

# Simulação Virtual da Evolução de Estratégias e do Controle Inteligente em Sistemas Multi-Robóticos

Gustavo Pessin, Fernando Osório  
Unisinos – PIPCA  
PPG em Computação Aplicada  
Av. Unisinos, 950  
São Leopoldo, RS - Brasil  
pessin@gmail.com, fosorio@gmail.com

Soraia Musse  
PUC-RS – FACIN  
Faculdade de Informática  
Av. Ipiranga, 6681  
Porto Alegre, RS - Brasil  
soraia.musse@pucrs.br

## Resumo

*O objetivo deste artigo é detalhar o projeto e o desenvolvimento de um sistema multi-agente que opera em um ambiente virtual de simulação realística<sup>1</sup>. Neste sistema, uma equipe de agentes autônomos trabalha cooperativamente a fim de realizar com sucesso a identificação e o combate de incêndios em áreas florestais, sem intervenção humana. O ambiente suporta uma série de características fundamentais para a simulação realística da operação, como terrenos irregulares, processos naturais e restrições físicas na criação e uso de robôs móveis. A operação multi-agente depende essencialmente de duas etapas: planejamento e ação. No planejamento, usamos Algoritmos Genéticos para evoluir estratégias de posicionamento de atuação dos robôs bombeiros. Para a ação foram criados robôs de combate fisicamente simulados, sendo as informações sensoriais de cada robô (e.g. GPS, bússola, sonar) usadas na entrada de uma Rede Neural Artificial (RNA). Esta RNA que controla os atuadores do veículo permite navegação com desvio de obstáculos. Os resultados das simulações demonstram que a Rede Neural controla satisfatoriamente os robôs móveis, que o uso de Algoritmos Genéticos configura a estratégia de combate ao incêndio de modo satisfatório e que o sistema multi-agente proposto pode vir a ter um papel muito importante no planejamento e execução de operações reais de combate a incêndios florestais.*

## 1. Introdução

Com a evolução das pesquisas em robótica, cada vez mais os robôs estão se tornando complexos em termos físicos. A grande variedade de estudos em

morfologia robótica tem desenvolvido variações de robôs dotados de diversos meios de locomoção (e.g. pernas, rodas, esteiras). Em paralelo a este desenvolvimento temos a evolução constante de uma gama extremamente grande de sensores (e.g. sistemas de visualização, posicionamento, detecção de obstáculos). O desenvolvimento de algoritmos e técnicas para coordenar estes conjuntos físicos em um ambiente dinâmico é um desafio extremamente complexo [3]. Dotar robôs autônomos de capacidade de raciocínio inteligente e de interação com o meio em que estão inseridos é uma área de pesquisa que tem atraído a atenção de um grande número de pesquisadores [2]. O uso de ambientes de realidade virtual torna a pesquisa em robótica muito mais ágil do que a pesquisa usando robôs reais.

No trabalho [18] propomos a idéia de um ambiente de simulação virtual para a identificação e combate de incêndios florestais, com agentes controlados por regras. Neste artigo apresentamos a evolução do ambiente, que foi remodelado usando a biblioteca de simulação física *Open Dynamics Engine* (ODE) e a aplicação de técnicas inteligentes nos agentes.

Existem diversas áreas onde a habilidade de um único agente não é suficiente ou eficiente para a realização de uma tarefa, em alguns destes casos, como patrulhamento, vigilância, resgate ou exploração o mais indicado é a aplicação de sistemas multi-robóticos. Sistemas multi-robóticos são sistemas onde robôs autônomos trabalham cooperativamente a fim de cumprir uma missão, podendo existir interação entre os robôs ou não [4].

Um problema fundamental da robótica é a navegação. O deslocamento de um lugar para outro depende de três aspectos fundamentais: localização, orientação e controle motor. Para conhecer tanto sua localização como sua orientação, um robô móvel deve possuir sensores próprios (e.g. GPS, bússola). Para o

<sup>1</sup> Código-fonte disponível em <http://pessin.googlepages.com>

controle motor, deve possuir um número adequado de motores (um veículo autônomo, por exemplo, usualmente possui um motor angular, para giro das rodas e um motor linear, para tração). Normalmente sensores e atuadores são sujeitos a erros e interferências, assim o controle das ações de um robô deve sempre levar em conta a imprecisão dos sensores e motores envolvidos. Um sistema robusto deve permitir que, mesmo com sensores e atuadores imprecisos, o agente cumpra o seu objetivo. Uma técnica de Aprendizado de Máquina indicada para este controle é a de Redes Neurais Artificiais, dada sua capacidade de aprendizado a partir de exemplos e a respectiva generalização e adaptação das saídas. É uma técnica muito utilizada no controle de navegação de sistemas reativos [34,43].

Na tarefa de controle de incêndio, uma das questões mais importantes tem relação com a formação do posicionamento dos agentes para atuação no combate. De acordo com as capacidades de atuação de cada agente, as condições climáticas (vento, chuva), a topografia e a vegetação, diferentes formações (Figura 1) podem ser sugeridas. Estas formações, quando sugeridas por um especialista, podem não levar em conta um número muito grande de variáveis, assim, a definição do posicionamento de formação poderia fazer uso de técnicas de Aprendizado de Máquina (*Machine Learning* - ML). Uma técnica de ML indicada para estes casos é a de Algoritmos Genéticos (*Genetic Algorithms* - GA) [7,8], que são algoritmos de otimização global que empregam estratégia de busca paralela e estruturada, embora aleatória, direcionada a busca de pontos de aptidão [9], permitem assim a realização de busca multi-critério em um espaço multidimensional e por serem não supervisionados, não necessitam de nenhuma base de informação de antemão, e se corretamente utilizados são capazes de escapar de mínimos locais [26].



**Figura 1. Formações para uma equipe de quatro agentes (da esquerda para a direita: linha, coluna, circular ou ferradura) [1].**

Para que seja possível a implementação física real, o sistema multi-agente que propomos deve ser projetado, desenvolvido e testado anteriormente em ambientes de simulação realísticos. Ambientes com terrenos 3D gerados com a biblioteca de programação *Open Scene Graph* (OSG) [28] junto com a biblioteca de programação *Demeter* [29] permitem combinar o mapa

de elevação juntamente com uma determinada distribuição de vegetação, criando assim um terreno bastante realístico [30]. Além disso, o uso da biblioteca de programação *Open Dynamics Engine* (ODE) [27] permite implementar atributos físicos (*e.g.* atrito, fricção, gravidade, colisão) tornando o sistema ainda mais realístico.

Neste trabalho, um grupo de agentes autônomos trabalha cooperativamente a fim de realizar com sucesso a identificação e o combate de incêndios em áreas florestais, sem intervenção humana. São simulados diferentes tipos de agentes, com níveis hierárquicos diferentes, assim, cada etapa do processo conta com agentes específicos que possuem um conjunto de ações determinadas. Em nossos estudos e pesquisas, pelo que pudemos constatar, não existe um ambiente de simulação aberto que combine todas as características requisitadas para este trabalho, que são: (i) Realismo físico de modelagem robótica; (ii) Realismo físico de interação de agentes em terrenos irregulares; (iii) Facilidades de comunicação entre os agentes; (iv) Capacidade de aplicação de métodos de controle (*e.g.* Redes Neurais Artificiais); (v) Capacidade de aplicação de métodos de evolução de estratégias (*e.g.* Algoritmos Genéticos); (vi) Simulação de processos naturais (como a propagação do fogo).

Nossas principais metas neste projeto são: (i) Recolher informações sobre dados florestais, tipos de vegetação, topografia, e comportamento de incêndios para criar o ambiente virtual mais realista possível; (ii) Simular incêndios em florestas, reproduzindo de forma bastante realista o ambiente e a propagação dos focos de incêndio; (iii) Pesquisar ferramentas e técnicas de combate à incêndios florestais utilizadas por bombeiros; (iv) Implementar a simulação de agentes móveis autônomos colaborativos capazes de formar uma brigada de combate a incêndios; (v) Estudar métodos de aprendizado de máquina e suas vantagens para o modelo; e (vi) Estudar a robustez das ações dos agentes pela leitura de dados de sensores sujeitos a erros. Como resultado principal, o ambiente virtual deve permitir ajudar no projeto e validação dos robôs reais a serem construídos (tanto morfologia como técnicas de controle) e na validação da operação multi-agente de identificação e combate de incêndios. O sucesso desta tarefa envolve o uso de uma variedade de tecnologias de diferentes campos, assim, a construção deste sistema é um bom estudo para avaliar eficiência de arquitetura multi-agente, estratégia cooperativa, fusão de sensores e modelagem robótica. A tarefa em si poderá ser generalizada para outras atividades práticas como acidentes nucleares ou desastres ambientais. Com técnicas de aprendizado de máquina esperamos obter agentes que suportem melhor um

ambiente dinâmico, fazendo assim o sistema mais flexível e autônomo possível.

Neste artigo apresentamos na Seção 2 uma pequena conceituação teórica de agentes e sistemas multi-agentes. A Seção 3 apresenta conceitos de Aprendizado de Máquina e das técnicas de IA escolhidas. Na Seção 4 detalhamos características de robótica móvel e conceitos de simulação e modelagem, bem como as bibliotecas selecionadas para o desenvolvimento do trabalho. Na Seção 5 descrevemos técnicas e operações reais de identificação e combate a incêndios florestais. Na Seção 6 descrevemos o ambiente desenvolvido, a operação multi-agente de identificação e combate, o desenvolvimento e a aplicação da Rede Neural Artificial e o desenvolvimento e a aplicação do Algoritmo Genético. Finalizamos apresentando a conclusão do trabalho realizado.

## 2. Agentes e Sistemas Multi-Agentes

Uma das primeiras e mais simples definições de agentes inteligentes é de que são sistemas computacionais que habitam um dado ambiente, sentem e agem autonomamente nesse ambiente [25]. Segundo [40] um agente de software pode ser visto como um sistema dinâmico, onde a percepção e a ação constituem processos simultâneos e inseparáveis. Em termos gerais, ambiente é onde um agente ou um conjunto de agentes está inserido, pode ser físico (como ambientes onde estão inseridos robôs), de software ou de realidade virtual (onde se faz simulação do ambiente físico) [38].

De acordo com [40] o projeto de um agente de software envolve a definição de três componentes que são rigorosamente interconectados e mutuamente interdependentes: (i) Definição do ambiente para atuação; (ii) Estabelecimento de comportamentos e tarefas a serem realizadas; (iii) Determinação da morfologia do agente. A concretização do projeto de um agente robótico deve priorizar quatro aspectos principais [41]: (i) Autonomia: Deve ser capaz de operar com o mínimo possível de intervenção, supervisão e instrução humana; (ii) Auto-suficiência: Deve ser capaz de operar por tempo prolongado, ou, pelo menos, suficiente para realizar a tarefa e buscar recarga de energia; (iii) Localização: Deve ser capaz de adquirir informação sobre o ambiente somente através de seus próprios sensores; (iv) Corporificarão: Deve ser desenvolvido como um sistema físico ou computacional, porém a corporificarão não se dá necessariamente pela materialização, como a apresentada em animais ou robôs físicos, mas por uma relação dinâmica com o ambiente, ou seja, a corporificarão pode ser realizada em ambientes de

simulação computacionais, desde que estes ambientes sejam realísticos do ponto de vista físico [41].

Sistemas multi-agentes são sistemas constituídos de múltiplos agentes que interagem ou trabalham em conjunto de forma a realizar um determinado conjunto de tarefas ou objetivos. Um sistema multi-agente pode ser visto como uma rede, fracamente acoplada, de solucionadores de problemas que trabalham em conjunto para resolver problemas que vão além da sua capacidade individual. Estes solucionadores de problemas são essencialmente autônomos e, muitas vezes, heterogêneos em sua natureza [36]. Um sistema multi-agente deve ter alguma forma de controle. A utilização de ambientes dinâmicos, robôs com mau funcionamento, múltiplas regras e restrições em comportamentos individuais adicionam complexidade ao problema de controle [4].

Cooperação pode ser entendida como a capacidade que os agentes têm de trabalhar em conjunto de forma a concluir tarefas de interesse comum [31]. Para que exista cooperação, deve existir alguma forma de interação entre os agentes, ou com humanos, através de alguma forma de comunicação [19]. A cooperação não depende obrigatoriamente de comunicação entre todos os agentes, visto que, como no mundo real, um coordenador (unidade de controle) pode delegar tarefas (enviar mensagens) para um grupo de pessoas (agentes) que não tenham comunicação direta entre si. Neste caso, as pessoas podem trabalhar em prol do mesmo objetivo sem ter conhecimento do que as outras pessoas estão fazendo. A morfologia de um robô deve, ao incluir a capacidade de comunicação, possuir um módulo de controle de comunicações, assim, deve possuir componentes de percepção (recepção de mensagens) e de ação (envio de mensagens). O módulo de comunicação está diretamente ligado ao módulo de controle do agente [31].

## 3. Aprendizado de Máquina

Aprendizado de Máquina (*Machine Learning* - ML) é uma área da Inteligência Artificial que tem como objetivo desenvolver técnicas computacionais de aprendizado e de aquisição de conhecimentos [5]. Essas técnicas devem exibir um comportamento inteligente e realizar tarefas complexas com um nível de competência equivalente ou superior ao de um especialista humano [6]. Para a construção do sistema proposto neste artigo utilizamos Redes Neurais Artificiais e Algoritmos Genéticos, assim, descrevemos sucintamente suas características a seguir.

### 3.1. Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são sistemas paralelos distribuídos, compostos por unidades de

processamento simples que calculam determinadas funções matemáticas, normalmente não-lineares [9]. Seus atributos básicos podem ser divididos em arquitetura e neurodinâmica. A arquitetura determina a estrutura da rede, ou seja, o número de neurônios e sua interconectividade e a neurodinâmica, por sua vez, define as propriedades funcionais da rede, ou seja, como ela aprende, recupera, associa e compara novas informações com o conhecimento já armazenado [37]. Matematicamente, RNAs são aproximadores universais, que realizam mapeamentos em espaços de funções multi-variáveis [32]. A capacidade de aprender e generalizar das RNAs é um dos seus maiores atrativos. O processamento da informação em uma RNA é feito por meio de estruturas neurais artificiais [5], sendo que esta estrutura, bem como o próprio neurônio artificial são uma analogia biológica ao funcionamento do cérebro.

O algoritmo *Backpropagation* [42] é um algoritmo supervisionado de aprendizado de RNA. A base de treinamento é um conjunto de dados que deve apresentar, para cada entrada, a saída prevista do sistema. Este tipo de aprendizado ocorre em várias épocas, sendo que cada época representa a apresentação do conjunto inteiro de dados à RNA para o ajuste dos pesos. O treinamento de uma RNA deve envolver várias rodadas de simulação, iniciando os pesos de forma aleatória. Outra questão importante é o grau de generalização, sendo medido usualmente através do uso de uma base de validação usada em paralelo a base de treinamento.

O *Stuttgart Neural Network Simulator* (SNNS) [10] é um simulador de Redes Neurais Artificiais criado no *Institute for Parallel and Distributed High Performance Systems* (IPVR) da Universidade de Stuttgart. O SNNS possui um grande número de algoritmos de aprendizado (e.g. *Backpropagation*, *Quickprop*, *RProp*). Um aplicativo do pacote SNNS, o SNNS2C, permite a conversão de uma RNA em código em C, que pode ser inserido em uma outra aplicação.

### 3.2. Algoritmos Genéticos

Algoritmos Genéticos (*Genetic Algorithms* - GA) [7,8] são técnicas de otimização global que empregam estratégia de busca paralela e estruturada, embora aleatória, direcionada a busca de pontos de aptidão [9]. Devido a ser uma técnica estocástica, é classificada como não supervisionado, visto que não necessita de nenhuma base de informação de antemão. Os GA utilizam procedimentos iterativos que simulam o processo de evolução de uma população de possíveis soluções de um determinado problema. Durante o processo evolutivo, cada indivíduo da população é

avaliado através de uma função de aptidão (*fitness*) que permite mensurar o quanto cada indivíduo está apto a resolver o problema. Nos indivíduos selecionados, é feito *crossover* e mutação, gerando descendentes para a próxima geração [9,12,13]. O *crossover* e a mutação transformam uma população através de sucessivas gerações e possibilitam assim que o GA percorra os melhores pontos do espaço de busca até chegar a um resultado satisfatório [11].

A *GALib* (<http://lancet.mit.edu/ga>) é uma biblioteca de software livre desenvolvida por Matthew Wall do *Massachusetts Institute of Technology* (MIT) em C++ que contém um conjunto bastante amplo de funções relacionadas a programação de Algoritmos Genéticos. É uma das mais completas e eficientes bibliotecas de software para a simulação de Algoritmos Genéticos, permitindo experiências com diferentes funções objetivo (*fitness*), representações genéticas, operadores genéticos e métodos de seleção.

## 4. Robótica Móvel

Um robô móvel é um dispositivo mecânico montado sobre uma base não fixa que age sob o controle de um sistema computacional, equipado com sensores e atuadores que o permitem interagir com o ambiente [34,39]. A interação com o ambiente se dá através de ciclos de percepção-ação que consistem em três passos fundamentais: (i) Obtenção de informação através de sensores; (ii) Processamento das informações para seleção de ação; e, (iii) Execução da ação através do acionamento dos atuadores. Esse conjunto de operações, em uma análise superficial, pode parecer simples, porém o controle robusto de sistemas robóticos tem complicações físicas, mecânicas, eletrônicas e computacionais que tornam a criação de um conjunto de regras uma tarefa árdua e sujeita a erros. Sensores são os mecanismos de percepção de um robô e realizam medições físicas (e.g. contato, distância, orientação, temperatura), provêm sinais ou dados crus que precisam ser interpretados pelo “cérebro” do robô. A interpretação destes sinais deve ser a única maneira de um robô autônomo entender o ambiente que o cerca para poder realizar as mudanças de ação necessárias [35]. Atuadores são os mecanismos de ação de um robô (e.g. motores, pistões, braços manipuladores), controlados por circuitos eletrônicos que recebem valores de ação, valores estes que devem ser calculados pelo “cérebro” do robô e devem estar de acordo com especificações de fabricantes [35].

### 4.1. Simulação e Modelagem

Experimentos em robótica móvel podem ser realizados de duas formas: diretos em um robô real ou

em um robô simulado em um ambiente virtual realístico [20]. Usualmente, experimentos em robótica móvel utilizando um robô real demandam enorme despendimento de tempo e de recursos financeiros. Para que seja possível a implementação física real, o sistema multi-agente que propomos deve ser projetado, desenvolvido e testado anteriormente em ambientes de simulação realísticos. A simulação de sistemas robóticos é especialmente necessária para robôs caros, grandes, ou frágeis [3], sendo uma ferramenta extremamente poderosa para agilizar o ciclo de desenvolvimento de sistemas de controle robóticos eliminando desperdício de recursos, tanto financeiros como computacionais. Para que uma simulação seja útil, entretanto, ele deve capturar características importantes do mundo físico, onde o termo *importantes* tem relação ao problema em questão [3]. No caso deste trabalho, é fundamental que existam restrições físicas no modelo e que exista a possibilidade de trabalho em um terreno irregular, provido de obstáculos. Para o desenvolvimento deste trabalho foram pesquisadas algumas ferramentas de simulação, como o *Microsoft Robotics Studio* ([msdn.microsoft.com/robotics](http://msdn.microsoft.com/robotics)), o *Webots* ([www.cyberbotics.com](http://www.cyberbotics.com)), o *Khepera Simulator* ([diwww.epfl.ch/lami/team/michel/khep-sim](http://diwww.epfl.ch/lami/team/michel/khep-sim)), o *Mission Simulation Facility* ([ase.arc.nasa.gov/msf](http://ase.arc.nasa.gov/msf)), o *JUICE* ([www.natew.com/juice](http://www.natew.com/juice)) e o *Simulator BOB* ([tuharburg.de/ti6/mitarbeiter/pst/Sim](http://tuharburg.de/ti6/mitarbeiter/pst/Sim)). No entanto, optamos pela criação de um simulador próprio que reproduzisse o mundo real em um ambiente virtual 3D com realismo físico. Como optamos por desenvolver nosso próprio ambiente, todas as bibliotecas de programação selecionadas para o desenvolvimento do ambiente e dos agentes robóticos são software livre, multi-plataforma e em linguagem C/C++. Uma pequena descrição de cada uma das bibliotecas utilizadas é fornecida a seguir: (i) *Open Dynamics Engine* (ODE) [27] é uma biblioteca desenvolvida para a simulação física de corpos rígidos articulados [33]. Uma estrutura articulada é criada quando corpos rígidos de vários tipos são conectados por algum tipo de articulação, como, por exemplo, um veículo terrestre que tem a conexão de rodas em um chassi. A ODE foi projetada para ser utilizada de modo interativo em simulações de tempo real e é especialmente indicada para a simulação de objetos móveis em ambientes dinâmicos. A ODE não tem como objetivo realizar simulações de outras dinâmicas além da dinâmica de corpos rígidos (e.g. partículas, roupas, ondas, fluidos, corpos flexíveis, fraturas). A ODE possui contatos rígidos, assim, quando dois corpos colidem não há penetração. O sistema de detecção de colisão é nativo e suporta diversas primitivas (e.g. esfera, caixa, cilindro, plano, raio). A

utilização da ODE no nosso ambiente é fundamental por fornecer restrições físicas, principalmente na definição da morfologia dos robôs; (ii) *Open Scene Graph* (OSG) [28] é uma biblioteca para desenvolvimento de aplicações gráficas 3D de alta performance. Baseada no conceito de grafos de cena, provê ao desenvolvedor um ambiente orientado a objeto sobre a *OpenGL* ([www.opengl.org](http://www.opengl.org)), liberando este da necessidade de implementação e otimização de chamadas gráficas de baixo nível; (iii) *DrawStuff* [27] é um ambiente de visualização de objetos 3D que tem o propósito de permitir a demonstração visual da ODE sendo uma biblioteca bastante simples e rápida para utilização; (iv) *Demeter* [29] é uma biblioteca desenvolvida para renderizar terrenos 3D, desenvolvida para ter rápida performance e boa qualidade visual, pode renderizar grandes terrenos em tempo-real sem necessidade de hardware especial, depende da *Simple DirectMedia Layer* ([www.libsdl.org](http://www.libsdl.org)) para realizar o tratamento das texturas do terreno e da *Geospatial Data Abstraction Library* ([www.gdal.org](http://www.gdal.org)) para carregar arquivos de elevação.

## 5. Incêndios em Ambientes Naturais

A fim de melhor entender como proceder no combate a incêndios florestais e assim planejar as estratégias a serem implementadas nos agentes autônomos, foi realizado um estudo sobre técnicas reais de operação. Este estudo teve como base os trabalhos de [14, 15, 16, 21, 22]. Para a implementação da propagação do fogo obtivemos de [17] medições reais de velocidades. O estudo dos modelos de florestas e resíduos florestais é de grande importância para o aprimoramento dos modelos de simulação a serem implementados em ambientes virtuais [18].

Os combustíveis florestais, que são produto da cobertura vegetal e sua dinâmica, são importantes parâmetros nos processos de ignição e propagação dos incêndios. O conhecimento das características básicas dos combustíveis (e.g. tipo, quantidade, continuidade, arranjo) é muito útil na previsão do comportamento dos incêndios [23]. Quanto aos tipos de combustíveis florestais, sugere-se os seguintes modelos [24]: Modelos 1, 2 e 3 são do tipo herbáceo, sendo pasto fino, contínuo, e contínuo espesso respectivamente, os incêndios propagam-se com grande velocidade; Modelos 4, 5, 6 e 7 são do tipo arbustivo, indo de árvores jovens até mato denso; Modelos 8, 9 e 10 são do tipo manta morta; Modelos 11 e 12 são do tipo resíduos lenhosos. Cada um destes modelos possui velocidade de propagação de fogo específica.

O estudo das técnicas reais de operação permite que se possa planejar melhor o formato da operação e as

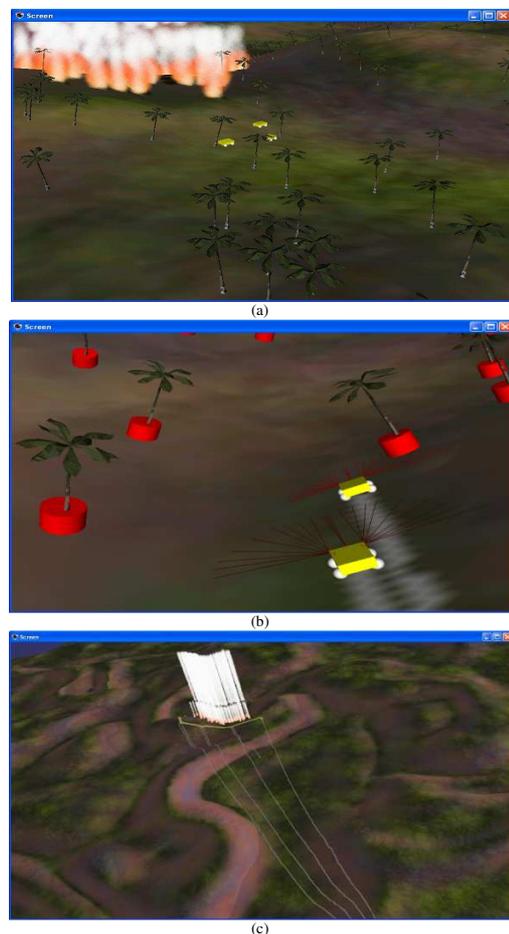
estratégias a serem implementadas no sistema. O conjunto das técnicas listadas a seguir é de consenso geral para aplicações em monitoramento e combate a incêndios. A operação de combate ou supressão de um incêndio envolve seis etapas distintas [21]: (i) Detecção; (ii) Comunicação; (iii) Mobilização; (iv) Deslocamento; (v) Planejamento; e (vi) Combate. Existem quatro métodos de combate ao fogo nos incêndios florestais [21]: (i) Método direto: usado quando a intensidade do fogo permite uma aproximação suficiente da brigada à linha de fogo, são usadas as seguintes técnicas e materiais: água (bombas costais, baldes ou moto-bombas); terra (pás); ou batidas (abafadores); (ii) Método paralelo ou intermediário: usado quando não é possível o método direto e a intensidade do fogo não é muito grande, consiste em limpar, com ferramentas manuais, uma estreita faixa, próxima ao fogo, para deter o seu avanço e possibilitar o ataque direto; (iii) Método indireto: usado em incêndios de intensidade muito grande, consiste em abrir aceiros com equipamento pesado (e.g. trator, motoniveladora); (iv) Método aéreo: usado nos incêndios de copa, de grande intensidade e área e em locais de difícil acesso às brigadas de incêndio, são usados aviões e helicópteros.

A rapidez e a eficiência na detecção e monitoramento dos incêndios florestais são fundamentais para a viabilização do controle do fogo, redução dos custos nas operações de combate e atenuação dos danos. Para países de grande extensão territorial, como o Brasil, o monitoramento dos incêndios florestais através de imagens de satélites é o meio mais eficiente e de baixo custo quando comparado com os demais meios de detecção [15]. O lançamento em 1972 do primeiro satélite *Landsat* possibilitou detectar alterações nas áreas florestais do espaço. Desde então, as imagens termais e de infravermelho têm sido usadas na detecção de incêndios e estudos de mapeamento, permitindo que áreas queimadas e não queimadas sejam detectadas através do contraste entre os gradientes térmicos [16].

## 6. Implementação do Protótipo

A operação multi-agente depende essencialmente de duas etapas, planejamento e ação, assim, foram implementados protótipos específicos para cada etapa. Ambos foram desenvolvidos em C++. O protótipo onde é realizada a ação dos robôs bombeiros, controlados por uma Rede Neural Artificial pode ser visto na Figura 2. Este protótipo tridimensional usa a biblioteca OSG, responsável pela saída gráfica do protótipo, a biblioteca Demeter, responsável pelo terreno irregular e a biblioteca ODE, responsável pelo realismo físico, tanto da morfologia robótica como da

colisão entre os objetos presentes no ambiente (e.g. robôs, árvores, inclinação de terreno). O uso da biblioteca ODE permite que os robôs simulados fisicamente respeitem questões como gravidade, inércia e atrito. Por exemplo, mantendo uma força  $f$  constante nos motores lineares (torque) um veículo terá velocidade  $v$  em regiões planas, em regiões de declive terá  $v$  maior e em aclives terá  $v$  menor. O protótipo onde é realizado o planejamento usa um Algoritmo Genético. Este não necessita obrigatoriamente de visualização, porém está implementado com a possibilidade de uma saída gráfica 2D desenvolvida com SDL, como mostra a Figura 4(b). A integração entre os protótipos se dá através de um arquivo texto, após realizar a evolução, o protótipo responsável cria um arquivo com as posições de atuação que é lido na inicialização do protótipo de ação no combate.



**Figura 2. Ambiente de simulação desenvolvido com OSG, ODE e Demeter.**

Para a simulação da vegetação e correta propagação do fogo, existe uma matriz oculta sob o terreno. Esta matriz possui, para cada área do terreno, o tipo de

vegetação presente, assim, considerando orientação do vento, intensidade do vento e tipo de vegetação de uma área podemos construir a simulação de propagação do fogo. A velocidade de propagação respeita dados do modelo retirados de [17]. Quanto ao vento, tanto a sua intensidade como a sua orientação podem ser geradas aleatoriamente ou configuradas a partir de dados parametrizados pelo usuário. O tempo de permanência do fogo em uma área é relacionado diretamente ao tipo da vegetação presente e se comporta baseado nos valores de tipo de vegetação, inclinação do terreno, intensidade e orientação do vento. Desta forma a propagação do fogo busca simular de modo bastante realístico a forma como o fogo se propagaria em um ambiente real. O comportamento do fogo é idêntico nos dois protótipos. Ambos os protótipos possuem um mapa que simula a integração das informações de vegetação, topografia e comportamento de fogo detalhados na Seção 5. A criação dos mapas teve como base cartas topográficas e o mapa de modelos de combustíveis florestais, que podem ser vistos em [18].

A comunicação entre os agentes robóticos existe para que alertem os atrasos na navegação, assim, o raio de atuação é recalculado a cada  $n$  fragmentos de tempo. A simulação desta comunicação faz uso de um *blackboard*. Além do alerta de atrasos, o sistema de comunicação simula a troca de mensagens entre o agente monitor e os agentes de combate. Existem casos de comunicação um para um e de um para todos. A fila usada como função de *blackboard* armazena as seguintes informações: indicador de remetente, indicador de destinatário, *timestamp*, e tipo da mensagem (e.g. aviso de incêndio, aviso de fim de incêndio, negociação de times, posição do incêndio).

No ambiente desenvolvido, cada árvore existe como um modelo OSG e como um cilindro ODE. Os sensores identificam os cilindros (Figura 3(a)) que são posicionados junto às árvores. O ambiente permite que, através de parâmetros, possamos habilitar ou não a exibição dos objetos físicos ao invés de apenas a sua representação gráfica.

### 6.1 Operação de Identificação e Combate

Usamos o ambiente para simular a seguinte operação: um agente monitor (satélite) monitora o terreno da área florestal, ao identificar uma área com foco de incêndio, ativa o módulo de evolução de estratégias (detalhado na Subseção 6.3). Após obter as coordenadas de atuação através do GA, o agente monitor envia mensagens aos agentes de combate informando suas posições de atuação (nesta implementação simulamos esta operação através de um arquivo texto). O comportamento dos agentes de combate é reativo, deslocando-se em direção a posição

de seu objetivo desviando de obstáculos. O método de combate de incêndio simulado é o método indireto. Os agentes de combate simulados são motoniveladoras que tem como finalidade cercar o foco de incêndio e criar um aceiro (área livre de vegetação onde o fogo se extingue pela falta de combustível). Esta operação pode ser entendida com a Figura 4(a). Quanto ao controle de posicionamento, um sensor do tipo GPS é simulado em cada robô. Em experimentos realizados com um GPS *Garmin Etrex* ([www.garmin.com](http://www.garmin.com)) obtivemos um erro médio de 18,6 metros. Considerando que cada agente possui seu próprio GPS, o tratamento deste erro é crucial na criação dos aceiros [18]. Tratamos esta informação da seguinte maneira: o erro médio deste sensor durante o deslocamento é usado somado a distância de criação do aceiro e também somado ao final da área de criação.

### 6.2 Morfologia dos Robôs Móveis

Os robôs móveis foram desenvolvidos com a biblioteca de simulação de corpos rígidos articulados ODE. As Figuras 3(a) e 3(b) apresentam o veículo desenvolvido. Dada a existência de restrições físicas, a única maneira de controlar este veículo é com a aplicação de forças em seus dois motores simulados, que são: um motor angular (para o giro do volante) e um motor linear (para o torque). Além do GPS, responsável pela obtenção da localização, cada robô possui também uma bússola, necessária para a obtenção da orientação do veículo.

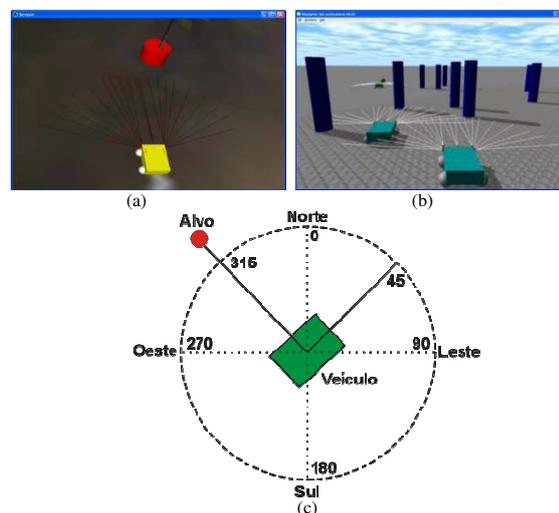


Figura 3. Sensores desenvolvidos: (a) e (b) Sensores de distância e (c) Bússola e GPS, responsáveis pela orientação, localização e obtenção do azimute.

O azimute (ângulo para o alvo) é obtido a partir da posição atual (GPS) e da posição do alvo (recebida por mensagem), como mostra a Figura 3(c). Os sensores de distância são sonares simulados, apresentando as características de capacidade de medir distâncias entre 15cm e 11m, como o *Polaroid 6500* (www.senscomp.com).

### 6.3. Evolução de Estratégias de Posicionamento

O mecanismo de planejamento usa um GA para definir as posições iniciais e finais de atuação de cada robô no combate ao incêndio. Considerando que os agentes de combate são motoniveladoras que tem como finalidade criar um aceiro, necessitamos que o GA retorne as seguintes informações: ângulo inicial e final e raio inicial e final para cada robô, ambos em relação ao ponto inicial do foco de incêndio. Esta operação pode ser entendida com a Figura 4. O genoma desenvolvido para as simulações iniciais pode ser visto na Tabela 1. Neste, estão presentes informações de todo o grupo dos agentes envolvidos, assim, o tamanho do genoma é dependente da quantidade de agentes no sistema. Realizamos simulações considerando a existência de 4 agentes de combate.

Tabela 1. Genoma dos indivíduos.

Gene	Função	Valor Mínimo	Valor Máximo
0	Posição inicial do agente 0	0,000°	360,000°
1	Posição final do agente 0 e inicial do agente 1	0,000°	360,000°
2	Posição final do agente 1 e inicial do agente 2	0,000°	360,000°
3	Posição final do agente 2 e inicial do agente 3	0,000°	360,000°
4	Posição final do agente 3	0,000°	360,000°
5	Raio inicial do agente 0	10,000m	100,000m
6	Raio final do agente 0 e inicial do agente 1	10,000m	100,000m
7	Raio final do agente 1 e inicial do agente 2	10,000m	100,000m
8	Raio final do agente 2 e inicial do agente 3	10,000m	100,000m
9	Raio final do agente 3	10,000m	100,000m

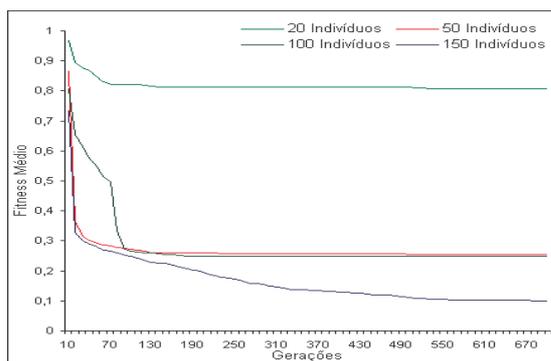


Gráfico 1. Evolução do *fitness* médio de acordo com o número de gerações, para diferentes quantidades de indivíduos.

O *fitness* desenvolvido acumula os seguintes valores finais de cada simulação: (i) Total de área queimada: busca minimizar a área queimada; (ii) Total de área

com aceiro: busca minimizar a área de trabalho dos robôs, e; (iii) Erro médio absoluto: busca minimizar a diferença entre a média geral de aceiros úteis em relação ao aceiro útil de cada indivíduo, assim, o tamanho das áreas de trabalho tende a se equalizar. Buscamos, na simulação, minimizar o valor do *fitness*, o Gráfico 1 mostra os resultados das simulações com diversos tamanhos de população. Podemos ver o *fitness* diminuindo com o passar das gerações.

A simulação com 150 indivíduos foi a que chegou ao menor *fitness*, podemos ver o resultado satisfatório desta simulação nas Figuras 4(a) e 4(b). O cromossomo resultante das 700 gerações com 150 indivíduos (Figuras 4(a) e 4(b)) apresentou, na última geração os seguintes valores:

225,7 | 199,2 | 174,1 | 155,2 | 136,3 | 27,3 | 30,4 | 33,9 | 38,0 | 35,9

As posições de atuação são calculadas aplicando no cromossomo acima as Equações 1 e 2.

$$x_f = x_a + r_i \times \cos(a_i) \quad (1)$$

$$y_f = y_a + r_i \times \sin(a_i) \quad (2)$$

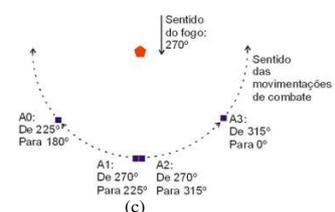
Onde  $(x_f, y_f)$  é a coordenada da posição final dos robôs,  $(x_a, y_a)$  é a coordenada da posição inicial do incêndio,  $r_i$  é o raio (gene 5 a 9) e  $a_i$  é o ângulo (gene 0 a 4). O raio, bem como o ângulo, são específicos para cada operação de cada robô (coordenada inicial e final de criação de aceiro, como mostra a Tabela 1).



(a)



(b)



(c)

Figura 4. (a) Resultado do planejamento usando GA com 150 indivíduos e 700 gerações, (b) Mesmo resultado em visualização 2D e (c) Modelo teórico de simulação.

## 6.4 Controle de Navegação

Quando um robô recebe a coordenada da posição do incêndio do agente monitor, ele deve realizar o deslocamento de sua posição atual até a posição de atuação. Para realizar esta navegação, desviando de obstáculos, o agente usa uma RNA do tipo *Backpropagation*. As entradas da RNA são obtidas pelos sensores presentes no robô e as saídas são as forças a serem aplicadas nos motores. A RNA possui 7 entradas: orientação do veículo e orientação para o alvo (obtidos através de bússola e GPS simulados) mais as proximidades de obstáculos medidas por 5 sensores do tipo sonar. As saídas são: força a aplicar no motor linear (torque, de 0.0 a 6.0) e força a aplicar no motor angular (giro da barra da direção, de -1.5 a 1.5).

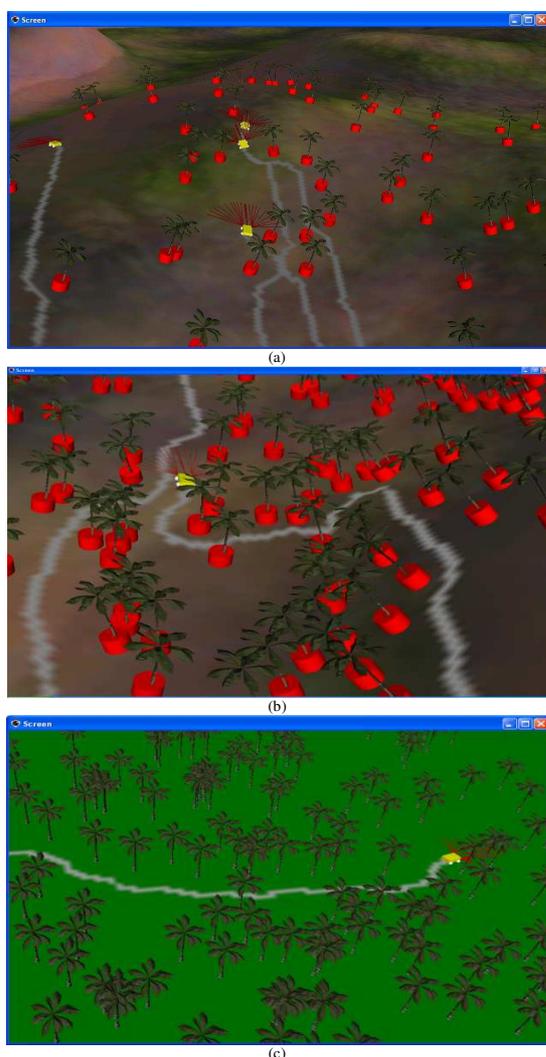


Figura 5. Trajetos realizados com simulações usando a RNA: (a) 2,5% de ocupação, (b) e (c) 10% de ocupação.

A base de dados para treinamento da RNA foi obtida a partir de um sistema de regras simples, rodamos 32 simulações com diferentes pontos de início e destino para obter dados específicos de navegação e de desvios de obstáculos, obtendo um total de 4.985 registros. Esta base de dados foi dividida em 70% para treino e 30% para teste, e, usando o SNNS foram geradas seis Redes Neurais Artificiais com 4, 9, 18, 24, 30 e 36 neurônios na camada oculta. Foi realizado o treino de cada RNA e análise de 5.000, 10.000, 20.000, 40.000, 60.000, 80.000 e 100.000 ciclos. A RNA que apresentou o menor *Mean Absolute Error* (MAE) tanto no giro como na velocidade foi a com 24 neurônios na camada oculta, com 80.000 ciclos de treino. Esta RNA foi convertida em um programa C e utilizada nos robôs. Após a implementação da RNA no controle de navegação com desvio de obstáculos nos robôs móveis, realizamos uma série de simulações com o ambiente contendo 2 robôs, com pontos de início e destino aleatórios e com uma quantidade parametrizada de árvores. Os resultados das simulações podem ser vistos na Tabela 2 e na Figura 5.

Tabela 2. Resultado das simulações.

Quantidade de Simulações de Navegação	Área Ocupada com Árvores	Resultados Satisfatórios com a RNA
50	10,00%	42 (84,00%)
50	5,000%	49 (98,00%)
50	2,500%	49 (98,00%)
50	0,625%	50 (100,0%)

## 7. Conclusão

O objetivo deste artigo foi detalhar o projeto e o desenvolvimento de um sistema multi-agente que opera em um ambiente virtual de simulação realística. Neste sistema, uma equipe de agentes autônomos trabalha cooperativamente a fim de realizar com sucesso a identificação e o combate de incêndios em áreas florestais, sem intervenção humana. O ambiente suporta uma série de características fundamentais para a simulação realística da operação, como terrenos irregulares, processos naturais e restrições físicas na criação e uso de robôs móveis. A operação multi-agente depende essencialmente de duas etapas: planejamento e ação. No planejamento, usamos Algoritmos Genéticos para evoluir estratégias de posicionamento de atuação dos robôs bombeiros. Para a ação foram criados robôs de combate fisicamente simulados, sendo as informações sensoriais de cada robô (e.g. GPS, bússola, sonar) usadas na entrada de uma Rede Neural Artificial (RNA). Esta RNA que controla os atuadores do veículo permite navegação com desvio de obstáculos. Os resultados das simulações demonstram que a Rede Neural controla satisfatoriamente os robôs móveis, que o uso de

Algoritmos Genéticos configura a estratégia de combate ao incêndio de modo satisfatório e que o sistema multi-agente proposto pode vir a ter um papel muito importante no planejamento e execução de operações reais de combate a incêndios florestais.

## 10. Referências

- [1] Balch, T., Arkin R. C., “Motor schema-based formation control for multiagent robot teams”, *IEEE International Conference on Multiagent Systems (ICMAS)*, 1995.
- [2] Dudek, G., Jenkin, M., *Computational Principles of Mobile Robotics*, Cambridge, London, UK: MIT Press, 2000.
- [3] GO, J. et. al., “Accurate and flexible simulation for dynamic, vision-centric robots”, *International Joint Conference on Autonomous Agents*, 2004.
- [4] Osagie, P., “Distributed Control for Networked Autonomous Vehicles”, *Dissertação de Mestrado*, KTH-CSC, Royal Institute of Technology, Sweden, 2006.
- [5] Rezende, S.O., *Sistemas inteligentes: fundamentos e aplicações*, Ed. Manole, São Paulo, 2003.
- [6] Nikolopoulos, C., *Expert Systems - Introduction to First and Second Generation and Hybrid Knowledge Based Systems*. New York, USA: Marcel Dekker Inc. Press, 1997.
- [7] Holland, J., *Adaptation in Natural and Artificial Systems*, Michigan Press, 1975.
- [8] Mitchell, M., *An Introduction to Genetic Algorithms*, Cambridge, MA, The MIT Press, 1996.
- [9] Carvalho, A. C. F., Braga, A. P., Ludermir, T. B., *Sistemas inteligentes: Fundamentos e aplicações*, Cap. Computação Evolutiva, p. 225-248, 2003.
- [10] Stuttgart Neural Network Simulator (SNNS), em [www.informatik.uni-tuebingen.de/SNNS/](http://www.informatik.uni-tuebingen.de/SNNS/), outubro de 2007.
- [11] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [12] Lacerda, E. G. M., Carvalho, A. C. P. L. F. “Introdução aos algoritmos genéticos”, *XVIII Jornada de Atualização em Informática*, XIX CSBC, v. 2, p. 51-126, 1999.
- [13] Nolfi, S., Floreano, D., *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*, The MIT Press, 2000.
- [14] Antunes, M. A. H., “Uso de satélites para detecção de queimadas e para avaliação do risco de fogo”, *Ação Ambiental*, 12:24-27, 2000.
- [15] Batista, A.C., “Detecção de incêndios florestais por satélite”, *Floresta*, Curitiba, Paraná, n 34, 237-241, 2004.
- [16] Rimmel, T. K., Perera, A. H. “Fire mapping in a northern boreal forest assessing AVHRR/NDVI methods”, *Forest Ecology and Management* 152, 2001.
- [17] Koproski, L.P., “O fogo e seus efeitos sobre a heperoto e a mastofauna terrestre no parque nacional de Ilha Grande”, *Dissertação de mestrado*, UFPR, 2005.
- [18] Pessin, G. et. al. “Simulação Virtual de Agentes Autônomos para a Identificação e Controle de Incêndios em Reservas Naturais”, *IX Symposium on Virtual and Augmented Reality (SVR)*, v. 1, p. 236-245, 2007.
- [19] Wooldridge, M.; Jennings, N., “Intelligent agents: Theory and practice”, *Knowledge Engineering Review*, 1995.
- [20] Pfeifer, R., Scheier, C., *Understanding Intelligence*, Cambridge, Massachusetts, USA: The MIT Press, 1999.
- [21] Laboratório de Incêndios Florestais, “Pesquisas e projetos em prevenção e combate de incêndios florestais”, UFPR, [www.floresta.ufpr.br/~firelab](http://www.floresta.ufpr.br/~firelab), setembro 2006.
- [22] Centro de previsão do tempo e estudos climáticos - Instituto nacional de pesquisas espaciais (CPTEC/INPE), [www.cptec.inpe.br/queimadas](http://www.cptec.inpe.br/queimadas), Acesso em outubro 2006.
- [23] Castro, F. X.; Tudela, A.; Sebastia, M. T. “Modeling moisture content in shrubs to predict fire risk in catalonia”, *Agricultural and Forest Meteorology*, Spain, v. 116, 2001.
- [24] Ministério da Agricultura do Brasil, *Guia método-lógico para elaboração de plano municipal/intermunicipal de defesa da floresta contra incêndios*, 2006
- [25] Maes, P. “Intelligent software: easing the burdens that computers put on people”, *IEEE Expert*, v. 11, 1996.
- [26] Heinen, M. R., Osório, F. S., “Algoritmos genéticos aplicados em roteamento de veículos” *Hífen*, 2006.
- [27] *Open Dynamics Engine*, Disponível em: <[www.ode.org](http://www.ode.org)>. Acesso em jun. 2007.
- [28] *Open Scene Graph*, OSG Community, Disponível em: <<http://www.openscenegraph.com>>. Acesso em jun. 2007.
- [29] *Demeter Terrain Engine*, Disponível em: <<http://www.tbgssoftware.com/>>. Acesso em jun. 2007.
- [30] Osório, F. S., et al. “Increasing Reality in Virtual Reality Applications through Physical and Behavioural Simulation”, *Proc. of Virtual Concept Conference*, 2006.
- [31] Reis, L. P., “Coordenação em sistemas multi-agente: aplicações na gestão universitária e futebol robótico”, *Tese de Doutorado*, Universidade do Porto, Portugal, 2003.
- [32] Hornik, K., Stinchcombe, M., White, H. “Multilayer feedforward networks are universal approximators”, *Neural Networks*, v. 2, p. 359-366, 1989.
- [33] Smith, R., *Open Dynamics Engine, User Guide*, 2006.
- [34] Marchi, J., “Navegação de robôs móveis autônomos: Estudo e implementação de abordagens”, *Dissertação de Mestrado*, Universidade Federal de Santa Catarina, 2001.
- [35] JPL/NASA. Jet Propulsion Laboratory, Disponível em: <<http://www.robotics.jpl.nasa.gov>>. Acesso em maio 2007.
- [36] Silveira, R. A., “Introdução a Sistemas Multi-agente”, Universidade Federal de Santa Catarina (UFSC), 2006.
- [37] Kartalopoulos, S. V., “Understanding Neural Networks and Fuzzy Logic - Basic Concepts and Applications”, *IEEE Press Understanding Science and Technology Series*, 1996.
- [38] Garcia, A. C. B., Sichman, J. S., *Sistemas inteligentes: Fundamentos e aplicações*. In: Barueri, SP, Brasil: Manole, Cap. Agentes e Sistemas Multi-Agentes, p. 269-306, 2003.
- [39] Bekey, G. A., *Autonomous Robots: From Biological Inspiration to Implementation and Control*, Cambridge, Massachusetts, USA: The MIT Press, 2005.
- [40] Pfeifer, R., Scheier, C., “From perception to action: The right direction?” *Proceedings of the From Perception to Action Conference*, *IEEE Press*, p. 1-11, 1994.
- [41] Pfeifer, R., Iida, F., Bongard, J. “New robotics: Design principles for intelligent systems”, *Artificial Life*, 2005.
- [42] Rumelhart, D., McClelland, J., *Parallel Distributed Processing*, Cambridge: MIT Press, 1986.
- [43] Osório, F. S. “INSS: Un Système Hybride Neuro-Symbolique pour l’Apprentissage Automatique Constructif”, *Tese de Doutorado*, Institut National Polytechnique de Grenoble (INPG), Grenoble, França, 1998.