



Uma abordagem Swarm-Fuzzy para a coordenação de VANTs

Trabalho de Conclusão de Curso
Engenharia de Computação

Aluno: Luiz Felipe Freitas de Oliveira

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho



Luiz Felipe Freitas de Oliveira

Uma abordagem Swarm-Fuzzy para a coordenação de VANTs

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco - Universidade de Pernambuco

Graduação em Engenharia de Computação

Escola Politécnica de Pernambuco

Universidade de Pernambuco

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho

Recife

Novembro de 2014

De acordo. Recife, 28 de novembro de 2014:

Prof. Dr. Carmelo José Albanez
Bastos Filho
Orientador

Recife
Novembro de 2014

Agradecimentos

Agradeço ao professor e orientador Carmelo Bastos-Filho, pelo acompanhamento e pela oportunidade de pesquisa em áreas interessantes e motivadoras.

Agradeço ao colega de pesquisa Diego Pinheiro, por sempre compartilhar sua experiência e pelo apoio constante.

Agradeço ao colega de pesquisa Fernando Baptistella, pela ajuda e empenho no desenvolvimento deste trabalho.

Resumo

Este trabalho de conclusão de curso apresenta um mecanismo híbrido que utiliza inteligência de enxames e lógica *fuzzy* para a coordenação de múltiplos veículos aéreos não-tripulados (VANTs). O objetivo deste trabalho é diminuir a influência estática de uma máquina de estados finitos, anteriormente utilizada como principal mecanismo de coordenação do VANT, e substituí-la por um mecanismo que seja capaz de melhor representar o comportamento dinâmico de sistemas reais. Para isso, foram implementados controladores *fuzzy* inspirados no controlador de Takagi-Sugeno responsáveis por definir os níveis de ativação dos componentes de coordenação do VANT, antes definidos pela máquina de estados. Para analisar o desempenho do modelo de coordenação proposto, foram utilizadas métricas de capacidade de rastreamento, anti-colisão e coesão do enxame de VANTs. Os resultados mostraram que o modelo proposto provocou uma diminuição considerável de colisões e um aumento na taxa de coesão, o que evidencia uma maior estabilidade na coordenação do enxame.

Abstract

This monography proposes a hybrid mechanism based upon fuzzy logic and swarm intelligence for the coordination of multiple unmanned aerial vehicles (UAVs). The goal of this project is to decrease the static influence of a finite state machine, used previously as the main coordination mechanism of the UAV, and replace it by a mechanism that makes it possible to better represent the dynamic behaviour of a real system. In order to achieve this, Takagi-Sugeno inspired fuzzy controllers were implemented to set the activation levels of the coordination components of the UAV. In order to analyse the performance of the proposed coordination model, tracking capability, anti-collision and cohesion metrics were used. The results indicate that the proposed model caused a considerable reduction of collisions and an increase in the cohesion rate, which denotes a better coordination consistency of the swarm.

Sumário

1	INTRODUÇÃO	1
1.1	Introdução e caracterização do problema	1
1.2	Objetivos	2
1.3	Estrutura do trabalho	2
2	FUNDAMENTAÇÃO TEÓRICA	4
2.1	Inteligência de enxames	4
2.2	Robótica de enxames e Veículos Aéreos não-Tripulados	7
2.3	Lógica <i>fuzzy</i>	10
2.3.1	Conjuntos <i>fuzzy</i>	10
2.3.2	Funções de pertinência	11
2.3.3	Operadores <i>fuzzy</i>	13
2.3.4	Lógica <i>fuzzy</i>	14
2.3.5	Controladores <i>fuzzy</i>	15
2.3.5.1	Componentes de um controlador <i>fuzzy</i>	16
2.3.5.2	Controlador <i>fuzzy</i> de Mamdani	16
2.3.5.3	Controlador <i>fuzzy</i> de Takagi-Sugeno	17
2.4	Modelo de coordenação utilizando inteligência de enxames	18
2.4.1	VANTs	18
2.4.2	Mecanismo de localização	18
2.4.3	Mecanismo de locomoção	19
2.4.4	Mecanismo de comunicação	19
2.4.5	Mecanismo de sincronismo	20
2.4.6	Mecanismo de percepção	21
2.4.7	Mecanismo de patrulha e de retorno à base	21
2.4.8	Mecanismo de coordenação	22
3	CONTRIBUIÇÃO	23
3.1	Modelo de coordenação híbrido Swarm-Fuzzy	23
3.2	Ferramenta de simulação e métricas de avaliação	29
3.2.1	Ferramenta de simulação de enxames de VANTs	29
3.2.2	Métricas de avaliação de desempenho do enxame	30
3.2.2.1	Taxa de ratreamento de alvos móveis	30
3.2.2.2	Taxa de colisões entre VANTs	31
3.2.2.3	Taxa de coesão do enxame	31
3.2.2.4	Área de cobertura territorial	31

4	RESULTADOS	33
4.1	Análise de desempenho do modelo híbrido Swarm-Fuzzy	33
4.2	Análise da influência de erros de posicionamento	34
5	CONCLUSÕES E TRABALHOS FUTUROS	50
	Referências	52

Lista de ilustrações

Figura 1 – Função de pertinência para conjuntos clássicos	12
Figura 2 – Função de pertinência do conjunto <i>fuzzy</i> alto	12
Figura 3 – Exemplos de funções de pertinência para conjuntos <i>fuzzy</i>	14
Figura 4 – Função de pertinência da variável D_{Com}	25
Figura 5 – Função de pertinência da variável D_{Col}	26
Figura 6 – Função de pertinência da variável D_{Rast}	27
Figura 7 – Interface gráfica da ferramenta de simulação para enxames de VANTs.	30
Figura 8 – Taxas de colisão (CL) ambientes com 10, 20 e 40 VANTs	35
Figura 9 – Taxas de rastreamento (TT) para 10, 20 e 40 VANTs	36
Figura 10 – Taxas de coesão (COE) para 10, 20 e 40 VANTs	37
Figura 11 – Taxas de colisão (CL) para 10 VANTs com máquina de estados	38
Figura 12 – Taxas de colisão (CL) para 10 VANTs com lógica <i>fuzzy</i>	39
Figura 13 – Taxas de colisão (CL) para 20 VANTs com máquina de estados	40
Figura 14 – Taxas de colisão (CL) para 20 VANTs com lógica <i>fuzzy</i>	41
Figura 15 – Taxas de rastreamento (TT) para 10 VANTs com máquina de estados	42
Figura 16 – Taxas de rastreamento (TT) para 10 VANTs com lógica <i>fuzzy</i>	43
Figura 17 – Taxas de rastreamento (TT) para 20 VANTs com máquina de estados	44
Figura 18 – Taxas de rastreamento (TT) para 20 VANTs com lógica <i>fuzzy</i>	45
Figura 19 – Taxas de coesão (COE) para 10 VANTs com máquina de estados	46
Figura 20 – Taxas de coesão (COE) para 10 VANTs com lógica <i>fuzzy</i>	47
Figura 21 – Taxas de coesão (COE) para 20 VANTs com máquina de estados	48
Figura 22 – Taxas de coesão (COE) para 20 VANTs com lógica <i>fuzzy</i>	49

Lista de tabelas

Tabela 1 – Tabela de estados do VANT	22
Tabela 2 – Tabela de parâmetros de simulação	34

Lista de abreviaturas e siglas

CL	Taxa de colisão
COE	Taxa de coesão
PSO	<i>Particle Swarm Optimization</i>
TT	Taxa de rastreamento
VANT	Veículo Aéreo não-Tripulado

1 Introdução

1.1 Introdução e caracterização do problema

O interesse e a utilização de Veículos Aéreos Não-Tripulados (VANTs) têm crescido significativamente tanto em aplicações comerciais quanto em aplicações militares, principalmente em missões de inteligência, vigilância e reconhecimento estratégico [1]. Outras áreas de utilização têm sido propostas, como análise de risco, monitoramento de desastres e mapeamento agrícola. Tal crescimento no interesse se deve ao fato de que estes veículos podem ser utilizados na execução de tarefas complexas principalmente em situações que coloquem a vida humana em perigo. Além disso, existe um interesse em tornar estes veículos amplamente autônomos, reduzindo até mesmo a necessidade de operação remota humana [2][3].

Entretanto, existem situações em que a utilização de apenas um VANT não é suficiente. Aplicações de busca em larga escala necessitam de atributos que apenas podem ser alcançados utilizando-se um enxame de VANTs, como escalabilidade, robustez e distributividade [4]. Um enxame de VANTs também executa tarefas de recolhimento de informação sobre o ambiente de maneira mais eficiente, mostrando-se como uma plataforma indispensável neste tipo de atividade. O controle supervisionado destes veículos é um processo complexo e visualmente cansativo, afetado frequentemente pelo fraco desempenho de operadores humanos e que podem tomar decisões inapropriadas para determinadas situações [5].

Um fator importante para maximizar a atuação de um enxame de VANTs no ambiente é a presença de um sistema de coordenação que garanta a autonomia desses veículos [6]. Uma das abordagens utilizadas é a aplicação de conceitos da Inteligência de Enxames em múltiplos robôs, introduzindo o conceito de enxames de robôs. Estes são caracterizados por um controle descentralizado, comunicação limitada entre os robôs, uso de informação local e emergência de comportamento global. Tais sistemas têm mostrado seu potencial para flexibilidade e robustez [7]. No entanto, os modelos de enxames de robôs existentes carecem de um sistema de coordenação que assegure heterogeneidade na tomada de decisões, necessária para a eficácia da atuação de um enxame de VANTs que pretende abordar a complexidade e não-linearidade requerida por sistemas do mundo real.

Lógica *fuzzy* é um sistema lógico, extensão da lógica multivalorada, que tem o propósito de servir como uma lógica de raciocínio aproximado. Lógica *fuzzy* é utilizada em sistemas que necessitam de maior generalização, de maior poder de expressão, e de uma capacidade de modelar um fenômeno do mundo real [8]. Um dos modelos deste

tipo de lógica é a abordagem proposta por Takagi-Sugeno, capaz de representar qualquer sistema não-linear dinâmico [9] que exhibe um comportamento complexo.

Um mecanismo de coordenação baseado em Inteligência de Enxames utilizando o paradigma de Otimização por Enxames de Partículas foi proposto em [10], e comprovou a escalabilidade, segregação, agregação, capacidade de rastreamento de alvos no ambiente e capacidade de desvio de obstáculos do modelo. No entanto, embora capaz de desviar de obstáculos, os VANTs ainda apresentam uma considerável taxa de colisão entre si, o que pode afetar a eficiência do modelo para simulações com um número menor de agentes.

Este trabalho apresenta um modelo de coordenação distribuída para VANTs utilizando Inteligência de Enxames em conjunto com um controlador *fuzzy* proposto por Takagi-Sugeno, o qual atua no comportamento do VANT considerando aspectos de comunicação, colisão, afastamento e rastreamento de alvos no ambiente visando melhorar os aspectos de coordenação dos veículos.

1.2 Objetivos

O objetivo deste trabalho é propor um modelo de coordenação de VANTs utilizando Inteligência de enxames em conjunto com um controlador *fuzzy* baseado no modelo proposto por Takagi e Sugeno.

Os objetivos específicos deste trabalho são:

- a) propor um mecanismo híbrido de coordenação baseado em Inteligência de Enxames e Lógica Fuzzy;
- b) fazer uso de métricas para avaliar o modelo de coordenação proposto, como taxa de colisão, taxa de rastreamento e taxa de coesão;
- c) propor um mecanismo de posicionamento no ambiente considerando erros de localização;
- d) analisar o impacto dos erros de localização no modelo de coordenação;
- e) comparar o modelo de coordenação híbrido com o modelo de coordenação baseado somente em Inteligência de Enxames proposto em [10].

1.3 Estrutura do trabalho

Este trabalho está organizado da seguinte maneira: o Capítulo 2 apresenta os fundamentos de Inteligência de Enxames e Lógica *Fuzzy*; o Capítulo 3 apresenta o modelo híbrido de coordenação proposto, a ferramenta de simulação utilizada e as métricas para avaliar o modelo; o Capítulo 4 apresenta os resultados obtidos da análise do modelo de

coordenação proposto e do impacto dos erros de localização no modelo; e finalmente, o Capítulo 5 apresenta as conclusões deste trabalho e propostas para trabalhos futuros.

2 Fundamentação teórica

Neste capítulo são apresentados os fundamentos para compreensão deste trabalho. A Seção 2.1 apresenta os fundamentos de inteligência de enxames. A Seção 2.2 apresenta os conceitos sobre enxames de robôs e veículos aéreos não-tripulados. A Seção 2.3 apresenta os conceitos de lógica *fuzzy* e a Seção 2.4 apresenta o modelo de coordenação utilizando Inteligência de Enxames proposto em [10], o qual é usado como base para a proposta deste trabalho.

2.1 Inteligência de enxames

Formalmente, um enxame pode ser definido como um grupo de agentes que podem comunicar-se uns com os outros (direta ou indiretamente), agindo no ambiente local [11]. As interações entre os agentes resultam em estratégias coletivas e distribuídas com a finalidade de resolver problemas complexos, considerados impossíveis ou muito difíceis de serem resolvidos por apenas um indivíduo. No campo da Engenharia, a abordagem utilizando enxames é um avanço em relação às abordagens tradicionais provendo efetividade e robustez na execução de tarefas [12]. O mecanismo que indivíduos não-sofisticados utilizam para superar suas próprias limitações cognitivas através da interação local com o ambiente, causando o surgimento de padrões coerentes, globais e funcionais é conhecido como inteligência de enxames.

Estudos de insetos e animais sociais têm resultado em diversos modelos computacionais de inteligência de enxames. Sistemas coletivos biológicos que têm inspirado modelos computacionais incluem formigas, cupins, abelhas, peixes e pássaros [13][14][15][16][17]. Nestes enxames, indivíduos são relativamente simples em estrutura, mas seu comportamento coletivo resulta em padrões capazes de desempenhar atividades complexas [18]. Essas estruturas, padrões e comportamentos vêm a existir sem nenhum tipo de sistema de controle coordenado, mas emergem das interações dos indivíduos com o ambiente e da interação entre agentes.

O objetivo computacional de modelos de inteligência de enxames é modelar o comportamento simples dos indivíduos e as interações locais com o ambiente e com os indivíduos vizinhos, para que se obtenha comportamentos mais complexos que possam ser utilizados para resolver problemas complexos. Um exemplo de problema bastante recorrente é o de otimização [11]. Por exemplo, Otimização por Enxames de Partículas (PSO, do inglês *Particle Swarm Optimization*) modela dois comportamentos simples: cada indivíduo move-se em direção ao seu mais próximo e melhor vizinho, e move-se para a posição em que o indivíduo considera ter sido a sua melhor até então. Como resultado disso, o

comportamento coletivo que emerge é aquele no qual todos os indivíduos convergem para o estado do ambiente que seja o melhor para todos os indivíduos.

As principais vantagens de se utilizar abordagens baseadas em enxames são as seguintes:

- a) eficiência computacional: a disponibilidade de múltiplos processadores para um enxame diminui a carga computacional;
- b) confiabilidade: a operação contínua do grupo, apesar de falhas individuais ou mudanças no ambiente, pode ser atribuída ao controle descentralizado, compartilhamento de dados sensoriais, redundância inerente ao uso de um grande número de indivíduos e a simplicidade destes. Além disso, não existe um ponto único de falha no sistema;
- c) escalabilidade: indivíduos podem ser adicionados ou removidos do enxame, dentro de limites pré-definidos, sem o requerimento de mudanças no modelo;
- d) auto-organização: um enxame não requer a necessidade de um líder;
- e) longevidade: múltiplos agentes certamente sobreviverão por mais tempo do que um único agente; e
- f) baixo custo: um projeto simples requer menos *hardware* e é mais adequado à produção em massa.

O algoritmo de Otimização por Enxames de Partículas [17] é um modelo de busca populacional baseado na simulação do comportamento de pássaros em bando. No PSO, os indivíduos movem-se pelo espaço de busca hiperdimensional. As mudanças na posição das partículas dentro do espaço de busca são baseadas nas tendências socio-cognitivas dos indivíduos de reproduzirem o sucesso de outros indivíduos. As mudanças que ocorrem em uma partícula são portanto influenciadas pela experiência, ou conhecimento, de seus vizinhos.

No PSO, o comportamento de busca de uma partícula é afetado pelo comportamento de outras partículas dentro do enxame, caracterizando-o como um algoritmo de cooperação simbiótica. A consequência de modelar este comportamento social é que o processo de busca é tal que as partículas tendem a retornar, estocasticamente, a regiões de sucesso no espaço de busca.

A depender da abordagem, o algoritmo do PSO pode ser desenvolvido de maneira diferente considerando o tamanho da vizinhança das partículas [19]. Estas vizinhanças podem ser: global (também conhecida como *gbest* PSO), onde a vizinhança de cada partícula é o enxame inteiro e reflete a topologia estrela (*star topology*); e local (também conhecida como *lbest* PSO), onde a vizinhança é definida por poucos vizinhos para cada partícula e utiliza a topologia em anel (*ring topology*).

As partículas locomovem-se pelo hiperespaço modificando seu posicionamento baseado em sua própria experiência e na experiência da vizinhança. Cada partícula $i \in 1, 2, \dots, N$ armazena as seguintes informações:

- $\vec{x}_i(t)$ – um vetor de valores reais que determina a posição da partícula i e são os parâmetros para a solução do problema;
- $\vec{v}_i(t)$ – um vetor que determina como se dará a movimentação da partícula i ;
- $\vec{P}_{best_i}(t)$ – a melhor posição encontrada pela partícula i ;
- $\vec{G}_{best_i}(t)$ – a melhor posição encontrada pela vizinhança da partícula i , ou seja, o $\vec{P}_{best_j}(t)$ da melhor partícula j ;
- *fitness* – o valor que define a qualidade da partícula. Geralmente, é o retorno da função a ser otimizada.

A posição $\vec{x}_i(t)$ é modificada pela adição do vetor velocidade $\vec{v}_i(t)$.

O comportamento do PSO consiste em, a cada iteração, alterar o vetor velocidade da partícula de acordo com as posições $\vec{P}_{best_i}(t)$ e $\vec{G}_{best_i}(t)$. No PSO, a velocidade das partículas é regida pela Equação (2.1):

$$\vec{v}_i(t+1) = \omega \vec{v}_i(t) + c_1 r_1 [\vec{P}_{best_i} - \vec{x}_i(t)] + c_2 r_2 [\vec{G}_{best} - \vec{x}_i(t)], \quad (2.1)$$

em que $\vec{v}_i(t+1)$ é a nova velocidade da partícula i , $\vec{v}_i(t)$ e $\vec{x}_i(t)$ são, respectivamente, a velocidade atual e a posição atual da partícula.

Existem duas constantes que influenciam o comportamento do PSO: a primeira pondera a componente cognitiva c_1 , que denota a influência cognitiva individual da partícula em relação a melhor posição encontrada a partir de sua experiência pessoal, \vec{P}_{best_i} ; a segunda pondera a componente social c_2 , que especifica a influência social em relação à melhor posição encontrada pela vizinhança, \vec{G}_{best} . Os termos r_1 e r_2 são variáveis geradas por uma distribuição de probabilidade uniforme no intervalo $[0,1]$. Por fim, o fator de inércia ω provê o balanceamento entre a busca em amplitude e profundidade tendo aumentado a eficácia das soluções encontradas em um grande número de aplicações [20]. A velocidade da partícula é utilizada no cálculo da nova posição da partícula. A posição atual é somada à nova velocidade, gerando assim a posição final da partícula de acordo com a Equação (2.2):

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t)\Delta t. \quad (2.2)$$

Como o tempo é a unidade de iteração e possui valor unitário no PSO, $\Delta t = 1$.

Em resumo, o PSO funciona de acordo com o Algoritmo 1.

Algoritmo 1: Pseudocódigo da técnica Otimização por Enxame de Partículas (PSO)

```

1 Inicializar aleatoriamente as partículas no espaço de busca;
2 Avaliar o fitness de cada partícula;
3 Determinar o  $\vec{G}_{best}$ ;
4 enquanto critério de parada não for alcançado faça
5   para cada partícula faça
6     Atualizar velocidade e posição (Equações (2.1) e (2.2));
7     Avaliar o fitness da partícula;
8     Atualizar o  $\vec{P}_{best_i}$ ;
9   Atualizar o  $\vec{G}_{best}$ ;
10 Retornar o  $\vec{G}_{best}$ .

```

2.2 Robótica de enxames e Veículos Aéreos não-Tripulados

Robótica de Enxames (do inglês *Swarm Robotics*) é uma abordagem para a coordenação de grandes grupos de robôs com inspiração no comportamento de insetos sociais, tais como abelhas, formigas e cupins, com o objetivo de desenvolver sistemas que sejam tolerante a falhas, escaláveis e robustos [21].

Enxames de robôs são sistemas auto-organizados que podem ser observados em dois níveis: o individual, também chamado de microscópico; e o coletivo, também conhecido como macroscópico. O nível individual representa o comportamento de um único robô, enquanto que o nível coletivo representa o comportamento mostrado pelo enxame e é o resultado da interação dos comportamentos individuais [22]. As capacidades de percepção e comunicação são locais e os robôs não têm acesso à informação global. Tipicamente, um enxame de robôs é composto de indivíduos homogêneos, embora alguns exemplos de enxames de robôs heterogêneos já existam [23].

De acordo com Dorigo *et al* [24], as características acima mencionadas em um enxame de robôs são consideradas para promover a implementação de sistemas que sejam:

- a) tolerantes a falhas: a robótica de enxames promove o desenvolvimento de sistemas que sejam capazes de lidar bem com a falha de um ou mais de um de seus robôs constituintes. A perda de robôs individuais não deve implicar na falha do sistema inteiro. A tolerância a falhas torna-se possível graças à alta redundância do enxame: o enxame não depende de nenhum tipo de entidade de controle, líderes, ou qualquer robô individual exercendo um papel pré-definido;
- b) escaláveis: a robótica de enxames também permite o desenvolvimento de sistemas que sejam aptos a lidar com mudanças no número de indivíduos do enxame. Idealmente, a introdução ou remoção de indivíduos não deve causar uma mudança drástica na desempenho do enxame. A escalabilidade é possível graças

à percepção e à comunicação: desde que a introdução ou remoção de robôs não modifique dramaticamente a densidade do enxame, cada robô continuará interagindo com aproximadamente o mesmo número de indivíduos, aqueles que estejam em seu alcance de percepção e comunicação;

- c) flexíveis: finalmente, a robótica de enxames também promove o desenvolvimento de sistemas que são capazes de lidar com um amplo espectro de ambientes e condições de operação. A flexibilidade é possível graças à natureza distribuída e auto-organizada de um enxame de robôs: em um enxame, os robôs são alocados dinamicamente em diferentes tarefas que correspondem aos requisitos específicos do ambiente e das condições de operação; além disso, robôs operam com base na percepção local e na comunicação e não dependem de infraestrutura pré-existente ou qualquer tipo de informação global.

De acordo com Sahin [25], os principais problemas já abordados com sucesso na área de robótica de enxames são os seguintes:

- agregação;
- dispersão;
- busca;
- auto-montagem;
- conexão durante movimento;
- transporte cooperativo;
- formação de padrões;
- construção auto-organizada.

Sistemas de enxames de robôs podem ser implementados em diversas plataformas, a depender do tipo de problema que se esteja tentando abordar. Os VANTs são bastante utilizados em análise de situação de risco, monitoramento de desastres, mapeamento agrícola e em aplicações militares [26][27]. Muitas pesquisas têm percebido a utilidade dos VANTs de pequeno porte, e com o desenvolvimento da Inteligência Artificial e das tecnologias de controle de sistemas embarcados, os VANTs têm sido empregados para servir como plataformas em diversos cenários, não apenas civis e militares, mas também para educação e pesquisa acadêmica.

De acordo com [2], o desenvolvimento do sistema de controle de um VANT ainda é considerado difícil em comparação ao projeto e à implementação de sua estrutura física. Primeiramente, o sistema de controle de vôo é complicado por ser constituído de partes

mecânicas, microprocessadores e *software*. Este sistema multidisciplinar requer especialistas em diferentes áreas e uma solução também multidisciplinar. Além disso, o processo de desenvolvimento requer um consumo considerável de tempo. Os pesquisadores precisam lidar com problemas de diversos tipos relacionados à engenharia, como consumo energético, interfaces elétricas, interferência eletromagnética, compatibilidade entre *software* e *hardware*, interfaces humano-máquina e recursos materiais. Entretanto, os VANTs podem executar muitas tarefas com um custo mais baixo, além de serem considerados boas ferramentas para a validação de resultados de pesquisa ainda em etapa de simulação.

As propriedades de um sistema de enxames de robôs fazem destes opções interessantes em diversos domínios de aplicação. O uso de robôs em atividades consideradas perigosas é claramente atraente por eliminar ou reduzir riscos para humanos. A natureza perigosa dessas atividades implica um alto risco de perda de recursos. Portanto, uma abordagem que seja tolerante a falhas é necessária, fazendo com que essas atividades de risco sejam um domínio e aplicação ideal para enxames de robôs [28].

Outras potenciais aplicações para enxames de robôs são aquelas em que é difícil ou até mesmo impossível estimar com antecedência os recursos necessários para se completar uma tarefa. Nestes casos, é importante que a solução seja escalável e flexível: robôs devem ser adicionados ou removidos em tempo, de forma a proporcionar a quantidade apropriada de recursos e atingir os requisitos da atividade específica. Exemplos de tarefas que podem exigir uma quantidade de recursos desconhecidos *a priori* são busca, resgate e rastreamento [29].

Atividades que precisam ser realizadas em ambientes grandes ou não-estruturados também estão entre as aplicações de enxames de robôs. Nestes ambientes não existe uma infraestrutura disponível que possa ser utilizada para controlar os robôs, como rede de comunicação ou sistema de posicionamento global. O enxame de robôs pode ser então empregado em tais aplicações porque é capaz de agir de maneira autônoma sem a necessidade de qualquer infraestrutura ou qualquer sistema de coordenação externa. Exemplos de aplicações em ambientes não-estruturados são exploração submarina, vigilância, busca e resgate [28].

Alguns ambientes podem mudar rapidamente com o tempo. Por exemplo, em uma situação pós-terremoto, estruturas prediais podem entrar em colapso—mudando o *layout* do ambiente e criando novos possíveis riscos. Neste caso, é necessário adotar soluções que sejam flexíveis e que possam reagir rapidamente a esses eventos. A robótica de enxames pode então ser utilizada para desenvolver sistemas flexíveis que possam adaptar-se rapidamente a novas condições de operação.

2.3 Lógica *fuzzy*

A lógica bivalorada, ou Booleana, é uma teoria bem definida e bastante utilizada. O raciocínio bivalorado é especialmente importante para a implementação de sistemas computacionais onde a informação, ou conhecimento a respeito de algum problema, é codificado binariamente. Além disso, a lógica Booleana também possui uma função importante no desenvolvimento dos primeiros sistemas de raciocínio com inteligência artificial, especialmente no mecanismo de inferência de sistemas especialistas [11]. Associada à lógica Booleana, tem-se a tradicional teoria dos conjuntos bivalorados, onde um elemento pode ou não pertencer a um conjunto, caracterizando uma pertinência precisa.

Entretanto, a compreensão humana de muitos dos processos físicos existentes é amplamente baseado no raciocínio impreciso. Essa imprecisão, quando comparada com os valores precisos requeridos por computadores, é nada mais do que um tipo de informação que pode ser de bastante utilidade para seres humanos. A capacidade de embarcar tal raciocínio, até então, em problemas complexos e intratáveis, é o critério pelo qual a eficácia da lógica *fuzzy* é julgada. Enquanto um certo grau de sucesso tem sido alcançado utilizando conjuntos tradicionais da lógica clássica, não é possível resolver todos os problemas apenas mapeando-se o domínio em variáveis bivalentes.

A lógica *fuzzy* abrange o conceito de conjuntos *fuzzy*, que são uma maneira de representar incertezas não-estatísticas e de raciocínio aproximado, incluindo as operações utilizadas para se fazer a inferência em lógica *fuzzy* [20]. Em conjuntos *fuzzy*, um elemento pertence ao conjunto em um determinado grau, o que indica certeza (ou incerteza) de pertinência.

Os fundamentos da lógica de múltiplos valores foram apresentados por Lotfi A. Zadeh em 1965 [8] em conjunto com a matemática teórica dos conjuntos *fuzzy*. Seguindo o trabalho de Zadeh, muitas pesquisas têm sido realizadas na teoria dos sistemas *fuzzy*, com aplicações em controle, sistemas de informação, reconhecimento de padrões e suporte à decisão [30]. Como os conceitos de lógica e conjuntos *fuzzy* têm sido utilizados para resolver problemas reais, eles também podem ser combinados com outros paradigmas de Inteligência Computacional para formar sistemas híbridos.

2.3.1 Conjuntos *fuzzy*

Diferente dos conjuntos clássicos, os elementos de um conjunto *fuzzy* possuem graus de pertinência para o conjunto. O grau de pertinência para um conjunto indica a certeza (ou incerteza) de que um elemento pertence ao conjunto. Formalmente definindo, suponha X como o domínio, ou universo do discurso, e $x \in X$ um elemento específico do domínio X . Então, um conjunto *fuzzy* A é caracterizado pela função de associação

$$\mu_A : X \rightarrow [0, 1]. \quad (2.3)$$

Portanto, para todo $x \in X$, $\mu_A(x)$ indica a certeza que o elemento x pertence ao conjunto *fuzzy* A . Para conjuntos bivalentes, $\mu_A(x)$ é 0 ou 1.

Os conjuntos *fuzzy* podem ser definidos para domínios contínuos ou discretos. A notação utilizada para denotar os conjuntos *fuzzy* pode variar dependendo do tipo de domínio sobre o qual o conjunto está definido. No caso de um domínio discreto X , o conjunto *fuzzy* pode tanto ser expresso na forma de um vetor n -dimensional ou utilizando a notação soma. Considerando $X = \{x_1, x_2, \dots, x_n\}$, então utilizando a notação de conjunto

$$A = \{(\mu_A(x_i)/x_i) | x_i \in X, i = 1, \dots, n\}. \quad (2.4)$$

Utilizando a notação soma,

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_{n_x})/x_{n_x} = \sum_{i=1}^{n_x} \mu_A(x_i)/x_i. \quad (2.5)$$

em que a soma não deve ser confundida com o somatório algébrico. A utilização da notação soma anteriormente apenas serve como indicação que A é um conjunto de pares ordenados. Um conjunto *fuzzy* contínuo, A , é denotado por

$$A = \int_X \mu(x)/x. \quad (2.6)$$

2.3.2 Funções de pertinência

A função de pertinência, ou função de associação, é a essência dos conjuntos *fuzzy*. A função de pertinência, também denominada como a função característica do conjunto *fuzzy*, define o conjunto *fuzzy*. Esta função é utilizada para associar um grau de pertinência para cada elemento do domínio ao conjunto *fuzzy* correspondente. Conjuntos de dois valores, ou bivalentes, também são caracterizados por uma função de pertinência [11]. Por exemplo, considere o domínio X de todos os números de ponto flutuante no intervalo $[0, 100]$. Defina o conjunto *crisp*, ou preciso, $A \subset X$ para todos os números de ponto flutuante no intervalo $[10, 50]$. Então, a função de pertinência para o conjunto *crisp* A é representado pela Figura 1. Todo $x \in [10, 50]$ possui $\mu_A(x) = 1$, enquanto todos outros números de ponto flutuante possuem $\mu_A(x) = 0$.

Funções de associação para conjuntos *fuzzy* podem apresentar qualquer formato ou tipo determinado por especialistas no domínio sobre o qual os conjuntos são definidos. Embora os projetistas de conjuntos *fuzzy* possuam muita liberdade na seleção de funções de pertinência apropriadas, estas funções devem satisfazer as seguintes restrições [11]: (i) o intervalo de uma função de pertinência deve ser $[0,1]$; e (ii) para cada $x \in X$, $\mu_A(x)$ deve ser único. Ou seja, ao mesmo elemento não se pode mapear diferentes graus de pertinência para o mesmo conjunto *fuzzy*.

Uma possível função de associação para o conjunto *fuzzy* *alto* pode ser definida de acordo com a Equação (2.7). A Figura 2 ilustra esta função de associação. Por exemplo,

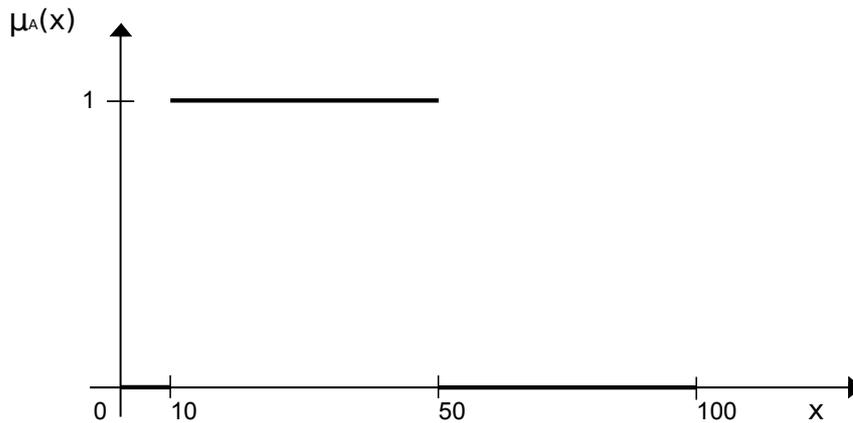


Figura 1: Ilustração da uma função de pertinência para conjuntos clássicos de dois valores.

Fonte: Engelbrecht, p. 455

para uma pessoa com altura de $1,75m$, $\mu_A(1,75) = 0,5$.

$$alto(x) = \begin{cases} 0, & altura(x) < 1,5m, \\ (altura(x) - 1,5m) \times 2,0m^{-1} & altura(x) \in [1,5m, 2,0m], \\ 1, & altura(x) > 2,0m. \end{cases} \quad (2.7)$$

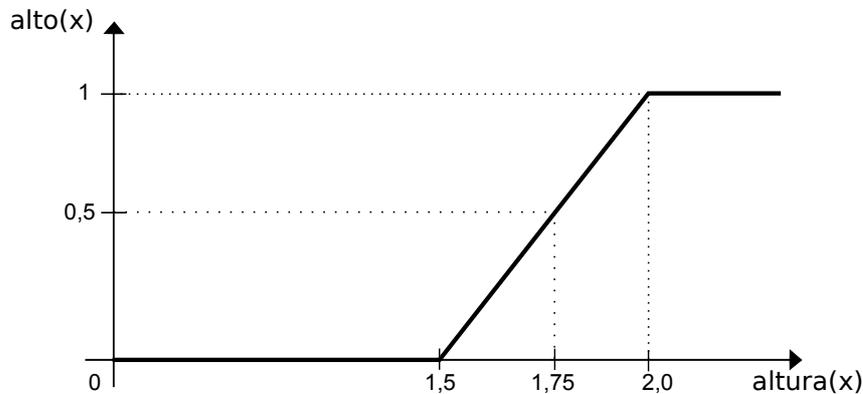


Figura 2: Ilustração da uma função de pertinência para o conjunto *fuzzy* alto.

Fonte: Engelbrecht, p. 455

Embora a função de associação anteriormente mencionada seja simples, podem ser utilizadas funções de pertinência mais complexas como, por exemplo:

a) função triangular (Figura 3 (a)) definida de acordo com a Equação (2.8):

$$\mu_A(x) = \begin{cases} 0, & x \leq \alpha_{min}, \\ \frac{x - \alpha_{min}}{\beta - \alpha_{min}}, & x \in (\alpha_{min}, \beta], \\ \frac{\alpha_{max} - x}{\alpha_{max} - \beta}, & x \in (\beta, \alpha_{max}), \\ 0, & x \geq \alpha_{max}; \end{cases} \quad (2.8)$$

b) função trapezoidal (Figura 3 (b)) definida de acordo com a Equação (2.9):

$$\mu_A(x) = \begin{cases} 0, & x \leq \alpha_{min}, \\ \frac{x - \alpha_{min}}{\beta_1 - \alpha_{min}}, & x \in (\alpha_{min}, \beta_1], \\ \frac{\alpha_{max} - x}{\alpha_{max} - \beta_2}, & x \in (\beta_2, \alpha_{max}), \\ 0, & x \geq \alpha_{max}; \end{cases} \quad (2.9)$$

c) funções Γ -pertinência de acordo com a Equação (2.10):

$$\mu_A(x) = \begin{cases} 0, & x \leq \alpha, \\ 1 - \exp[-\gamma(x - \alpha)^2], & x > \alpha; \end{cases} \quad (2.10)$$

d) função logística (Figura 3 (c)) de acordo com a Equação (2.11):

$$\mu_A(x) = \frac{1}{1 + \exp[-\gamma x]}; \quad (2.11)$$

e) funções de associação exponenciais de acordo com a Equação (2.12):

$$\mu_A(x) = \frac{1}{1 + \gamma(x - \beta)^2}, \quad (2.12)$$

em que $\gamma > 1$;

f) função gaussiana (Figura 3 (d)) de acordo com a Equação (2.13):

$$\mu_A(x) = e^{-\gamma(x - \beta)^2}. \quad (2.13)$$

2.3.3 Operadores *fuzzy*

Assim como nos conjuntos *crisp*, as relações e os operadores são definidos para os conjuntos *fuzzy*. Cada relação e operador é definido a seguir. Para este propósito, deve-se assumir que X é o domínio, ou universo, e A e B são conjuntos *fuzzy* definidos sobre o domínio X .

- a) igualdade de conjuntos *fuzzy*: Dois conjuntos *fuzzy* A e B são iguais se e somente se os conjuntos possuírem o mesmo domínio e $\mu_A(x) = \mu_B(x)$ para todo $x \in X$. Ou seja, $A = B$;
- b) Contenção de conjuntos *fuzzy*: O conjunto *fuzzy* A é um subconjunto do conjunto *fuzzy* B se e somente se $\mu_A(x) \leq \mu_B(x)$, para todo $x \in X$. Ou seja, $A \subset B$;
- c) complemento de um conjunto *fuzzy* (NÃO): O complemento do conjunto A consiste em todos os elementos do conjunto A mas com graus de pertinência diferentes. Denotando $\neq A$ o complemento do conjunto A , para todo $x \in X$, $\mu_{\neq A}(x) = 1 - \mu_A(x)$;

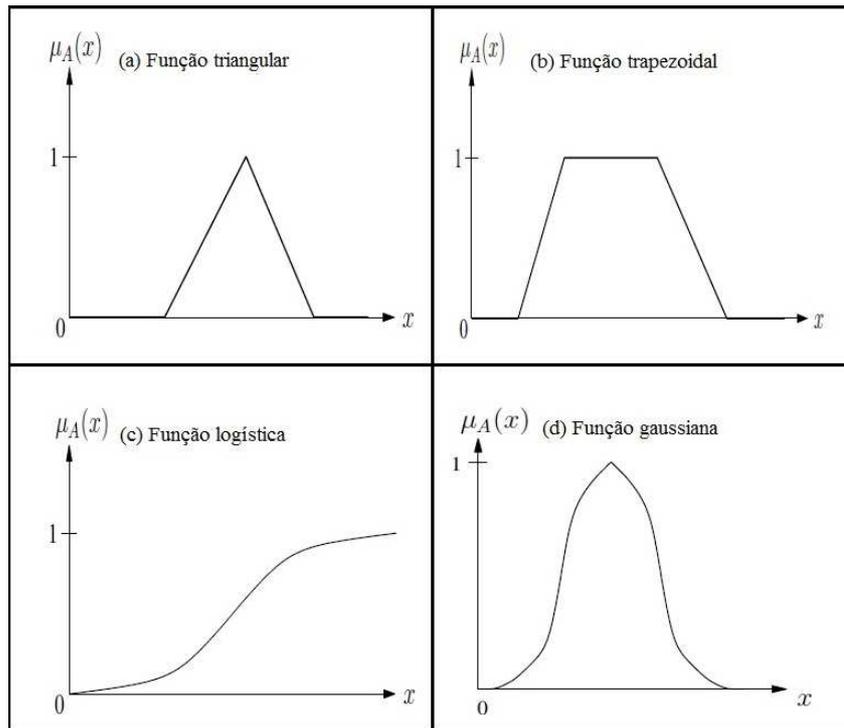


Figura 3: Ilustração de funções de pertinência para conjuntos *fuzzy*. A função (a) corresponde a função triangular, a função (b) corresponde a função trapezoidal, a função (c) corresponde a função logística e a função (d) corresponde a função gaussiana.

Fonte: Engelbrecht, p. 458

- d) interseção de conjuntos *fuzzy* (E): A abordagem mais comum para calcular a interseção entre dois conjuntos *fuzzy* é o *min-operator*: $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$, $\forall x \in X$;
- e) união de conjuntos *fuzzy* (OU): A abordagem mais comum para calcular a interseção entre dois conjuntos *fuzzy* é o *max-operator*: $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$, $\forall x \in X$.

2.3.4 Lógica *fuzzy*

Zadeh define a lógica *fuzzy* como uma extensão da lógica multivalorada, destinada a servir como uma lógica para o raciocínio aproximado [8]. Os dois conceitos mais importantes na lógica *fuzzy* são os de variável linguística e de regras *fuzzy*, mais precisamente do tipo *se-então*.

Variáveis linguísticas são variáveis capazes de assumir valores que podem ser representados por palavras ou frases da linguagem natural ou artificial. Estas variáveis permitem a transição a partir da linguagem natural para os valores numéricos.

Em geral, o comportamento dinâmico dos sistemas *fuzzy* em geral é caracterizado

por um conjunto de regras linguísticas. Essas regras são consolidadas a partir do conhecimento prévio do problema e de seus domínios específicos. As regras *fuzzy* possuem a forma geral

$$\text{se antecedente(s), então consequente(s)}. \quad (2.14)$$

O antecedente e o consequente de uma regra *fuzzy* são proposições contendo variáveis linguísticas. Em geral, uma regra *fuzzy* é expressada como

$$\text{se } A \text{ é } a \text{ e } B \text{ é } b \text{ então } C \text{ é } c, \quad (2.15)$$

onde A e B são conjuntos *fuzzy* com um universo de discurso X_1 e C é um conjunto *fuzzy* com um universo de discurso X_2 . Dessa forma, o antecedente de uma regra forma uma combinação de conjuntos *fuzzy* através da aplicação de operadores lógicos. A parte consequente de uma regra é geralmente um único conjunto *fuzzy*, com uma função de pertinência correspondente. Múltiplos conjuntos *fuzzy* podem também ocorrer no termo consequente, quando serão combinados utilizando operadores lógicos.

As regras e os conjuntos formam a base de um sistema de indução *fuzzy*. Adicionalmente, um sistema *fuzzy* consiste em três outros componentes, onde cada um deles possui uma tarefa específica no processo de indução: *fuzzificação*, *inferência* e *defuzzificação*. O processo de *fuzzificação* consiste em encontrar uma representação *fuzzy* para valores de entrada não-*fuzzy*. Isso é possível através da aplicação de regras de pertinência associadas à cada conjunto *fuzzy* nos antecedentes de cada regra. A inferência consiste em avaliar as relações existentes entre os conjuntos de entrada através dos operadores lógicos, e então atribuir um grau de pertinência aos consequentes das regras baseado nessas avaliações. Finalmente, a *defuzzificação* consiste em converter as saídas das regras *fuzzy* em valores escalares, não-*fuzzy*, dado um conjunto de regras que foram devidamente ativadas e os seus valores de disparo correspondentes.

2.3.5 Controladores *fuzzy*

A área de Sistemas de controle é um dos maiores campos de aplicação da lógica *fuzzy*. O controle é expresso como um conjunto de regras linguísticas ao invés de equações matemáticas diferenciais. As regras linguísticas são mais familiares à compreensão humana do que a utilização de sistemas de equações matemáticas.

Um controlador *fuzzy* pode ser considerado como uma função não-linear estática que mapeia as entradas do controlador às suas saídas. O sistema a ser controlado possui uma resposta desejada, que deve ser mantida independentemente do valor de entrada. As entradas do sistema podem, entretanto, mudar o estado deste, o que causa uma mudança na resposta. A função do controlador consiste em tomar ações corretivas para que o sistema obtenha a saída desejada.

2.3.5.1 Componentes de um controlador *fuzzy*

Um controlador *fuzzy* consiste de quatro componentes principais:

- a) banco de regras *fuzzy*: também conhecido como *knowledge base*, contém as regras *fuzzy* que expressam a estratégia de controle não-linear do sistema;
- b) interface de condição (*fuzzificador*): contém as saídas do sistema e transforma os valores não-*fuzzificados* em graus de pertinência do conjunto *fuzzy* correspondente;
- c) mecanismo de inferência: realiza a inferência com relação aos valores de entrada *fuzzificados*, a fim de produzir uma saída *fuzzificada*.
- d) interface de ação (*defuzzificador*): a interface de ação *defuzzifica* o resultado do mecanismo de inferência, a fim de produzir um valor não-*fuzzy* que representa a resposta de controle a ser aplicada ao sistema;

As principais causas da não-linearidade de um controlador *fuzzy* são:

- o processo de *fuzzificação*, caso funções de pertinência não-lineares sejam utilizadas;
- o conjunto de regras, dado que as regras expressam uma estratégia de controle não-linear;
- o mecanismo de inferência;
- o processo de *defuzzificação*.

2.3.5.2 Controlador *fuzzy* de Mamdani

Enquanto que existe um número de diferentes tipos de controladores *fuzzy*, todos eles possuem os mesmos componentes e envolvem os mesmos passos de construção. As diferenças entre eles estão principalmente na implementação do mecanismo de inferência e do *defuzzificador*.

Mamdani e Assilian [31] desenvolveram o primeiro controlador *fuzzy* em 1975. O projeto de um controlador *fuzzy* do tipo Mamdani segue os passos seguintes:

- a) identificar e nomear as variáveis linguísticas de entrada e definir a extensão de seus valores numéricos;
- b) identificar e nomear as variáveis linguísticas de saída e definir a extensão de seus valores numéricos;
- c) definir um conjunto de funções de pertinência *fuzzy* para cada uma das variáveis de entrada, assim como para as variáveis de saída;
- d) construir uma base de regras que representa a estratégia de controle;

- e) executar a *fuzzificação* das variáveis de entrada;
- f) executar o mecanismo de inferência a fim de determinar os valores de disparo das regras ativadas;
- g) *Defuzzificar*, utilizando a centróide de gravidade, para determinar a ação correspondente a ser executada.

2.3.5.3 Controlador *fuzzy* de Takagi-Sugeno

As saídas de um controlador Mamdani são constituídas de *singletons* (um único conjunto), ou combinações de *singletons*, onde essas combinações são obtidas através da aplicação de operações entre os conjuntos *fuzzy*. Os conjuntos de saída podem, entretanto, também ser combinações lineares das entradas. Takagi e Sugeno propuseram uma abordagem que permite a implementação desses sistemas mais complexos, conhecida como controladores *fuzzy* Takagi-Sugeno [9]. Em geral, a estrutura das regras para controladores Takagi-Sugeno é a seguinte:

$$\text{se } f_1(a_1 \text{ é } A_1, a_2 \text{ é } A_2, \dots, a_n \text{ é } A_n,) \text{ então } C = f_2(a_1, a_2, \dots, a_n,) \quad (2.16)$$

onde f_1 é uma função lógica, e f_2 é uma função matemática das entradas; C é o consequente, ou a variável de saída sendo inferida, a_i é um antecedente, ou variável de entrada, e A_i é um conjunto *fuzzy* representado pela função de pertinência μ_{A_i} . A base de regras completa é definida por n_k regras.

O valor de disparo de cada regra é calculado utilizando o *min-operator*:

$$\alpha_k = \min_{\forall i | a_i \in \mathcal{A}_k} \{ \mu_{A_i}(a_i) \}. \quad (2.17)$$

onde \mathcal{A}_k é o conjunto de antecedentes da regra k . Alternativamente, o produto pode ser utilizado para calcular os valores de disparo das regras:

$$\alpha_k = \prod_{\forall i | a_i \in \mathcal{A}_k} \mu_{A_i}(a_i). \quad (2.18)$$

A saída do controlador é então determinada por

$$C = \frac{\sum_{k=1}^{n_k} \alpha_k f_2(a_1, \dots, a_n)}{\sum_{k=1}^{n_k} \alpha_k}. \quad (2.19)$$

A principal vantagem de um controlador Takagi-Sugeno é a quebra da abordagem de ciclo fechado existente nos controladores de Mamdani. Enquanto que nos controladores de Mamdani o sistema é estaticamente descrito pelas regras, os controladores de Takagi-Sugeno possuem um controle mais dinâmico pelo fato de que o consequente das regras é constituído por funções matemáticas.

2.4 Modelo de coordenação utilizando inteligência de enxames

O modelo de coordenação descrito a seguir propõe um sistema baseado principalmente no sincronismo entre VANTs de modo a evitar perda de comunicação e a evitar colisões, e no rastreamento de alvos com utilização do algoritmo de Inteligência de Enxames conhecido como PSO. O comportamento dos VANTs é definido a partir da interação destes com os outros elementos do sistema e por uma máquina de estados finitos.

2.4.1 VANTs

Os VANTs são os principais elementos do sistema, responsáveis por patrulhar o ambiente e rastrear possíveis alvos invasores. Cada VANT possui sensores, os quais dão a capacidade ao VANT de perceber o ambiente. Os principais sensores do VANT são:

- a) sensor de localização: responsável por fornecer a posição do VANT no ambiente;
- b) sensor anti-colisão: responsável por fornecer informações a fim de que colisões sejam evitadas;
- c) sensor de comunicação: responsável por possibilitar a comunicação entre VANTs;
- d) sensor de percepção: responsável por detectar alvos invasores no ambiente.

2.4.2 Mecanismo de localização

A posição do VANT é representada a partir de coordenadas globais em relação à sua posição atual no ambiente, desfazendo-se a necessidade da utilização de um mapa interno com informações a respeito deste ambiente. A localização do VANT na iteração é fornecida pelo sensor de localização e é essencial para as tarefas de patrulha e sensoria-mento do enxame. O mecanismo de localização foi modelado de modo a permitir o efeito de erros ou imprecisões na obtenção da posição do VANT pelo sensor de localização. Os erros de localização inteferem na transmissão de localização do VANT aos seus vizinhos, fazendo com que a posição compartilhada não seja necessariamente a posição real em que o VANT se encontra. O erro do ruído de localização é obtido através da Equação (2.20)

$$r = \mathcal{U}[0, 1] \cdot e_{x-loc}, \quad (2.20)$$

onde $\mathcal{U}[0, 1]$ é uma variável aleatória distribuída normalmente com média igual a 0 e desvio padrão igual a 1, e e_{x-loc} é o valor do erro de localização em metros e que pode ser definido junto com os parâmetros de simulação.

A localização do VANT no ambiente é determinada de acordo com a Equação (2.21):

$$\vec{x}_{uav-i}(t) = \vec{x}_{real-uav}(t) + r \quad (2.21)$$

onde $\vec{x}_{real-uav}(t)$ é a posição real do VANT i no ambiente na iteração t , e r é o ruído de localização.

2.4.3 Mecanismo de locomoção

O mecanismo de locomoção tem como objetivo representar os aspectos de aerodinâmica dos VANTs de modo a possibilitar a locomoção destes no ambiente. Os componentes do mecanismo de locomoção são descritos a seguir:

- a) aceleração ($\vec{a}_{uav-i}(t)$): Vetor que determina a aceleração do VANT i na iteração t em m/s^2 ;
- b) velocidade ($\vec{v}_{uav-i}(t)$): Vetor que determina a velocidade do VANT i na iteração t em m/s ;
- c) posição ($\vec{x}_{uav-i}(t)$): Vetor que determina a posição do VANT i na iteração t em m .

A locomoção do VANT no ambiente é determinada de acordo com a Equação (2.22):

$$\vec{x}_{uav-i}(t+1) = \vec{x}_{uav-i}(t) + \vec{v}_{uav-i}(t)\Delta t, \quad (2.22)$$

em que $\Delta t = 1$ e $\vec{v}_{uav-i}(t)$ é a velocidade do VANT i na iteração t calculada de acordo com a Equação (2.23):

$$\vec{v}_{uav-i}(t+1) = \vec{v}_{uav-i}(t) + \vec{a}_{uav-i}(t)\Delta t, \quad (2.23)$$

em que $\Delta t = 1$ e $\vec{a}_{uav-i}(t)$ é a aceleração do VANT, descrita na Seção 2.4.8.

2.4.4 Mecanismo de comunicação

O mecanismo de comunicação tem como objetivo possibilitar a conexão e troca de informações entre os VANTs. Para possibilitar a comunicação por todo enxame levando em consideração aspectos de consumo energético, o sistema de comunicação utilizado é do tipo *ad hoc*. Neste sistema, cada VANT atua como um roteador de uma rede de comunicações e a mensagem é transmitida pelo VANT aos seus vizinhos e estes, por sua vez, retransmitem-na para seus respectivos vizinhos.

O conteúdo da mensagem transmitida possui 20 Bytes. A transmissão é do tipo *broadcasting* em cada iteração. A mensagem contém as seguintes informações:

- a) 4 Bytes, os quais contêm a identificação do VANT remetente;
- b) 8 Bytes, os quais contêm as coordenadas da posição do VANT remetente;
- c) 8 Bytes, os quais contêm a posição do alvo móvel rastreado;

O VANT i é considerado conectado de acordo com a Equação (2.24):

$$n_{nei-i}(t) = n_{nei-max}, \quad (2.24)$$

em que $n_{nei-max}$ determina o número máximo de vizinhos e $n_{nei-i}(t)$ representa o número de VANTs vizinhos conectados ao VANT i na iteração t . Para garantir o roteamento de informações, $n_{nei-i}(t) > 1$.

2.4.5 Mecanismo de sincronismo

O mecanismo de sincronismo tem como objetivo evitar colisões entre VANTs e obstáculos, e ao mesmo tempo, evitar a perda de conectividade com os vizinhos. Este mecanismo consiste na combinação dos mecanismos de anti-colisão e comunicação, de acordo com a Equação (2.25):

$$\vec{a}_{syn-i}(t) = \gamma_{col}\vec{a}_{col-i}(t) + \gamma_{com}\vec{a}_{com-i}(t), \quad (2.25)$$

onde γ_{col} e γ_{com} são os coeficientes anti-colisão e anti-afastamento, respectivamente. Estes coeficientes determinam a ativação ou desativação do mecanismo.

O mecanismo anti-colisão tem como objetivo evitar colisões de VANTs com obstáculos e outros VANTs. Este mecanismo é composto pelos seguintes parâmetros:

- a) raio de alcance (r_{col}): Determina o raio de alcance do sensor anti-colisão;
- b) limiar (l_{col}): Determina o início da atuação efetiva do sensor anti-colisão;
- c) segurança (s_{col}): Determina a distância mínima de segurança na qual é desejado atuar repulsivamente com intensidade igual a $a_{uav-max}$.

Por sua vez, o mecanismo anti-afastamento tem como objetivo evitar que os VANTs percam a conectividade com seus vizinhos devido à distância na qual se encontram. Este mecanismo é constituído pelos parâmetros a seguir:

- a) raio de alcance (r_{com}): Este parâmetro determina o raio de alcance do sensor de comunicação;
- b) limiar (l_{com}): Este parâmetro determina o início da atuação efetiva do sensor de comunicação;
- c) segurança (s_{com}): Este parâmetro determina a distância mínima na qual é desejado atuar com intensidade igual a $a_{uav-max}$ por segurança;
- d) número máximo de vizinhos ($n_{nei-max}$): Este parâmetro determina o número máximo de vizinhos aos quais um VANT pode se conectar.

2.4.6 Mecanismo de percepção

O mecanismo de percepção é responsável pela detecção e rastreamento de alvos móveis no ambiente. Este mecanismo é inspirado no compartilhamento de informações existente no algoritmo do PSO mencionado na Seção 2.1. O mecanismo de percepção é calculado de acordo com a Equação (2.26):

$$\vec{a}_{per-i}(t) = \omega \vec{v}_{uav-i}(t) + c_1 r_1 [\vec{P}_{best-i}(t) - \vec{x}_{uav-i}(t)] + c_2 r_2 [\vec{L}_{best-i}(t) - \vec{x}_{uav-i}(t)], \quad (2.26)$$

onde:

- ω é o fator de inércia, responsável por prover o balanceamento entre a busca em amplitude e busca em profundidade;
- $\vec{x}_{uav-i}(t)$ é a localização do VANT i no ambiente na iteração t ;
- c_1 é a aceleração cognitiva, responsável por determinar a influência individual de cada VANT na percepção do alvo;
- c_2 é a aceleração social, responsável por determinar a influência social da informação comunicada pelos VANTs vizinhos;
- r_1 e r_2 são variáveis aleatórias uniformemente distribuídas no intervalo entre zero e um, $r_1, r_2 \in U(0, 1)$;
- $\vec{P}_{best-i}(t)$ é a melhor posição encontrada pelo VANT i na iteração t ;
- $\vec{G}_{best-i}(t)$ é a melhor posição encontrada e comunicada pelos vizinhos do VANT i na iteração t .

Para calcular e atualizar as melhores soluções encontradas, o PSO faz uso de uma função de aptidão denominada *fitness*. Os valores dos termos $\vec{P}_{best-i}(t)$ e $\vec{G}_{best-i}(t)$ são calculados de acordo com a função de aptidão apresentada na Equação (2.27):

$$f_{uav-i}(t) = |d_{per-ij}(t)|, \quad (2.27)$$

que, por sua vez, é a distância euclidiana entre a posição do VANT i e a posição do alvo j . A função *fitness* define o VANT qualitativamente: quanto menor for o valor desta função, melhor é a qualidade da posição, pois o VANT i está mais próximo do alvo j .

2.4.7 Mecanismo de patrulha e de retorno à base

O mecanismo de patrulha tem como objetivo garantir a cobertura territorial do ambiente a partir de uma aceleração de patrulha $\vec{a}_{pat-i}(t)$ para o VANT i na iteração t . A

aceleração de patrulha é inicializada por um vetor unitário cuja direção e sentido foram gerados aleatoriamente a partir de uma distribuição uniforme.

O mecanismo de retorno à base tem como objetivo direcionar o VANT para as coordenadas da base no ambiente em caso de afastamento dos limites do ambiente durante a simulação. A aceleração de retorno à base \vec{a}_{bas-i} possui intensidade igual à aceleração máxima do VANT.

2.4.8 Mecanismo de coordenação

O mecanismo de coordenação consiste na combinação dos mecanismos de anti-colisão, comunicação, percepção, patrulha e retorno à base aérea. O mecanismo de coordenação dos VANTs proposto em [10] é diretamente influenciado pela máquina de estados do sistema: esta é responsável por definir qual o comportamento a ser assumido pelo VANT, e por conseguinte, quais os mecanismos a serem ativados de forma a constituírem o mecanismo de coordenação correspondente ao estado atual do VANT. O mecanismo de coordenação é calculado conforme a Equação (2.28):

$$\vec{a}_{uav-i}(t) = \gamma_{col}\vec{a}_{col-i}(t) + \gamma_{com}\vec{a}_{com-i}(t) + \gamma_{per}\vec{a}_{per-i}(t) + \gamma_{pat}\vec{a}_{pat-i}(t) + \gamma_{bas}\vec{a}_{bas-i}(t), \quad (2.28)$$

onde $\vec{a}_{col-i}(t)$ e $\vec{a}_{com-i}(t)$ são os mecanismos de anti-colisão e comunicação, que juntos formam o mecanismo de sincronismo; $\vec{a}_{per-i}(t)$ é o mecanismo de percepção; $\vec{a}_{pat-i}(t)$ é o mecanismo de patrulha; $\vec{a}_{bas-i}(t)$ é o mecanismo de retorno à base; e γ_{col} , γ_{com} , γ_{per} , γ_{pat} , e γ_{bas} são os coeficientes de ativação desses mecanismos, respectivamente.

A Tabela 1 apresenta os estados que o VANT pode assumir e os respectivos valores dos coeficientes da Equação (2.28). Nesta tabela, é possível observar, por exemplo, que os coeficientes γ_{col} , γ_{com} e γ_{pat} são os únicos ativados no estado de patrulha.

Tabela 1: Estados assumidos pelo VANT no modelo de coordenação proposto e os respectivos valores assumidos pelos coeficientes da Equação (2.28).

Estado	γ_{col}	γ_{com}	γ_{per}	γ_{pat}	γ_{bas}
Patrulha	1	1	0	1	0
Rastreamento	1	1	1	0	0
Retorno à Base	1	0	0	0	1
Recarregamento de Energia	0	0	0	0	0

Fonte: Silva, p. 18

3 Contribuição

Apesar de abranger com sucesso os comportamentos aqui descritos para um modelo de VANT, o mecanismo de coordenação apresentado na Seção 2.4.8 não consegue abordar com fidelidade a complexidade e a não-linearidade de sistemas reais. A utilização de uma máquina de estados para modelar o mecanismo de coordenação de um veículo que necessite agir e reagir considerando fatores diversos no ambiente, como possíveis colisões, prejudica a sua atuação por causa da rigidez da máquina estados. Em busca de diminuir esta influência estática, definindo quais coeficientes de coordenação estão ativos a depender do estado atual do VANT, foi desenvolvido um mecanismo de controle para o cálculo dos coeficientes γ_{col} , γ_{com} e γ_{per} baseado no controlador de Takagi-Sugeno.

3.1 Modelo de coordenação híbrido Swarm-Fuzzy

O mecanismo de controle dos coeficientes é consituído pelas três variáveis contínuas a seguir: a distância de comunicação entre VANTs, a distância de colisão entre VANTs e, por fim, a distância entre um VANT e um alvo. Estas variáveis estão descritas a seguir.

A distância de comunicação entre o VANT i e o VANT vizinho j na iteração t é medida pelo sensor de comunicação de acordo com a Equação (3.1):

$$\vec{d}_{com-ij}(t) = \vec{x}_{nei-j}(t) - \vec{x}_{uav-i}(t), \quad (3.1)$$

em que $\vec{x}_{nei-j}(t)$ é a posição do VANT j na iteração t .

A distância de colisão entre o VANT i e o VANT j dentro do alcance do sensor de anti-colisão na iteração t é medida de acordo com a Equação (3.2):

$$\vec{d}_{col-ij}(t) = \vec{x}_{uav-i}(t) - \vec{x}_{uav-j}(t), \quad (3.2)$$

em que $\vec{x}_{uav-i}(t)$ é a posição do VANT i na iteração t e $\vec{x}_{uav-j}(t)$ é a posição do VANT j na iteração t .

A distância de percepção entre um VANT i e um alvo j dentro do alcance do sensor de percepção é calculada pela Equação (3.3)

$$\vec{d}_{per-ij}(t) = \vec{x}_{uav-i}(t) - x_{tgt-j}(t), \quad (3.3)$$

em que $x_{tgt-j}(t)$ é a posição do alvo j na iteração t .

O mecanismo de controle proposto para o cálculo dos coeficientes procura considerar a possibilidade de imprecisões e incertezas que podem existir em sistemas reais. Para isso, faz uso de variáveis linguísticas devido à capacidade destas em representar estas

imprecisões. Para cada variável contínua, foi criada uma variável linguística, as quais são descritas a seguir:

- a) distância de comunicação (D_{Com}): representa os valores assumidos pela variável $\vec{d}_{com-ij}(t)$ por meio dos seguintes conjuntos: (i) *Close*: representa uma distância de comunicação entre VANTs considerada próxima; e (ii) *Far*: representa uma distância de comunicação entre VANTs considerada distante;
- b) distância de colisão (D_{Col}): representa os valores assumidos pela variável $\vec{d}_{col-ij}(t)$ por meio dos conjuntos: (i) *Unsafe*: representa uma distância de colisão entre VANTs considerada insegura; (ii) *Near*: representa uma distância de colisão entre VANTs considerada razoável; e (iii) *Safe*: representa uma distância de colisão entre VANTs considerada segura;
- c) distância de retreamento (D_{Rast}): representa os valores assumidos pela variável $\vec{d}_{per-ij}(t)$ por meio dos conjuntos a seguir: (i) *TargetClose*: representa uma distância entre um VANT e um alvo considerada próxima; e (ii) *TargetFar*: representa uma distância entre um VANT e um alvo considerada distante.

As funções de pertinência, como já descritas na Seção 2.3.2, determinam o grau de associação de um determinado valor ao seu conjunto, onde este grau de associação pode assumir valores $\in [0, 1]$. As funções de pertinência da variável D_{Com} para os conjuntos *fuzzy Close* e *Far* são $f_{DCom-Close}$ e $f_{DCom-Far}$, respectivamente. Estas funções estão definidas de acordo com as Equações (3.4) e (3.5) apresentadas a seguir:

$$f_{DCom-Close}(\vec{d}_{com-ij}(t)) = \begin{cases} 1, & \vec{d}_{com-ij}(t) \leq 4, \\ \frac{10-\vec{d}_{com-ij}(t)}{6} & \vec{d}_{com-ij}(t) \in (4, 10), \\ 0, & \vec{d}_{com-ij}(t) \geq 10, \end{cases} \quad (3.4)$$

$$f_{DCom-Far}(\vec{d}_{com-ij}(t)) = \begin{cases} 0, & \vec{d}_{com-ij}(t) \leq 8, \\ \frac{\vec{d}_{com-ij}(t)-4}{8} & \vec{d}_{com-ij}(t) \in (8, 16], \\ 1, & \vec{d}_{com-ij}(t) > 16. \end{cases} \quad (3.5)$$

As funções de pertinência da variável linguística D_{Com} podem ser visualizadas na Figura 4.

As funções de pertinência da variável D_{Col} para os conjuntos *fuzzy Unsafe*, *Near* e *Safe* são $f_{DCol-Unsafe}$, $f_{DCol-Near}$ e $f_{DCol-Safe}$, respectivamente. Estas funções estão definidas de acordo com as Equações (3.6), (3.7) e (3.8) a seguir:

$$f_{DCol-Unsafe}(\vec{d}_{col-ij}(t)) = \begin{cases} 1, & \vec{d}_{col-ij}(t) \leq 3, \\ \frac{5-\vec{d}_{col-ij}(t)}{2} & \vec{d}_{col-ij}(t) \in (3, 5), \\ 0, & \vec{d}_{col-ij}(t) \geq 5, \end{cases} \quad (3.6)$$

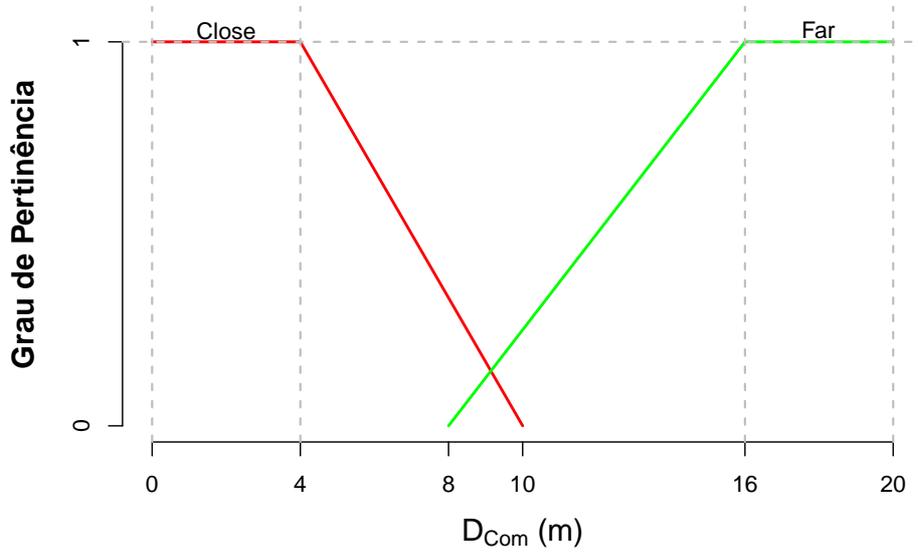


Figura 4: Representação das funções de pertinência *fuzzy* para a variável linguística D_{Com} e seus respectivos conjuntos *Close* e *Far*.

$$f_{DCol-Near}(\vec{d}_{col-ij}(t)) = \begin{cases} 0, & \vec{d}_{col-ij}(t) \leq 3, \\ \frac{\vec{d}_{col-ij}(t)-3}{3} & \vec{d}_{col-ij}(t) \in (3, 6], \\ \frac{9-\vec{d}_{col-ij}(t)}{3} & \vec{d}_{col-ij}(t) \in (6, 9], \\ 0, & \vec{d}_{col-ij}(t) \geq 9, \end{cases} \quad (3.7)$$

$$f_{DCol-Safe}(\vec{d}_{col-ij}(t)) = \begin{cases} 0, & \vec{d}_{col-ij}(t) \leq 7, \\ \frac{\vec{d}_{col-ij}(t)-7}{2} & \vec{d}_{col-ij}(t) \in (7, 9], \\ 1, & \vec{d}_{col-ij}(t) > 9. \end{cases} \quad (3.8)$$

As funções de pertinência da variável linguística D_{Col} podem ser visualizadas na Figura 5.

As funções de pertinência da variável D_{Rast} para os conjuntos *fuzzy* *TargetFar* e *TargetClose* são $f_{DRast-Far}$ e $f_{DRast-Close}$, respectivamente. Estas funções, por sua vez, são calculadas de acordo com as Equações (3.9) e (3.10) apresentadas a seguir:

$$f_{DRast-Close}(\vec{d}_{per-ij}(t)) = \begin{cases} 1, & \vec{d}_{per-ij}(t) \leq 2, \\ \frac{6-\vec{d}_{per-ij}(t)}{4} & \vec{d}_{per-ij}(t) \in (2, 6), \\ 0, & \vec{d}_{per-ij}(t) \geq 6, \end{cases} \quad (3.9)$$

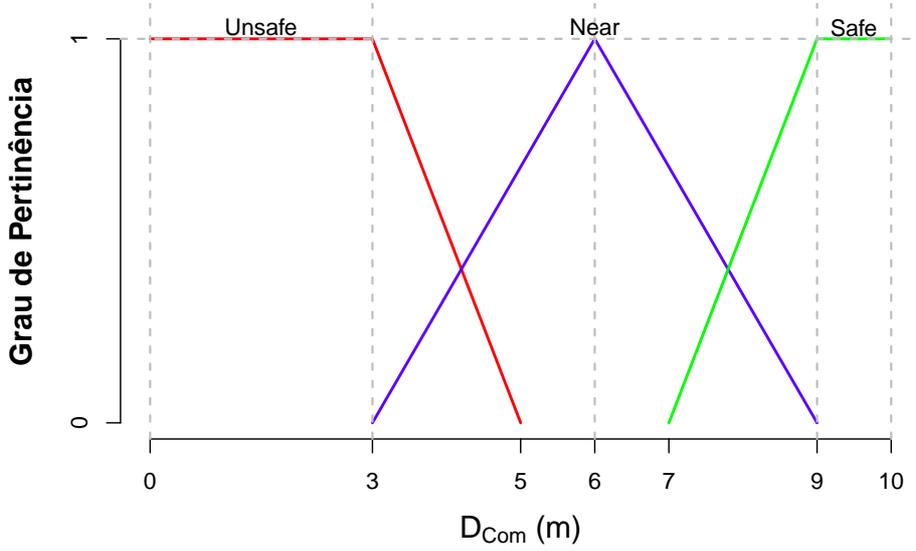


Figura 5: Representação das funções de pertinência *fuzzy* para a variável linguística D_{Col} e seus respectivos conjuntos *Unsafe*, *Near* e *Safe*.

$$f_{DRast-Far}(\vec{d}_{per-ij}(t)) = \begin{cases} 0, & \vec{d}_{per-ij}(t) \leq 3, \\ \frac{\vec{d}_{per-ij}(t)-3}{5} & \vec{d}_{per-ij}(t) \in (3, 8], \\ 1, & \vec{d}_{per-ij}(t) > 8. \end{cases} \quad (3.10)$$

As funções de pertinência da variável linguística D_{Rast} podem ser visualizadas na Figura 6.

O método de cálculo do valor dos coeficientes γ_{col} , γ_{com} e γ_{per} utilizado neste trabalho é determinado pelas saídas de controladores inspirados pelo controlador de Takagi-Sugeno apresentado na Seção 2.3.5.3. Cada controlador é constituído por um conjunto de regras, as quais recebem as variáveis linguísticas como entrada em seus antecedentes, e os seus consequentes são constituídos por equações matemáticas. O controlador do coeficiente de colisão γ_{col} é constituído pelas três regras a seguir:

- a) SE $D_{Col} \in Unsafe$, ENTÃO $y_1 = 1$,
- b) SE $D_{Col} \in Near$, ENTÃO $y_2 = 2 + col_{unsafe}$,
- c) SE $D_{Col} \in Safe$, ENTÃO $y_3 = 0,25 + col_{near}$,

onde col_{unsafe} representa o grau de pertinência da variável D_{Col} para o conjunto de pertinência *Unsafe*, enquanto que col_{safe} representa o grau de pertinência da variável D_{Col} para o conjunto de pertinência *Safe*. O cálculo do coeficiente é então obtido de acordo

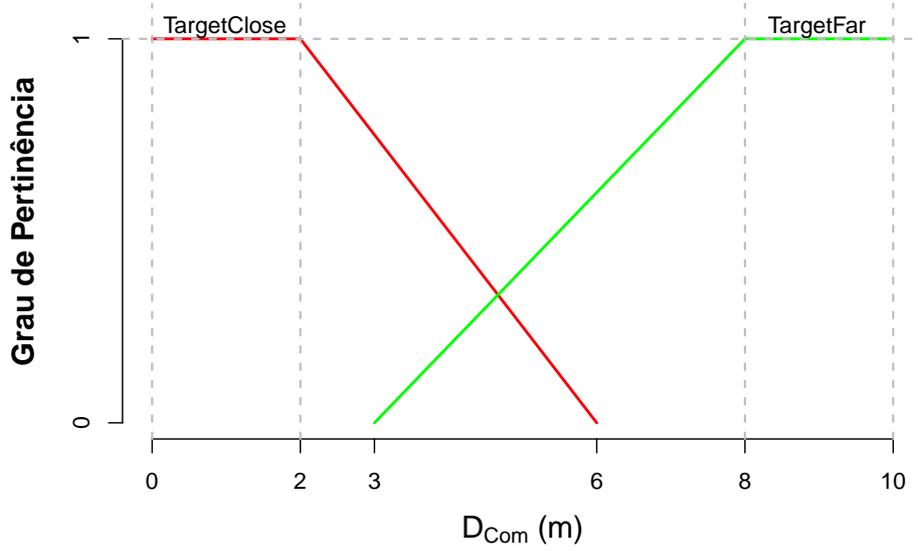


Figura 6: Representação das funções de pertinência *fuzzy* para a variável linguística D_{Rast} e seus respectivos conjuntos *TargetClose* e *TargetFar*.

com a Equação 3.11

$$\gamma_{col} = \frac{\alpha_1 \cdot 1 + \alpha_2 \cdot (2 + col_{unsafe}) + \alpha_3 \cdot (0,25 + col_{near})}{\alpha_1 + \alpha_2 + \alpha_3}, \quad (3.11)$$

onde α_1 , α_2 e α_3 são os valores de disparo das regras a, b e c, respectivamente.

As regras a e b representam a necessidade de desvio com aceleração máxima em distâncias inseguras ou próximas da insegurança do VANT, enquanto que a regra c representa a necessidade de afastamento moderado dado que o VANT encontra-se a uma distância segura.

Para o controlador do coeficiente de comunicação γ_{com} tem-se as regras a seguir:

- a) SE $D_{Com} \in Close$ E $D_{Col} \in Unsafe$, ENTÃO $y_1 = 1 - 2 \cdot com_{close}$,
- b) SE $D_{Com} \in Far$, ENTÃO $y_2 = 1$,
- c) SE $D_{Com} \in Close$ E $D_{Col} \in Near$, ENTÃO $y_3 = com_{close} - 0,25 \cdot col_{near}$,
- d) SE $D_{Com} \in Close$ E $D_{Col} \in Safe$, ENTÃO $y_4 = com_{close} - 0,5 \cdot col_{safe}$,

onde com_{close} representa o grau de pertinência da variável D_{Com} para o conjunto de pertinência *Close*, col_{near} representa o grau de pertinência da variável D_{Col} para o conjunto de pertinência *Near* e col_{safe} representa o grau de pertinência da variável D_{Col} para o conjunto de pertinência *Safe*. O cálculo do coeficiente é então obtido de acordo com a

Equação (3.12)

$$\gamma_{com} = \frac{\alpha_4 \cdot (1 - 2 \cdot com_{close}) + \alpha_5 \cdot 1 + \alpha_6 \cdot (com_{close} - 0,25 \cdot col_{near})}{\alpha_4 + \alpha_5 + \alpha_6 + \alpha_7} + \frac{\alpha_7 \cdot (com_{close} - 0,5 \cdot col_{safe})}{\alpha_4 + \alpha_5 + \alpha_6 + \alpha_7}, \quad (3.12)$$

onde α_4 , α_5 , α_6 e α_7 são os valores de disparo das Regras a, b, c e d, respectivamente.

As regras a representa a necessidade de diminuição da aceleração, dado que o VANT já encontra-se a uma distância de comunicação próxima, porém a distância de colisão é insegura; a regra b representa a necessidade de aproximação com aceleração máxima, dado que o VANT encontra-se a uma distância de comunicação distante; as regras c e d representam a necessidade de aproximação moderada, já que o VANT encontra-se a uma distância de comunicação segura e a distâncias de colisão próximas e seguras, respectivamente.

Finalmente, para o controlador do coeficiente de percepção γ_{per} , temos as seguintes regras:

- a) SE $D_{Rast} \in TargetClose$, ENTÃO $y_1 = 1 - 0,5 \cdot target_{close}$,
- b) SE $D_{Rast} \in TargetFar$, ENTÃO $y_2 = 1$,

onde tar_{close} representa o grau de pertinência da variável D_{Rast} para o conjunto de pertinência $TargetClose$. O cálculo do coeficiente é então obtido de acordo com a Equação (3.13)

$$\gamma_{per} = \frac{\alpha_8 \cdot (1 - 0,5 \cdot target_{close}) + \alpha_9 \cdot 1}{\alpha_8 + \alpha_9} \quad (3.13)$$

onde α_8 e α_9 são os valores de disparo das Regras a e b, respectivamente.

A regra a representa a necessidade de diminuição da aceleração de rastreamento para evitar colisões com outros VANTs que estejam rastreando o mesmo alvo, dado que o VANT encontra-se a uma distância de rastreamento próxima; e a regra b representa a necessidade de aproximação do alvo a uma aceleração de rastreamento máxima, dado que a distância entre o VANT e o alvo é distante.

A cada iteração t durante a simulação, os controladores calculam um novo valor para os coeficientes γ_{col} , γ_{com} e γ_{per} . Como os coeficientes podem assumir valores $\in [0, 1]$, as acelerações $\vec{a}_{col-i}(t)$, $\vec{a}_{com-i}(t)$ e $\vec{a}_{per-i}(t)$ não estão mais sujeitas à máquina de estados do sistema, e sim ao conjunto das circunstâncias na qual o VANT se encontra, representadas pelas regras dos controladores de cada coeficiente. As regras e seus consequentes foram estabelecidos de modo a representar o tipo de atuação desejada pelas acelerações mencionadas, nas situações descritas por cada uma das regras. Os controladores, então, são os responsáveis por especificar com que intensidade as acelerações $\vec{a}_{col-i}(t)$, $\vec{a}_{com-i}(t)$ e $\vec{a}_{per-i}(t)$ serão utilizadas. Este mecanismo de coordenação *fuzzy* é utilizado em conjunto

com o mecanismo de rastreamento inspirado no PSO descrito na Seção 2.4.6, caracterizando um sistema híbrido de coordenação para um enxame de VANTs.

É importante mencionar que os parâmetros de cada variável linguística foram estabelecidos a partir de testes de simulação com o objetivo de maximizar as métricas utilizadas. Apesar de representar satisfatoriamente o comportamento desejado no enxame, as configurações dos parâmetros ainda são bastante sensíveis devido à não-utilização de métodos matemáticos mais sofisticados para estabelecê-los.

3.2 Ferramenta de simulação e métricas de avaliação

Esta Seção tem como objetivo apresentar a ferramenta de simulação utilizada para avaliar o mecanismo proposto e as métricas de avaliação de desempenho do enxame de VANTs utilizadas pela ferramenta.

3.2.1 Ferramenta de simulação de enxames de VANTs

O mecanismo de coordenação proposto neste trabalho foi testado em um simulador de enxames de VANTs desenvolvido em [10]. Este simulador foi construído na linguagem de programação Java. A interface gráfica do simulador pode ser visualizada na Figura 7. A interface gráfica é constituída de dois painéis principais: o painel esquerdo é constituído pelos atributos paramétricos do modelo de simulação, enquanto que o painel direito é constituído pelo ambiente de simulação visual. O ambiente de simulação é o espaço por onde os VANTs e alvos podem trafegar. Os principais componentes do ambiente estão descritos a seguir:

- a) área de extensão: área total do ambiente na qual os VANTs e alvos são permitidos trafegar;
- b) quantidade de VANTs: número máximo de VANTs contidos no ambiente;
- c) quantidade de alvos: número máximo de alvos contidos no ambiente;
- d) base terrestre: posição da base terrestre no ambiente. Neste modelo, a base terrestre está localizada no centro do ambiente.

Durante a execução de uma simulação, além de ser possível observar o comportamento do enxame no painel direito, é possível obter de forma imediata estatísticas de simulação a respeito do desempenho do enxame. As métricas utilizadas para o cálculo dos dados estatísticos são apresentadas na seção a seguir.

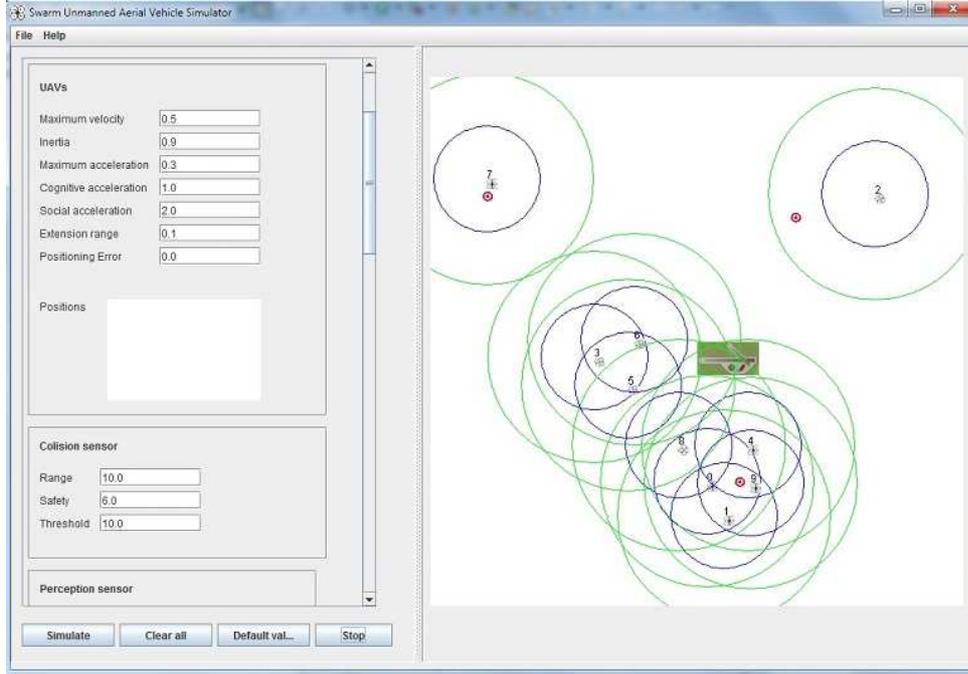


Figura 7: Interface gráfica da ferramenta de simulação para enxames de VANTs.

3.2.2 Métricas de avaliação de desempenho do enxame

Esta Seção introduz as métricas propostas em [10], utilizadas para avaliar o desempenho do mecanismo de coordenação proposto.

3.2.2.1 Taxa de rastreamento de alvos móveis

A taxa de rastreamento de alvos móveis (TT) representa o percentual de alvos rastreados pelos VANTs durante o tempo de simulação. Um alvo j é considerado como rastreado pelo VANT i na iteração t quando a distância entre a posição do alvo, $\vec{x}_{tgt-j}(t)$, e a posição do VANT i , $\vec{x}_{uav-i}(t)$, é menor que o raio de atuação do sensor de percepção, r_{per} . Essa métrica é calculada acordo com a Equação (3.14)

$$TT = \frac{1}{i_{max} \cdot n_{tgt}} \cdot \sum_{t=1}^{i_{max}} \sum_{j=1}^{n_{tgt}} tt_{tgt-j}(t), \quad (3.14)$$

onde i_{max} representa o número máximo de iterações, n_{tgt} representa o número de alvos e $tt_{tgt-j}(t)$ representa se o alvo j na iteração t está sendo rastreado, sendo $tt_{tgt-j}(t) = 1$, ou se não está sendo rastreado, sendo $tt_{tgt-j}(t) = 0$.

Um enxame de VANTs ideal segundo a métrica TT apresentaria uma taxa de rastreamento de 100%.

3.2.2.2 Taxa de colisões entre VANTs

A taxa de colisões representa o percentual de colisões entre VANTs ocorridas durante a simulação. Essa métrica é calculada de acordo com a Equação (3.15):

$$CL = \frac{1}{i_{max} \cdot n_{uav}} \sum_{t=1}^{i_{max}} n_{uav-crt}(t), \quad (3.15)$$

onde i_{max} representa o número máximo de iterações, n_{uav} representa o número máximo de VANTs e $n_{uav-crt}(t)$ representa o número total de VANTs que colidiram até a iteração t .

Um enxame de VANTs ideal segundo a métrica CL apresentaria uma taxa de colisões de 0%.

3.2.2.3 Taxa de coesão do enxame

A taxa de coesão representa a proporção de conexão do enxame de VANTs. Essa métrica é calculada de acordo com a Equação (3.16):

$$COE = \frac{1}{i_{max} \cdot n_{uav}} \sum_{t=1}^{i_{max}} n_{uav-con}(t), \quad (3.16)$$

onde i_{max} é o número máximo de iterações, n_{uav} é o número máximo de VANTs e $n_{uav-con}(t)$ é o número de VANTs que estão com o número máximo de conexões estabelecidas na iteração t .

Um enxame de VANTs ideal segundo a métrica COE apresentaria uma taxa de coesão de 100%.

3.2.2.4 Área de cobertura territorial

A métrica patrulha territorial (COA) é calculada a partir da divisão da área do ambiente em a_n sub-áreas quadradas de tamanho a_i , cujo lado é igual ao diâmetro do sensor de percepção do VANT ($2r_{per}$). Todas as sub-áreas são inicializadas como sub-áreas não visitadas ($coa_i(t) = 0$). Dessa forma, se a distância entre o VANT e o centro de qualquer sub-área for menor ou igual ao raio de alcance do sensor de percepção, então a sub-área é marcada como visitada. Ao final, a soma das sub-áreas marcadas pelos VANTs é dividida pela quantidade total de sub-áreas do ambiente conforme a Equação (3.17):

$$COA = \frac{1}{a_n} \cdot \sum_{t=1}^{a_n} coa_{a-i}(t), \quad (3.17)$$

em que a_n é a quantidade total de sub-áreas e $coa_{a-i}(t)$ representa se a sub-área i foi visitada, $coa_{a-i}(t) = 1$, ou não foi visitada, $coa_{a-i}(t) = 0$, na iteração t .

Contudo, é necessário definir o número de iterações segundo o qual as sub-áreas serão inicializadas novamente como sub-áreas não visitadas. Ou seja, COA é recalculada a cada i_{coa} iterações.

Um enxame de VANTs ideal segundo a métrica COA apresentaria uma taxa de cobertura territorial de 100%.

Apesar de ser uma métrica importante para avaliar o desempenho do enxame, COA foi desativada durante as simulações para este trabalho devido à complexidade computacional significativa exigida por esta métrica.

4 Resultados

Este capítulo tem como objetivo apresentar os experimentos utilizados e os resultados obtidos para o modelo proposto. A Seção 4.1 apresenta a análise da influência do modelo proposto em comparação com o modelo descrito na Seção 2.4 e proposto em [10]. A Seção 4.2 apresenta a análise da influência de erros de posicionamento tanto no desempenho do modelo proposto como no modelo de coordenação que utiliza apenas a máquina de estados.

4.1 Análise de desempenho do modelo híbrido Swarm-Fuzzy

O objetivo desta análise é avaliar os impactos causados pelo modelo proposto no desempenho do enxame. Foram realizadas 30 simulações para cada conjunto experimental. As métricas utilizadas foram: taxa de colisão (CL), taxa de rastreamento (TT) e taxa de coesão (COE).

Em cada conjunto de simulações foi analisado o desempenho de 10, 20 e 40 VANTs em ambientes com 1, 3 e 5 alvos móveis. A ferramenta de simulação utilizada foi a mesma descrita na Seção 3.2.1. Os parâmetros utilizados para todas as simulações encontram-se na Tabela 2. A Figura 8 apresenta os gráficos com as taxas de colisão para o modelo de coordenação utilizando apenas a máquina de estados e para o modelo proposto neste trabalho. A Figura 9 apresenta os gráficos em relação à taxa de rastreamento destes dois modelos de coordenação, enquanto que a Figura 10 ilustra os gráficos em relação à taxa de coesão apresentada por estes modelos.

Na Figura 8 é possível observar uma redução na quantidade de colisões para 0%, com a presença de *outliers*, em comparação com o modelo utilizando apenas máquina de estados em um ambiente com 10 ou 20 VANTs. Para um ambiente com 40 VANTs observa-se um aumento na quantidade de colisões até mesmo no modelo proposto. Entretanto, é importante mencionar que a área do ambiente não foi aumentada para comportar a maior quantidade de VANTs utilizados. Ainda assim, enquanto que o modelo utilizando apenas máquina de estados dobrou a média de colisões de 10% para 20% com o aumento de 20 para 40 VANTs, o modelo híbrido proposto apresentou um aumento na taxa de colisões de, em média, 5%. É importante mencionar que para que os resultados de colisão pudessem ser mantidos neste nível, foi permitido que o coeficiente de colisão atingisse níveis duas vezes mais altos que os outros coeficientes em ambientes com 20 ou mais VANTs. Isso permitiu que a aceleração de colisão atingisse valores suficientes que permitiram evitar conflitos entre os VANTs.

Tabela 2: Parâmetros utilizados nas simulações relacionadas às análises de desempenho do modelo híbrido Swarm-Fuzzy.

Parâmetro	Descrição	Valor
t_{max}	Número máximo de iterações t	3600
env	Área do ambiente	10.000 m^2
$v_{uav-max}$	Velocidade máxima do VANT	0,5 m/s
$a_{uav-max}$	Aceleração máxima do VANT	0,3 m/s^2
e_{uav}	Raio de extensão do VANT	0,1 m
w	Peso inercial	0,9
a_{cog}	Aceleração cognitiva	1,0
a_{soc}	Aceleração social	2,0
$n_{nei-max}$	Número máximo de vizinhos conectados	2
r_{col}	Raio de alcance do sensor anti-colisão	10 m
t_{col}	Limiar do sensor anti-colisão	10 m
s_{col}	Segurança do sensor anti-colisão	6 m
r_{prc}	Raio de alcance do sensor de percepção	10 m
t_{prc}	Limiar do sensor de percepção	10 m
r_{com}	Raio de alcance do sensor de comunicação	20 m
l_{com}	Limiar do sensor de comunicação	0 m
s_{com}	Segurança do sensor de comunicação	7 m
$v_{tgt-max}$	Velocidade máxima dos alvos	0,5 m/s
$a_{tgt-max}$	Aceleração máxima dos alvos	0,3 m/s^2
t_{coa}	Número de iterações para calcular área de cobertura	60

A partir dos resultados contidos na Figura 9 pode-se observar que a taxa de rastreamento de alvos não apresentou mudanças expressivas entre os dois modelos de coordenação. É importante salientar que, apesar de manter taxas de rastreamento satisfatórias para 1, 3 e 5 alvos, o modelo utilizando apenas máquina de estados pode ter seu desempenho de rastreamento prejudicado em um ambiente com uma quantidade maior de alvos devido ao fato de possuir taxas mais elevadas de colisão.

Na Figura 10 é possível observar que o modelo híbrido proposto apresentou um aumento de em média 10% em comparação com o modelo anterior, podendo chegar até 25% em um ambiente com 10 VANTs. É importante mencionar que o modelo proposto é capaz de manter uma maior conectividade entre os VANTs com quantidade mais reduzida de colisões mesmo com o aumento de VANTs no ambiente, o que indica um comportamento de sincronismo mais estável que o do modelo utilizando apenas máquina de estados.

4.2 Análise da influência de erros de posicionamento

Como descrito na Seção 2.4.2, o mecanismo de localização permite que seja adicionada a influência da imprecisão dos sensores de localização no sensor de localização do VANT. O objetivo desta análise é avaliar os impactos dos ruídos no desempenho do exame tanto no modelo proposto quanto no modelo utilizando apenas máquina de estados.

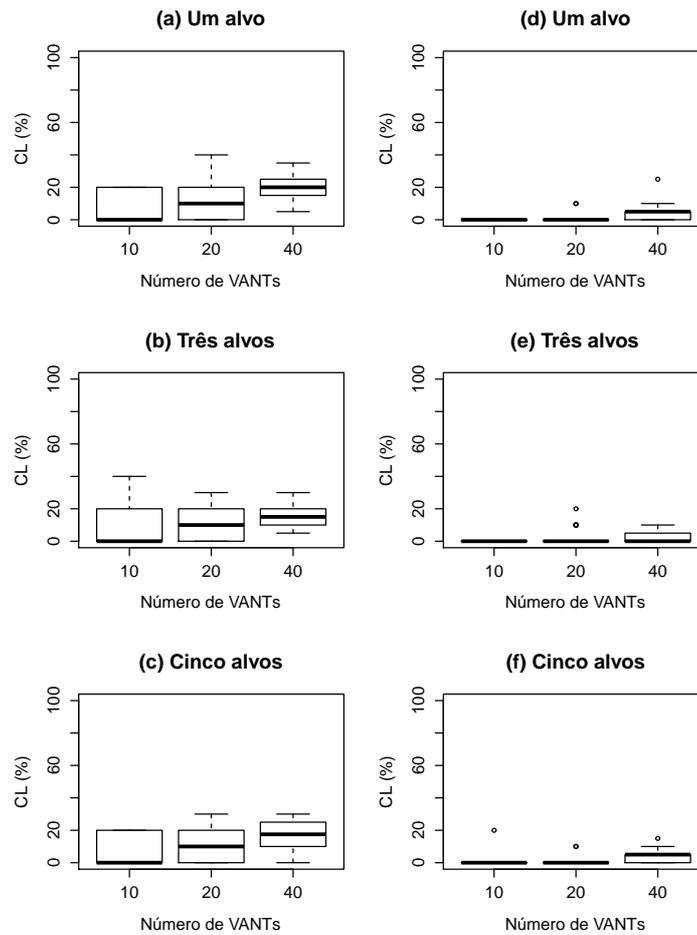


Figura 8: *Boxplots* das taxas de colisão (CL) para um, três e cinco alvos em ambientes com 10, 20 e 40 VANTs. Os *boxplots* (a), (b) e (c) correspondem aos resultados obtidos com o modelo utilizando apenas máquina de estados; os *boxplots* (d), (e) e (f) correspondem aos resultados obtidos utilizando o modelo híbrido.

As métricas utilizadas foram: taxa de colisão (CL), taxa de rastreamento (TT) e taxa de coesão (COE).

Em cada conjunto de simulações foi analisado o desempenho de 10, 20 e 40 VANTs em ambientes com 1, 3 e 5 alvos móveis. Os valores de erro de posicionamento utilizados para o ruído foram: 0; 0,1; 0,2; 0,25; 0,3; 0,5; 1; 3; 5 e 10 metros. Foram realizadas 30 simulações para cada conjunto experimental. A ferramenta utilizada e os parâmetros de simulação foram os mesmos utilizados nas simulações descritas na Seção 4.1. As Figuras 11, 13, 15, 17, 19 e 21 ilustram os gráficos com as taxas de colisão, rastreamento e coesão para o modelo de coordenação utilizando apenas máquina de estados. As Figuras 12, 14, 16, 18, 20 e 22 apresentam as taxas de colisão, rastreamento e coesão para o modelo proposto neste trabalho.

Na Figura 11 é possível observar taxas de colisões de até 20% para erros de posicionamento com valores de até 0,5 metros, e de uma média de 60% para erros de posicio-

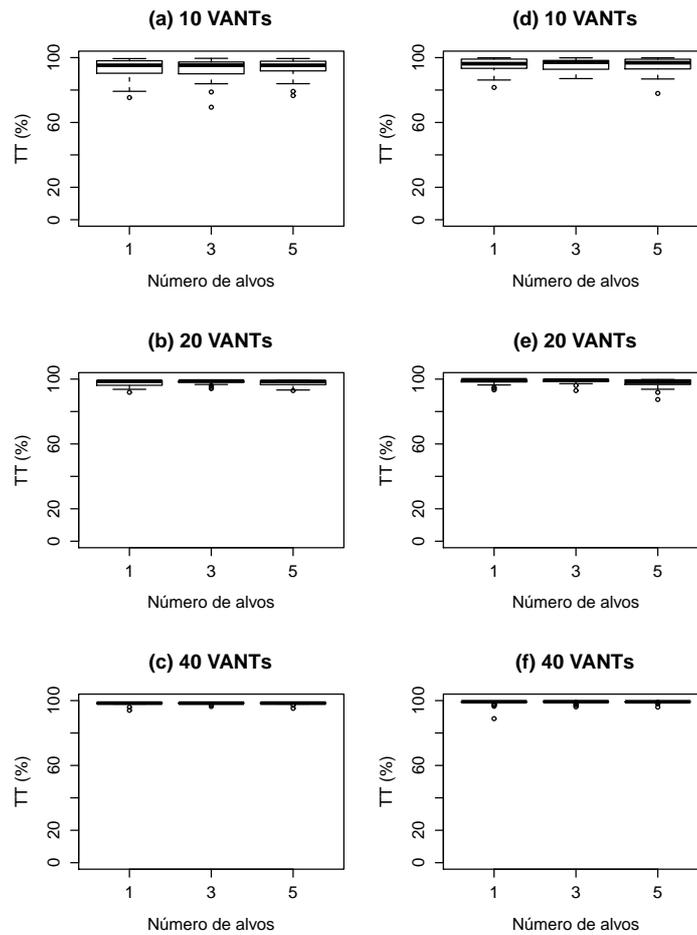


Figura 9: *Boxplots* das taxas de rastreamento (TT) para 10, 20 e 40 VANTs em ambientes com um, três e cinco alvos. Os *boxplots* (a), (b) e (c) correspondem aos resultados obtidos com o modelo utilizando máquina de estados; os *boxplots* (d), (e) e (f) correspondem aos resultados obtidos utilizando o modelo híbrido.

namento maiores que 1 metro em ambientes com 1, 3 e 5 alvos, com uma pequena queda para erros a partir de 5 metros. O mesmo tipo de comportamento pode ser observado para 20 VANTs nas Figura 13, porém é importante mencionar um aumento para 20% na média de colisões para valores de erro de posicionamento de até 0,5 metros. Na Figura 12, por sua vez, a utilização do modelo proposto provoca uma queda para 0% com alguns *outliers* para erros de posicionamento até 0,5 metros, e para uma média de 40% para valores a partir de 1 metro. Na Figura 14 é possível observar que, apesar do aumento do número de VANTs para 20, a taxa de colisões mantém-se em 0% na maioria das simulações com erro de posicionamento de até 0,5 metros.

Nas Figuras 15 e 17 observa-se que as taxas de rastreamento mantêm-se em valores razoáveis para erros de posicionamento de até 0,5 metros, e de maneira ainda mais acentuada para ambientes com 20 VANTs. A Figura 16 mostra que já é possível obter resultados melhores no modelo proposto utilizando apenas 10 VANTs, com erros de po-

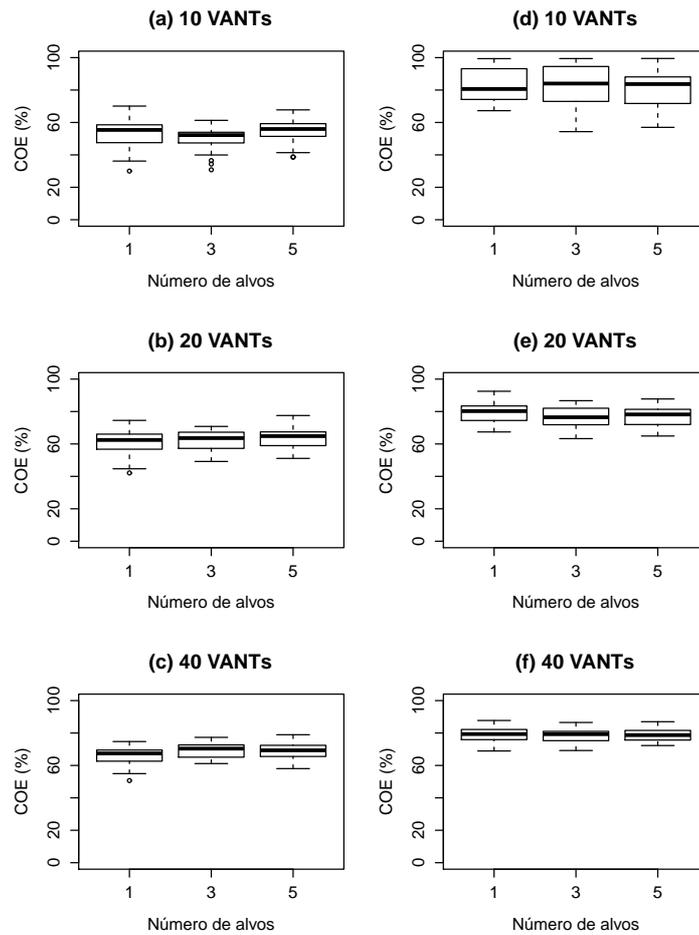


Figura 10: *Boxplots* das taxas de coesão (COE) para um, três e cinco alvos em ambientes com 10, 20 e 40 VANTs. Os *boxplots* (a), (b) e (c) correspondem aos resultados obtidos com o modelo utilizando apenas máquina de estados; os *boxplots* (d), (e) e (f) correspondem aos resultados obtidos utilizando o modelo híbrido.

sicionamento de até 0,5 metros. Os resultados repetem-se na Figura 18, onde os valores tornam-se ainda mais acentuados com uma média próxima de 100% de rastreamento para erros de posicionamento de até 0,5 metros. Entretanto, uma queda mais brusca pode ser observada ainda na Figura 18 para erros de posicionamento maiores que 0,5.

Nas Figuras 19 e 21 as taxas de coesão mantêm-se, em média, entre 55% e 60%, com uma queda neste valor a partir de erros de posicionamento de 0,5 metros. Nas Figuras 20 e 22, por sua vez, é possível observar que as taxas de coesão mantêm-se, em média, sempre acima de 70% para valores com erro de posicionamento de até 0,5 metros, o que confirma a maior robustez do modelo proposto na manutenção da conexão entre os indivíduos do enxame.

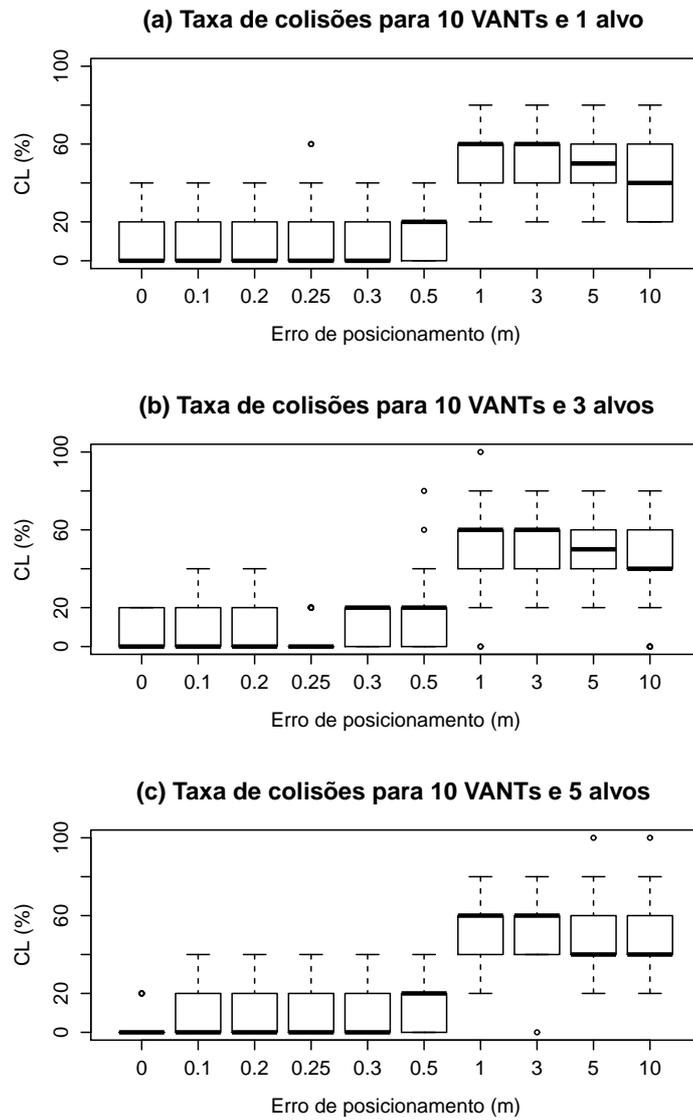


Figura 11: *Boxplots* das taxas de colisão (CL) para (a) um, (b) três e (c) cinco alvos em ambientes com 10 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo de máquina de estados.

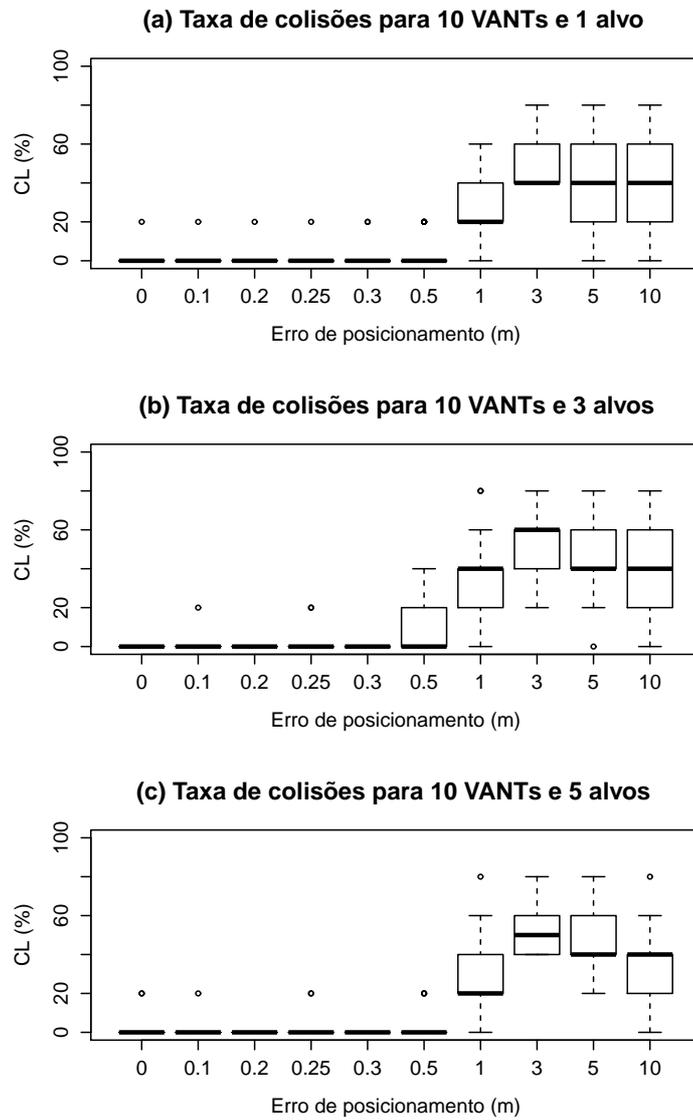


Figura 12: *Boxplots* das taxas de colisão (CL) para (a) um, (b) três e (c) cinco alvos em ambientes com 10 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo híbrido *Swarm-Fuzzy*.

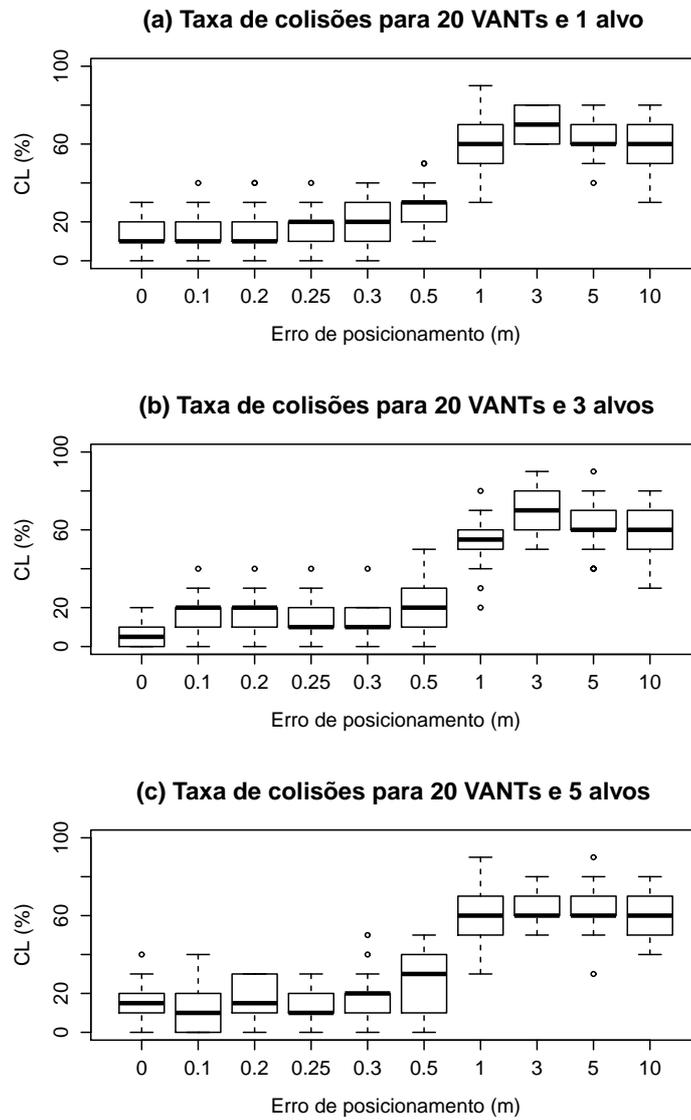


Figura 13: *Boxplots* das taxas de colisão (CL) para (a) um, (b) três e (c) cinco alvos em ambientes com 20 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo utilizando apenas máquina de estados.

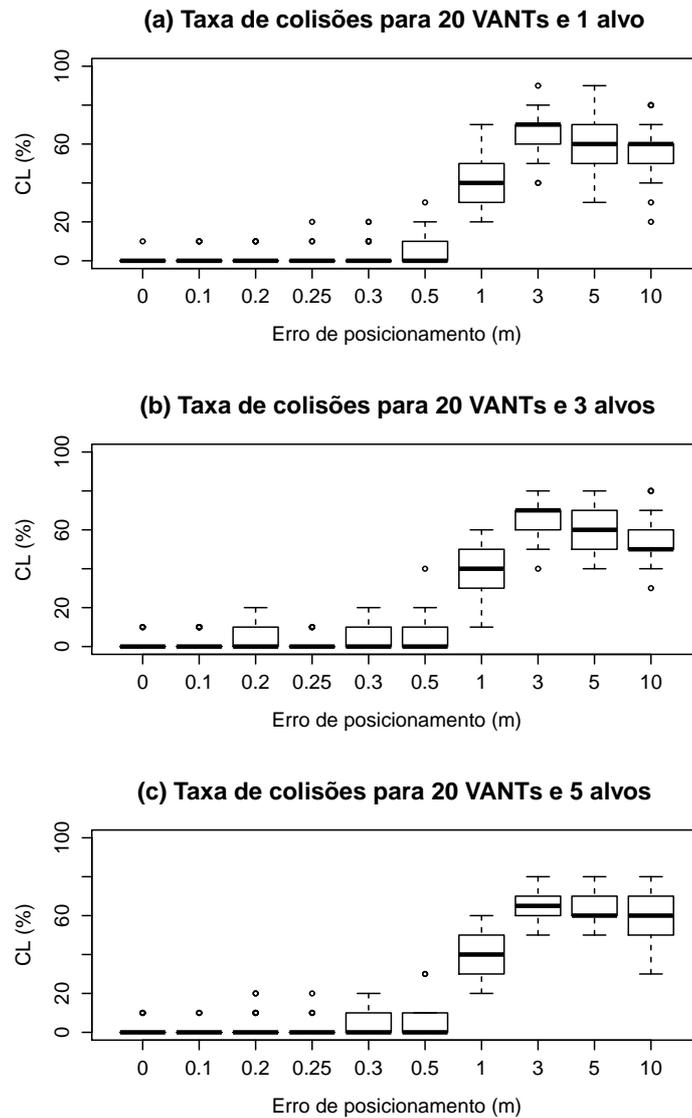


Figura 14: *Boxplots* das taxas de colisão (CL) para (a) um, (b) três e (c) cinco alvos em ambientes com 20 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo híbrido *Swarm-Fuzzy*.

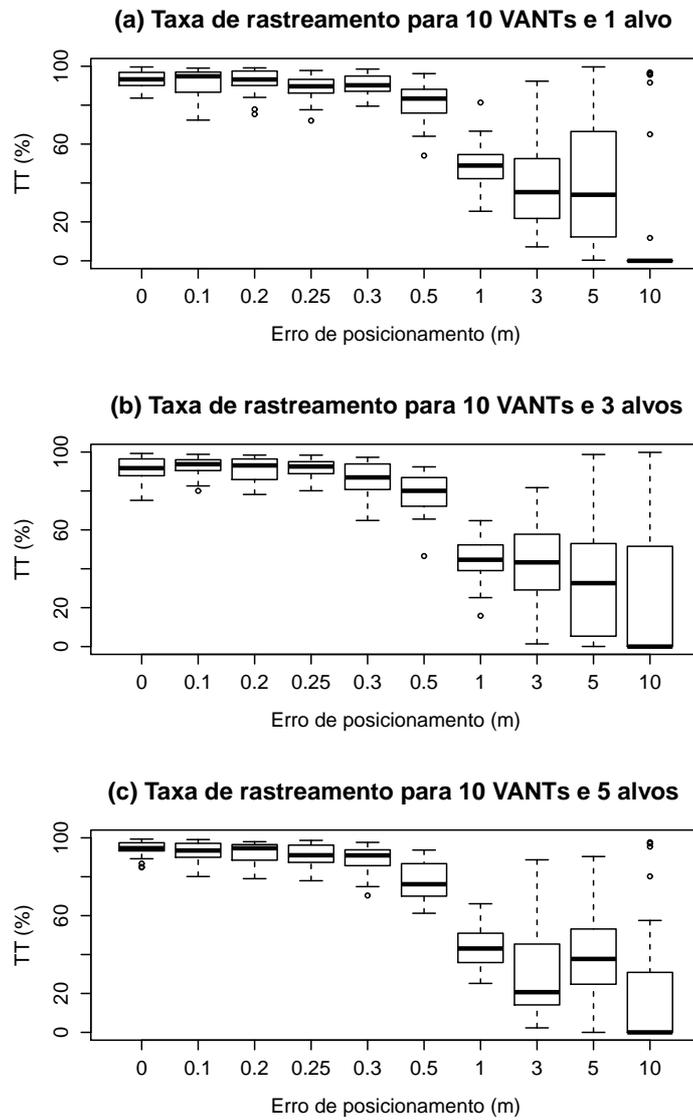


Figura 15: *Boxplots* das taxas de rastreamento (TT) para (a) um, (b) três e (c) cinco alvos em ambientes com 10 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo utilizando apenas máquina de estados.

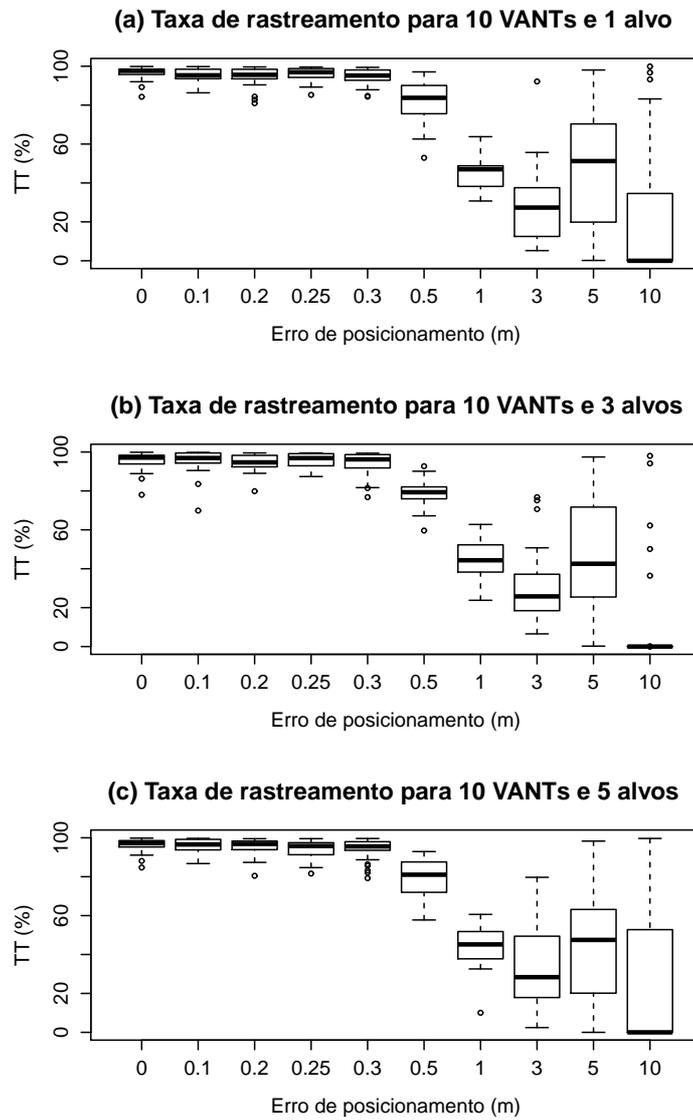


Figura 16: *Boxplots* das taxas de rastreamento (TT) para (a) um, (b) três e (c) cinco alvos em ambientes com 10 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo híbrido *Swarm-Fuzzy*.

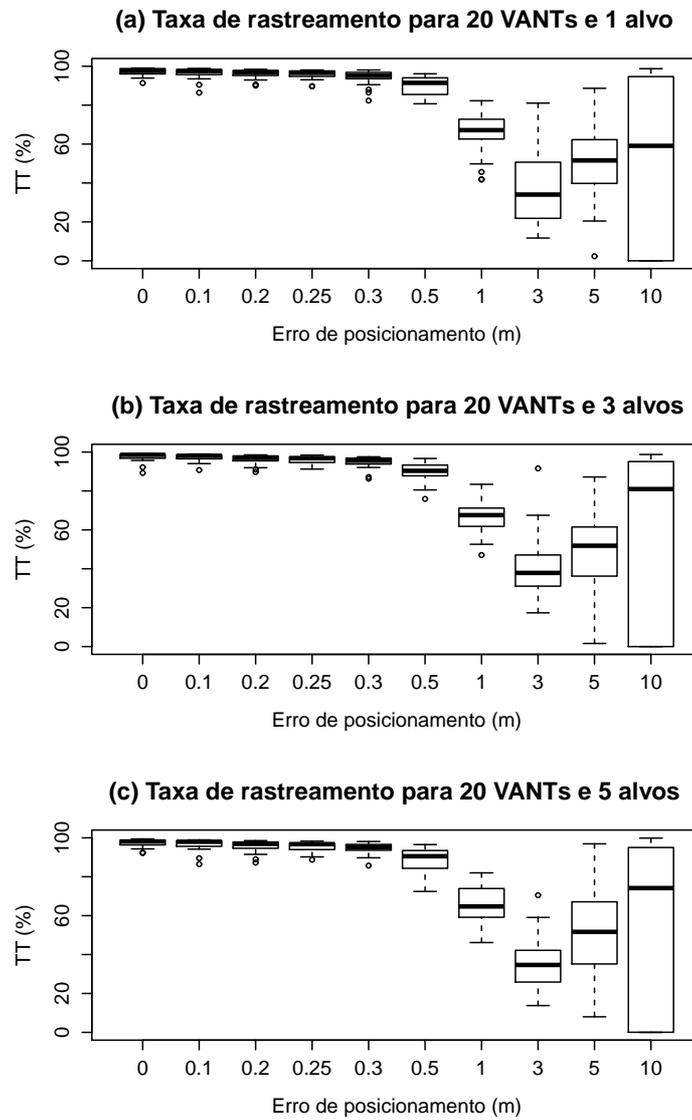


Figura 17: *Boxplots* das taxas de rastreamento (TT) para (a) um, (b) três e (c) cinco alvos em ambientes com 20 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo utilizando apenas máquina de estados.

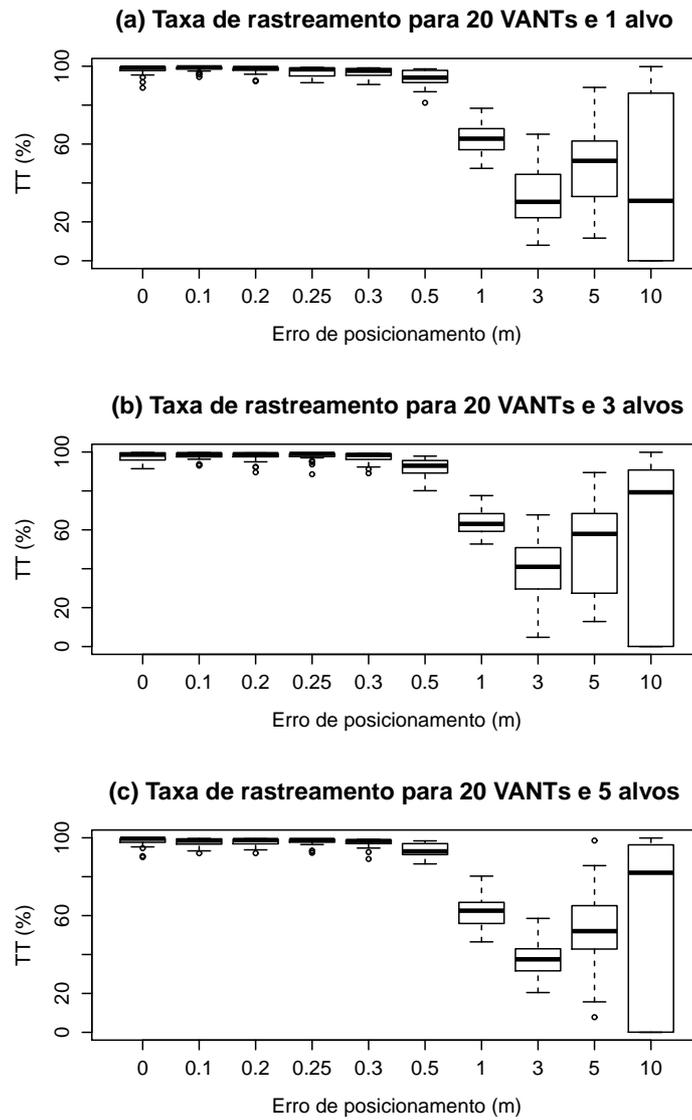


Figura 18: *Boxplots* das taxas de rastreamento (TT) para (a) um, (b) três e (c) cinco alvos em ambientes com 20 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo híbrido *Swarm-Fuzzy*.

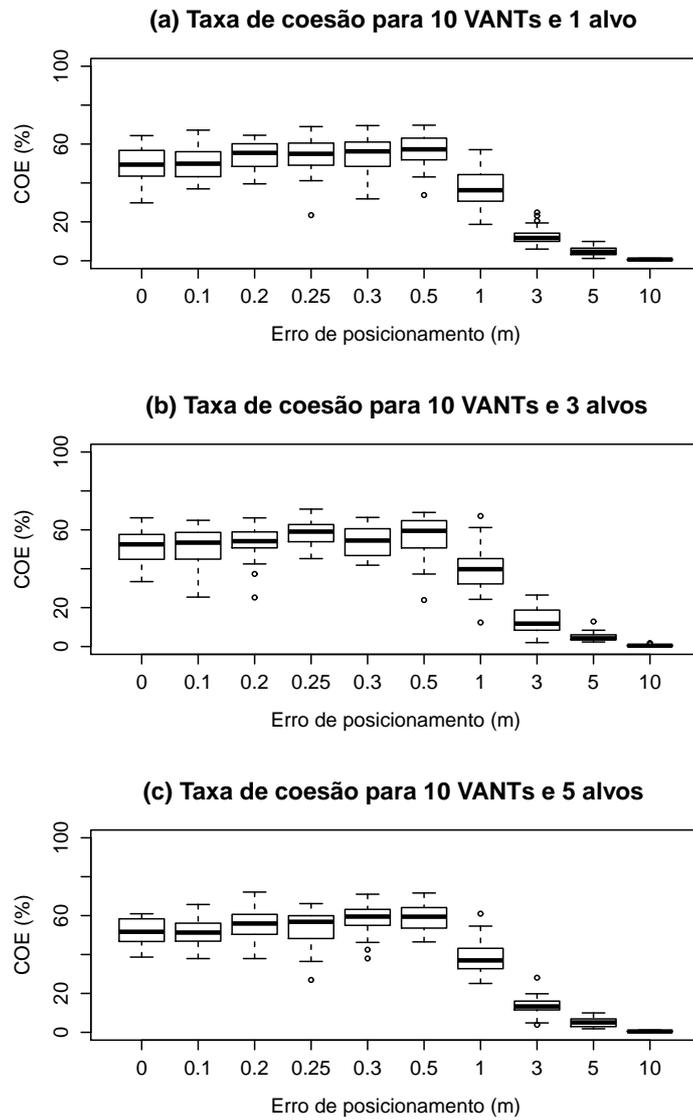


Figura 19: *Boxplots* das taxas de coesão (COE) para (a) um, (b) três e (c) cinco alvos em ambientes com 10 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo utilizando apenas máquina de estados.

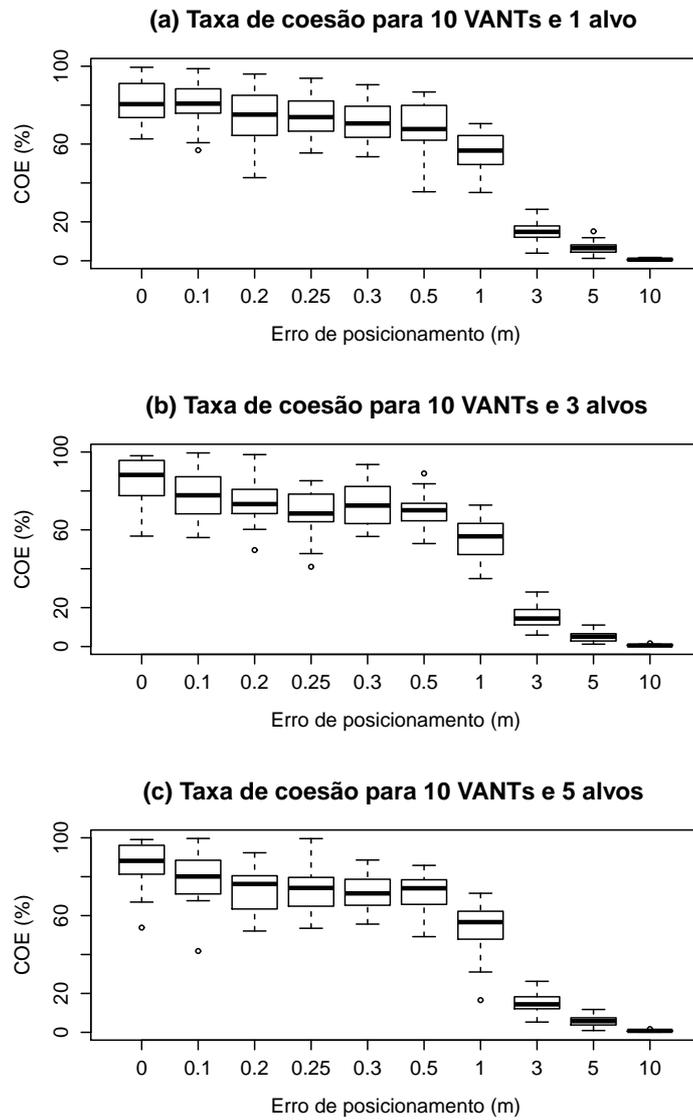


Figura 20: *Boxplots* das taxas de coesão (COE) para (a) um, (b) três e (c) cinco alvos em ambientes com 10 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo híbrido *Swarm-Fuzzy*.

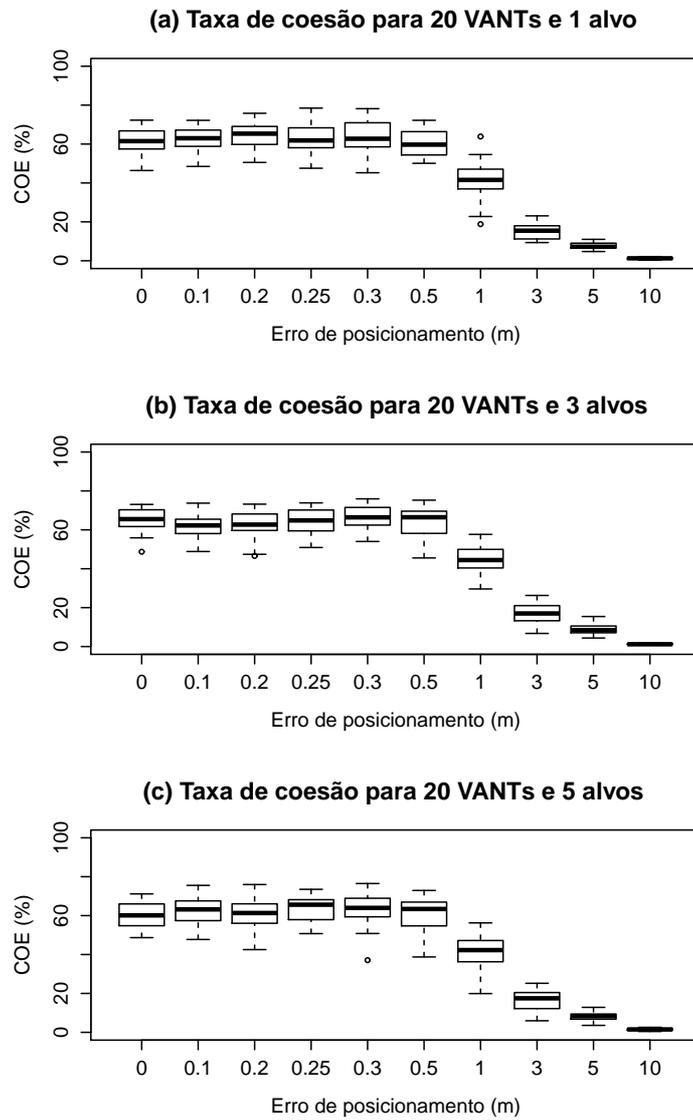


Figura 21: *Boxplots* das taxas de coesão (COE) para (a) um, (b) três e (c) cinco alvos em ambientes com 20 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo utilizando apenas máquina de estados.

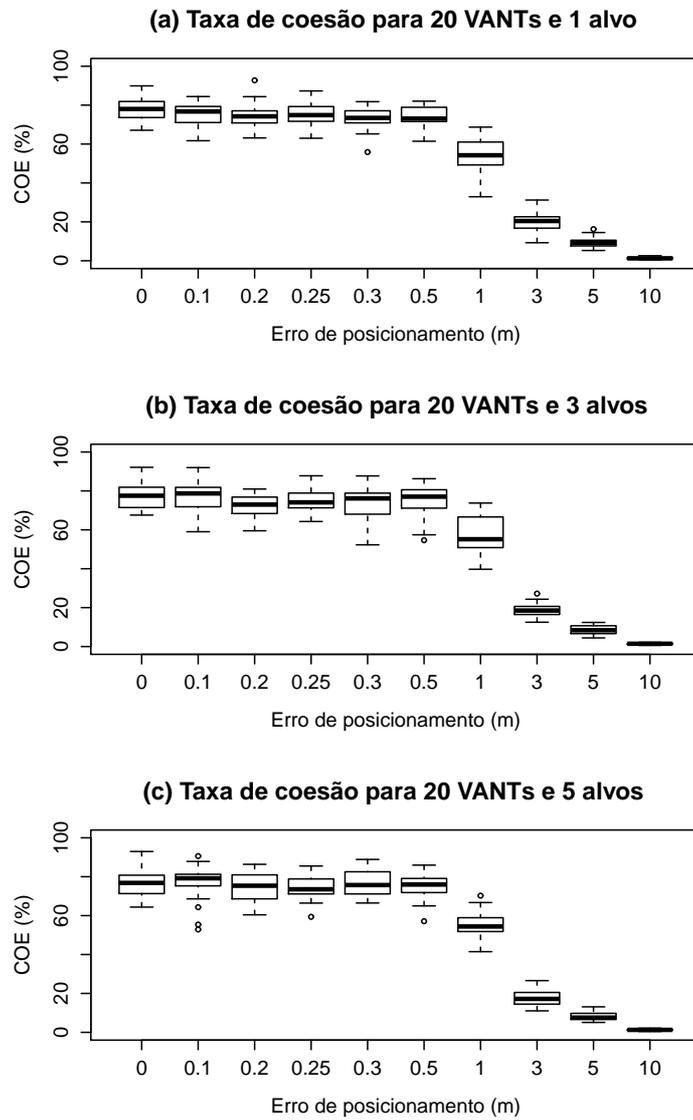


Figura 22: *Boxplots* das taxas de coesão (COE) para (a) um, (b) três e (c) cinco alvos em ambientes com 20 VANTs e erro de posicionamento variando de 0 a 10 metros. Resultados correspondentes ao modelo híbrido *Swarm-Fuzzy*.

5 Conclusões e trabalhos futuros

Neste trabalho de conclusão de curso foi proposto um modelo de coordenação inspirado no controlador *fuzzy* de Takagi-Sugeno para o cálculo de coeficientes de acelerações que antes eram obtidos através de uma máquina de estados. A utilização de controladores *fuzzy* permite que o VANT não esteja mais sujeito à variação dos estados impostos pela máquina de estados, mas sim pelas circunstâncias externas que podem interferir no grau de ativação dos coeficientes. Além do mais, a utilização de um controlador *fuzzy* de Takagi-Sugeno torna o modelo mais próximo de um sistema real, onde sua atuação não é caracterizada pela transição de seus estados, mas pelo conjunto de contextos ao qual o sistema encontra-se inserido.

Os experimentos realizados tiveram como objetivo analisar o desempenho do enxame em termos de colisões, rastreamento de alvos móveis e coesão do enxame em diferentes cenários em que se variou a quantidade tanto de VANTs quanto de alvos. Também foi analisado o desempenho do enxame sob a interferência, em diferentes níveis, de erros de posicionamento no mecanismo de localização do VANT.

Em todos os experimentos realizados, o modelo proposto provou-se mais robusto e estável que o modelo de máquinas de estados, destacando-se uma queda considerável nas taxas de colisão e um aumento nas taxas de coesão do enxame. Apesar de apresentar taxas de rastreamento equiparáveis, o modelo de máquinas de estado pode ter seu desempenho penalizado em ambientes com uma quantidade maior de alvos devido às elevadas taxas de colisão que apresenta.

O modelo proposto apresentou um desempenho superior com relação ao modelo de máquinas de estados nas análises de erro de posicionamento. Isto se deve ao fato de o modelo híbrido proposto conseguir reagir de uma forma mais dinâmica às imprevisibilidades do que o modelo utilizando máquinas de estados, e desta maneira garante a robustez na coordenação e sincronismo do enxame. Entretanto, é importante mencionar que, para a manutenção do bom desempenho do enxame em ambientes com um número de VANTs maior ou igual a 20, foi necessário permitir que o coeficiente de colisão γ_{col} atingisse até duas vezes mais que os outros coeficientes de modo que a aceleração de colisão $\vec{a}_{col-i}(t)$ pudesse ser suficiente para evitar um possível conflito entre VANTs.

Em trabalhos futuros, os seguintes requisitos podem ser destacados:

- a) implementar um mecanismo de controle *fuzzy* semelhante para outros coeficientes do modelo de coordenação e assim eliminar por completo a influência da máquina de estados do modelo;

- b) avaliar o desempenho do modelo proposto considerando o consumo energético do VANT e a possibilidade de carga e descarga de bateria;
- c) avaliar o desempenho do modelo proposto em cenários específicos que exijam a contração e a expansão do enxame;
- d) implementar erros e sinais de ruído em outros sensores do VANT, além do sensor de localização;
- e) implementar um protocolo de comunicação mais robusto que se aproxime de protocolos já existentes.

Referências

- 1 KIM, M. et al. A system-of-systems framework for the improved capability of insurgent tracking missions involving unmanned aerial vehicles. IEEE, Loughborough, Reino Unido, p. 1–6, junho 2010.
- 2 DENG, Z.; MA, C.; ZHU, M. A reconfigurable flight control system architecture for small unmanned aerial vehicles. IEEE, Vancouver, Canadá, p. 1–4, março 2012.
- 3 FLYNN, E. Low-cost approaches to uav design using advanced manufacturing techniques. IEEE, Princeton, Estados Unidos, p. 1–4, março 2013.
- 4 GADE, S.; JOSHI, A. Heterogeneous uav swarm system for target search in adversarial environment. IEEE, Thiruvananthapuram, Índia, p. 358–363, dezembro 2013.
- 5 FORTMANN, F.; LUDTKE, A. An intelligent sa-adaptive interface to aid supervisory control of a uav swarm. IEEE, Bochum, Alemanha, p. 768–773, julho 2013.
- 6 TEACY, W. et al. Maintaining connectivity in uav swarm sensing. IEEE, Miami, Estados Unidos, p. 1771–1776, dezembro 2010.
- 7 AL, M. D. et. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *Robotics Automation Magazine, IEEE*, v. 20, n. 4, p. 60–71, dezembro 2013.
- 8 ZADEH, L. A. International conference on acoustics, speech, and signal processing. IEEE, Adelaide, Austrália, p. 1520–6149, abril 1994.
- 9 TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-15, n. 1, p. 116–132, 1985.
- 10 SILVA, D. M. P. F. *Coordenação de veículos aéreos não Tripulados utilizando inteligência de enxames*. 2012, 83 p. Dissertação de Mestrado - Engenharia de Sistemas, UPE, Recife-PE.
- 11 ENGELBRECHT, A. P. *Computational Intelligence: An Introduction*. 2. ed. [S.l.]: John Wiley & Sons, 2007. 628 p.
- 12 ZHU, Y.-f.; TANG, X.-m. Overview of swarm intelligence. IEEE, Guangzhou, China, p. 400–403, outubro 2010.
- 13 DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, v. 1, n. 4, p. 28–39, 2006.
- 14 HEDAYATZADEH, R. et al. Termite colony optimization: A novel approach for optimizing continuous problems. IEEE, Isfahan, Irã, p. 553–558, maio 2010.
- 15 KARABOGA, D. An idea based on honey bee swarm for numerical optimization. *TECHNICAL REPORT-TR06, Erciyes University, Engineering Faculty*, v. 1, 2005.
- 16 BASTOS-FILHO, C. et al. On the influence of the swimming operators in the fish school search algorithm. IEEE, San Antonio, Texas, p. 5012–5017, outubro 2009.

- 17 KENNEDY, J.; EBERHART, R. Particle swarm optimization. IEEE, Perth, Austrália, v. 4, p. 1942–1948, 1995.
- 18 KENNEDY, J.; EBERHART, R.; SHI, Y. *Swarm Intelligence*. 1. ed. San Francisco, Califórnia, USA: Morgan Kaufmann Publishers, 2001.
- 19 KENNEDY, J. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. v. 3, p. 1931–1938, julho 1999.
- 20 EBERHART, R.; SHI, Y. *Computational Intelligence: concepts to implementations*. Burlington, EUA: Elsevier Inc, 2007.
- 21 WEIXING, F. et al. Novel algorithms for coordination of underwater swarm robotics. IEEE, Luoyang, China, v. 3, p. 654–659, junho 2006.
- 22 BRAMBILLA, M.; DORIGO, M.; BIRATTARI, M. Property-driven design for robot swarms: A design method based on prescriptive modeling and model checking. *IRIDIA - Technical Report Series*, n. 3, janeiro 2014.
- 23 AL, M. dorigo et. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *Robotics & Automation Magazine, IEEE*, v. 20, n. 4, dezembro 2013.
- 24 DORIGO, M.; BIRATTARI, M.; BRAMBILLA, M. *Swarm robotics*. Bruxelas, Bélgica, 2014. Disponível em: <http://www.scholarpedia.org/article/Swarm_robotics>.
- 25 SAHIN, E. Swarm robotics: From sources of inspiration to domains of application. Springer Berlin Heidelberg, v. 3342, p. 10–20, janeiro 2005.
- 26 LIN, J. et al. Practical application of unmanned aerial vehicles for mountain hazards survey. IEEE, Chengdu, China, p. 1–5, junho 2010.
- 27 KIM, M.; PUCHATY, E.; SEVCOVIC, M. A system-of-systems framework for the improved capability of insurgent tracking missions involving unmanned aerial vehicles. IEEE, West Lafayette, Estados Unidos, p. 1–6, junho 2010.
- 28 JOORDENS, M. et al. Applications and prototype for system of systems swarm robotics. IEEE, San Antonio, Estados Unidos, p. 2049–2055, outubro 2008.
- 29 YUNLONG, Z.; JIANCHAO, X. S. Z.; JING, D. Target position estimation aided swarm robotic search under conditions of relative localization mechanism. IEEE, Taiyuan, China, p. 183–186, julho 2012.
- 30 ISERMANN, R. On fuzzy logic applications for automatic control, supervision, and fault diagnosis. IEEE, Alemanha, p. 221–235, março 1998.
- 31 MAMDANI, E.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, v. 7, n. 1, p. 1–13, 1975.