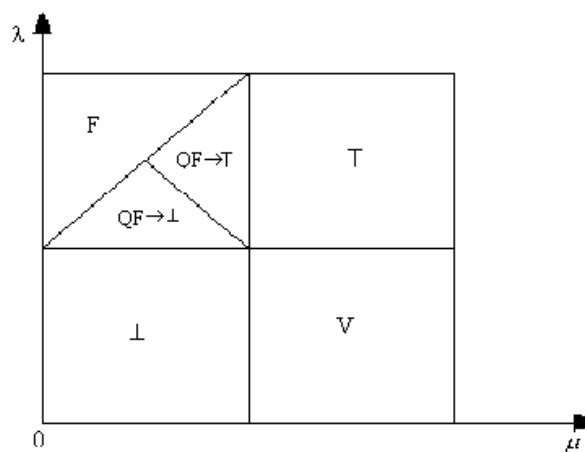


Figura 4.2 – Reticulado com os estados lógicos utilizado pelo robô *Emmy II*.

A verificação dos valores do grau de evidência favorável e do grau de evidência contrária, tomada de decisão e movimentação dos motores é feita de forma sequencial. Tal seqüência de ações é quase imperceptível ao se observar o robô movimentando-se.

Para cada estado, a respectiva decisão é a seguinte:

- Estado V: Ir para frente. Os motores 1 e 2 são acionados para frente¹ ao mesmo tempo.
- Estado F: Ir para trás. Os motores 1 e 2 são acionados para trás ao mesmo tempo.
- Estado \perp : Girar para a direita. Apenas o motor 1 é acionado para frente. Permanecendo o motor 2 parado.
- Estado T: Girar para a esquerda. Apenas o motor 2 é acionado para frente. Permanecendo o motor 1 parado.
- Estado $QF \rightarrow \perp$: Girar para a direita. Apenas o motor 2 é acionado para trás. Permanecendo o motor 1 parado.
- Estado $QF \rightarrow T$: Girar para a esquerda. Apenas o motor 1 é acionado para trás. Permanecendo o motor 2 parado.

As justificativas para as escolhas são as seguintes:

- Quando o estado for Verdadeiro (V), significa que a frente do robô está livre. Assim, o robô pode seguir em frente.
- Na Inconsistência (T), μ e λ assumem valores altos (i.e., pertencentes à região T). Isto significa que S_1 está distante de um obstáculo e S_2 está próximo de um obstáculo, sendo que o lado esquerdo está mais livre que o direito. Assim, a

¹ Significa obviamente que as respectivas rodas giram fazendo com que o robô se movimente para frente. Este e outros abusos de linguagem são cometidos ao longo do trabalho.

ação recomendada é girar para a esquerda. Aciona-se apenas o motor 2 para a frente e mantendo o motor 1 parado.

- Quando for detectado o estado de Paracompleteza (\perp), μ e λ assumem valores baixos. Isto significa que S_1 está próximo de um obstáculo e S_2 está distante de um obstáculo, sendo que o lado direito está mais livre que o esquerdo. Portanto, a decisão deve ser girar para a direita. Aciona-se apenas o motor 1 para frente e mantendo o motor 2 parado.
- No estado de Falsidade (F), a frente do robô está obstruída por um o obstáculo muito próximo do robô. Portanto, a decisão é recuar.
- No estado Quase-falso tendendo ao Inconsistente ($QF \rightarrow T$), a frente do robô continua obstruída, com as seguintes características: o obstáculo não está tão próximo como na Falsidade e o lado esquerdo está um pouco mais livre do que o direito. A decisão é girar para a esquerda acionando-se apenas o motor 1 para trás e mantendo-se o motor 2 parado.
- No estado Quase-falso tendendo ao Paracompleto ($QF \rightarrow \perp$), a frente do robô continua obstruída, do seguinte modo: o obstáculo não está tão próximo como na falsidade e o lado direito está um pouco mais livre do que o esquerdo. A decisão é girar para a direita, acionando-se apenas o motor 2 para trás e mantendo-se o motor 1 parado.

4.3 – Controle de velocidade

Utiliza-se o Valor de Controle de Veracidade (V_{cve}), o Valor de Controle de Falsidade (V_{cfa}), o Valor de Controle de Inconsistência (V_{cic}) ou o Valor de Controle de Paracompleteza (V_{cpa}) para determinar a velocidade com que se realiza cada movimento.

A figura 4.3 mostra o segmento de reta perfeitamente indefinido, ou seja, $G_{ce} \equiv 0$.

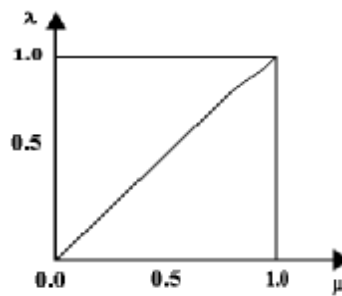


Figura 4.3 – Segmento de reta perfeitamente indefinido.

Quando $G_{ce}(\mu, \lambda) = \mu - \lambda > 0$, o grau de certeza se “distancia” do segmento perfeitamente indefinido em “direção” ao estado Verdadeiro (V), como mostrado abaixo na figura 4.4.

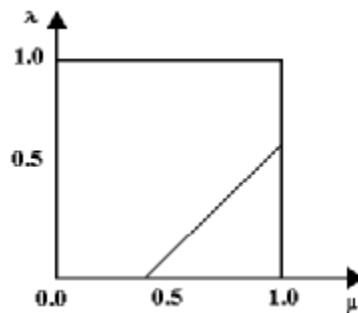


Figura 4.4 – Segmento de reta $G_{ce}(\mu, \lambda) = k$, $0 < k < 1$.

Em consequência, a velocidade do movimento varia proporcionalmente ao Valor de Controle de Veracidade (V_{cve}), sendo mínima quando G_{ce} assume valores do segmento de reta perfeitamente indefinido e máxima quando G_{ce} assume o valor 1.

Nos estados Falso, Quase-falso tendendo ao Inconsistente e Quase-falso tendendo ao Paracompleto, ou seja, quando $G_{ce}(\mu, \lambda) = \mu - \lambda < 0$, o Grau de Certeza se “distancia” do segmento de reta perfeitamente indefinido em “direção” ao estado de Falsidade, como mostrado na figura 4.5.

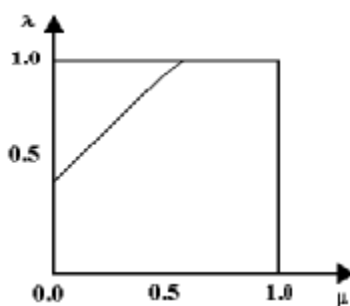


Figura 4.5 – Segmento de reta $G_{ce}(\mu, \lambda) = k$, $-1 < k < 0$.

Assim, a velocidade do movimento depende do Valor de Controle de Falsidade (V_{cfa}).

A figura 4.6 mostra a o segmento de reta perfeitamente definido. Ele é caracterizado pela equação $G_{in}(\mu, \lambda) = \mu + \lambda - 1 = 0$.

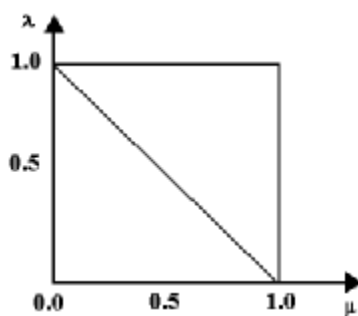


Figura 4.6 – Segmento de reta perfeitamente definido.

Na situação $G_{in}(\mu, \lambda) = \mu + \lambda - 1 = k$, $0 < k < 1$, o Grau de Incerteza se “distancia” do segmento de reta perfeitamente definido em “direção” ao estado de inconsistência, como mostra-se na figura 4.7.

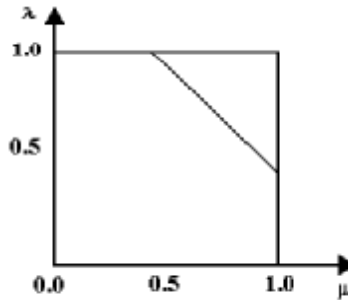


Figura 4.7 – Segmento de reta para $G_{in}(\mu, \lambda) = k$, $0 < k < 1$.

Assim, nessa situação, a velocidade do movimento depende do valor do Valor de Controle de Inconsistência (G_{cic}), sendo mínima quando G_{in} assume valores do segmento de reta perfeitamente definido e máxima quando G_{in} assume o valor 1.

Quando $G_{in}(\mu, \lambda) = \mu + \lambda - 1 = k$, $-1 < k < 0$ o Grau de Incerteza se “distancia” do segmento de reta perfeitamente definido em direção ao estado Paracompleto, como mostra-se na figura 4.8.

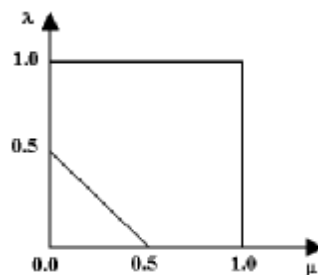


Figura 4.8 – Segmento de reta para $G_{in} = k$, $-1 < k < 0$.

Nessa situação, a velocidade de movimentação depende do módulo do Grau de

Incerteza ou do módulo do Grau de Paracompleteza.

Resumidamente:

- Para $\mu > 0.5$ e $\lambda \leq 0.5$

$G_{ce}(\mu, \lambda)$ ou $G_{ve}(\mu, \lambda)$ determina a velocidade do movimento.

- Para $\mu \leq 0.5$ e $\lambda > 0.5$

$|G_{ce}(\mu, \lambda)|$ ou $|G_{fa}(\mu, \lambda)|$ determina a velocidade do movimento.

- Para $\mu \leq 0.5$ e $\lambda \leq 0.5$

$|G_{in}(\mu, \lambda)|$ ou $|G_{pa}(\mu, \lambda)|$ determina a velocidade do movimento.

- Para $\mu > 0.5$ e $\lambda > 0.5$

$G_{in}(\mu, \lambda)$ ou $G_{ic}(\mu, \lambda)$ determina a velocidade do movimento.

Capítulo 5

Construção Física do Robô Móvel Autônomo

Emmy II

5.1 – Considerações iniciais

Com o objetivo de avaliar o *Paracontrol* proposto no capítulo anterior, implementou-se um robô móvel autônomo. Tal robô denominou-se *Emmy II*.

A plataforma utilizada para a montagem do robô *Emmy II* possui aproximadamente 25cm de diâmetro e 23cm de altura. Os componentes principais do robô *Emmy II* são um microcontrolador da família 8051, dois sensores de ultra-som e dois motores de corrente contínua.

Os sensores de ultra-som são responsáveis pela verificação de obstáculos à frente do robô. Os sinais gerados pelos sensores são enviados para o microcontrolador. Os valores do grau de evidência favorável (μ) e do grau de evidência contrária (λ) na proposição “A frente do robô está livre” são determinados pelo microcontrolador com base nos sinais recebidos dos sensores de ultra-som. O microcontrolador também determina o movimento a ser executado pelo robô, ou seja, qual motor deve ser acionado, com base na decisão proveniente do *Paracontrol*.

A figura 5.2 mostra o diagrama em blocos simplificado do robô móvel autônomo *Emmy II*.

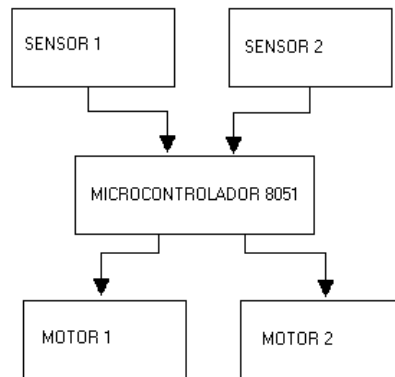


Figura 5.2 – Diagrama em blocos simplificado do robô *Emmy II*.

A seguir, faz-se uma descrição detalhada do robô móvel *Emmy II*.

5.2 – Descrição do robô móvel autônomo *Emmy II*.

A descrição do robô é feita em quatro etapas. Tais são elas:

- Circuito de alimentação.
- Circuito dos sensores.
- Circuito de controle.
- Circuito de atuação.

5.2.1 – Circuito de alimentação

O circuito de alimentação possui como objetivo fornecer as tensões elétricas necessárias para garantir o funcionamento dos demais circuitos do robô. Os vários circuitos do robô *Emmy II* necessitam de tensões de 5, 12 ou 16,8 Volts. O circuito de alimentação deve ser compacto o suficiente para não atrapalhar o funcionamento do robô.

Para a geração dos 12 Volts necessários para o acionamento dos motores, optou-se pela ligação em série de duas baterias gelatinosas com capacidade de geração de 6 Volts cada uma. Assim, conseguiu-se gerar a tensão elétrica necessária para o acionamento dos motores com um circuito leve o suficiente para garantir o funcionamento do robô móvel autônomo.

Em alguns pontos do circuito elétrico do robô móvel necessita-se de tensão elétrica maior que 12 Volts. Para tanto, ligou-se quatro pilhas recarregáveis de 1,2 Volts em série com as baterias de 6 Volts, conseguindo-se, assim, os 16,8 Volts.

Os circuitos integrados utilizados no robô móvel *Emmy II* são alimentados com 5 Volts. Para se conseguir essa tensão utilizou-se um regulador de tensão serial monolítico 7805. À sua entrada conecta-se 12 Volts e na sua saída obtêm-se 5 Volts.

A figura 5.3 mostra o esquema elétrico do circuito de alimentação utilizado no robô *Emmy II*.

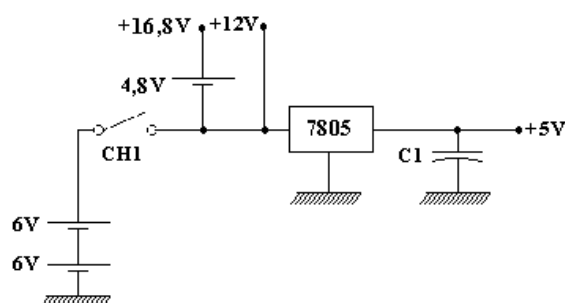


Figura 5.3 - Esquema elétrico do circuito de alimentação utilizado no robô *Emmy II*.

O capacitor eletrolítico C1 tem a finalidade de evitar ruídos na saída de 5 Volts. O seu valor é de 3000 μ F, 25V.

5.2.2 – Circuito dos sensores

As informações sobre o ambiente por onde o robô se locomove são obtidas através de dois sensores de ultra-som. Essas informações são enviadas para o microcontrolador onde são processadas de acordo com o *Paracontrol* e como resultado obtém-se a decisão de qual movimento o robô deve realizar.

O “Polaroid 6500 Series Sonar Ranging Module” [POL 96] é o sensor de ultra-som utilizado no robô *Emmy II*. Esse sensor possui quatro terminais que devem ser conectados ao circuito eletrônico do robô. Os terminais são os seguintes: Vcc, Gnd, Init e Echo.

Durante o funcionamento do robô o Vcc e o Gnd devem ser constantemente conectados ao +5 Volts contínuo e ao terra respectivamente, pois esses são os terminais de alimentação do sensor.

O sensor dispara um trem de pulsos ultra-sônico no momento em que o microcontrolador envia um sinal de +5 Volts contínuo para o Init. Esse trem de pulsos ultra-sônico gera eco após se chocar com algum obstáculo. Quando o eco retorna ao sensor, o terminal Echo passa a ter +5 Volts contínuo, por sua vez, envia-se este sinal para o microcontrolador. Portanto, o microcontrolador deve calcular a distância entre o sensor e o obstáculo utilizando o intervalo de tempo entre o envio do sinal para o Init e o recebimento do sinal originado no Echo. No próximo capítulo descreve-se o funcionamento do programa armazenado na memória do microcontrolador.

A figura 5.4 mostra o circuito de sensoriamento utilizado pelo robô *Emmy II*.

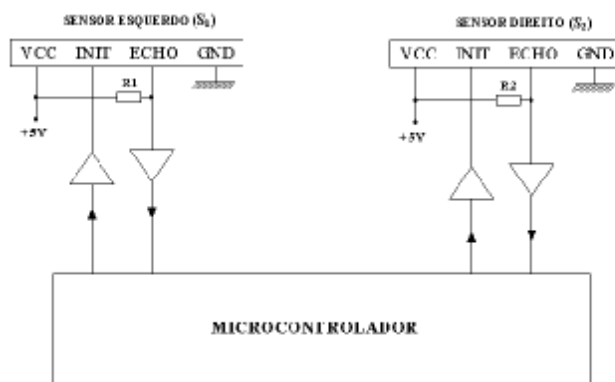


Figura 5.4 – Circuito de sensoriamento utilizado no robô *Emmy II*.

O funcionamento adequado dos sensores é garantido com a conexão de um resistor de 4700Ω entre o Vcc e o Echo [Pol 96]. A interface entre os sensores e o microcontrolador realiza-se com “buffers”.

5.2.3 – Circuito de controle

Essa é a etapa principal do circuito do robô *Emmy II*, pois ela é responsável pelo cálculo da distância entre os sensores de ultra-som e obstáculos, transformação dessas distâncias em valores de graus de evidência favorável e contrária, execução do algoritmo *Para-analisador* e geração dos sinais de acionamento dos motores.

O componente principal dessa etapa é um microcontrolador da família 8051. Escolheu-se o 89C52 de fabricação da Atmel Corporation. Portanto, o bom funcionamento do robô depende do programa armazenado na memória de programa do microcontrolador. A descrição detalhada desse programa será realizada em capítulo posterior.

A porta de comunicação paralela 1 do microcontrolador da família 8051 é utilizada para enviar e receber os sinais para o circuito de sensoriamento e acionamento. “Buffers”

fazem a interface entre o circuito de controle e as outras etapas do circuito do robô *Emmy*

II.

A porta de comunicação paralela 1 do microcontrolador da família 8051 possui 8 pinos. A função de cada pino é a seguinte:

- Pino 0: Init do sensor S_1 .
- Pino 1: Echo do sensor S_1 .
- Pino 2: Init do sensor S_1 .
- Pino 3: Echo do sensor S_2 .
- Pino 4: Em nível lógico alto (+5 Volts) aciona o motor 1.
- Pino 5: Em nível lógico alto (+5 Volts) aciona o motor 2.
- Pino 6: Em nível lógico baixo (0 Volts) faz com que o motor 1 gire para frente quando o pino 5 estiver em nível lógico alto (+5 Volts). Em nível lógico alto (+5 Volts) faz com que o motor 1 gire para trás quando o pino 5 estiver em lógico alto (+5 Volts).
- Pino 7: Em nível lógico baixo (0 Volts) faz com que o motor 2 gire para frente quando o pino 6 estiver em nível lógico alto (+5 Volts). Em nível lógico alto (+5 Volts) faz com que o motor 2 gire para trás quando o pino 6 estiver em lógico alto (+5 Volts).

A figura 5.5 mostra as conexões do microcontrolador.

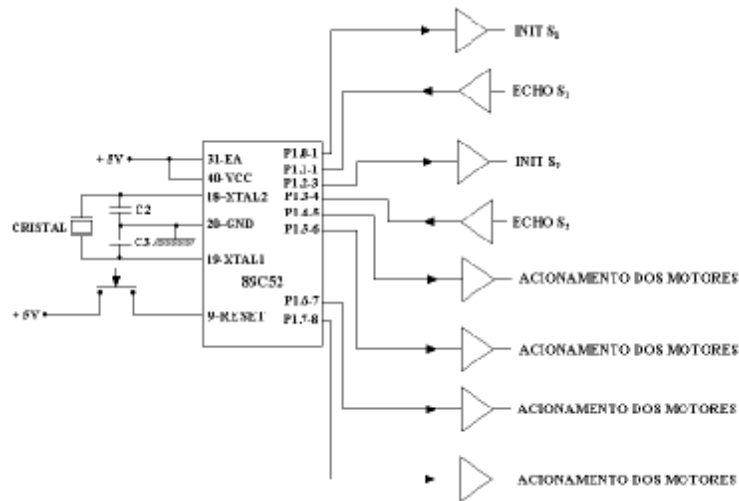


Figura 5.5 – Conexões do microcontrolador utilizado no robô *Emmy II*.

O pino 31 (External Access Enable Programming Supply Voltage) deve ser conectado ao +5Volts para que o microcontrolador busque na memória interna o programa a ser executado.

Os pinos 19 e 18 (XTA1 e XTAL2) são as entradas para o amplificador oscilador. O circuito oscilador é composto pelo cristal de 1059 MHz e pelos capacitores cerâmicos C2 e C3 de 33 pF. Esse circuito é responsável pela geração do sinal de relógio utilizado pelo microcontrolador.

5.2.4 – Circuito de acionamento

Dois motores alimentados com uma tensão elétrica contínua de 12 Volts são responsáveis pelos movimentos do robô móvel *Emmy II*. Os motores podem ser acionados para frente ou para trás. O *Paracontrol*, através do microcontrolador é responsável por determinar qual motor deve ser acionado e para qual sentido ele deve girar. O microcontrolador não pode alimentar diretamente os motores, pois a tensão de saída das

portas de comunicação paralela é de apenas 5 Volts. Além disso, a corrente consumida pelos motores é muito alta para que o microcontrolador possa supri-la.

Criou-se, então, um circuito de interface de potência. Esse circuito é composto basicamente por transistores de efeito de campo - MOSFETs. A figura 5.6 mostra o circuito de interface entre o microcontrolador e os motores.

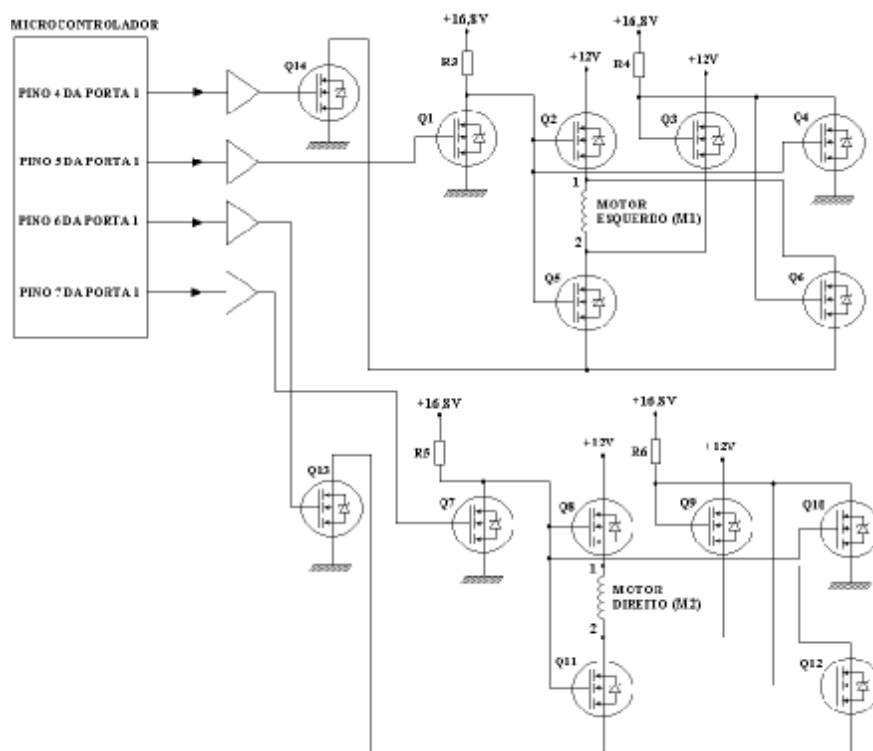


Figura 5.6 – Circuito de interface de potência utilizado no robô *Emmy II*.

O motor 1 gira para frente quando o pino 4 da porta paralela 1 do microcontrolador 89C52 estiver em nível lógico alto (+5 Volts) e o pino 5 estiver em nível lógico baixo (0 Volts). O terminal porta do transistor Q14 está conectado ao pino 4 da porta paralela 1 do microcontrolador, portanto, este transistor está saturado. Assim, o terra passa a alimentar o terminal fonte dos transistores Q5 e Q6. O terminal porta do transistor Q1 está conectado ao

pino 5 da porta paralela 1 do microcontrolador, como este pino está com nível lógico baixo (0 Volts), conseqüentemente, o transistor Q1 está em corte. O fato de Q1 estar em corte nos garante que os transistores Q2, Q4 e Q5 estão saturados, pois seus terminais portas estão alimentados com 16,8 Volts. A saturação de Q4 faz com que o terra seja conectado ao terminal porta de Q3 e Q6, garantindo o corte desses transistores. Portanto, o motor 1 é alimentado com +12 Volts no terminal 1 e com o terra no terminal 2, fazendo com que o motor 1 gire para a frente.

Para o motor 1 girar para trás os pinos 4 e 5 da porta paralela 1 do microcontrolador 89C52 devem estar com nível lógico alto (+5 Volts). Assim, o transistor Q14, que está conectado ao pino 4, satura, fazendo com que os terminais fontes dos transistores Q5 e Q6 sejam conectados ao terra. O transistor Q1 também satura, pois, o seu terminal porta está conectado ao pino 5 da porta paralela 1 do microcontrolador. A saturação de Q1 faz com que o terra seja conectado aos terminais portas dos transistores Q2, Q4 e Q5, fazendo com que eles fiquem em corte. O corte de Q4 garante a saturação de Q3 e Q6, pois, seus terminais portas são alimentados com 16,8 Volts. Portanto, o terminal 2 do motor 1 é alimentado com +12Volts e o terminal 1 do motor 1 é alimentado com o terra. Assim, o motor 1 gira para trás.

O motor 2 gira para frente quando o pino 6 da porta paralela 1 do microcontrolador 89C52 estiver em nível lógico alto (+5 Volts) e o pino 7 estiver em nível lógico baixo (0 Volts). O terminal porta do transistor Q13 está conectado ao pino 6 da porta paralela 1 do microcontrolador, portanto, significa que tal transistor está saturado. Assim, o terra passa a alimentar o terminal fonte dos transistores Q11 e Q12. O terminal porta do transistor Q7 está conectado ao pino 7 da porta paralela 1 do microcontrolador, e como este pino está com nível lógico baixo (0 Volts), em conseqüência, o transistor Q7 está em corte.

O fato de Q7 estar em corte nos garante que os transistores Q8, Q10 e Q11 estão saturados, pois seus terminais portas estão alimentados com 16,8 Volts. Em particular, a saturação de Q10 faz com que o terra seja conectado ao terminal porta de Q9 e Q12, garantindo-se assim o corte desses transistores. Portanto, o motor 2 é alimentado com +12 Volts no terminal 1 e com o terra no terminal 2, fazendo com que o motor 2 gire para a frente.

Para o motor 2 girar para trás, os pinos 6 e 7 da porta paralela 1 do microcontrolador 89C52 devem estar com nível lógico alto (+5 Volts). Assim, o transistor Q13, que está conectado ao pino 6, satura-se, fazendo com o que os terminais fontes dos transistores Q11 e Q12 sejam conectadas ao terra. O transistor Q7 também satura, pois o seu terminal porta está conectado ao pino 7 da porta paralela 1 do microcontrolador. A saturação de Q7 faz com que o terra seja conectado aos terminais portas dos transistores Q8, Q10 e Q11, fazendo com que eles fiquem em corte. O corte de Q10 garante a saturação de Q9 e Q12, pois, seus terminais portas são alimentados com 16,8 Volts. Portanto, o terminal 2 do motor 2 é alimentado com +12Volts e o terminal 1 do motor 2 é alimentado com o terra, fazendo com que o motor 2 gire para trás.

A alimentação de uma tensão maior que 12 Volts nas portas dos transistores Q2, Q3, Q4, Q5, Q6, Q8, Q9, Q10, Q11 e Q12 garante suas saturações. Como as portas desses transistores consomem pouca corrente elétrica, escolheu-se ligar quatro pilhas recarregáveis de 1,2 Volts cada uma em série com a bateria de 12 Volts.

Os resistores R3, R4, R5 e R6 são necessários apenas para evitar curto-circuito no momento da saturação de Q1, Q4, Q7 ou Q10. Seus valores são de 100 K Ω .

5.3 – Circuito completo do robô *Emmy II*

Montou-se todos os componentes eletrônicos do circuito do robô móvel *Emmy II* numa placa de 10 cm de comprimento por 15 cm de largura. O fato dos componentes eletrônicos ocuparem um espaço relativamente pequeno permitiu a montagem de um robô compacto. A figura 5.7 mostra o esquema completo do circuito utilizado no robô *Emmy II*.

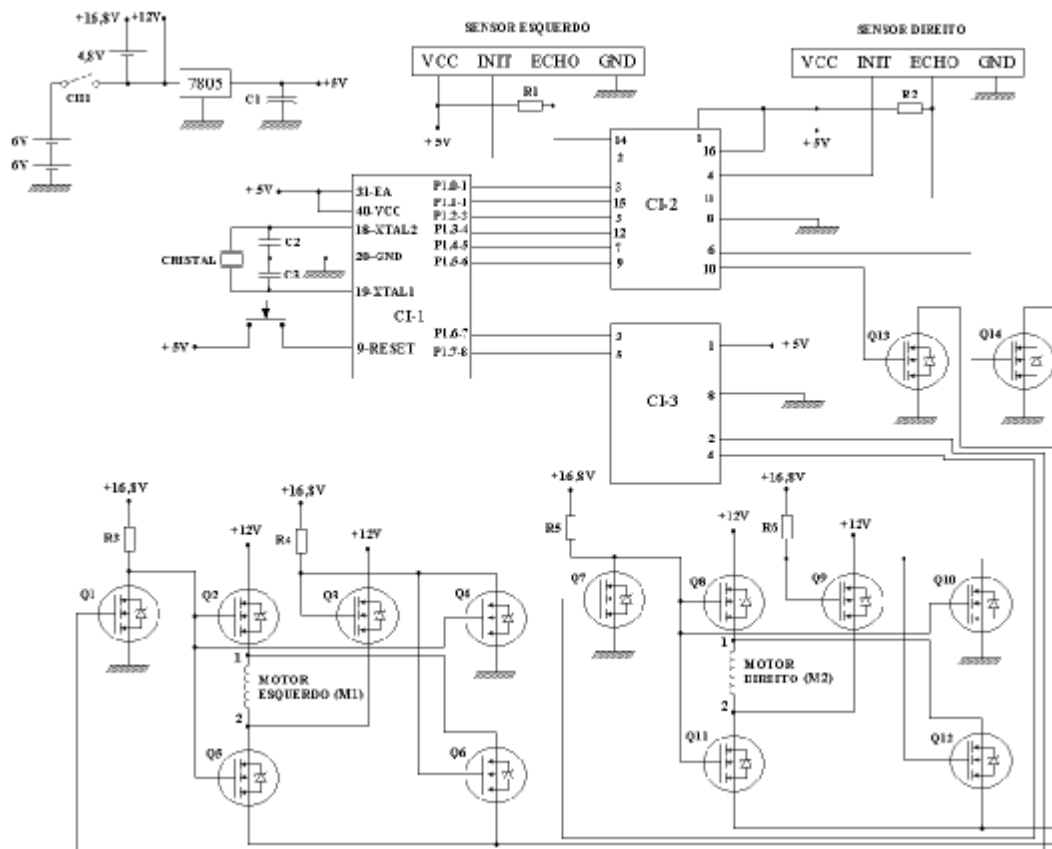


Figura 5.7 – Esquema da parte elétrica do robô *Emmy II*.

Os componentes utilizados são os seguintes:

- C1: Capacitor eletrolítico – 3000 μ F, 25V.
- C2 e C3: Capacitores cerâmicos de 33 pF.

- CI 1: 89C52 - Microcontrolador da família 8051 com 8 K bites de memória de programa interna.
- CI 2 e CI 3: CD4050BC – Amplificador não-inversor.
- Cristal de 11059 MHz.
- Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12, Q13 e Q14: IRF640 – MOSFETs.
- R1 e R2: Resistores de 5 K Ω .
- R3, R4, R5 e R6: Resistores de 100 K Ω .

A figura 5.8 mostra a vista frontal do robô *Emmy II*, a figura 5.9 mostra os dois sensores ultrassônicos S₁ e S₂ e a figura 5.10 mostra os motores do robô.

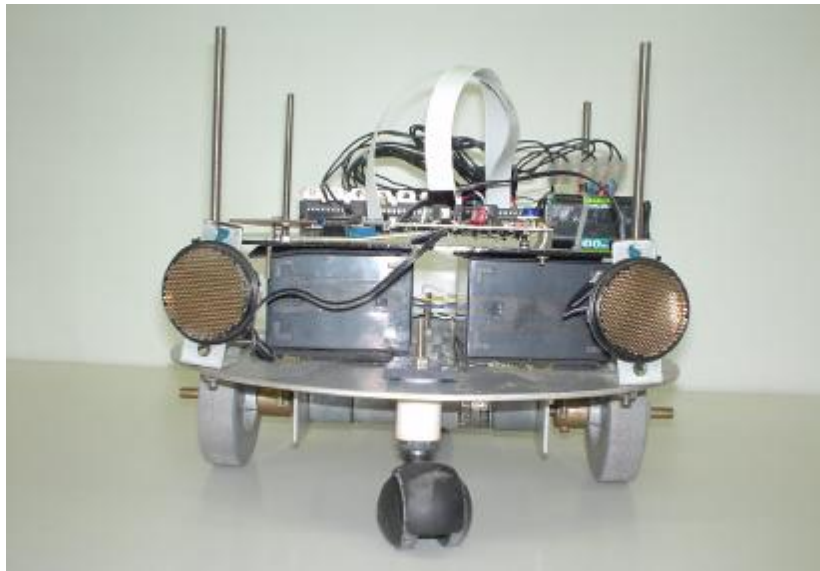


Figura 5.8 – Vista frontal do robô *Emmy II*.

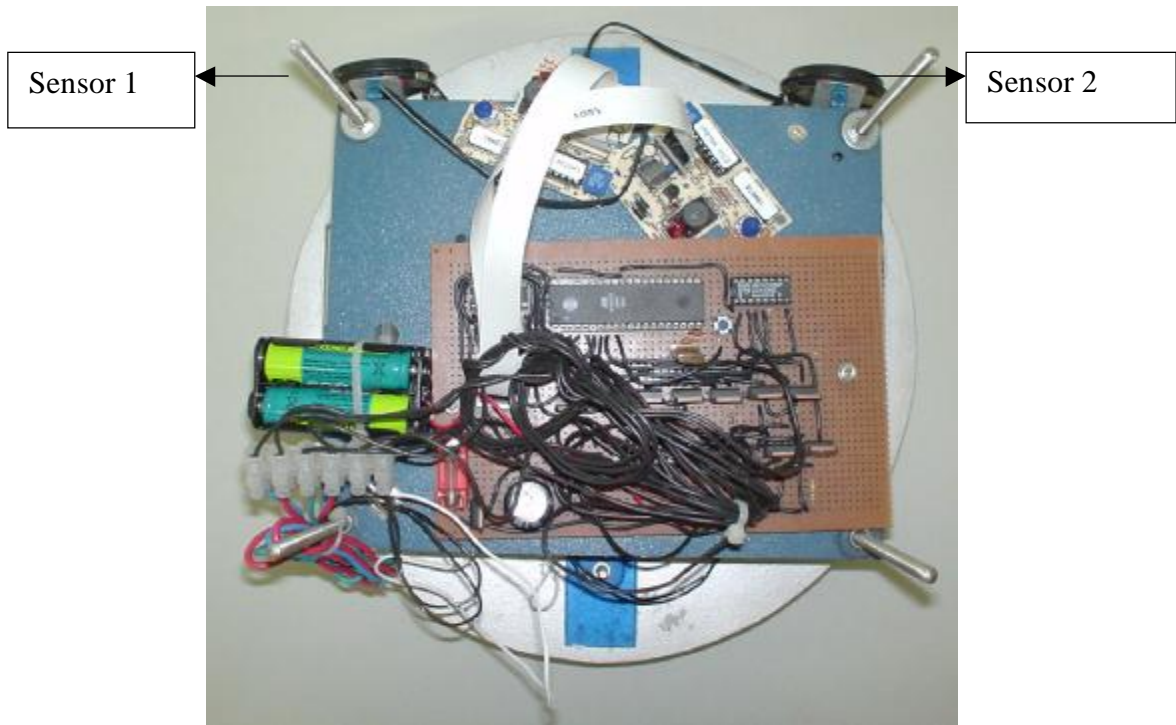


Figura 5.9 – Vista superior do robô *Emmy II*.

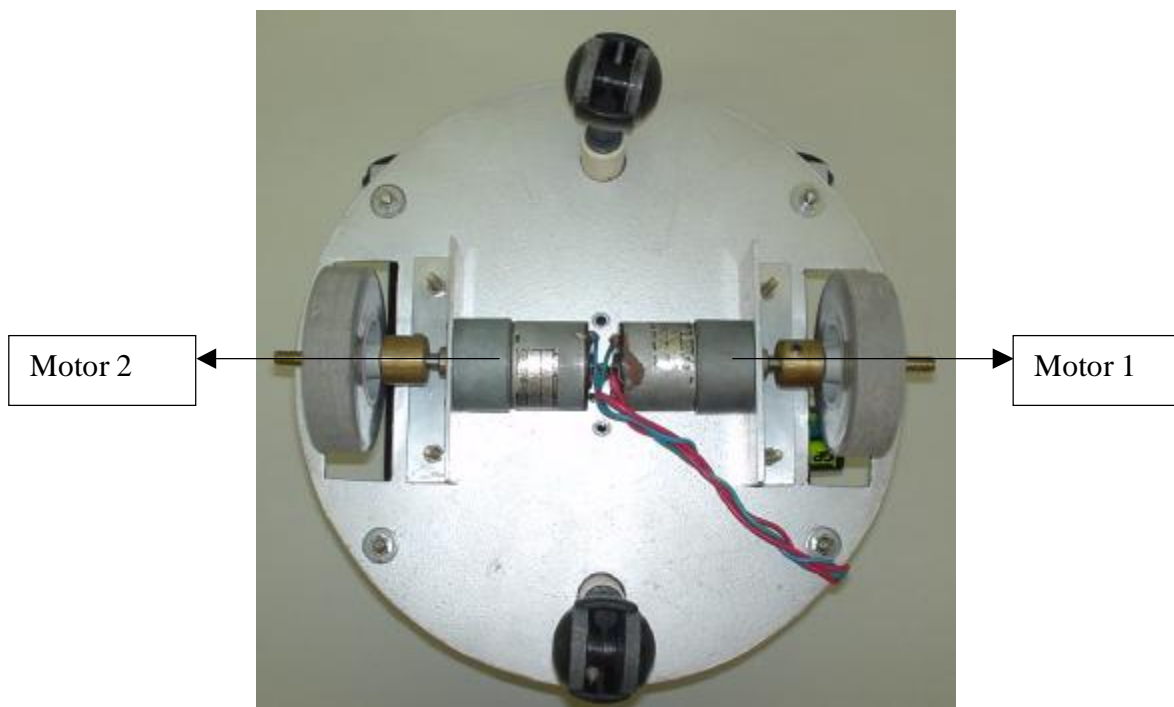


Figura 5.10 – Vista inferior do robô *Emmy II*.

Capítulo 6

Programação do Robô *Emmy II*

6.1 – Considerações iniciais

Como se expôs no capítulo anterior, o componente principal do robô *Emmy II* é o microcontrolador 89C52, pois ele é responsável por determinar as distâncias entre os sensores de ultra-som e os obstáculos localizados à frente do robô, calcular os valores dos graus de evidência favorável e evidência contrária na proposição “A frente do robô está livre”, executar o algoritmo *Para-analisador* e gerar sinais para o acionamento dos motores.

Armazena-se o programa na memória interna do microcontrolador 89C52. Descreve-se em seguida as várias partes do programa utilizado no robô móvel autônomo *Emmy II*.

6.2 – Determinação do grau de evidência favorável

A memória interna do microcontrolador 89C52 pode armazenar números de até 8 bits, ou seja, representando o número em hexadecimal, pode se armazenar qualquer valor entre 00h e FFh. Escolheu-se a posição 30h da memória para armazenar o valor do grau de evidência favorável.

A partir do momento que o pino 0 da porta de comunicação paralela 1 do microcontrolador 89C52 é colocado em nível lógico alto (+5 Volts), ou seja, quando o Init

do sensor 1 é acionado, portanto, quando o trem de pulsos ultra-sônicos parte do sensor 1, a posição de memória 30h passa a ser incrementada em uma unidade a cada 118 μ seg.

O Echo do sensor 1 está conectado ao pino 1 da porta paralela 1. Assim, quando o pino 1 da porta paralela 1 estiver em nível alto (+5 Volts), significa que o eco retornou ao sensor 1. Nesse momento a posição de memória 30h pára de ser incrementada.

O tempo para o som percorrer 4 centímetros é 118 μ seg. Portanto, o valor armazenado na posição 30h multiplicado por dois representa a distância em centímetros entre o sensor 1 e o obstáculo à sua frente.

O valor máximo que a posição de memória 30h pode alcançar é 3Fh. Portanto, o sensor 1 pode medir distâncias entre 00h e 3Fh, ou, 0 cm e 126 cm.

Assim, 00h armazenado na posição de memória 30h significa que o valor do grau de evidência favorável (μ) na proposição “A frente do robô está livre” é 0. E 3Fh armazenado na posição de memória 30h significa que o valor do grau de evidência favorável (μ) na proposição “A frente do robô está livre” é 1. Da mesma forma, valores intermediários entre 00h e 3Fh armazenados na posição 30h significam valores intermediários do grau de evidência favorável.

6.3 – Determinação do grau de evidência contrária

Quando o pino 2 da porta de comunicação paralela 1 do microcontrolador 89C52 é colocado em nível lógico alto (+5 Volts), ou seja, quando o Init do sensor 2 é acionado, portanto, quando o trem de pulsos ultra-sônicos parte do sensor 2, a posição de memória 31h passa a ser incrementada em uma unidade a cada 118 μ seg.

O Echo do sensor 2 está conectado ao pino 3 da porta paralela 1. Assim, quando o pino 3 da porta paralela 1 estiver em nível alto (+5 Volts), significa que o eco retornou ao sensor 3. Nesse momento a posição de memória 31h pára de ser incrementada.

O tempo para o som percorrer 4 centímetros é 118 μ seg. Portanto, o valor armazenado na posição 31h multiplicado por dois representa a distância em centímetros entre o sensor 2 e o obstáculo à sua frente.

O valor máximo que a posição de memória 31h pode alcançar é 3Fh. Portanto, o sensor 2 pode medir distâncias entre 00h e 3Fh, ou, 0 cm e 126 cm.

Complementa-se o conteúdo da posição de memória 31, pois ela representa o grau de evidência contrária (λ) na proposição “A frente do robô está livre”.

Assim, 00h armazenado na posição de memória 31h significa que o valor do grau de evidência contrária (λ) na proposição “A frente do robô está livre” é 0. E 3Fh armazenado na posição de memória 31h significa que o valor do grau de evidência contrária (λ) na proposição “A frente do robô está livre” é 1. Da mesma forma, valores intermediários entre 00h e 3Fh armazenados na posição 31h significam valores intermediários do grau de evidência favorável.

6.4 – Algoritmo *Para-analisador*

De posse dos valores do grau de evidência favorável e do grau de evidência contrária é possível calcular Grau de Inconsistência, Grau de Paracompleteza, Grau de Veracidade e Grau de Falsidade. Essas são as variáveis de entrada do algoritmo *Para-analisador*.

A figura 6.1 mostra o reticulado utilizado na primeira implementação do robô *Emmy*

II.

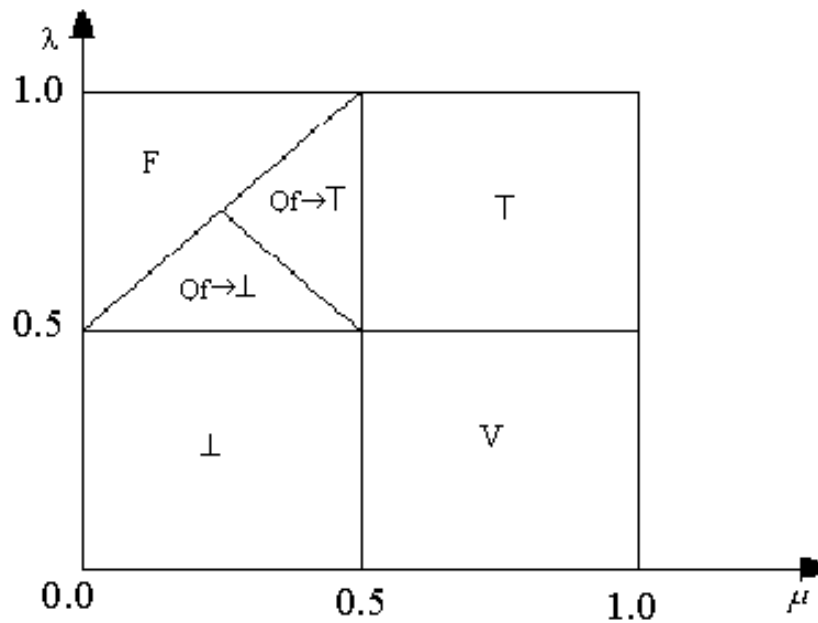


Figura 6.1 – Reticulado utilizado na primeira versão do robô *Emmy II*.

A decisão tomada para cada estado é a seguinte:

- Estado V: Ir para frente. Os motores 1 e 2 são acionados para frente ao mesmo tempo.
- Estado F: Ir para trás. Os motores 1 e 2 são acionados para trás ao mesmo tempo.
- Estado \perp : Girar para a direita. Apenas o motor 1 é acionado para frente.
- Estado T: Girar para a esquerda. Apenas o motor 2 é acionado para frente.
- Estado $QF \rightarrow \perp$: Girar para a direita. Apenas o motor 2 é acionado para trás.
- Estado $QF \rightarrow T$: Girar para a esquerda. Apenas o motor 1 é acionado para trás.

Cada movimento tem a duração aproximada de 0,4 segundos.

Os primeiros testes mostraram que o robô praticamente nunca detecta os estados $QF \rightarrow \perp$ e $QF \rightarrow T$.

Por isso, implementou-se uma segunda versão do algoritmo *Para-analisador* onde as regiões de $QF \rightarrow \perp$ e $QF \rightarrow T$ são maiores. A figura 6.2 mostra o reticulado utilizado nessa segunda versão do *Para-analisador*.

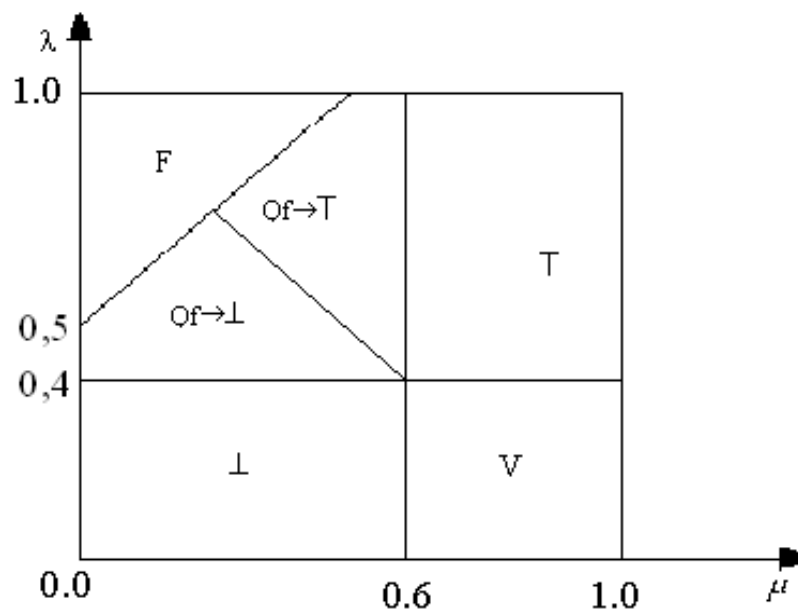


Figura 6.2 – Reticulado utilizado na segunda versão do robô *Emmy II*.

6.5 – Testes

Com o objetivo de verificar a funcionalidade do robô *Emmy II* executou-se quatro testes. Esses testes constituem-se basicamente por contar quantas vezes o robô se choca a algum obstáculo quando o robô se locomove por um determinado ambiente. A figura 6.3 mostra o ambiente onde os testes com o robô *Emmy II* foram realizados.

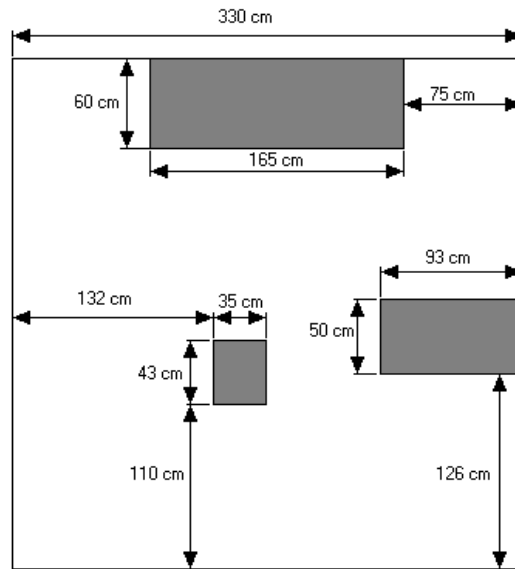


Figura 6.3 – Ambiente de testes do robô *Emmy II*.

O tempo de duração e o resultado de cada teste são o seguinte:

- Teste 1:

Duração: 3 minutos e 50 segundos.

Resultado: 13 colisões.

- Teste 2:

Duração: 3 minutos e 10 segundos.

Resultado: 7 colisões

- Teste 3:

Duração: 3 minutos e 30 segundos.

Resultado: 10 colisões.

- Teste 4:

Duração: 2 minutos e 45 segundos.

Resultado: 10 colisões.

Os sensores de ultra-som utilizados pelo robô *Emmy II* não detectam obstáculos a uma distância de 7,5 cm ou menos [Pol 96]. Os sensores de ultra-som emitem ondas sonoras e esperam o retorno dessas ondas (eco) para determinar a distância entre o obstáculo e o sensor; mas nem sempre essas ondas retornam, às vezes elas refletem para uma outra direção. Essas são as causas principais das colisões ocorridas durante os testes. Isto pode ser resolvido com a colocação de mais sensores e algumas modificações no *Paracontrol*.

As causas das colisões são as seguintes:

- Teste 1:

Total de colisões: 13.

Colisões causadas por causa do reflexo das ondas sonoras: 4.

Colisões causadas pela proximidade do obstáculo: 9.

- Teste 2:

Total de colisões: 7.

Colisões causadas por causa do reflexo das ondas sonoras: 2.

Colisões causadas pela proximidade do obstáculo: 5.

- Teste 3:

Total de colisões: 10.

Colisões causadas por causa do reflexo das ondas sonoras: 5.

Colisões causadas pela proximidade do obstáculo: 5.

- Teste 4:

Total de colisões: 10.

Colisões causadas por causa do reflexo das ondas sonoras: 4.

Colisões causadas pela proximidade do obstáculo: 6.

Uma outra possibilidade de colisão existe quando se detecta o estado falso. Nessa situação o robô vai para trás por 0,4 segundos, como não existe sensoriamento atrás do robô, a possibilidade de colisão é grande.

Capítulo 7

Conclusões

Neste trabalho propôs-se uma nova versão do *Paracontrol* (controlador lógico paraconsistente) baseado na lógica paraconsistente evidencial $E\tau$. Implementou-se esta nova versão do *Paracontrol* no robô móvel autônomo que foi denominado *Emmy II*. Como novidade, neste novo projeto do *Paracontrol* se verificou a possibilidade do controle da velocidade dos movimentos do robô baseado numa configuração especial do reticulado subjacente τ .

Uma das diferenças significativas entre as arquiteturas dos robôs *Emmy I* e *Emmy II* é que o segundo permite a realização de movimentos retrógrados. Isto se deve ao fato de que a interface entre o microcontrolador e os motores, no robô *Emmy II*, é realizada através de MOSFETs. Este tipo de interface permite a construção de um robô móvel compacto e eficiente.

Conforme o ensaio realizado verificou-se que o funcionamento do robô *Emmy II* mostrou-se superior ao apresentado pelo robô *Emmy I* em termos de menor número de colisões, o que nos parece bastante satisfatório.

Os resultados obtidos neste trabalho demonstram a importância da utilização da Lógica Paraconsistente em projetos de robôs móveis autônomos e controladores de processo. Conseguiu-se, assim, a implementação de um robô móvel autônomo sem supervisão externa e com possibilidade de navegação em ambiente não estruturado a um custo financeiro relativamente baixo.

O controlador apresentado neste trabalho permite uma gama de melhorias na tomada de decisão de robôs cada vez mais sofisticados. Com efeito, por exemplo, se dada uma configuração original do reticulado com os respectivos estados lógicos, pode-se tomar a decisão de considerar um novo reticulado que permite “adequar” a uma nova situação e tomar uma nova decisão. Esta ação claramente pode ser recursiva até que se encontre uma decisão satisfatória, tudo dentro dos limites e capacidades físicas oferecidas pelos circuitos, possibilitando assim pesquisas ulteriores interessantes.

Referências

- [Abe 92] ABE, J.M., “*Fundamentos da Lógica Anotada*” (Foundations of Annotated Logics), in Portuguese, Ph. D. Thesis, University of São Paulo, São Paulo, 1992.
- [Abe 97] ABE, J.M., Some Aspects of Paraconsistent Systems and Applications, *Logique et Analyse*, 157(1997), 83-96.
- [Abe 01] ABE, J.M., Annotated logics $Q\tau$ and model theory, in Logic, Artificial Intelligence, and Robotics, Proc. 2nd Congress of Logic Applied to Technology – LAPTEC’2001, Edts. J.M. Abe & J.I. Da Silva Filho, Frontiers in Artificial Intelligence and Its Applications, IOS Press, Amsterdam, Ohmsha, Tokyo, Vol. 71, ISBN 1 58603 206 2 (IOS Press), 4 274 90476 8 C3000 (Ohmsha), ISSN 0922-6389, 1-12, 287p., 2001.
- [Abe & Akama 99] ABE, J.M. & S. AKAMA, A Logical System for Reasoning with Fuzziness and Inconsistencies, Proceedings of the IASTED *International Conference on Artificial Intelligence and Soft Computing* (ASC’99), August 9-12, Honolulu, Hawaii, USA, 221-225, 1999.
- [Abe, Ávila & Nakamatsu 99] ABE, J.M., B.C. ÁVILA & K. NAKAMATSU, Paraconsistent Annotated Logic Programming - Paralog, *International Journal of Computing Anticipatory Systems*, vol. 6, 51-65, 1999.
- [Abe & Da Silva Filho 98] ABE, J.M. & J.I. DA SILVA FILHO, Inconsistency and Electronic Circuits, *Proceedings of The International ICSC Symposium on Engineering of Intelligent Systems* (EIS’98), Volume 3, Artificial Intelligence, Editor: E. Alpaydin, ICSC Academic Press International Computer Science Conventions Canada/Switzerland, ISBN 3-906454-12-6, 191-197, 1998.
- [Abe & Da Silva Filho 01a] ABE, J.M. & J.I. DA SILVA FILHO, A Para-Analyser Method to Increase Robot Availability Through Maintenance, Proceedings of the “*International Conference on Industrial Logistic 2001*”, 9-12 July 2001, Okinawa, Japan, 327-337, 2001.
- [Abe & Da Silva Filho 01b] ABE, J.M. & J.I. DA SILVA FILHO, Simulating Inconsistencies in a Paraconsistent Logic Controller, *Fifth International Conference on Computing Anticipatory Systems*, CASYS’2001, Organized by the non-profit association CHAOS, Centre for Hyperincursion and Anticipation in Ordered Systems CHAOS asbl, Institut de Mathématique, Université de Liège, Liège, Belgium, 2001.
- [Abe & Da Silva Filho 03] ABE, J.M. & J.I. DA SILVA FILHO, Manipulating Conflicts and Uncertainties in Robotics, *Journal of Multiple-Valued Logic and Soft Computing*, Philadelphia, USA, Old City Publishing, Vol. 9, Number 2, ISSN 1542-3980, 147-169, 2003.

- [Akama & Abe 00] AKAMA, S. & J.M. ABE, Fuzzy annotated logics, Proc. of 8th *International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems*, IPMU'2000, Organized by: Universidad Politécnica de Madrid (Spain), July 3-7, 2000, Madrid, Espanha, Vol. 1, 504-508, 2000.
- [Ávila 96] ÁVILA, B.C., *Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança*, PhD thesis, in Portuguese, University of São Paulo, São Paulo, 1996.
- [Ávila, Abe & Prado 97] ÁVILA, B.C., J.M. ABE & J.P.A. PRADO, ParaLog-e: A Paraconsistent Evidential Logic Programming Language, *XVII International Conference of the Chilean Computer Science Society*, IEEE Computer Society Press, pp 2-8, Valparaíso, Chile, 1997.
- [Da Costa, Abe, Da Silva Filho, Murolo & Leite 99] DA COSTA, N.C.A., J.M. ABE, J.I. DA SILVA FILHO, A.C. MUROLO & C.F.S. LEITE, *Lógica Paraconsistente Aplicada*, in Portuguese, ISBN 85-224-2218-4, Editôra Atlas, 214 pp., 1999.
- [Da Costa, Abe & Subrahmanian 91] DA COSTA, N.C.A., J.M. ABE & V.S. SUBRAHMANIAN, Remarks on Annotated Logic, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, Vol.37, pp.561-570, 1991.
- [Da Costa, Prado, Abe, Ávila & Rillo 95] DA COSTA, N. C. A., PRADO, J. P. A., ABE, J. M., ÁVILA, B. C., RILLO, M. Paralog: Um Prolog Paraconsistente Baseado Em Lógica Anotada. Coleção Documentos Série Lógica e Teoria da Ciência. IEA - USP, São Paulo - SP: USP - Universidade de São Paulo, Instituto de Estudos Avançados, n.18, p.1 - 28, 1995.
- [Da Silva Filho 97] DA SILVA FILHO, J.I., *Implementação de Circuitos Lógicos Fundamentados em Uma Classe de Lógicas Paraconsistentes Anotada*, in Portuguese, Master Thesis, University of São Paulo, São Paulo, 1997.
- [Da Silva Filho 99] DA SILVA FILHO, J.I., *Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com Dois Valores LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos*, in Portuguese, Ph. D. Thesis, University of São Paulo, São Paulo, 1999.
- [Da Silva Filho & Abe 99a] DA SILVA FILHO, J.I. & J.M. ABE, Para-Fuzzy Logic Controller – Part I: A New Method of Hybrid Control Indicated for Treatment of Inconsistencies Designed with the Junction of the Paraconsistent Logic and Fuzzy Logic, Proceedings of the *International ICSC Congress on Computational Intelligence Methods and Applications - CIMA'99*, Rochester Institute of Technology, RIT, Rochester, N.Y., USA, ISBN 3-906454-18-5, Editors: H. Bothe, E. Oja, E. Massad & C. Haefke, ICSC Academic Press, International Computer Science Conventions, Canada/Switzerland, 113-120, 1999.

- [Da Silva Filho & Abe 99b] DA SILVA FILHO, J.I. & J.M. ABE, Para-Fuzzy Logic Controller – Part II: A Hybrid Logical Controller Indicated for Treatment of Fuzziness and Inconsistencies, Proceedings of the *International ICSC Congress on Computational Intelligence Methods and Applications - CIMA'99*, Rochester Institute of Technology, RIT, Rochester, N.Y., USA, ISBN 3-906454-18-5, Editors: H. Bothe, E. Oja, E. Massad & C. Haefke, ICSC Academic Press, International Computer Science Conventions, Canada/Switzerland, 106-112, 1999.
- [Da Silva Filho & Abe 99c] DA SILVA FILHO, J.I. & J.M. ABE, Para-Analyser and Inconsistencies in Control Systems, Proceedings of the IASTED *International Conference on Artificial Intelligence and Soft Computing (ASC'99)*, August 9-12, Honolulu, Hawaii, USA, 78-85, 1999.
- [Da Silva Filho & Abe 01a] DA SILVA FILHO, J.I. & J.M. ABE, Emmy: a paraconsistent autonomous mobile robot, in Logic, Artificial Intelligence, and Robotics, Proc. 2nd *Congress of Logic Applied to Technology – LAPTEC'2001*, Edts. J.M. Abe & J.I. Da Silva Filho, Frontiers in Artificial Intelligence and Its Applications, IOS Press, Amsterdam, Ohmsha, Tokyo, Vol. 71, ISBN 1 58603 206 2 (IOS Press), 4 274 90476 8 C3000 (Ohmsha), ISSN 0922-6389, 53-61, 287p., 2001.
- [Da Silva Filho & Abe 01b] DA SILVA FILHO, J.I. & J.M. ABE, Paraconsistent electronic circuits, *International Journal of Computing Anticipatory Systems*, vol. 9, ISSN 1373-5411, ISBN 2-9600262-1-7, 337-345, 2001.
- [Da Silva Filho & Abe 01c] DA SILVA FILHO, J.I. & J.M. ABE, Paraconsistent analyser module, *International Journal of Computing Anticipatory Systems*, vol. 9, ISSN 1373-5411, ISBN 2-9600262-1-7, 346-352, 2001.
- [Da Silva Filho & Abe 01d] DA SILVA FILHO, J.I. & J.M. ABE, *Fundamentos das Redes Neurais Paraconsistentes – Destacando Aplicações em Neurocomputação*, in Portuguese, Editôra Arte & Ciência, ISBN 85-7473-045-9, 247 pp., 2001.
- [Da Silva Filho & Abe 01e] DA SILVA FILHO, J.I. & J.M. ABE, Para-Control: An Analyser Circuit Based On Algorithm For Treatment of Inconsistencies, Proc. of the *World Multiconference on Systemics, Cybernetics and Informatics*, ISAS, SCI 2001, Vol. XVI, Cybernetics and Informatics: Concepts and Applications (Part I), ISBN 980-07-7556-0, 199-203, Orlando, Florida, USA, 2001.
- [Dempster 68] DEMPSTER, A.P, Generalization of Bayesian inference, *Journal of the Royal Statistical Society*, Séries B-30, 205-247, 1968.
- [Kubota, Hisajima, Kojima & Fukuda 03] KUBOTA, N., D. HISAJIMA, F. KOJIMA & T. FUKUDA, Fuzzy and Neural Computing for Communication of a Partner Robot, *Journal of Multiple-Valued Logic and Soft Computing*, Philadelphia, USA, Old City Publishing, Vol. 9, Number 2, ISSN 1542-3980, 221-239, 2003.

- [Mills, Walker & Himebaugh 03] MILLS, J. W., T. WALKER & B. HIMEBAUGH, Lukasiwicz' Insect: Continuous-Valued Robotic Control After Ten Years, *Journal of Multiple-Valued Logic and Soft Computing*, Philadelphia, USA, Old City Publishing, Vol. 9, Number 2, ISSN 1542-3980, 131-146, 2003.
- [Nakamatsu, Abe & Suzuki 99c] NAKAMATSU, K., J.M. ABE & A. SUZUKI, "Defeasible Reasoning Between Conflicting Agents Based on VALPSN", American Association for Artificial Intelligence - AAAI'99 Workshop on Agents' Conflicts, ISBN 1-57735-092-8, TR WS-99-08, AAAI Press – American Association for Artificial Intelligence, Menlo Park, California, USA, 20-27, 1999.
- [Nakamatsu, Abe & Suzuki 01] NAKAMATSU, K., J.M. ABE & A. SUZUKI, Annotated Semantics for Defeasible Deontic Reasoning, Rough Sets and Current Trends in Computing, The Second International Conference on Rough Sets and Current Trends in Computing - RSCTC'2000, Banff, Canada, *Lecture Notes in Artificial Intelligence* series, LNAI 2005, Springer-Verlag, 470-478, 2000.
- [Pol 96] Manual do sensor de ultra-som com sistema POLAROID 500, 1996.
- [Prado 96] PRADO, J.P.A., *Uma Arquitetura em IA Baseada em Lógica Paraconsistente*, PhD Thesis, in Portuguese, University of São Paulo, Brazil, 1996.
- [Shoval, Ulrich & Borenstein 03] SHOVAL, S., I. Ulrich & J. Borenstein, Robotics-Based Obstacle-Avoidance Systems for the Blind and Visually Impaired, *IEEE Robotics & Automation Magazine*, 9-20, March 2003.
- [Sossai & Chemello 03] SOSSAI, C & G. CHEMELLO, Coherent Functions in Autonomous Systems, *Journal of Multiple-Valued Logic and Soft Computing*, Philadelphia, USA, Old City Publishing, Vol. 9, Number 2, ISSN 1542-3980, 171-194, 2003.
- [Subrahmanian 97] SUBRAHMANIAN, V.S., *On the Semantics of Quantitative Logic Programs*, Proc. 4th IEEE Symposium on Logic Programming, Computer Society Press, Washington D.C., 173-182, 1987.
- [Sylvan & Abe 96] SYLVAN, R. & J.M. ABE, On general annotated logics, with an introduction to full accounting logics, *Bulletin of Symbolic Logic*, 2, 118-119, 1996 (Abstract).

Anexo I

Programa em Linguagem Assembly do microcontrolador 8051 utilizado na versão final do robô *Emmy II*:

```
org    0000h
ljmp   ini

org    000Bh
ljmp   intr

org    0030h
mov    34h,#03fh    ;número 1
mov    35h,#02eh    ;valor limite do grau de evidência favorável
mov    36h,#010h    ;valor limite do grau de evidência contrária
mov    37h,#0e1h    ;valor limite do grau de certeza para determinação
                        ;de tipos de falsidade (E1h=-0,5)

ini:   clr    90h
        clr    92h
        clr    94h
        clr    95h
        clr    96h
```

```

clr    97h

lcall  sens1      ;coloca no acumulador a distância em centímetros
                    ;do objeto ao sensor1

cjne   a,#03fh,cor10 ;comparação com o número 1
nop

cor10: jc    cor11
        mov  a,34h

cor11: clr    c
        mov  30h,A      ;grau de evidência favorável

lcall  sens2      ;coloca no acumulador a distância em centímetros
                    ;do objeto ao sensor2

rot31: cjne   a,#03fh,cor00 ;comparação com o número 1
nop

cor00: jc    cor01
        mov  a,34h

cor01: clr    c
        mov  31h,A
        mov  a,34h
        subb a,31h
        mov  31h,a      ;grau de evidência contrária

clr    c

```

```

mov    a,30h
subb   a,31h
mov    32h,a      ;grau de certeza

clr    c

mov    a,30h
add    a,31h
subb   a,34h
mov    33h,a      ;grau de incerteza

mov    a,30h
cjne   a,#026h,comp00 ;comparação com o valor limite do grau de evidência
favorável
nop
comp00:  jc    comp01
mov    a,31h
cjne   a,#016h,comp02 ;comparação com o valor limite do grau de evidência
;contrária
nop
comp02:  jc    rotverd
lcall  incons
ljmp   ini

rotverd:lcall  verd

```

```

        ljmp  ini
comp01: mov  a,31h
        cjne  a,#016h,comp03    ;comparação com o valor limite do grau de evidência
contrária
        nop
comp03:  jc   rotparac
        ljmp  rotfalso

rotparac: lcall parac
        ljmp  ini

rotfalso:  mov  a,32h
        cjne  a,#0e1h,fff00 ;se o grau de certeza for menor do que o valor limite de
                                ;grau de certeza o estado é falso. Caso contrario deve se
                                ;verificar se o estado é quase falso tendendo
                                ;a inconsistente ou paracompleto.

        nop
fff00:  jnc  fff01
        lcall falso
        ljmp  ini

fff01:  mov  a,033h

```



```

    jb     0e7h,fff03    ;se o grau de contradição é positivo então o estado é quase
                        ;falso tendendo a inconsistente. Caso contrario é falso
                        ;tendendo a paracompleto

    lcall  qftinc

    ljmp   ini

fff03: lcall  qftpar

    ljmp   ini

verd:  setb  94h

        setb  95h

        lcall  atraso

        ret

incons:setb  95h

        lcall  atraso

        ret

parac: setb  94h

        lcall  atraso

        ret

falso: setb  95h        ;estado falso

```

```

    setb    97h

    setb    96h

    setb    94h

    lcall   atraso

    ret

qftinc: setb    96h        ;estado quase falso tendendo a inconsistente

    setb    94h

    lcall   atraso

    ret

qftpar: setb    95h        ;estado quase falso tendendo a paracompleto

    setb    97h

    lcall   atraso

    ret

atraso:  mov    R3,#03h

dlay00: mov    R2,#0ffh

dlay10: mov    R1,#0ffh

dlay20: djnz   R1,dlay20

    djnz   R2,dlay10

    djnz   R3,dlay00

    ret

sens1:  mov    r7,34h        ; limita a distância no número 1

```

```

mov  r6,#00h      ; zera o registrador 6
mov  a,#00h      ; zera o acumulador
clr  8ch          ; desligar o t/c 0
mov  tmod,#01h   ; define o modo 1 para o t/c 0
mov  tcon,#00h   ; temporizadores desligados e interrupções externas
                    ; desativadas
mov  ip,#00h     ; nenhuma interrupção possui prioridade
mov  ie,#082h    ; habilita a interrupção pedida por t/c 0
mov  th0,#0ffh   ; tempo para o som percorrer 4cm (4 * 29,41us [tempo
mov  tl0,#09bh   ; para o som percorrer 1 cm]) = 117,64µs (118 ciclos de
                    ; máquinas [1 ciclo de máquina = 1µs])
setb 91h         ; prepara bit 1 para leitura de nível alto do echo
setb 8ch         ; liga o t/c 0
setb 90h         ; liga init
repete: jb 91h,rot3 ; desvia se p1.1 estiver em 1
        cjne r6,#02h,repete; desvia para rot5 se o registrador 6 conter o
                    ; valor 02h
        nop
rot3:  ret
                    ; enquanto não houver retorno do echo (p1.0="1")
                    ; fica em loop
                    ; mas sai daqui se for gerada interrupção interna
                    ; pelo contador
                    ; ou seja, 'contou' 127cm sem acusar obstáculo

```

```

intr: inc  a      ; incrementa o acumulador em 1 para cada cm ida e
          ;volta sem detecção de objeto

mov  th0,#0ffh   ; tempo para o som percorrer 2cm (2 * 29,41us [tempo
          ;para o som percorrer 1 cm]) = 58,82us [59 ciclos de
          ;máquinas [1 ciclo de

mov  tl0,#0a1h   ;máquina = 1us]] + 11 ciclos de maquinas [equivalentes
          ;as instruções executadas durante a interrupção]

djnz r7,rot4     ; limita a distância máxima em 7f

mov  r6,#02h     ; coloca o valor 02h no registrador 6

rot4: reti

sens2: mov  r7,34h      ; limita a distância máxima em 7Fh cm
        mov  r6,#00h    ; zera o registrador 6
        mov  a,#00h     ; zera o acumulador
        clr  8ch        ; desligar o t/c 0
        mov  tmod,#01h  ; define o modo 1 para o t/c 0
        mov  tcon,#00h  ; temporizadores desligados e interrupções externas
          ; desativadas
        mov  ip,#00h    ; nenhuma interrupção possui prioridade
        mov  ie,#082h   ; habilita a interrupção pedida por t/c 0
        mov  th0,#0ffh  ; tempo para o som percorrer 4cm (4 * 29,41us [tempo para o
          ;som percorrer 1 cm]) = 117,64us (118 ciclos de máquinas [1
          ;ciclo de máquina = 1us])

        mov  tl0,#09bh

```

```

setb 93h ; prepara bit 1 para leitura de nível alto do echo
setb 8ch ; liga o t/c 0
setb 92h ; liga init
repete1:jb 93h,sai ; desvia se p1.1 estiver em 1
cjne r6,#02h,repete1 ; desvia para rot5 se o registrador 6 conter o valor 02h
nop
sai: ret ; desvio incondicional enquanto não houver retorno do echo (p1.0="1") fica
; em loop mas sai daqui se for gerada interrupção interna pelo contador ou
; seja, 'contou' 127cm sem acusar obstáculo

```