

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Pedro Morangueira Carlson

**Estratégia Baseada em Sistema Fuzzy
Aplicada no Domínio de Futebol de
Robôs**

São Carlos
2012

Pedro Morangueira Carlson

Estratégia Baseada em Sistema Fuzzy Aplicada no Domínio de Futebol de Robôs

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos, da
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase
em Eletrônica

ORIENTADOR: Prof. Dr. Ivan Nunes da Silva

São Carlos
2012

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Atendimentos ao Usuário do Serviço de
Biblioteca – EESC/USP.

C284e Carlson, Pedro Morangueira
Estratégia baseada em sistema fuzzy aplicada no
domínio de futebol de robôs. / Pedro Morangueira
Carlson; orientador ; coorientador Ivan Nunes da
Silva. São Carlos, 2012.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2012.

1. Sistemas fuzzy. 2. Estratégias inteligentes. 3.
RoboCup soccer simulation 2D. I. Título.

FOLHA DE APROVAÇÃO

Nome: Pedro Morangueira Carlson

Título: "Estratégia Baseada em Sistema Fuzzy Aplicada no Domínio de Futebol de Robôs"


Trabalho de Conclusão de Curso defendido e aprovado
em 28 / 06 / 2012,

com NOTA 10,0 (DEZ , ZERO), pela comissão julgadora:


Prof. Associado Ivan Nunes da Silva (Orientador)-- EESC/USP


Prof. Dr. Danilo Hernane Spatti - EESC/USP


Prof. Dr. Ricardo Augusto Souza Fernandes - EESC/USP


Prof. Associado Homero Schiabel
Coordenador da CoC-Engenharia Elétrica
EESC/USP

dedicatória

Aos meus pais, por toda paciência e incentivo ao longo desses anos.

À minha família, Carlos e Magda, pelo apoio incondicional e paciência durante a execução deste trabalho.

Ao Prof. Dr. Ivan Nunes da Silva, pela orientação, ajuda e prestatividade ao longo de toda a graduação.

Aos moradores e agregados de casa, por tornarem o pandemônio habitável.

Aos amigos da turma, por todas viagens e momentos vividos juntos.

Aos amigos do Warthog Robotics, por todas madrugadas passadas em claro nas vésperas de campeonatos, e por me ensinarem o verdadeiro significado de engenharia.

Aos demais amigos apreciadores de um bom *Rock'n'Roll*.

Sumário

Lista de Tabelas

Lista de Figuras

Resumo	
Abstract	
1 Introdução	p. 21
1.1 Objetivo	p. 21
1.2 Organização	p. 22
2 Futebol de Robôs	p. 23
2.1 <i>RoboCup</i>	p. 23
2.2 Simulação 2D	p. 26
3 Sistemas <i>Fuzzy</i>	p. 29
3.1 Conjuntos <i>Fuzzy</i>	p. 30
3.2 Lógica <i>Fuzzy</i>	p. 33
3.3 Sistemas de Inferência <i>Fuzzy</i>	p. 34
4 Abordagem <i>Fuzzy</i> para o Posicionamento dos Agentes	p. 37
4.1 Características do Time	p. 37
4.2 Características do Sistema <i>Fuzzy</i> Implementado	p. 38
4.3 Resultados Obtidos	p. 42
5 Conclusões	p. 47
Referências Bibliográficas	p. 49

Lista de Tabelas

4.1	Média de Gols	p. 46
4.2	Desempenho (Vitórias-Empates-Derrotas)	p. 46
4.3	Aproveitamento (%)	p. 46
4.4	Gols Prós	p. 46
4.5	Gols Contra	p. 46

Lista de Figuras

2.1	Robôs da categoria <i>Humanoid</i> da <i>RoboCup</i> [6].	p. 24
2.2	Robôs da categoria <i>Middle Size</i> da <i>RoboCup</i> [6].	p. 24
2.3	Robôs da categoria <i>Small Size</i> da <i>RoboCup</i> [6].	p. 25
2.4	Robôs da categoria <i>Standard Platform</i> da <i>RoboCup</i> [6].	p. 25
2.5	Robôs da categoria <i>Simulation 2D</i> da <i>RoboCup</i> [6].	p. 26
2.6	Robôs da categoria <i>Simulation 3D</i> da <i>RoboCup</i> [6].	p. 26
2.7	Arquitetura da categoria Simulação 2D da <i>RoboCup</i> [11].	p. 27
2.8	Monitor utilizado na Simulação 2D da <i>RoboCup</i> [13].	p. 28
3.1	Exemplos de diferentes formas de funções de pertinência [21].	p. 31
3.2	Sistema de Inferência <i>fuzzy</i> [15].	p. 35
4.1	Formação 4-4-2 [39].	p. 38
4.2	Formação 4-3-3 [39].	p. 39
4.3	Formação 4-2-3-1 [39].	p. 39
4.4	Função de Pertinência - Tempo.	p. 40
4.5	Função de Pertinência - Posição de bola.	p. 40
4.6	Função de Pertinência - Ataques bem Sucedidos (%).	p. 41
4.7	Função de Pertinência - Defesas bem Sucedidas (%).	p. 41
4.8	Função de Pertinência - Formação Tática.	p. 41
4.9	Fluxo de Dados no Sistema <i>Fuzzy</i>	p. 42
4.10	Resultados dos jogos - <i>Agent2D</i> x Nadco	p. 43
4.11	Resultados dos jogos - <i>Agent2D</i> x MarliK	p. 44
4.12	Resultados dos jogos - <i>Agent2D</i> x Helios	p. 44
4.13	Resultados dos jogos - Time Desenvolvido x Nadco	p. 44
4.14	Resultados dos jogos - Time Desenvolvido x MarliK	p. 45

4.15 Resultados dos jogos - Time Desenvolvido x Helios	p. 45
4.16 Resultados dos jogos - Time Desenvolvido x <i>Agent2D</i>	p. 46

Resumo

Um dos maiores campos de pesquisa dentro da área da robótica consiste em encontrar soluções para a navegação e movimentação dos agentes robóticos em ambientes estáticos ou dinâmicos. Tais soluções se baseiam em ferramentas inteligentes e heurísticas, com a finalidade de se adaptarem rapidamente às possíveis mudanças no entorno do robô, garantindo uma rota otimizada e segura. Dentre as estratégias mais encontradas na literatura destacam-se os sistemas *fuzzy*, que são muito utilizados para mapeamento de ambientes complexos, onde o conhecimento especialista é imprescindível. Neste trabalho foi implementado um sistema *Fuzzy* para a navegação e escolha de posicionamento de robôs autônomos, utilizando um ambiente simulado no domínio de futebol de robôs, altamente dinâmico e complexo.

Palavras-chave: Sistemas *Fuzzy*, Estratégias Inteligentes, *RoboCup Soccer Simulation 2D*

Abstract

One of the biggest research fields in robotics is in finding solutions to the navigation and movements of robots in static or dynamic environments. These solutions are based on intelligent approaches and heuristics, with the purpose of quickly adapting to possible changes in the surroundings of the robotic agent, in order to guarantee an optimal and safe route. Fuzzy systems have many approaches among the most common strategies found in the literature, since they are useful in mapping complex environments, where expert knowledge is indispensable. In this work, a Fuzzy system was implemented to evaluate autonomous robots navigation and positioning, using a dynamic and complex simulated robot soccer environment.

Keywords: Fuzzy Systems, Intelligent Strategies, RoboCup Soccer Simulation 2D

1 *Introdução*

Uma das áreas de maior desenvolvimento técnico e científico nos últimos anos é a robótica. Esse desenvolvimento começou nas indústrias, com a implementação de robôs para automatizar processos e serviços, visto que eram capazes de executar tarefas repetitivas sem a preocupação de que ocorresse algum imprevisto durante a execução, uma vez que fossem devidamente programados.

Com o crescente desenvolvimento tecnológico, os robôs deixaram de ser utilizados somente nas indústrias e passaram a serem utilizados em outros propósitos, como por exemplo, medicina, forças armadas, entretenimento e lazer. Além disso, deixaram de ser somente fixos e se tornaram robôs móveis e autônomos capazes de executar tarefas mais complexas.

Por conta disso, surgiu o conceito de futebol de robôs [1], com o pretexto de desenvolvimento de inteligência artificial para aplicações robóticas, onde as mais diversas abordagens inteligentes são implementadas e testadas. Entre elas, pode-se destacar:

- Algoritmos Genéticos;
- Redes Neurais Artificiais;
- Sistemas *Fuzzy*;
- Aprendizado por reforço;
- Campos Potenciais Aplicados à Navegação.

1.1 **Objetivo**

O objetivo deste trabalho é avaliar a eficiência de um sistema de inferência *fuzzy* em um ambiente de futebol de robôs simulado. Para isso, o sistema *fuzzy* irá inferir posicionamento de robôs móveis autônomos e os resultados serão comparados com um time base pré-definido, usado como plataforma de testes.

1.2 Organização

Este trabalho está organizado conforme segue abaixo. No Capítulo 2 serão abordados os conceitos de futebol de robôs, seus focos de pesquisa e diferentes categorias. O Capítulo 3 tratará sobre sistemas *fuzzy* e suas diversas aplicações. O Capítulo 4 apresenta a metodologia proposta para o posicionamento de agentes no domínio de futebol de robôs. As conclusões são descritas no Capítulo 5, as quais se referem aos resultados obtidos e aos avanços nas áreas de sistemas *fuzzy* e da robótica aplicada no domínio de futebol de robôs. Posteriormente, apresentam-se as Referências Bibliográficas utilizadas durante a pesquisa desenvolvida neste trabalho.

2 *Futebol de Robôs*

Na busca de uma plataforma para o desenvolvimento de inteligência artificial para aplicações robóticas, em 1993 surgiu o conceito de futebol de robôs autônomos, introduzido por Itsuki Noda [1]. A competição de futebol de robôs é um ambiente dinâmico e abrangente para a aplicação de diferentes tipos de tecnologias e técnicas inteligentes. As abordagens mais encontradas no domínio de futebol de robôs são aprendizado por reforço baseado em heurísticas [2], redes neurais artificiais [3], algoritmos evolutivos [4] e sistemas *fuzzy* [5].

Neste capítulo, aspectos gerais da *RoboCup* serão abordados na Seção 2.1. A Seção 2.2 tratará sobre a arquitetura e conceitos do ambiente simulado de futebol de robôs e da categoria de Simulação 2D.

2.1 *RoboCup*

A *RoboCup* (*Robot World Cup*) é uma iniciativa científica internacional com o objetivo de avançar o estado da arte de robôs inteligentes. Quando criada no Japão, em 1997, tem como meta montar uma equipe de robôs humanóides capazes de ganhar contra os campeões da Copa do Mundo (humanos) em 2050. Esse objetivo ainda continua, mas desde então a *RoboCup* se expandiu para outros domínios relevantes, com base nas necessidades da sociedade moderna. Além do futebol, a *RoboCup* tem competições na área de resgate (*RoboCupRescue*), na área doméstica (*RoboCup@Home*) e para crianças e adolescentes (*RoboCupJunior*). O domínio de futebol de robôs é composto por cinco categorias, que são *Simulation*, *Humanoid*, *Middle Size*, *Small Size* e *Standard Platform* [6].

Na categoria *Humanoid* (Figura 2.1), robôs autônomos com aparência semelhante a seres humanos jogam futebol uns contra os outros. Dentre as áreas de pesquisa envolvidas com essa categoria se destacam a dinâmica de andar, correr e chutar sem perder o equilíbrio, percepção visual da bola dos jogadores, do campo e cooperação entre agentes. Esta categoria é dividida em três divisões, de acordo com os tamanhos dos robôs. São elas *Teen Size*, *Kid Size* e *Adult Size* [6].

A categoria *Middle Size* (Figura 2.2) é composta por robôs que não podem exceder 50 cm de diâmetro e jogam futebol em equipes de até seis robôs, com uma bola de tamanho



Figura 2.1: Robôs da categoria *Humanoid* da *RoboCup* [6].

oficial definido pela FIFA [7] em um campo semelhante a um campo de futebol em escala. Todos os sensores são embarcados e os robôs se comunicam por rede *wireless*. O foco das pesquisas nesta categoria são autonomia e cooperação em termos de planejamento e percepção [6].

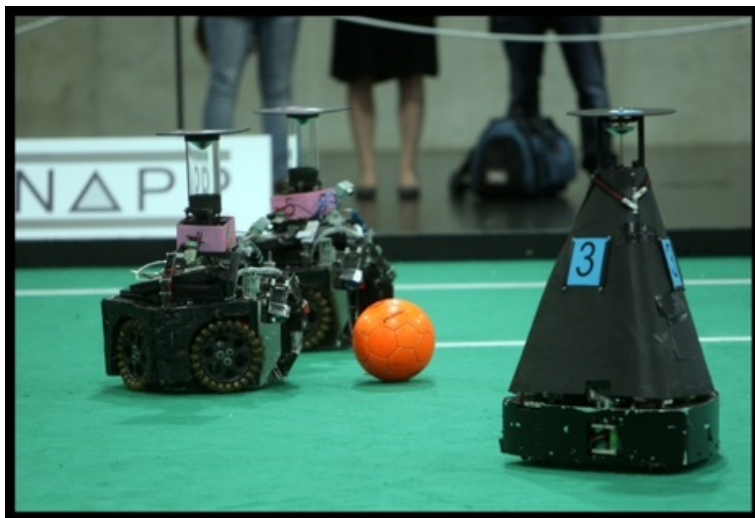


Figura 2.2: Robôs da categoria *Middle Size* da *RoboCup* [6].

A categoria *Small Size* (Figura 2.3) é uma das mais antigas da *RoboCup*. Seu foco é cooperação inteligente de sistemas multi-agentes e controle em um ambiente altamente dinâmico, com um sistema centralizado/distribuído híbrido [6].

Na categoria *Standard Platform* (Figura 2.4), todas as equipes utilizam robôs idênticos, portanto, as equipes se concentram no desenvolvimento de inteligência artificial. Não é permitido visão omnidirecional, forçando a utilização de tomada de decisões e troca de informações para a localização dos agentes e da bola. O robô padrão desta categoria era o AIBO [8], mas foi substituído pelo NAO [9], baseado em humanóides [6].

Assim como a *Small Size*, a categoria *Simulation* é uma das mais antigas da *Robo-*



Figura 2.3: Robôs da categoria *Small Size* da *RoboCup* [6].



Figura 2.4: Robôs da categoria *Standard Platform* da *RoboCup* [6].

Cup. O foco é o desenvolvimento de inteligência artificial, cooperação em sistemas multi-agentes, navegação e tomada de decisões em sistemas dinâmicos. Cada jogador (agente) é um software que joga futebol num campo virtual simulado. Existem duas divisões, 2D (Figura 2.5) e 3D (Figura 2.6) [6].

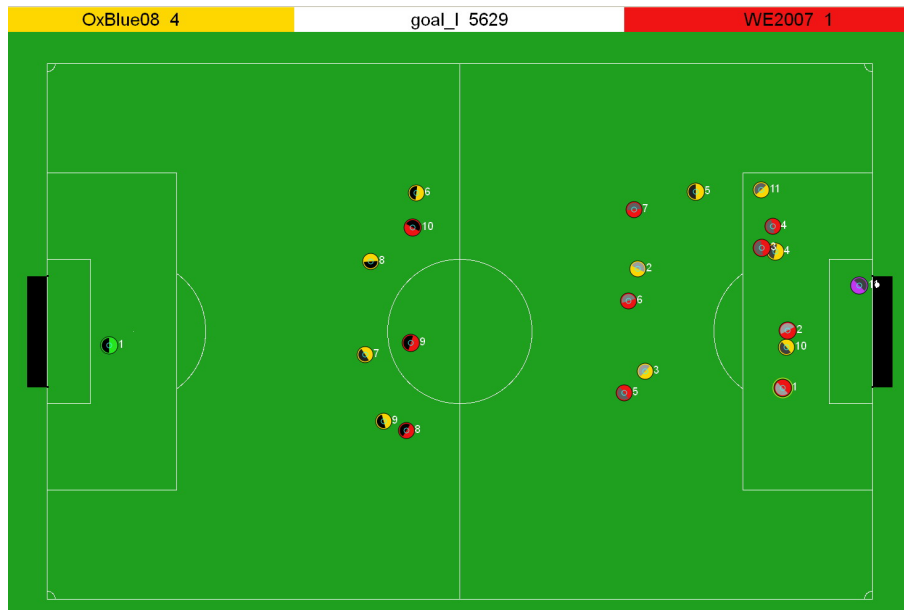


Figura 2.5: Robôs da categoria *Simulation 2D* da *RoboCup* [6].

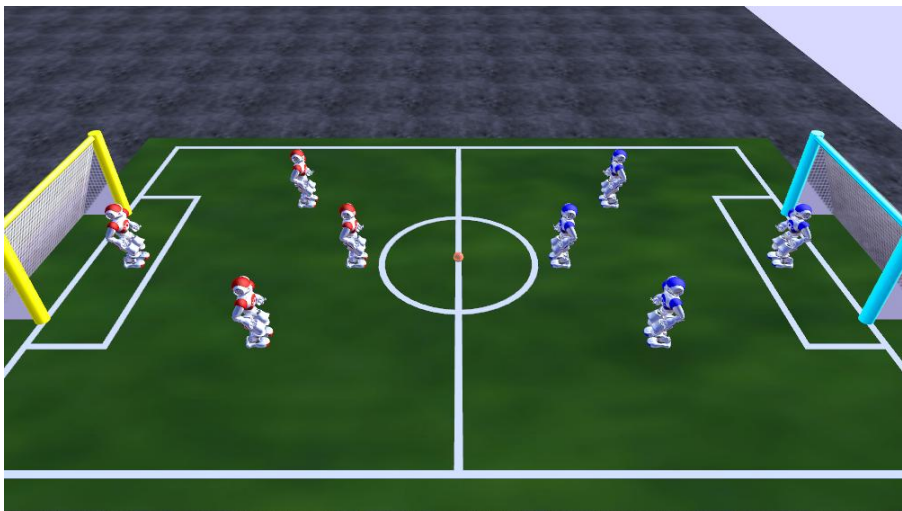


Figura 2.6: Robôs da categoria *Simulation 3D* da *RoboCup* [6].

2.2 Simulação 2D

Por não necessitar de robôs físicos, as categorias de simulação da *RoboCup* são as que possuem maior pesquisa e desenvolvimento nas áreas de inteligência artificial e estratégia da equipe. Na categoria de Simulação 2D, duas equipes de onze *softwares* autônomos (chamados de agentes) jogam futebol em um estádio de futebol bidimensional virtual,

representado por um servidor central, chamado *SoccerServer*. A arquitetura da categoria Simulação 2D é representada pela Figura 2.7. Os jogos são exibidos em um monitor chamado *SoccerMonitor*, conforme a Figura 2.8 [10].

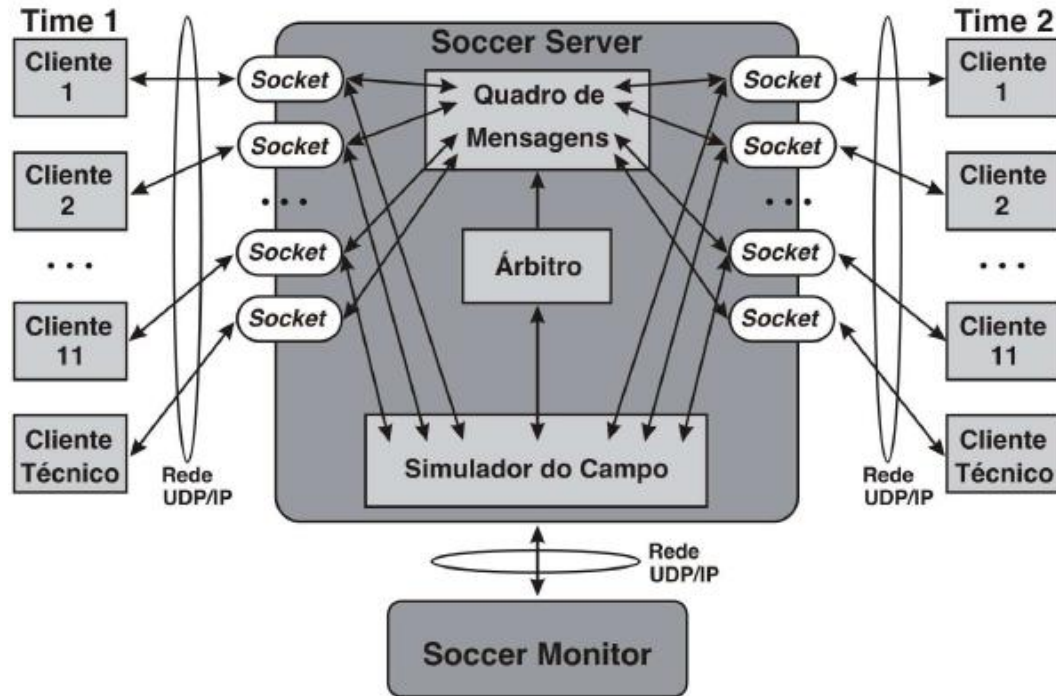


Figura 2.7: Arquitetura da categoria Simulação 2D da *RoboCup* [11].

Este servidor gerencia todas as informações do mundo virtual, como a posição atual de todos os jogadores e da bola e a física. A partida é realizada no formato cliente/servidor: Um servidor, *SoccerServer*, fornece um campo virtual e simula todos os movimentos de bola e jogadores. Cada cliente controla os movimentos de um jogador. A comunicação entre o servidor e cada agente é feita via *UDP/IP* (*User Datagram Protocol/Internet Protocol*, que é um protocolo orientado à conexão, com mecanismos para começar e finalizar conexões, negociar e retransmitir pacotes corrompidos [12]). Cada agente robótico recebe uma entrada relativa e ruidosa de seus sensores virtuais (visual, acústica e física), interpretam esses pacotes e executam algumas ações básicas (como correr, chutar, olhar, escutar) a fim de influenciar o seu ambiente. O servidor proporciona um ambiente desafiador, de maneira que o grande desafio nas categorias de Simulação é avaliar de forma inteligente as informações recebidas pelos sensores de cada agente, para inferir a melhor ação possível a ser executada [10].

O monitor (Figura 2.8) é uma ferramenta de visualização que permite ver o que está acontecendo dentro do servidor durante um jogo. As informações mostradas incluem o placar, os nomes da equipe, as posições de todos os jogadores e da bola. O monitor também é uma interface simplificada para o servidor, com informações da situação de jogo, permitindo que um árbitro humano intervenha no jogo [10].



Figura 2.8: Monitor utilizado na Simulação 2D da *RoboCup* [13].

Cada cliente que se conecta no servidor é um processo separado. Sua conexão é feita por uma única porta, onde é responsável por atuar no ambiente simulado. Os jogadores enviam ao simulador quais as ações que serão tomadas, o qual as executam, desde que sejam válidas. Na sequência, o servidor envia as informações do ambiente virtual para cada agente, que devem filtrar os ruídos e, por meio de alguma inteligência, decidir qual a próxima ação a ser tomada [11].

O simulador é um sistema em tempo real que trabalha com intervalos discretos (ciclos). A duração de cada ciclo é de 100ms, onde o servidor fica aguardando as informações dos agentes. Assim que todas as informações são recebidas o mundo simulado é atualizado, iniciando um novo ciclo. A duração de uma partida é de 6000 ciclos. A estrutura do simulador é hierárquica, onde o ambiente de baixo nível é responsável por se comunicar com o servidor e processar ações e informações. No ambiente de alto nível é onde as tomadas de decisões são feitas, por intermédio de abordagens inteligentes [11].

3 *Sistemas Fuzzy*

O raciocínio em lógica *fuzzy* tem como pretensão se assemelhar ao raciocínio humano. Ele permite que, por intermédio de valores aproximados e inferências, assim como dados incompletos ou nebulosos, seja possível chegar na solução de problemas complexos. Ele foi inicialmente proposto por Zadeh [14], com o objetivo de fornecer um ferramental matemático para o tratamento de informações de caráter impreciso ou vago [15], mediante o uso de variáveis linguísticas. Essas variáveis simplificam a modelagem de mecanização e controle de tarefas, visto que o problema pode ser escrito em termos que operadores humanos podem entender, ao invés de equações e valores exatos, que nem sempre podem ser aplicados dependendo da natureza da tarefa. A lógica *fuzzy*, baseada nessa teoria, foi inicialmente construída a partir dos conceitos já estabelecidos de lógica clássica, onde operadores foram definidos à semelhança dos tradicionalmente utilizados, enquanto outros foram introduzidos ao longo do tempo, por necessidades de caráter eminentemente prático [15].

Um sistema fuzzy pode ser descrito como um conjunto de regras lógicas *fuzzy* ou equações *fuzzy*. Essas regras lógicas podem ser entendidas como proposições associadas a distribuições de possibilidades, formados pelos conjuntos *fuzzy*. As equações *fuzzy* fornecem uma representação para sistemas com parâmetros e entradas *fuzzy*. Restrições e metas *fuzzy* também são consideradas [16].

Dentre as várias aplicações de sistemas *fuzzy*, destacam-se as aplicações em controle e automatização de processos, onde as variáveis linguísticas permitem que seja aproveitado a experiência e conhecimento dos operadores na modelagem do processo, como feito por Mamdani [17], Jantzen [18], entre outros. No domínio de futebol de robôs, sistemas de inferência *fuzzy* foram aplicados na navegação [19] e na escolha de comportamentos [20].

Neste capítulo, definições de conjuntos *fuzzy* serão apresentados na Seção 3.1. Na Seção 3.2 será tratada a lógica *fuzzy* e, na Seção 3.3 serão abordados sistemas de inferência *fuzzy*.

3.1 Conjuntos *Fuzzy*

Um conjunto *fuzzy* pode ser definido matematicamente atribuindo, a cada possível indivíduo no universo em questão, um valor que representa o seu nível de pertinência nesse conjunto. Este nível corresponde a quanto que esse indivíduo é semelhante ou compatível com o conceito representado pelo conjunto *fuzzy*. Assim, os indivíduos podem pertencer ao conjunto com diferentes graus de compatibilidade, dependendo de seu nível de pertinência [21].

Na teoria clássica dos conjuntos, o conceito de pertinência de um elemento a um conjunto é bem definido. Dado um conjunto A em um universo X , os elementos deste universo simplesmente pertencem ou não pertencem àquele conjunto, ou seja, seus níveis de pertinência são representados por números reais, com valores variando no intervalo fechado entre 0 e 1, conforme a função característica (3.1) [15]:

$$f_A(x) = \begin{cases} 1 & \longleftrightarrow x \in A \\ 0 & \longleftrightarrow x \notin A \end{cases} \quad (3.1)$$

Para um conjunto *fuzzy*, Zadeh [14] propôs uma caracterização mais ampla, generalizando a função característica de modo que ela pudesse assumir um número infinito de valores dentro do intervalo $[0,1]$. Desta maneira, um conjunto A em um universo X é definido por uma função de pertinência dada por:

$$\mu_A(x) : X \rightarrow [0, 1] \quad (3.2)$$

onde $\mu_A(x)$ indica o grau de compatibilidade de x com o conjunto A [15].

As variáveis linguísticas constituem o “vocabulário” da lógica *fuzzy*, trazendo toda a incerteza presente no pensamento e na expressão oral do ser humano, para sistemas de decisão que priorizam o padrão e respeitam determinada metodologia durante o cálculo computacional envolvido [22]. Sua principal função é fornecer uma maneira sistemática para uma caracterização aproximada de fenômenos complexos ou mal definidos. A utilização do tipo de descrição linguística empregada por seres humanos, e não de variáveis quantificadas, permite o tratamento de sistemas que são muito complexos para serem analisados por meio de termos matemáticos convencionais [15].

Funções de pertinência são generalizações, onde os valores atribuídos aos elementos do universo assumem uma determinada faixa, indicando o nível de adesão destes elementos do conjunto em questão. Valores maiores indicam níveis mais elevados de adesão ao conjunto. Dependendo do conceito a ser representado, do contexto em que serão utilizadas e da experiência do usuário, elas podem ter diferentes formas, representadas pela Figura 3.1. Os conjuntos definidos pelas funções de pertinência são os conjuntos *fuzzy* [21].

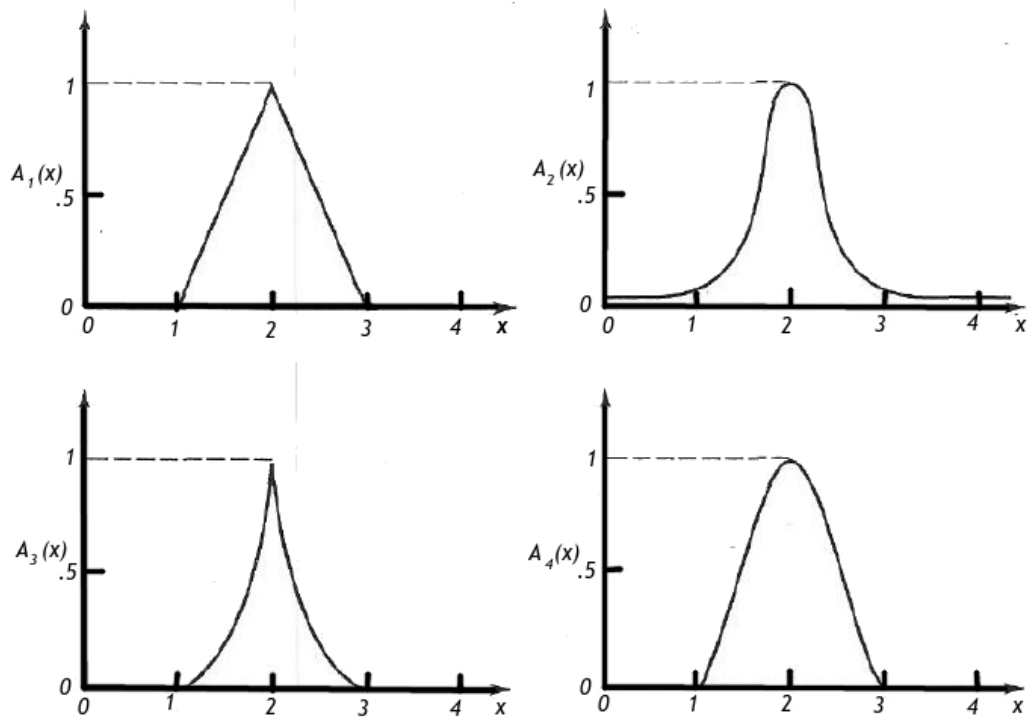


Figura 3.1: Exemplos de diferentes formas de funções de pertinência [21].

Por se tratarem de conjuntos ordinários, uma série de definições, operações e propriedades podem ser aplicadas nos conjuntos *fuzzy*. Dentre as mais relevantes para o assunto, pode-se destacar [23]:

- Um conjunto *fuzzy* A em X é **vazio** se e somente se sua função de pertinência é igual a zero sobre todo X :

$$A = \emptyset \iff \mu_A(x) = 0, \forall x \in X \quad (3.3)$$

- O complemento A' de um conjunto *fuzzy* A é dado por:

$$\mu_{A'}(x) = 1 - \mu_A(x), \forall x \in X \quad (3.4)$$

- Dois conjuntos fuzzy A e B em X são iguais se suas funções de pertinência forem iguais sobre todo X :

$$A = B \iff \mu_A(x) = \mu_B(x), \forall x \in X \quad (3.5)$$

- Um conjunto fuzzy A é um subconjunto de B se sua função de pertinência for menor ou igual à de B sobre todo X :

$$A \subset B \text{ se } \mu_A(x) \leq \mu_B(x), \forall x \in X \quad (3.6)$$

- Involução:

$$(A')' = A \quad (3.7)$$

- Idempotência:

$$\begin{cases} A \cap A = A \\ A \cup A = A \end{cases} \quad (3.8)$$

- Comutatividade:

$$\begin{cases} A \cap B = B \cap A \\ A \cup B = B \cup A \end{cases} \quad (3.9)$$

- Associatividade:

$$\begin{cases} (A \cap B) \cap C = A \cap (B \cap C) \\ (A \cup B) \cup C = A \cup (B \cup C) \end{cases} \quad (3.10)$$

- Distributividade:

$$\begin{cases} A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \end{cases} \quad (3.11)$$

- Absorção:

$$\begin{cases} A \cap (A \cup B) = A \\ A \cup (A \cap B) = A \end{cases} \quad (3.12)$$

- Lei Transitiva:

$$\text{se } A \subset B \text{ e } B \subset C \rightarrow A \subset C \quad (3.13)$$

- Leis de De Morgan:

$$\begin{cases} (A \cap B)' = A' \cup B' \\ (A \cup B)' = A' \cap B' \end{cases} \quad (3.14)$$

- Conjunto Vazio:

$$\begin{cases} A \cap \emptyset = \emptyset \\ A \cup \emptyset = A \end{cases} \quad (3.15)$$

- Conjunto Universo:

$$\begin{cases} A \cap X = A \\ A \cup X = X \end{cases} \quad (3.16)$$

Para conjuntos *fuzzy* ordinários, uma *relação* exprime a presença ou a ausência de uma associação (ou interação) entre elementos de dois ou mais conjuntos. Elas representam o nível de associação entre elementos de dois ou mais conjuntos *fuzzy*. Dados dois universos X e Y , a relação R definida em $X \times Y$ é um subconjunto do produto cartesiano dos dois universos, de tal forma que $R : X \times Y \rightarrow \{0, 1\}$. Ou seja, se algum $x \in X$ e $y \in Y$ estiverem relacionados, $R(x, y) = 1$, caso contrário, $R(x, y) = 0$, conforme a função de pertinência bivalente (3.17) [15]:

$$f_R(x) = \begin{cases} 1 & \longleftrightarrow (x, y) \in R \\ 0 & \text{caso contrário} \end{cases} \quad (3.17)$$

Proposições *fuzzy* são frases compostas de variáveis *fuzzy* com seus respectivos valores. Elas podem ser combinadas por meio de diferentes operadores, tais como conectivos lógicos E e OU , a negação $NÃO$ e o operador de implicação $SE \rightarrow ENTÃO$. A representação dessas proposições se dá mediante o produto cartesiano das variáveis linguísticas e da utilização das relações *fuzzy*. O conectivo E pode ser usado com variáveis linguísticas em diferentes universos, enquanto que o conectivo OU pode conectar valores linguísticos de uma mesma variável que estão no mesmo universo. A operação $NÃO$ é semanticamente sinônima da negação em linguagem natural. O operador $SE \rightarrow ENTÃO$ é uma *declaração condicional fuzzy* e descreve a dependência do valor de uma variável linguística em relação ao valor de outra [15].

3.2 Lógica *Fuzzy*

Regras são expressas por intermédio de implicações lógicas, que representam uma relação entre antecedentes e consequentes. A função de pertinência associada a esta relação é definida pelo operador de implicação, que deve ser escolhido apropriadamente. O conceito de implicação está relacionado a um ramo da matemática conhecido como lógica proposicional, que é necessário para a compreensão do conceito de implicação na lógica *fuzzy* [15].

Lógica é o estudo dos métodos e princípios de raciocínio em todas as suas formas possíveis. A lógica clássica trata com proposições que podem ser verdadeiras ou falsas, e essas proposições devem ter valores de verdade opostos. A lógica proposicional lida com combinações de variáveis que representam proposições arbitrárias. Estas variáveis são chamadas de variáveis lógicas. Na medida que cada variável representa uma proposição hipotética, ela pode então assumir qualquer um dos dois valores de verdade, isto é, a variável não está comprometida com qualquer valor absoluto de verdade, a menos que uma nova proposição substitua seu valor. Uma das preocupações da lógica proposicional é o estudo de regras pelas quais novas variáveis lógicas podem ser produzidas, como funções de variáveis lógicas. A estrutura interna das proposições que as variáveis lógicas representam não são consideradas [21]. Por sua vez, a lógica *fuzzy* difere dos sistemas lógicos convencionais na medida em que visa prover modelos aproximados em vez de raciocínio e modelagens precisas. Não é necessário calcular os verdadeiros valores absolutos das proposições, visto que uma proposição *fuzzy* induz uma distribuição de possibilidades ao longo de um universo de discurso, gerando-se um conjunto *fuzzy* [24].

Dentro da lógica proposicional, as combinações efetuadas para a formação de novas proposições se baseiam em funções lógicas simples, chamadas de primitivas lógicas. As primitivas mais utilizadas são conjunção, disjunção, implicação, negação e equivalência. Através dessas primitivas pode-se avaliar se as implicações formadas pelas combinações das proposições não relacionadas são verdadeiras. As relações entre proposições são nor-

malmente mostradas por meio de tabelas verdade [15].

A diferença fundamental entre proposições clássicas e proposições *fuzzy* é a faixa de valores de suas verdades. Enquanto cada proposição clássica precisa de um valor absoluto, tal como verdadeiro ou falso, nas proposições *fuzzy* a verdade é uma questão de nível de verdade. Supondo que a verdade e falsidade são expressas pelos valores 1 e 0, respectivamente, o grau de verdade de cada proposição *fuzzy* é expressa por um número no intervalo unitário $[0,1]$ [21].

A lógica *fuzzy* pode ser dividida em duas direções [25]. Em seu sentido amplo, ela é utilizada como ferramenta para aplicações em controle e *soft-computing*, isto é, métodos computacionais tolerantes a incertezas que geram soluções simples, rápidas e suficientemente boas [26]-[29]. A outra direção da lógica *fuzzy*, em seu sentido estreito, é a lógica simbólica com noção comparativa da verdade, desenvolvido totalmente no espírito da lógica clássica (sintaxe, semântica, axiomatização, dedução, integridade) [30]. Esta lógica *fuzzy* serve como base para novos trabalhos utilizando a lógica *fuzzy* em um sentido amplo, além de seu interesse lógico independente [31]-[33].

3.3 Sistemas de Inferência *Fuzzy*

Inferência *fuzzy* é o processo de elaboração do mapeamento de uma dada entrada para uma saída utilizando a lógica *fuzzy*. O mapeamento fornece uma base a partir da qual as decisões podem ser feitas, ou padrões discernidos. O processo de inferência *fuzzy* envolve todas as definições que são descritas nas seções anteriores: funções de pertinência, operações lógicas e preposições, sendo formado por cinco partes: fuzzificação das variáveis de entrada, aplicação de operadores *fuzzy* no antecedente, implicação do antecedente ao consequente, agregação dos consequentes através das regras e defuzzificação.

Em um sistema de inferência *fuzzy*, conforme a Figura 3.2, consideram-se entradas precisas, resultantes de medições e observações. Portanto, é necessário mapear essas entradas para os conjuntos *fuzzy* de interesse. Esse estágio é chamado de fuzzificação, e nele também ocorre a ativação das regras relevantes para cada situação [15].

Uma vez obtido o conjunto *fuzzy* de saída por meio do processo de inferência, no estágio de defuzzificação é efetuada uma interpretação dessa informação, avaliando se a saída está precisa, ou seja, se o sinal está de acordo com a modelagem proposta. Dentre os métodos de defuzzificação encontrados na literatura, os mais empregados são o centro de gravidade e a média dos máximos [34]. Neste, a saída precisa é obtida tomando-se a média entre os dois elementos extremos no universo que correspondem aos maiores valores da função de pertinência do consequente. Com o centro de área, a saída é o valor no universo que divide a área sob a curva da função de pertinência em duas partes iguais [15].

As regras podem ser fornecidas por especialistas, em forma de sentenças linguísti-

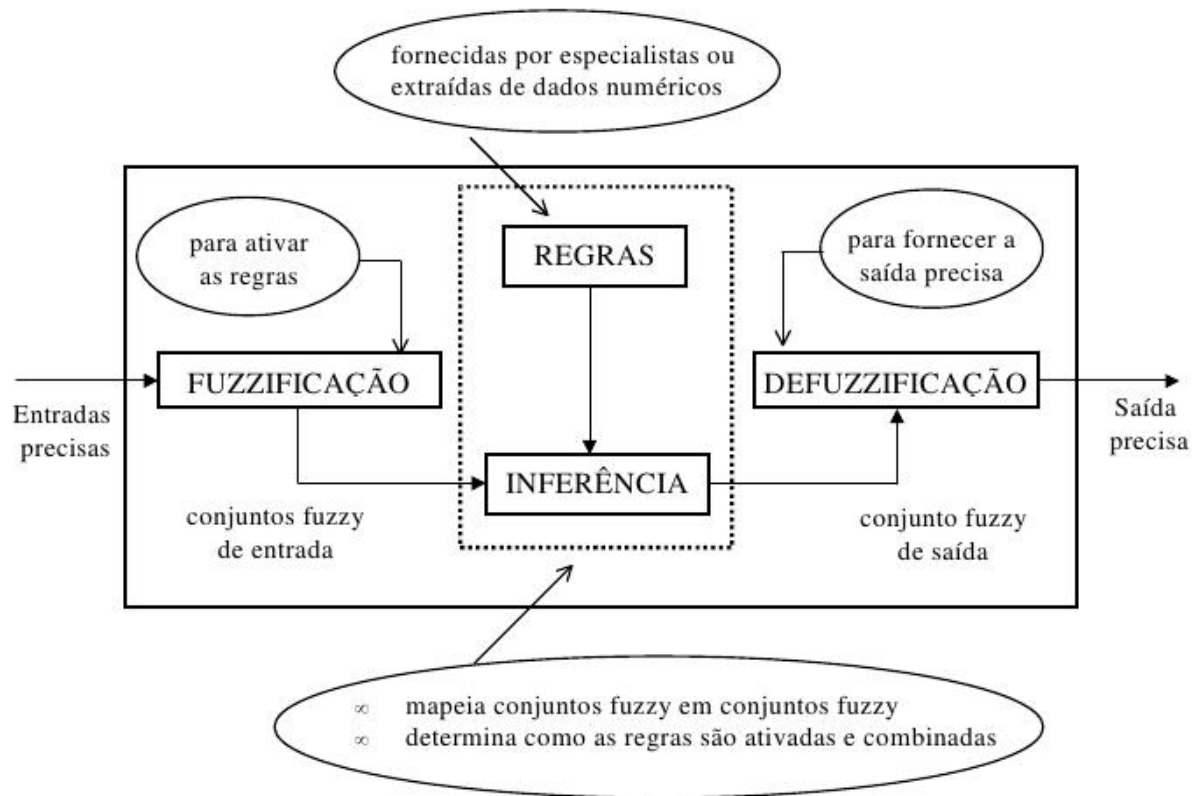


Figura 3.2: Sistema de Inferência *fuzzy* [15].

cas, e constituem um aspecto fundamental no desempenho de um sistema de inferência *fuzzy*. Extrair regras de especialistas por meio das proposições pode ser complexo, por mais conhecedores que eles sejam do problema em questão. Alternativamente ao uso de especialistas para a definição da base de regras, existem métodos de extração de regras a partir de dados numéricos. Estes métodos são particularmente úteis em problemas de classificação e previsão de séries temporais. No estágio de inferência ocorrem as operações com os conjuntos *fuzzy*. Esses conjuntos podem ser previamente definidos, ou utilizando métodos automáticos, tais como algoritmos genéticos [15], [35] e rede neurais [36].

Existem diversos tipos de inferência *fuzzy*. Os mais utilizados são Mamdani e Sugeno. A inferência fuzzy de Mamdani foi um dos primeiros sistemas de controle construídos usando a teoria dos conjuntos *fuzzy*. Ele foi proposto em 1975 por Mamdani [17] como uma tentativa de controlar uma combinação de motor a vapor e caldeira, sintetizando um conjunto de regras de controle linguísticas obtidas a partir de experientes operadores humanos. Após o processo de agregação, existe um conjunto fuzzy para cada variável de saída que necessita de defuzzificação. O método de Sugeno [37] é similar ao método de Mamdani em todas as etapas, exceto pelas funções de pertinência de saída, que ao invés de serem um conjunto *fuzzy* são funções lineares ou constantes.

4 *Abordagem Fuzzy para o Posicionamento dos Agentes*

Para este trabalho foi utilizada uma abordagem *fuzzy* para definir o posicionamento de robôs móveis autônomos. O método consiste em um sistema *fuzzy*, onde são avaliadas as funções de pertinência e, por meio da saída obtida, será atribuída para cada agente robótico um novo posicionamento. Esse método foi aplicado no domínio de futebol de robôs simulados, utilizando o time *Agent2D* [38] como base de comparação para avaliação dos resultados.

O propósito é encontrar um melhor posicionamento tático para cada agente robótico, de maneira que, para cada jogo ou adversário diferente, uma nova posição seja inferida para cada robô, otimizando-se as possíveis ações a serem tomadas. É importante ressaltar que o foco é otimizar o posicionamento do time como um todo, e não apenas de um jogador.

Neste capítulo, as propriedades e características do time desenvolvido estão na Seção 4.1. Na Seção 4.2 serão tratadas as características do sistema *fuzzy* e os resultados obtidos serão apresentados na Seção 4.3.

4.1 Características do Time

O time *Agent2D* [38] foi utilizado como base para aplicação e avaliação do sistema *fuzzy* implementado. Seu código fonte possui alguns comportamentos e ações básicas, além de implementações de baixo nível, como comunicação com o servidor e entre jogadores.

A equipe é composta por onze jogadores, e cada jogador tem uma função e posicionamento específicos. Esse posicionamento varia conforme a formação tática escolhida. Dentre as inúmeras formações utilizadas em jogos de futebol, destacam-se 4-4-2 (Figura 4.1), 4-3-3 (Figura 4.2) e 4-2-3-1 (Figura 4.3) [39].

A formação 4-4-2 foi a mais comum no futebol na década de 1990 e início de 2000. Os meio-campistas são obrigados a trabalhar muito a fim de apoiar tanto a defesa como o ataque: tipicamente um dos meio-campistas centrais está previsto para ir ao ataque o mais rápido possível, a fim de apoiar os jogadores de frente, enquanto o outro protege a defesa. Os dois meio-campistas mais abertos pelas laterais devem subir pelos flancos em

direção ao ataque para efetuar cruzamentos e ainda assim proteger os defensores laterais [40].



Figura 4.1: Formação 4-4-2 [39].

O jogador extra no meio-campo da formação 4-3-3 permite uma defesa mais forte, e os meio-campistas podem ser alocados para diferentes funções. Os três meio-campistas podem jogar em conjunto para proteger a defesa, se moverem lateralmente através do campo como uma unidade coordenada, e até apoiarem os atacantes. Esta formação é considerada uma formação ofensiva [41].

A formação 4-2-3-1 é amplamente utilizado por equipes espanholas e francesas. Embora pareça defensiva, é uma formação flexível, com ambos os laterais e meias defensivo juntando-se ao ataque. Ela é utilizada para manter a posse de bola e interromper ataques opostos, por meio do controle do meio de campo [39].

A abordagem proposta leva em consideração variáveis de jogo, tais como placar do jogo, posição da bola e porcentagens de ataques e defesas bem sucedidas a fim de inferir uma posição tática adequada à atual situação de jogo. Desta maneira, tem-se um sistema *fuzzy* de quatro entradas e uma saída, onde uma formação tática é escolhida, conforme será tratado na Seção 4.2.

4.2 Características do Sistema *Fuzzy* Implementado

O sistema de inferência *fuzzy* implementado possui quatro entradas e uma saída. As entradas são o tempo de jogo, posição da bola durante o jogo, porcentagem de ataques



Figura 4.2: Formação 4-3-3 [39].



Figura 4.3: Formação 4-2-3-1 [39].

bem sucedidos e porcentagem de defesas bem sucedidas. A saída é uma função de inferência *fuzzy*, já que o sistema de inferência utilizado foi o de Mamdani [17], que define a formação tática a ser escolhida. A saída é defuzzificada pelo método do centro de área. As funções de pertinência foram arbitrariamente definidas como triangulares, com a finalidade de facilitar a representação por variáveis linguísticas.

A variável de entrada "tempo de jogo" foi dividida em três termos linguísticos: começo, meio e fim. O tempo varia de 0 até 6000 ciclos, que é o tempo de duração de cada partida. Sua função de pertinência é dada pela Figura 4.4.

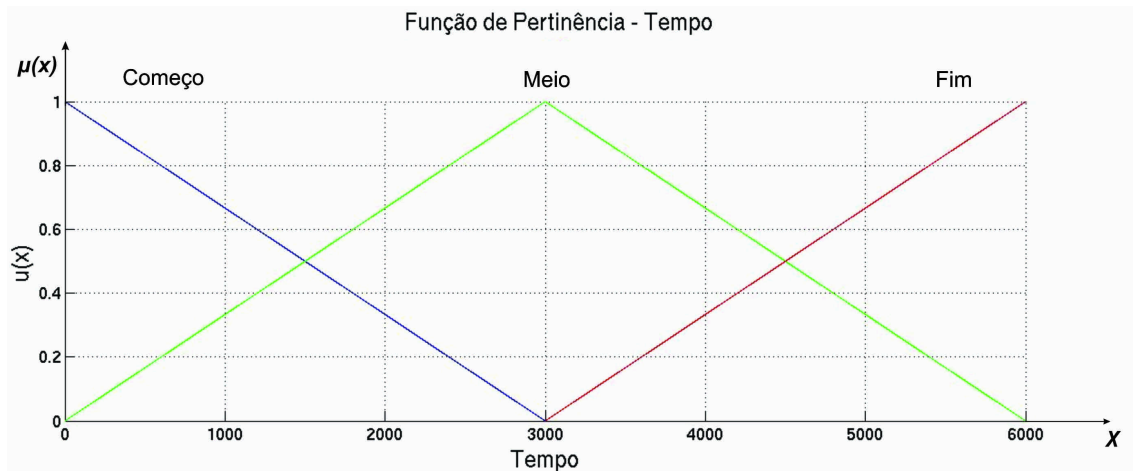


Figura 4.4: Função de Pertinência - Tempo.

A variável de entrada "posição de bola" foi também dividida em três termos linguísticos: perto, longe e muito longe. Ela varia de 0 até 100, representando o comprimento total do campo de futebol. Sua função de pertinência é dada pela Figura 4.5.

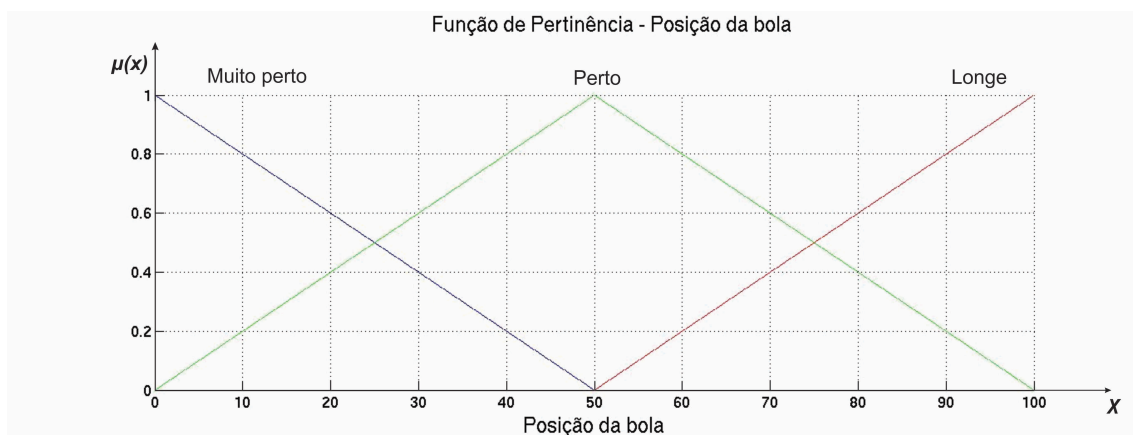


Figura 4.5: Função de Pertinência - Posição de bola.

A variável de entrada "porcentagem de ataques bem sucedidos" foi também dividida em três termos linguísticos: baixo, médio e alto. Essa porcentagem mostra o aproveitamento de ataque do time. Sua função de pertinência é dada pela Figura 4.6.

A variável de entrada "porcentagem de defesas bem sucedidos" foi também dividida

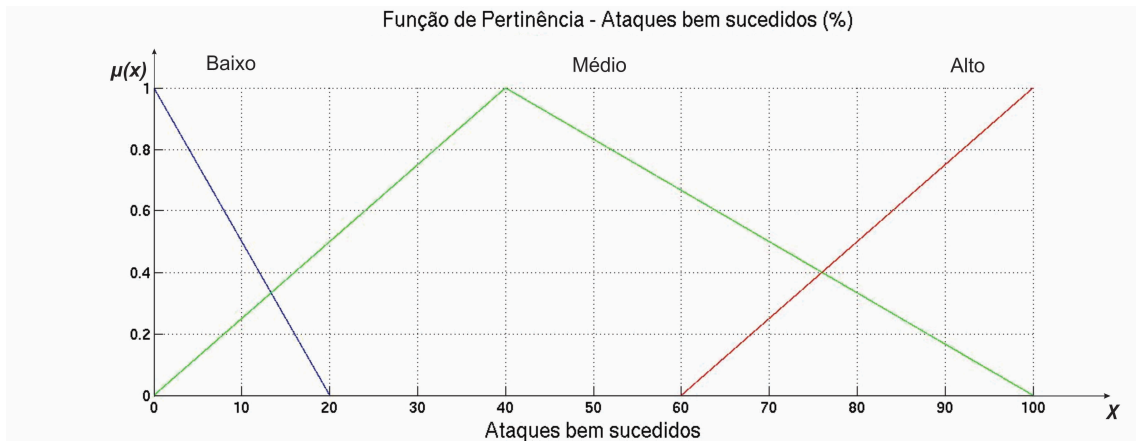


Figura 4.6: Função de Pertinência - Ataques bem Sucedidos (%).

em três termos linguísticos: baixo, médio e alto. Essa porcentagem mostra o aproveitamento de defesa do time. Sua função de pertinência é dada pela Figura 4.7.

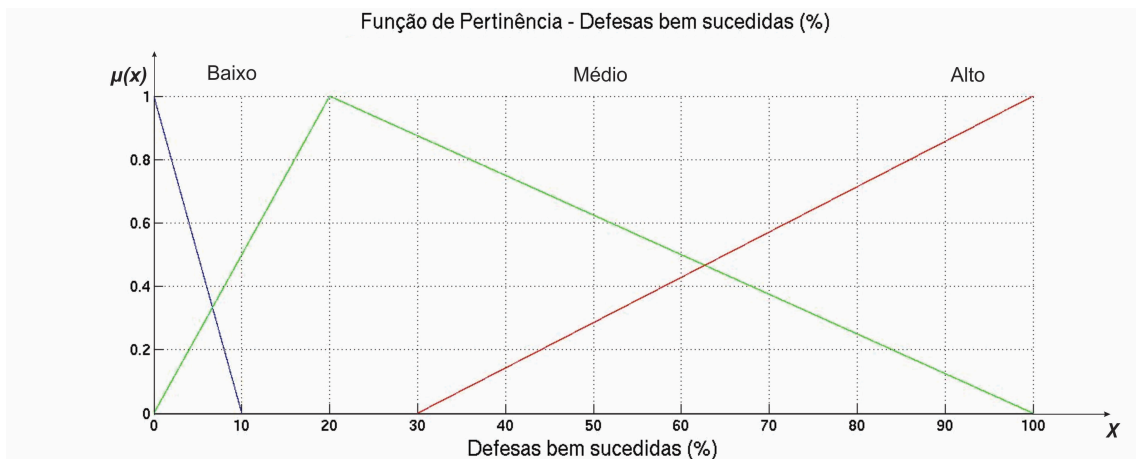


Figura 4.7: Função de Pertinência - Defesas bem Sucedidas (%).

A variável de saída "formação tática" foi dividida em três termos linguísticos: 4-4-2, 4-3-3 e 4-2-3-1. Elas indicam a formação tática escolhida pelo sistema *fuzzy*. Sua função de pertinência é dada pela Figura 4.8.

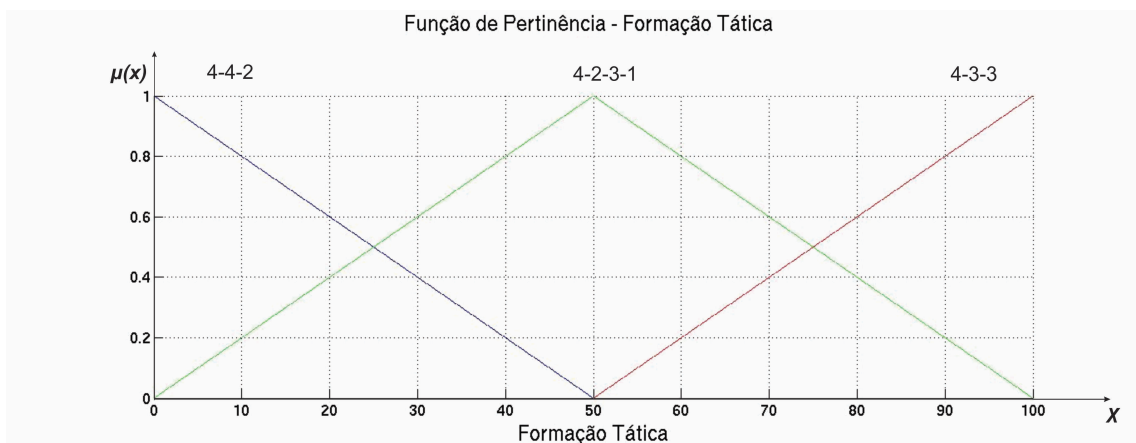


Figura 4.8: Função de Pertinência - Formação Tática.

O sistema desenvolvido funciona conforme a Figura 4.9. Ele calcula as funções de pertinência *fuzzy* a cada 250 ciclos e retorna uma formação tática. Como o jogo possui 6000 ciclos, a cada jogo é possível trocar de formação até 24 vezes. O responsável pela escolha da formação é o técnico. Quanto mais perto a bola está do goleiro, mais defensiva fica a formação. O tempo de jogo influencia na agressividade, ou seja, no começo do jogo as escolhas são bem prudentes e defensivas, mas conforme o decorrer do jogo, as escolhas tendem a ser mais agressivas, buscando pressionar o oponente. Se a equipe está com um bom aproveitamento no ataque, ou seja, se a porcentagem de ataques bem sucedidos é alta, o time torna-se mais ofensivo. Se essa porcentagem é baixa, o time assume uma postura defensiva. Uma quantidade alta de defesas bem sucedidas torna a equipe mais ofensiva, buscando aproximar a bola do goleiro adversário.

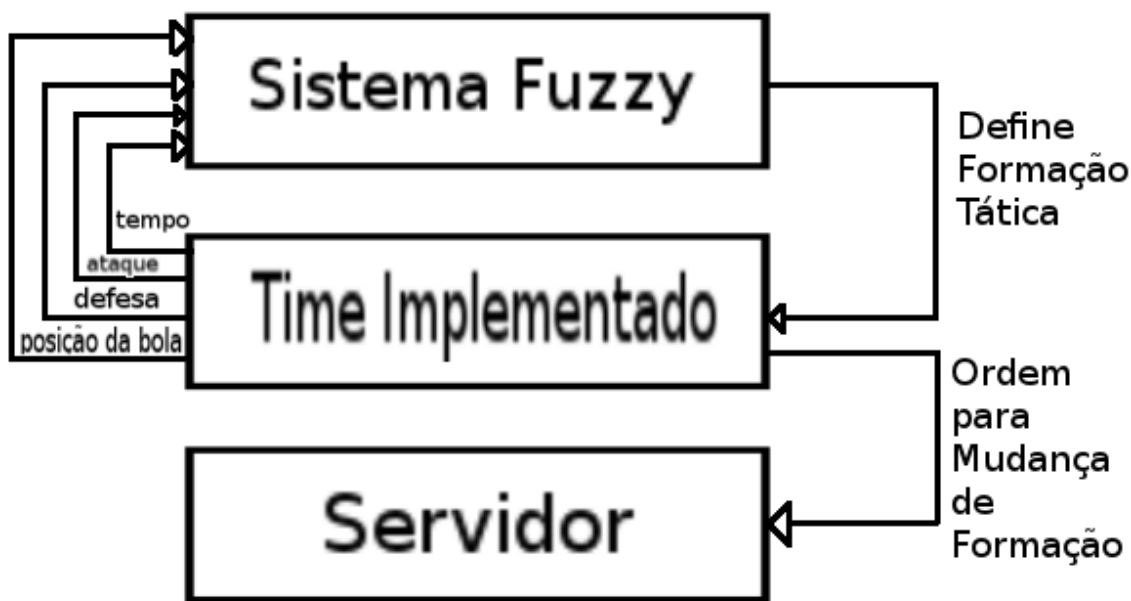


Figura 4.9: Fluxo de Dados no Sistema *Fuzzy*.

4.3 Resultados Obtidos

Para poder avaliar o desempenho do sistema *fuzzy* proposto, foram então escolhidos três times com bons desempenhos na última *RoboCup*. São eles os seguintes:

- *HELIOS* - Time desenvolvido por Hidehisa Akiyama and Hiroki Shimora, na Universidade de Tokyo, Japão. Utilizam técnicas de posicionamento baseado em triangulação *Delaunay* e redes neurais. Foram campeões da *RoboCup* em 2010 e vice-campeões em 2011 e 2009 [43].
- *NADCO* - Time desenvolvido por Mohammad Marasht, da Universidades do Teerã, Irã. O time possui técnicas de análises de estratégia e criação de novos tipos de jogador. Terminou a *RoboCup* entre os cinco primeiros nos últimos anos [44].

- *MARLIK* - Time desenvolvido por Amar Tavafi *et al.*, na Universidade de Guilan, Irã. O foco do time é sistemas de previsão para interceptação de passes e marcação. Foi o terceiro colocado da *RoboCup* 2011 [45].

Para a validação dos resultados, foram aqui realizados trinta jogos entre cada um dos times selecionados e o *Agent2D*, trinta jogos entre cada uma das equipes selecionadas e o time desenvolvido, e um confronto direto de trinta jogos entre o time base *Agent2D* e o time proposto. Os dados analisados são a média dos resultados dos jogos, o aproveitamento em cada jogo e a quantidade de gols prós e contras durante cada partida. Serão apresentadas figuras com os resultados dos jogos efetuados e, posteriormente, tabelas serão apresentadas destacando os dados relevantes.

Na Figura 4.10, observam-se os resultados entre *Agent2D* e Nadco. Os resultados são equilibrados, com doze vitórias para os dois times. A maior diferença de gols foi quatro.

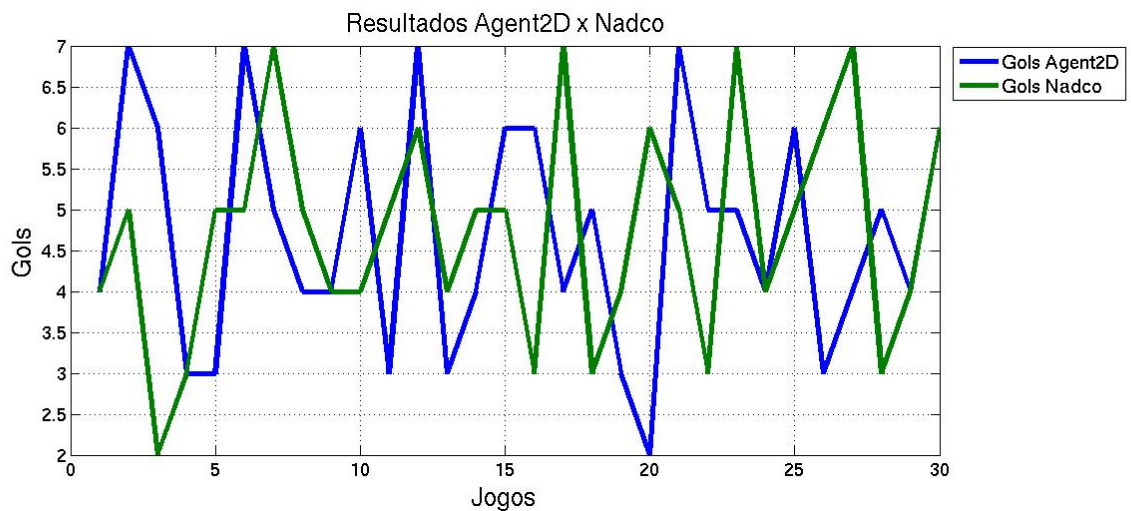


Figura 4.10: Resultados dos jogos - *Agent2D* x Nadco

Os resultados entre *Agent2D* e MarliK estão representados na Figura 4.11. O time *Agent2D* só conseguiu marcar quatro gols em trinta jogos, perdendo todos.

As partidas disputadas entre *Agent2D* e Helios foram contabilizadas na Figura 4.12. Nessas partidas ocorreram os maiores números de gols marcados, evidenciando a grande diferença entre os dois times. O *Agent2D* marcou cinco gols no total.

Como primeiro dado de comparação, tem-se na Figura 4.13 os resultados das partidas entre o time desenvolvido e Nadco. Os resultados apresentam placares menos elásticos, comparativamente com o *Agent2D*. O time desenvolvido obteve doze vitórias, contra 6 vitórias do Nadco, mostrando uma melhora no desempenho do time.

Na Figura 4.14, observam-se os resultados entre o time desenvolvido e MarliK. Este último sofreu doze gols, ou seja, o triplo do que tinha sofrido contra o time base. Além disso, o time desenvolvido ganhou quatro jogos e empatou mais quatro. Percebe-se uma grande evolução nos resultados.

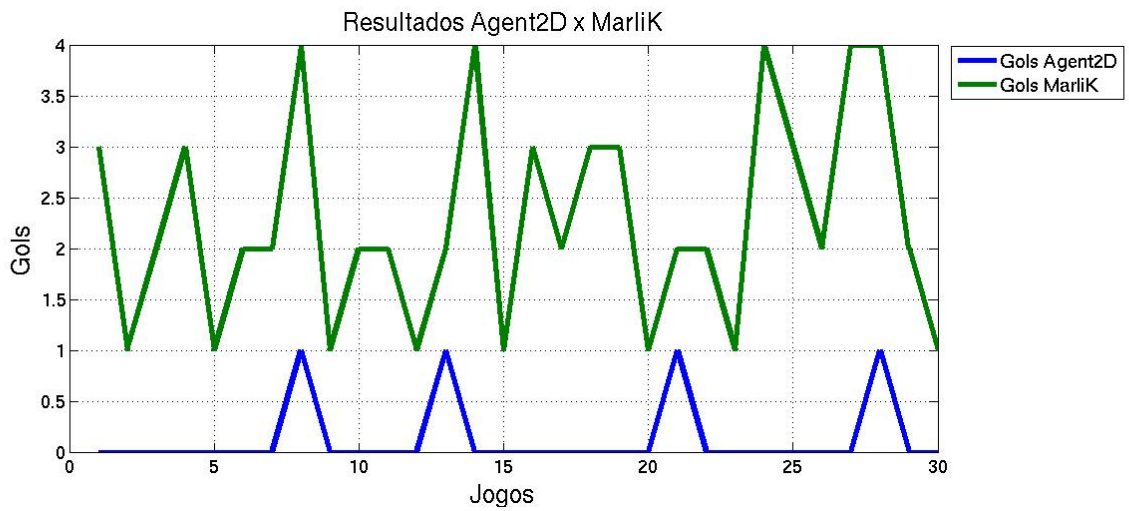


Figura 4.11: Resultados dos jogos - *Agent2D* x MarliK

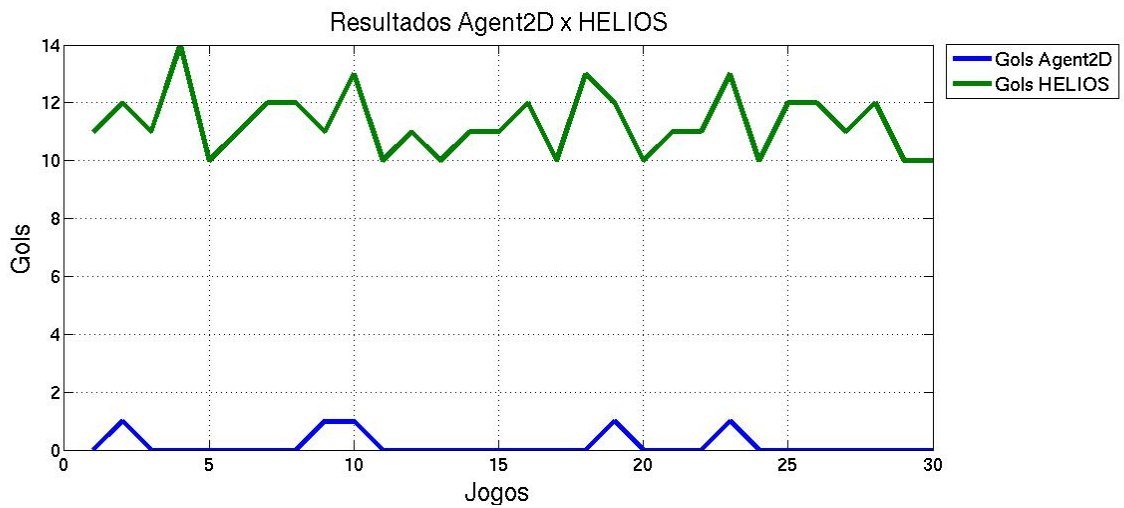


Figura 4.12: Resultados dos jogos - *Agent2D* x Helios

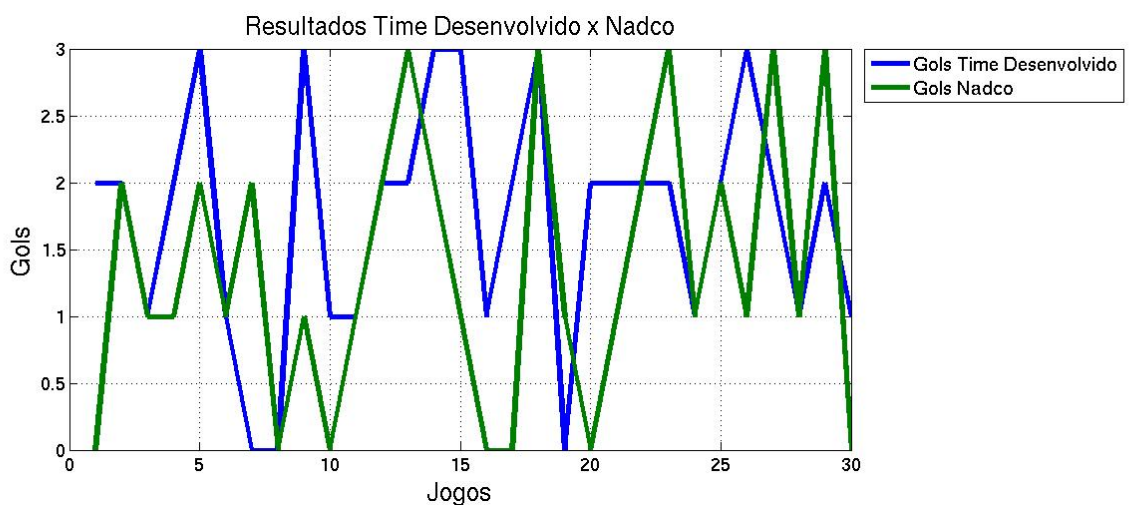


Figura 4.13: Resultados dos jogos - Time Desenvolvido x Nadco

Na comparação dos resultados entre o time proposto e Helios (Figura 4.15), nota-se uma diminuição nos gols sofridos, comparados com o time base. De uma média inicial de

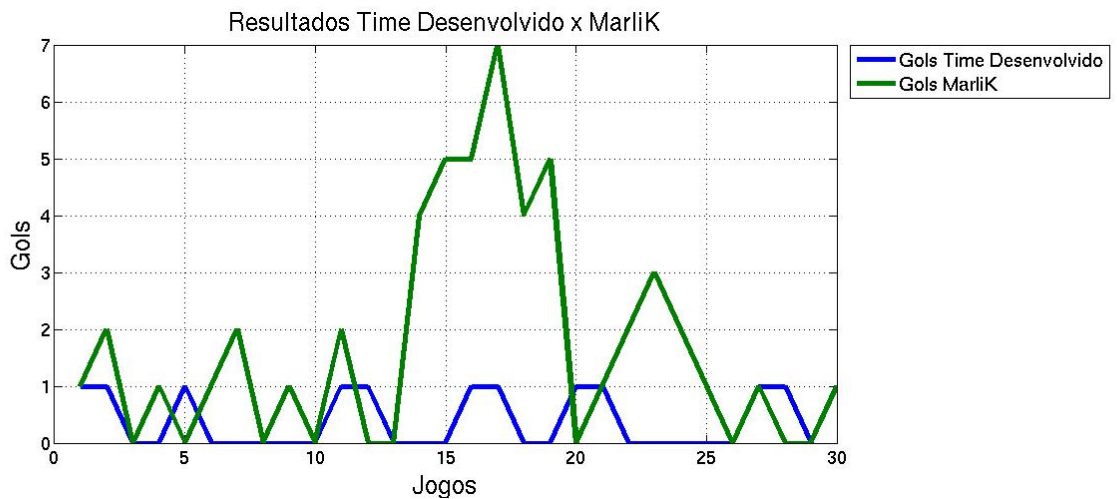


Figura 4.14: Resultados dos jogos - Time Desenvolvido x MarliK

gols sofridos de pouco mais de onze gols por jogo, a média diminuiu para pouco menos de seis gols por jogo. O número de gols marcados aumentou de cinco para sete.

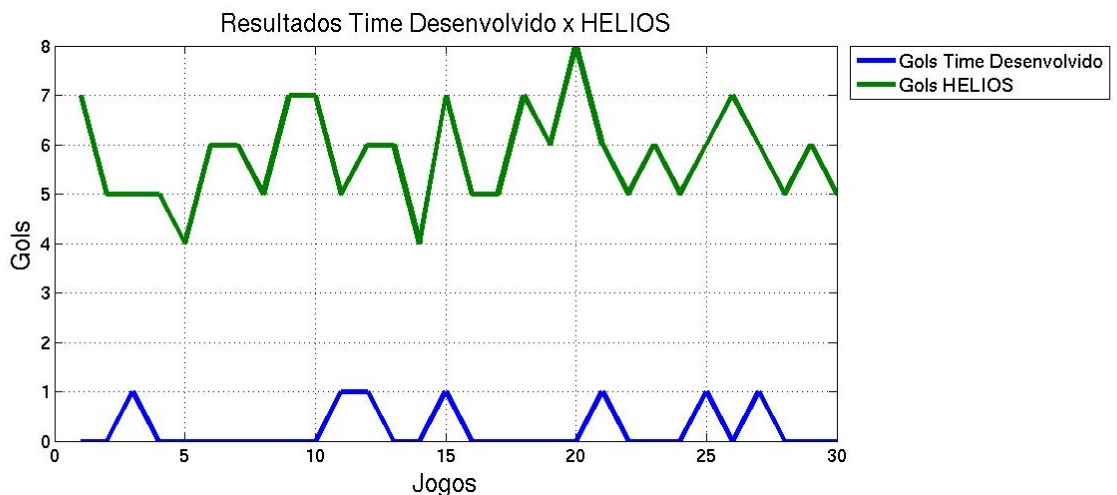


Figura 4.15: Resultados dos jogos - Time Desenvolvido x Helios

Na Figura 4.16, observa-se os resultados da comparação direta entre *Agent2D* e o time desenvolvido. Pode-se notar um resultado amplamente favorável ao time desenvolvido, vencendo todos os jogos e sofrendo apenas seis gols.

A Tabela 4.1 exibe um placar médio dos jogos. Observa-se que o número de gols sofridos diminuiu consideravelmente. Apesar do número de gols marcados ter diminuído, o resultado do jogo tornou-se favorável ao time desenvolvido, exibindo uma melhora nos resultados. Uma análise do desempenho dos times é apresentada na Tabela 4.2. Os valores representam o número de Vitórias-Empates-Derrotas. Pode-se notar que o time desenvolvido com o sistema *fuzzy* consegue melhores resultados, que são expressos em termos percentuais na Tabela 4.3. A melhoria nos resultados é notável, conseguindo até reverter uma situação que era apenas de derrotas em algumas vitórias e empates. As Tabelas 4.4 e 4.5 exibem a quantidade de gols marcados e sofridos, respectivamente. O número de

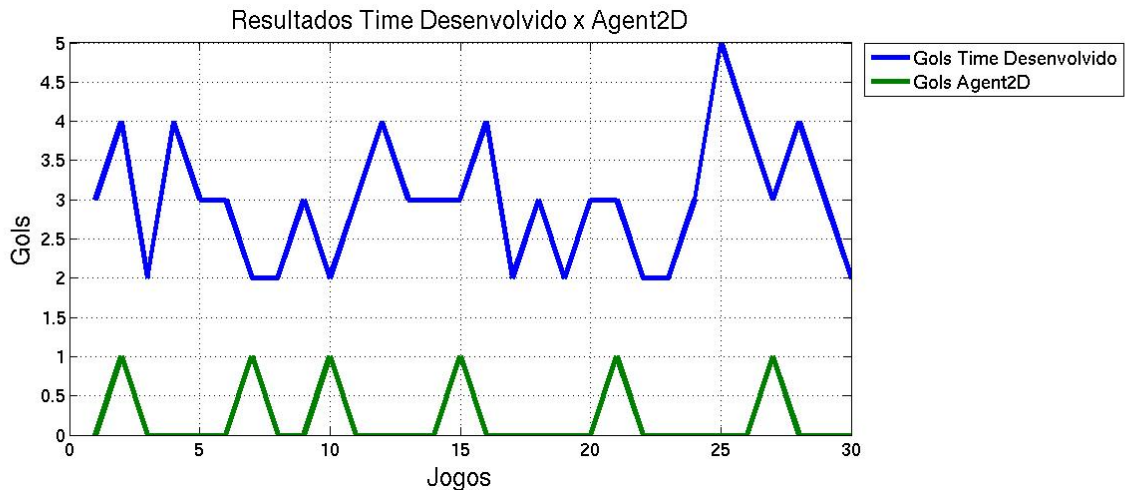


Figura 4.16: Resultados dos jogos - Time Desenvolvido x *Agent2D*

gols sofridos diminuiu consideravelmente, mostrando um melhor posicionamento do setor defensivo. O número de gols pró também diminuiu, porém os resultados foram melhores, indicando uma postura mais defensiva em determinadas situações de jogo.

	Nadco	Marlik	Helios
<i>Agent2D</i>	5x5	0x2	0x11
Time Desenvolvido	2x1	0x1	0x6

Tabela 4.1: Média de Gols

	Nadco	Marlik	Helios
<i>Agent2D</i>	12-6-12	0-0-30	0-0-30
Time Desenvolvido	12-12-6	4-4-22	0-0-30

Tabela 4.2: Desempenho (Vitórias-Empates-Derrotas)

	Nadco	Marlik	Helios
<i>Agent2D</i>	46.66	0	0
Time Desenvolvido	53,33	22.22	0

Tabela 4.3: Aproveitamento (%)

	Nadco	Marlik	Helios
<i>Agent2D</i>	141	4	5
Time Desenvolvido	52	12	7

Tabela 4.4: Gols Prós

	Nadco	Marlik	Helios
<i>Agent2D</i>	142	68	339
Time Desenvolvido	40	51	173

Tabela 4.5: Gols Contra

5 Conclusões

Sistemas de inferência *fuzzy* podem ser aplicados em diversos problemas relacionados com engenharia e ciências exatas. Tais sistemas permitem o mapeamento de ambientes complexos que não podem ser expressos apenas em termos de equações. A utilização de variáveis linguísticas, que são características de sistemas *fuzzy*, em conjunto com o conhecimento especialista, permitiu uma modelagem robusta e ainda de fácil compreensão, garantindo uma extração de dados objetiva e uma resposta precisa.

Ao final deste Trabalho de Conclusão de Curso foi desenvolvido um sistema de inferência *fuzzy* aplicado ao posicionamento de agentes robóticos no domínio de futebol de robôs. O sistema de inferência utilizado foi o de Mamdani [17], onde a saída é uma função de pertinência, expressa em variáveis linguísticas. Durante a modelagem e fase de testes do sistema, o *software* MATLAB foi utilizado e, posteriormente, o sistema foi implementado em linguagem de programação C++, utilizando um ambiente GNU/Linux.

Um time base foi utilizado como plataforma para a realização dos testes, permitindo uma comparação justa e precisa. Tais testes foram realizados, mostrando uma melhora significativa no desempenho do time desenvolvido, em relação ao time base *Agent2D*, validando a abordagem *fuzzy* sugerida. Assim, este trabalho fornece uma contribuição para pesquisas em inteligência artificial no domínio de futebol de robôs simulados, que é um ambiente dinâmico e complexo, com várias áreas e temas de pesquisa ainda a serem investigados.

Além dos testes realizados, esse sistema também foi utilizado em competições nacionais e internacionais de robótica. Durante a Competição Brasileira de Robótica realizada na cidade de São João del Rey em 2011, o time obteve a segunda colocação. Em competições internacionais, o time desenvolvido sagrou-se campeão da Latin American Robotics Competition, realizada em São Bernardo do Campo no ano de 2010 e obteve a décima oitava posição na *RoboCup* 2012, realizada na Cidade do México.

Como direções futuras para este trabalho pode ser ressaltado a utilização de diferentes entradas e formas das funções de pertinência. Uma outra possibilidade é a implementação de um sistema híbrido, tal como *ANFIS*, em que redes neurais são treinadas para otimizar as funções de pertinência, assim como *Genetic Fuzzy Systems*, onde o sistema *fuzzy* define qual parte da base de conhecimentos vai ser otimizada pelos algoritmos genéticos.

Referências Bibliográficas

- [1] H. Kitano *et al.*, *The Robot World Cup Initiative*, *IJCAI-95 Workshop on Entertainment and AI/Alife*, 1995.
- [2] K. Shi, X. Chen, *Action-Driven Markov Decision Process and the Application in RoboCup*, 2011.
- [3] J. Zhou *et al.*, *AUA 2D Soccer Simulation Team Description Paper*, 2011.
- [4] E. Fraccaroli, *Análise de Desempenho de Algoritmos Evolutivos no Domínio de Futebol de Robôs*, 2010.
- [5] X. Lin *et al.*, *LsuAmoyNQ 2009 RoboCup Soccer Simulation 2D Team Description*, 2009.
- [6] RoboCup, *The Official RoboCup Site*, <http://www.robocup.org/robocup-soccer/>, Acessado em Março/2012.
- [7] FIFA - Fédération Internationale de Football Association, *The FIFA Site*, <http://www.fifa.com>, Acessado em Março/2012.
- [8] Aldebaran Robotics, *The Aldebaran Robotics Site*, <http://www.aldebaran-robotics.com/en/>, 2012.
- [9] SONY, *The SONY Site*, <http://www.sony.com/>, Acessado em Março/2012.
- [10] M. Chen *et al.* *The RoboCup Soccer Server*, 2003.
- [11] L.P. REIS, *Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico*, 2003.
- [12] J. Kurose, K. Ross, *Computer Networking: A Top-Down Approach*, *Pearson Education*, 2010.

- [13] *RoboCup SoccerMonitor*, <http://code.google.com/p/sebbot/>, Acessado em Março/2012.
- [14] L.A. Zadeh, *Fuzzy Sets*, 1965.
- [15] R. Tanscheit, *Sistemas Fuzzy*, 2009.
- [16] D. Dubois, H. Prade, *Fuzzy Sets and Systems - Theory and Applications*, 1980.
- [17] E.H. Mamdani, S. Assilian, *An experiment in linguistic synthesis with a fuzzy logic controller*, 1975.
- [18] J. Jantzen, *Tuning Of Fuzzy PID Controllers*, 1998.
- [19] A. Palmer *et al.*, *2011 Team Description Paper: UBC Thunderbots*, 2011.
- [20] A. Saffiotti, Z. Wasik, *Using Hierarchical Fuzzy Behaviors in the RoboCup Domain*, 2003.
- [21] G. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, 1995.
- [22] C. Von Altrock, *Fuzzy Logic and neuroFuzzy applications in business and finance*, 1996.
- [23] W. Pedrycz, F. Gomide, *Fuzzy Systems Engineering: Toward Human-Centric Computing*, 2007.
- [24] R. Bellman, L.A. Zadeh, *Modern Uses of Multiple-Valued Logic*, 1977.
- [25] L.A. Zadeh, *Fuzzy Logic Technology and Applications*, 1994.
- [26] V. Novak, *Fuzzy Sets and their Applications*, 1989.
- [27] H.J. Zimmermann, *Fuzzy Set Theory and its Applications*, 1991.
- [28] G. Klir, B. Yuan, *Fuzzy Sets, Fuzzy Logic and Fuzzy System: Selected Papers by Lofti A. Zadeh*, 1996.

- [29] H.T. Nguyen, E. Walker *First Course in Fuzzy Logic*, 1999.
- [30] P. Hajek, *Fuzzy Logic*, 2010.
- [31] P. Hajek, *Metamathematics of Fuzzy Logic*, 1998.
- [32] E. Turunem, *Mathematics Behind Fuzzy Logic*, 1999.
- [33] V. Novak, *Mathematical Principles of Fuzzy Logic*, 2000.
- [34] W. Van Leekwijck, E. Kerre *Defuzzification: Criteria and Classification*, 1997.
- [35] O. Cordon *et al.*, *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*, 2001.
- [36] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, 1992.
- [37] M. Sugeno, *Industrial Applications of Fuzzy Control*, 1985.
- [38] H. Akiyama, *Agent2D Simulation League Team*, 2012.
- [39] Soccer Formations, [http://en.wikipedia.org/wiki/Formation \(Association Football\)](http://en.wikipedia.org/wiki/Formation_(Association_Football)), Acessado em Março/2012.
- [40] Formations: 4-4-2, *BBC News*, 2005.
- [41] Formations: 4-3-3, *BBC News*, 2005.
- [42] D. Sewell, A. Edmonson *Relationships between field position and pre-match competitive state anxiety in soccer and field hockey.*, 1996.
- [43] H. Akiyama *et al.*, *2011 Team Description Paper: Helios*, 2011.
- [44] M. Marasht *et al.*, *NADCO-2D Soccer 2D Simulation Team Description Paper*, 2011.
- [45] A. Tafavi *et al.*, *MarliK 2011 Soccer 2D Simulation Team Description Paper*, 2011.