

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ**

Thiago Steckert

**UMA ARQUITETURA PARA COORDENAÇÃO DE SISTEMAS
MULTIRROBÓTICOS APLICADA AO FUTEBOL DE ROBÔS**

Araranguá, julho de 2013.

Thiago Steckert

**UMA ARQUITETURA PARA COORDENAÇÃO DE SISTEMAS
MULTIRROBÓTICOS APLICADA AO FUTEBOL DE ROBÔS**

Trabalho de Conclusão de Curso
submetido à Universidade Federal de
Santa Catarina, como parte dos
requisitos necessários para a obtenção
do Grau de Bacharel em Tecnologias
da Informação e Comunicação.

Orientador: Prof. Dr. Anderson Luiz
Fernandes Perez.

Araranguá, julho de 2013.

Thiago Steckert

UMA ARQUITETURA PARA COORDENAÇÃO DE SISTEMAS MULTIRROBÓTICOS APLICADA AO FUTEBOL DE ROBÔS

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de Bacharel em Tecnologias da Informação e Comunicação, e aprovado em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, julho de 2013.

Prof. Vilson Gruber, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Anderson Luiz Fernandes Perez, Dr.
Orientador
Universidade Federal de Santa Catarina

Prof. Ricardo Moraes, Dr.
Universidade Federal de Santa Catarina

Prof. Fábio de La Rocha, Dr.
Universidade Federal de Santa Catarina

Dedico este trabalho a minha família e a todos os meus amigos que me apoiaram e me incentivaram, tornando esta jornada mais proveitosa e prazerosa.

AGRADECIMENTOS

Agradeço a Deus pela força e pela vida.

Agradeço aos meus pais que incentivaram e apoiaram meus estudos.

Ao professor e orientador Dr. Anderson Luiz Fernandes Perez que, sempre disposto a auxiliar no desenvolvimento do trabalho, contribuiu para meu crescimento pessoal e intelectual.

Aos professores Ricardo Moraes e Fábio De La Rocha, membros da banca de avaliação, pelas contribuições dadas ao trabalho.

Aos colegas do Laboratório de Automação e Robótica Móvel, sempre dispostos a ajudar e a compartilhar conhecimento.

À Universidade Federal de Santa Catarina, por me conceder ensino de qualidade.

À Andréia Schneider, pelo incentivo e motivação.

E a todos os amigos e colegas de curso que tornaram essa jornada mais prazerosa.

Não tentes ser bem sucedido, tente antes ser um homem de valor.

Albert Einstein

RESUMO

Sistemas robóticos podem realizar tarefas impossíveis de serem realizadas por um ser humano. Sistemas multirrobóticos surgiram para potencializar a capacidade de resolução de problemas dos robôs, além de poder oferecer maior robustez e flexibilidade para as aplicações. Este trabalho descreve uma proposta de arquitetura para a coordenação de sistemas multirrobóticos que atuam de modo cooperativo, em ambientes de tempo real, realizando tarefas fortemente acopladas. Tal arquitetura preocupa-se em definir planos globais de equipe adequados aos objetivos de longo prazo da aplicação. Cada plano é decomposto em tarefas individuais que são alocadas aos robôs que, globalmente, possuem maior aptidão para executá-las. Aplica-se a arquitetura em um ambiente simulado de futebol de robôs, que segue os padrões da *Small Size League* da RoboCup. São realizados testes que avaliam o aproveitamento geral dos recursos e a alocação de tarefas entre os robôs do grupo.

Palavras-Chave: Robótica, Sistemas Multirrobóticos, Futebol de Robôs, Cooperação.

ABSTRACT

Robotic systems can perform impossible tasks performed physically or intellectually by a human being. Multi robot systems have emerged to enhance the ability to solve robot's problems, moreover they can provide strength and flexibility to the applications. This work describes a proposed architecture for the coordination of multi robot systems that operate cooperatively in real time environments, performing strongly coupled tasks. This architecture is concerned with defining global team plans suited to long-term goals of the application. Each plan is broken down into individual tasks that are allocated to the robots that, overall, have higher capability to execute them. Applies the architecture in a simulated robot soccer environment, which follow the standards of RoboCup Small Size League. Tests are performed to evaluate the overall resource utilization and the task allocation among the robots in the group.

Keywords: Robotics, Multi Robot Systems, Robot Soccer, Cooperation.

LISTA DE FIGURAS

Figura 1 - Rover Curiosity	34
Figura 2 - Exemplos de robôs. A) Robô móvel militar Guardium MK III. B) Robô manipulador KUKA. C) Robô híbrido antibomba Mini Andros II.	40
Figura 3 - Fluxo de execução de uma arquitetura de software hierárquica.	43
Figura 4 - Fluxo de execução de uma arquitetura de software reativa.	44
Figura 5 - Fluxo de Execução de uma arquitetura de software híbrida.	45
Figura 6 - Taxonomia para a classificação de SMR.....	49
Figura 7 - Políticas de definição e aplicação de tarefas em SMR.	51
Figura 8 - Tripulação Robótica para Construção (RCC - <i>Robotic Construction Crew</i>).	56
Figura 9 - Grupo de robôs marsupiais para busca e resgate em ambientes urbanos.	57
Figura 10 - Partida de Futebol de Robôs realizada durante a CBR 2011.....	59
Figura 11 - Dimensões em centímetros de um campo oficial da RoboCup SSL.	61
Figura 12 - Mecânica de um robô da equipe RoboFEI, categoria RoboCup SSL.	62
Figura 13 - A e B. Roda holonômica, motor e sistema de redução da equipe Parsian Robotics. C. Chassi com sistema de chute da equipe RoboFEI.	62
Figura 14 - Interface Gráfica do <i>SSL-Vision</i>	64
Figura 15 - Representação de uma partida de futebol de robôs da <i>RoboCup SSL</i>	65
Figura 16 - Classificação da arquitetura conforme a taxonomia proposta por Farinelli et al. (2004).	66
Figura 17 - Arquitetura Híbrida individual.	68
Figura 18 - Arquitetura de Grupo.	70
Figura 19 - Descrição de um plano de equipe.	71
Figura 20 - Início do leilão.	73
Figura 21 - Fim do leilão e alocação de tarefas.	74
Figura 22 - Representação de Pacote de dados.	75
Figura 23 – Interface principal do simulador <i>grSim</i>	77
Figura 24 - Máquina de estados do goleiro.	80
Figura 25 - Máquina de estados do defensor.	81
Figura 26 - Máquina de estados do criador.	82
Figura 27 - Máquina de estados do atacante.	83
Figura 28 - Plano de equipe definido pelo "planejador de missão"	85

LISTA DE TABELAS

Tabela 1 - Classificação geral dos sensores.	36
Tabela 2 - Classificação e Características de Atuadores.	38
Tabela 3 - Comparação entre o Xadrez e o Futebol de Robôs.	58
Tabela 4 - Fatores para a quantificação de utilidade.	84
Tabela 5 - Métricas para quantificação de utilidade.	84
Tabela 6 - Estado dos fatores de quantificação de utilidade do grupo de robôs.	86
Tabela 7 - Matriz de utilidade.	86
Tabela 8 - Matriz de utilidade com alocação com perda de mensagens.	86

LISTA DE ABREVIATURAS E SIGLAS

ACSM - Arquitetura de Coordenação para Sistemas Multirroboicos
AUV - *Autonomous Underwater Vehicle*
CBR - Competição Brasileira de Robótica
CCD - *Charge Coupled Device*
CMOS - *Complementary Metal Oxide Semiconductor*
FIFA - Federação Internacional de Futebol Associado
FIRA - *Federation of International Robot-soccer Association*
GPB - *Google Protocol Buffers*
IA - *Instantaneous Assignment*
NASA - *National Aeronautics and Space Administration*
MT - *Multi-robot Tasks*
MR - *Multi-task Robots*
MRTA - *Multi Robot Task Allocation*
SSL - Small Size League
SMR - Sistema Multirroboico
SR - *Single-robot Task*
ST - *Single-task Robot*
TA - *Time-extended Assignment*
UAV - *Unmanned Air Vehicle*
UDP - *User Datagram Protocol*
ULV - *Unmanned Land Vehicle*

SUMÁRIO

1. INTRODUÇÃO	27
1.1. PROBLEMÁTICA E JUSTIFICATIVA.....	28
1.2. OBJETIVOS.....	29
1.2.1. Objetivo Geral	29
1.2.2. Objetivos Específicos	29
1.3. METODOLOGIA.....	29
1.4. ORGANIZAÇÃO DO DOCUMENTO.....	30
2. FUNDAMENTOS DE ROBÓTICA	32
2.1. BREVE HISTÓRICO DA ROBÓTICA.....	32
2.2. CONCEITOS E DEFINIÇÕES DE ROBÓTICA	34
2.3. PRIMITIVAS DE SISTEMAS ROBÓTICOS	40
2.4. PARADIGMAS DE ARQUITETURAS PARA CONTROLE DE SISTEMAS ROBÓTICOS.....	42
2.4.1. Paradigma Hierárquico	42
2.4.2. Paradigma Reativo	43
2.4.3. Paradigma Híbrido	44
3. SISTEMAS MULTIRROBÓTICOS	47
3.1. FUNDAMENTOS DE SISTEMAS MULTIRROBÓTICOS.....	47
3.2. ARQUITETURAS DE SISTEMAS MULTIRROBÓTICOS ...	50
3.3. TAREFAS EM SISTEMAS MULTIRROBÓTICOS	52
3.4. ALOCAÇÃO DE TAREFAS EM SISTEMAS MULTIRROBÓTICOS.....	52
3.5. DOMÍNIOS DE APLICAÇÕES DE SISTEMAS MULTIRROBÓTICOS.....	54
3.5.1. Futebol de Robôs	57
3.5.2. Small Size League	60
4. ARQUITETURA DE COORDENAÇÃO DE SISTEMAS MULTIRROBÓTICOS	66
4.1. DEFINIÇÕES PRELIMINARES.....	66
4.2. ARQUITETURA INDIVIDUAL	67
4.3. VISÃO GERAL DA ARQUITETURA DE GRUPO.....	69
4.3.1. Planejador de Missão	70

4.3.2. Alocador de Tarefas	72
4.3.3. Protocolo de Comunicação	75
5. ANÁLISE DA ARQUITURA DE COORDENAÇÃO PARA COORDENAÇÃO DE SISTEMAS MULTIRROBÓTICOS.....	77
5.1. DEFINIÇÕES PRELIMINARES.....	77
5.2. CENÁRIO DE APLICAÇÃO DA ARQUITETURA AO FUTEBOL DE ROBÔS	85
6. CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS.....	88
6.1. PROPOSTAS PARA TRABALHOS FUTUROS.....	89
REFERÊNCIAS	90

1. INTRODUÇÃO

O desenvolvimento da tecnologia em conjunto com a necessidade de automatização dos meios de produção para o aumento da produtividade e qualidade foram responsáveis pela inserção e popularização da robótica nas mais diversas áreas. Robôs foram idealizados para a realização de tarefas perigosas, insalubres e maçantes, assim como tarefas impossíveis de serem realizadas por um ser humano. Em relação ao ser humano, eles podem realizar serviços com maior eficiência e rapidez, além de ter custo operacional mais baixo.

Entretanto, existem tarefas que são complexas demais para serem realizadas por um único robô. Outras podem ser realizadas com maior efetividade por um grupo de robôs trabalhando de forma cooperativa. Além disso, a utilização de um grupo de robôs pode aumentar a robustez da aplicação por meio da redundância, flexibilidade e tolerância a falhas. Estes e outros fatores foram responsáveis pelo surgimento e pesquisas em sistemas multirrobóticos.

Tarefas inerentemente distribuídas, tanto em espaço, quanto em tempo ou funcionalidade podem beneficiar-se do uso de sistemas multirrobóticos. Dentre os problemas do mundo real é possível citar os que são compostos por tarefas de grande escala, onde grandes áreas de espaço devem ser percorridas, como por exemplo, exploração planetária e subaquática, detecção e combate a incêndios, busca e resgate de pessoas e remoção de resíduos tóxicos e também os que são compostos por tarefas complexas que necessitam do trabalho coordenado de mais de um robô, como por exemplo o transporte e manipulação cooperativa de objetos e a observação de múltiplos alvos.

Sistemas multirrobóticos oferecem vantagens em termos de custo e tempo de execução. Um grupo de robôs pode cobrir uma demanda em um tempo muito menor que o necessário para um único robô. Tratando-se de tarefas complexas, um único robô exigiria maior capacidade de processamento, maior autonomia e robustez do que seria necessário para cada robô em um sistema multirrobótico. Outra grande vantagem está na redundância oferecida, pois se um robô falha, o restante do grupo pode continuar a execução da tarefa.

Este trabalho visa propor e desenvolver uma arquitetura para coordenação de sistemas multirrobóticos que seja capaz de gerenciar de forma inteligente um sistema multirrobótico cooperativo, composto por robôs autônomos heterogêneos, que atuam em um ambiente dinâmico, na realização de tarefas fortemente acopladas. Ela será aplicada e testada em uma plataforma de futebol robótico simulado, que se baseia nas

regras da *Small Size League* (SSL), uma categoria de competição de robôs especificada pela *RoboCup* (ROBOCUP, 2013).

1.1. PROBLEMÁTICA E JUSTIFICATIVA

A utilização de robôs autônomos para a resolução de problemas da sociedade é um desafio bastante complexo. Uma das principais dificuldades está em como controlá-los e coordená-los de maneira eficiente. Muitas tarefas são de nível crítico, não permitem falhas e exigem um controle e execução precisos das ações. Existe também, a necessidade de controlar diversos sensores e atuadores simultaneamente em tempo real, tendo de lidar com situações inesperadas e com a presença de ruído (KORTENKAMP; SIMMONS, 2008).

Faria (2006), explica que a complexidade de um sistema de controle robótico cresce ainda mais quando se está lidando com múltiplos robôs. Pois além de ter de lidar com as questões referentes ao controle individual de cada robô, deve-se preocupar-se também com questões globais que referem-se a interação e resolução de conflitos entre os robôs. Aplicações envolvendo sistemas multirrobóticos exigem uma arquitetura robusta e versátil que opere o sistema de modo coerente na execução das tarefas. Para isto é necessário que sejam definidos políticas de coordenação que devem determinar como as tarefas serão decompostas e alocadas a cada robô, o modo como o conhecimento e modelagem de ambiente serão representados além do modo como ocorrerá a comunicação entre o grupo.

O futebol de robôs é um desafio complexo que proporciona uma rica plataforma padrão de teste e desenvolvimento para sistemas multirrobóticos. Sua problemática consiste em partidas de futebol, realizadas por robôs móveis autônomos, com regras semelhantes as impostas pela FIFA (Federação Internacional de Futebol Associado). Neste contexto estão inseridos conceitos de inteligência artificial, sistemas distribuídos, visão computacional, comunicação entre agentes, além da própria robótica e demais disciplinas que a compõem.

No futebol de robôs, mais especificamente na categoria SSL, a maioria das arquiteturas de controle e coordenação são organizadas de maneira centralizada, onde uma estação de controle é responsável por controlar robôs escravos, que obedecem ordens, possuindo pouco planejamento local. A aplicação de uma arquitetura de coordenação hierárquica ao ambiente de futebol de robôs possibilitará eliminar a utilização da estação de controle, sem abdicar das vantagens do planejamento global centralizado e da robustez e flexibilidade de

aplicações distribuídas. Um robô poderá ser definido remotamente como líder por meio de um operador que supervisiona a aplicação, ou ainda os robôs podem eleger um líder por meio de um algoritmo de eleição. O líder será responsável pelo planejamento global e atribuição de tarefas, além de seu próprio planejamento individual.

1.2. OBJETIVOS

Os objetivos deste trabalho estão divididos em geral e específicos e serão descritos a seguir.

1.2.1. Objetivo Geral

Propor uma arquitetura para a coordenação de sistemas multirrobóticos que atuam de modo cooperativo, em ambientes de tempo real, na realização de tarefas fortemente acopladas e avaliá-la, aplicando-a ao futebol de robôs.

1.2.2. Objetivos Específicos

Têm-se como objetivos específicos:

- a) Levantar o estado da arte sobre robótica, sistemas de multirrobóticos e futebol de robôs;
- b) Estudar as técnicas de controle e coordenação, e os paradigmas de arquiteturas aplicados a robôs móveis e sistemas de múltiplos robôs;
- c) Propor uma arquitetura de coordenação para múltiplos robôs móveis aplicando-a a um ambiente de futebol de robôs;
- d) Realizar testes com o sistema desenvolvido em ambiente simulado a fim de avaliar a arquitetura proposta.

1.3. METODOLOGIA

Este trabalho consiste em uma pesquisa de natureza aplicada. Como parte inicial do projeto, pode-se citar a caracterização do problema de pesquisa. Sua problemática remete ao planejamento global

e alocação efetiva de tarefas a sistemas multirrobóticos. Após esta etapa são realizadas pesquisas bibliográficas sobre os temas abordados, a fim de levantar o estado da arte de robótica, sistemas multirrobóticos e futebol de robôs para que seja possível propor a arquitetura de coordenação.

A arquitetura proposta será organizada de maneira hierárquica, onde um robô, definido como líder, é responsável por tomar as decisões de nível global, sendo de responsabilidade de cada robô seu controle local. A arquitetura destina-se a grupos de robôs heterogêneos, que atuam de modo cooperativo, realizando tarefas fortemente acopladas. Para a alocação de tarefas será utilizado uma metodologia baseada em mercado, onde os robôs, através de comunicação explícita, negociarão as demandas com base em quantificadores de utilidade.

A arquitetura de software projetada pode ser considerada como aberta, pois foi baseado em plataformas de desenvolvimento não proprietárias. As ferramentas utilizadas para o desenvolvimento do trabalho foram Linux, linguagem de programação C++ e bibliotecas de código aberto. O sistema foi testado no simulador de código aberto grSim (MONAJJEMI et al., 2012), que simula o ambiente de uma partida da *RoboCup*, na categoria SSL. Os testes foram realizados levando-se em conta o desempenho das técnicas utilizadas em relação ao planejamento global e alocação efetiva de tarefas a um grupo de robôs em cenários de uma partida de futebol de robôs. Com base nos testes foi possível identificar e propor melhorias para serem implementadas em trabalhos futuros.

1.4. ORGANIZAÇÃO DO DOCUMENTO

Este trabalho está dividido em 6 (seis) capítulos, incluindo esta introdução. No Capítulo 2 apresenta-se uma revisão de literatura sobre os fundamentos da robótica, os quais incluem um breve histórico, definições, conceitos e motivações para sua pesquisa. Também são descritos os principais conceitos envolvidos no processo de controle de robôs, como as primitivas da robótica: percepção, planejamento e atuação, e os paradigmas de inteligência em arquiteturas de controle de sistemas robóticos.

O Capítulo 3 aborda os principais fundamentos de sistemas multirrobóticos. Dentre estes são apresentadas as abordagens de arquiteturas de grupo, o conceito e os tipos de tarefas, a problemática de alocação de tarefas em sistemas multirrobóticos assim como os principais domínios de aplicações destes sistemas. Também é descrito o

futebol de robôs, um desafio que oferece um domínio de testes para a investigação e pesquisa de sistemas multirrobóticos.

O Capítulo 4 descreve a arquitetura proposta, decompondo-a em duas partes: a arquitetura individual e a arquitetura de grupo. A arquitetura de grupo refere-se as estratégias globais de coordenação, troca de informações e atribuições de tarefas entre os robôs, enquanto arquitetura individual esta focada na divisão dos módulos de responsabilidades individuais de cada robô.

O capítulo 5 apresenta os resultados da aplicação da arquitetura de coordenação proposta no contexto do futebol de robôs. O ambiente de testes será o simulador *grSim*, que segue as regras e padrões da *RoboCup Small Size League*. Com a simulação de situações reais de uma partida será possível analisar o aproveitamento geral dos recursos e a alocação de tarefas entre os robôs da Arquitetura de Coordenação de Sistemas Multirrobóticos (ACSM), proposta no Capítulo 4.

Por fim, no Capítulo 6 apresentam-se as considerações finais e algumas sugestões para trabalhos futuros.

2. FUNDAMENTOS DE ROBÓTICA

Neste capítulo apresenta-se uma revisão de literatura sobre os fundamentos da robótica, os quais incluem um breve histórico, definições, conceitos e motivações para sua pesquisa. Também são descritos os principais conceitos envolvidos no processo de controle de robôs, como as primitivas da robótica: percepção, planejamento e atuação, e os paradigmas de inteligência em arquiteturas de controle de sistemas robóticos.

2.1. BREVE HISTÓRICO DA ROBÓTICA

A palavra "robô" surgiu em 1920, na peça teatral "Universal Rossum Robots (R.U.R.)" escrita pelo dramaturgo tcheco Karel Capek. Seu significado deriva da palavra eslava "Robota" que significa trabalho subordinado. A peça apresentava seres humanóides dotados de inteligência que eram fabricados para trabalhar para seres humanos. O termo "Robótica" foi idealizado e popularizado com obras de ficção científica do escritor Isaac Asimov. Em seu livro de ficção científica Eu Robô, lançado em 1950, ele cria as leis da robótica, que segundo ele deveriam ser respeitadas por todos os robôs (STONE, 2010).

A Segunda Guerra Mundial impulsionou o surgimento de dois conceitos importantes da robótica: o sensoriamento artificial e o controle autônomo. No sensoriamento artificial podem-se citar o radar, instrumento utilizado para rastrear e detectar a presença de inimigos, e tanques americanos com sistemas de detecção de minas explosivas. Já o controle autônomo, era realizado por bombas teleguiadas alemãs, que eram capazes de corrigir sua trajetória em pleno voo (SOLUTIONS, 2010).

Em 1942 os EUA iniciaram o projeto Manhattan, para o desenvolvimento de bombas nucleares. Um dos problemas que os cientistas e engenheiros encontraram estava na manipulação e processamento de materiais radioativos, como o Urânio e o Plutônio que ofereciam riscos graves a saúde dos trabalhadores. Foi necessário desenvolver um mecanismo que isolasse o ser humano do contato direto com tais elementos. A solução para o problema foram os dispositivos teleoperadores (MURPHY, 2000a). Eles consistiam em braços escravos projetados de forma a imitar a mecânica do braço humano, possuíam pouca percepção do ambiente onde atuavam e seu controle era muito primitivo.

Com o fim da Segunda Guerra Mundial, vários países passaram a se interessar na produção de armas nucleares e na exploração da energia nuclear. Devido a necessidade de desenvolver dispositivos para manipulação e aos avanços nas teorias de controle, começaram a surgir os primeiros robôs manipuladores. O sucesso da automação na indústria nuclear despertou o interesse de usar estes dispositivos para a indústria de produção geral (MURPHY, 2000a).

Os primeiros robôs, construídos aproximadamente na década de 1960, foram resultado da junção e convergência de duas tecnologias: as máquinas de controle numérico para fabricação de peças de alta precisão e os teleoperadores para manuseio remoto de material radiativo (SICILIANO; KATHIB, 2008). Na indústria dois tipos tornaram-se comuns, os braços robóticos manipuladores, que são capazes de realizar movimentos repetitivos com elevado grau de precisão e rapidez e veículos autônomos para transporte de peças em linhas de produção.

Na metade da década de 1960, o Instituto de pesquisa de Stanford desenvolveu o Shakey, o primeiro robô móvel autônomo controlado por inteligência artificial. Ele possuía sensores táteis e de visão, contava com um sistema de navegação e era capaz de desviar de obstáculos e tomar suas próprias decisões (ANGELO, 2007).

Nesta época a robótica passou a ter papel importante em pesquisas de exploração espacial. A NASA (*National Aeronautics and Space Administration*) buscava desenvolver plataformas para exploração de terrenos hostis, pois enviar astronautas a planetas remotos e trazê-los com segurança de volta a terra seria uma tarefa muito complexa. Desta forma passaram a utilizar sondas espaciais não tripuladas, que coletam e transmitem informações de volta para a terra. No programa de exploração *Mariner*, sondas espaciais foram utilizadas para coletar dados do planeta Vênus em 1962. No programa *Surveyor*, sondas aterrisadoras (*landers*), foram enviadas para a lua para coletar dados de sua superfície. Mais recentemente, a NASA passou a utilizar veículos de exploração, denominados rovers, como o *Sojourner*, o *Opportunity* e o *Curiosity* (KURFESS, 2010), (PEDERSEN et al., 2003). A Figura 1 ilustra o rover *Curiosity*, enviado para Marte com o objetivo de investigar o clima e geologia daquele planeta.

Figura 1 - Rover Curiosity



Extraído de: Grotzinger et al. (2012).

Ao final dos anos 70, com o avanço tecnológico na mecatrônica, a criação dos circuitos integrados, a produção de componentes miniaturizados e de peças com maior precisão assim como a necessidade de aumentar a produtividade e qualidade dos meios de produção, os robôs passaram a ser componentes essenciais na automação de sistemas de produção, expandindo-se para outras áreas da indústria como, por exemplo, a de alimento, metal, química e eletrônica, dando origem ao termo "robô industrial" (SICILIANO; KATHIB, 2008).

Com desenvolvimento da computação, passou-se a buscar o elo de ligação entre a inteligência humana e as máquinas dando início a uma era de grande produtividade e avanços em pesquisas na área de Inteligência Artificial possibilitando que robôs fossem projetados para realizarem tarefas cada vez mais complexas, que exigem maiores níveis de autonomia. Nas últimas décadas, com o avanço da robótica móvel, foram criadas aplicações para os robôs em diversas áreas, como medicina, geografia, segurança e instalações militares.

2.2. CONCEITOS E DEFINIÇÕES DE ROBÓTICA

Segundo Niku (2011), a robótica é a arte de projetar, aplicar e utilizar robôs ou sistemas robóticos em empreendimentos humanos. Por ser interdisciplinar, envolve conhecimentos de engenharias mecânica,

elétrica e eletrônica, ciências da computação, ciências cognitivas, biologia e varias outras áreas. O termo sistema robótico refere-se ao robô e aos dispositivos e sistemas que são utilizados em conjunto com este.

Russell e Norvig (2004) definem robôs ou agentes robóticos como agentes físicos que interagem em ambientes reais com o objetivo de realizar uma tarefa. Para tanto, eles são dotados de sensores, que realizam a percepção do ambiente, atuadores e efetuares, que possibilitam que eles exerçam forças físicas sobre o ambiente em que irão atuar e controladores, onde softwares responsáveis pela tomada de decisão e gerenciamento do hardware são executados.

Tabela 1 - Classificação geral dos sensores.

Classificação geral	Sensor	PC ou EC	A ou P
Sensores Táteis (detecção de contato físico ou proximidade, interruptores de segurança)	Interruptores de contato Barreiras ópticas Sensores de proximidade	EC EC EC	P A A
Sensores de velocidade e posição de roda/motor	Encoder de escova Potenciômetro Synchros, resolvedores Encoder óptico Encoder magnético Encoder indutivo Encoder capacitivo	PC PC PC PC PC PC PC	P P A A A A A
Sensores de rumo (orientação do robô em relação a um quadro de referência fixo)	Compasso Giroscópio Inclinômetro	EC PC EC	P P A/P
Balizas terrestres (localização em um quadro de referência fixa)	GPS Ópticos ativos ou balizas RF Balizas ultrassônicas ativas Balizas refletivas	EC EC EC EC	A A A A
Sensores de alcance ativo (refletividade, tempo de voo e triangulação geométrica)	Sensor de refletividade Sensor ultrassônico Telêmetro laser Triangulação óptica (1D) Luz estruturada (2D)	EC EC EC EC EC	A A A A A
Sensores de movimento e velocidade (velocidade relativa de um objeto fixo ou em movimento)	Radar doppler Som doppler	EC EC	A A
Sensores baseados em visão	Câmera CCD/CMOS	EC	P

Legenda: A - ativo; P - passivo; A/P - ativo e passivo; PC - proprioceptivo; EC - exteroceptivo.

Adaptado de: Siegwart e Nourbakhsh (2004).

Os sensores são responsáveis por extrair dados do ambiente, que são interpretados e transformados em informações úteis. De acordo com Siegwart e Nourbakhsh (2004), eles podem ser classificados com base em 2 (dois) parâmetros: o primeiro refere-se ao domínio de atuação do sensor, se ele atua captando dados internos ao robô ele pode ser classificado como proprioceptivo, mas se os dados medidos são externos ao robô, relativos ao mundo onde atua, ele é classificado como exteroceptivo; o segundo parâmetro refere-se a atuação do sensor no ambiente, se ele capta energia do ambiente ele é considerado passivo, mas se ele emite energia ao ambiente ele é considerado ativo. A Tabela 1 mostra uma relação dos principais tipos de sensores e suas classificações.

Os controladores provêm o hardware e o software de controle, que tornam o robô autônomo. Mataric (2007), explica que um controlador pode utilizar tanto informações obtidas pela leitura de sensores como informações definidas em sua memória para realizar tomadas de decisão, controlar e gerenciar seus atuadores. Geralmente um robô possui vários controladores, pois além de um controlador principal, muitos de seus sensores e atuadores possuem um controlador dedicado.

Tabela 2 - Classificação e Características de Atuadores.

Classificação Geral	Atuador	Vantagens	Desvantagens
Eletromecânico	Motor de corrente alternada (AC) Motor de corrente contínua (DC) Motor de Passo Servo Motor	Controle; Precisão; Confiabilidade; Pouca manutenção.	Necessita sistemas de freio e redução; Baixa rigidez
Hidráulico	Motor Hidráulico Válvulas	Maior relação força/peso; Suporta cargas pesadas; Não necessita redução	Custo; Ruidoso; Necessita de bomba, reservatório, mangueiras; Manutenção frequente
Pneumático	Compressor de ar Válvulas	Custo; Simplicidade;	Baixa Precisão; Baixo poder para cargas;

Adaptado de: Niku (2011).

Os atuadores são mecanismos que, conectados as juntas e articulações dos robôs, são responsáveis por gerar a força necessária para movimentar o sistema robótico, agindo como músculos. Eles podem ser classificados de acordo com a fonte de energia a qual são alimentados conforme ilustra a Tabela 2.

Os atuadores de um robô devem ter potência suficiente para acelerar e desacelerar suas articulações, suportando a quantidade de carga exigida em sua aplicação. Devem ser leves, econômicos, precisos, ágeis, confiáveis e de fácil manutenção (NIKU, 2011).

Segundo Mataric (2007), os efetuadores são dispositivos que permitem ao robô produzir um efeito, impacto ou influência no ambiente. Eles atuam em conjunto com os atuadores, interfaceando-os com o mundo físico. Possuem duas funções principais, a locomoção em torno do ambiente de atuação e a manipulação de objetos. Efetuadores

podem ser rodas, garras, dedos, asas e várias outras partes do corpo semelhantes as do ser humano e animais.

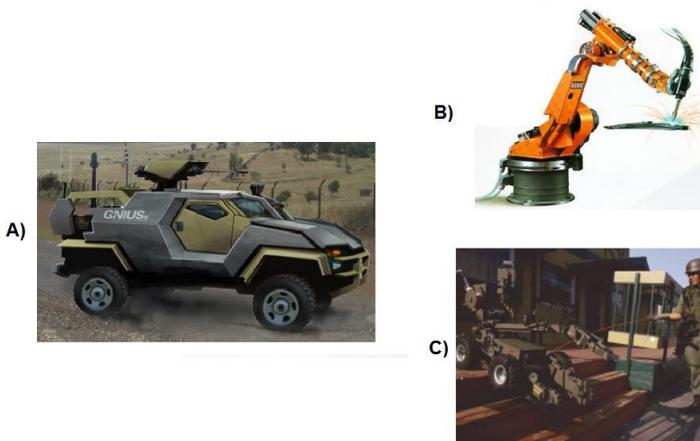
Um robô distingue-se de um mecanismo remotamente controlado devido a sua autonomia e diferencia-se de um computador devido a sua capacidade de agir no mundo físico. De acordo com Berry (2012), o nível de autonomia de um robô é definido com base na necessidade de interação deste com um ser humano, que pode variar de acordo com a sua aplicação.

Um robô pode ser semiautônomo ou totalmente autônomo, sendo que existem dois tipos de níveis semiautônomo: supervisão e compartilhado. No nível de supervisão o ser humano atribui tarefas ao robô e o observa a execução das mesmas sem intervenção. Já no modo compartilhado atribui-se tarefas, podendo intervir com novas entradas de dados ou cancelar a execução. Por fim, no nível totalmente autônomo não há nenhum tipo de intervenção humana, o robô percebe, planeja e atua de forma independente.

Os robôs atuais classificam-se em 3 (três) categorias principais: manipuladores, móveis e híbridos. Os robôs manipuladores caracterizam-se por estarem fixados no ambiente em que atuam e, geralmente, possuem um conjunto de articulações controláveis que os permitem atuarem em qualquer posição dentro de sua área de trabalho.

Os robôs móveis possuem a capacidade de se locomover em seu ambiente por meio de rodas, pernas ou mecanismos similares. Eles podem ser classificados em 3 (três) subcategorias, com base no meio em que atuam: veículo terrestre não tripulado (ULV - *Unmanned Land Vehicle*), veículo aéreo não tripulado (UAV - *Unmanned Air Vehicle*) e veículo subaquático autônomo (AUV - *Autonomous Underwater Vehicle*). A terceira categoria é composta por robôs híbridos que são, basicamente, robôs móveis equipados com dispositivos manipuladores, nesta categoria estão presentes os robôs humanoides (RUSSELL; NORVIG, 2004). A Figura 2 ilustra um exemplo de cada uma destas categorias.

Figura 2 - Exemplos de robôs. A) Robô móvel militar Guardian MK III. B) Robô manipulador KUKA. C) Robô híbrido antibomba Mini Andros II.



Adaptados de: A) Kuka (2013), B) G-Nius (2013), C) Kumagai (2002).

2.3. PRIMITIVAS DE SISTEMAS ROBÓTICOS

As funções realizadas por robôs podem ser classificadas em (MURPHY, 2000a): Percepção, Planejamento e Atuação. A percepção é caracterizada como o processo de agentes robóticos extraírem e interpretarem informações do mundo onde atuam. É realizada por meio de sensores que captam algum tipo de informação útil do ambiente servindo de entrada para um determinado algoritmo de controle. Estas podem ser utilizadas com base em duas abordagens. A primeira é a extração de características, na qual, a partir da entrada sensória, são detectadas informações relevantes sobre determinados atributos do ambiente. A segunda abordagem é baseada em modelo, onde são utilizados os dados de entrada para a criação de um modelo de mundo (RUSSELL; NORVIG, 2004).

O planejamento é o ato de definir e organizar um curso de ações voltadas para atingir uma determinada meta da maneira mais adequada. É indispensável para a realização de tarefas e objetivos complexos e ocorre, geralmente, antes e durante o processo de atuação. Por ser complicado, consumir tempo e ter um custo que pode ser alto, muitas vezes o custo benefício não permite atingir o planejamento ideal, sendo viável chegar apenas a soluções aproximadas (GHALLAB et al., 2004).

LaValle (2006) argumenta que o planejador pode ser uma máquina, com a execução de algoritmos de planejamento, ou um ser humano, que calcula um plano por meio de entradas. Este plano irá especificar ações de baixo nível que o robô deverá seguir, e após ser definido, ele poderá ser utilizado de 3 (três) maneiras. Pode ser executado em um ambiente de simulação ou no mundo real em um robô, pode ser refinado, para atingir um plano melhor, ou então incluso em uma estrutura hierárquica tornando-se parte de um plano global.

Para a definição de um plano de ações deve-se levar em consideração diversos fatores. Dentre eles estão o estado inicial e todos os demais estados possíveis de serem atingidos pelo agente, as ações que podem ser tomadas e o período de tempo em que estas devem ser definidas e executadas, além da meta global a ser alcançada. LaValle (2006) afirma ainda que para a definição de um plano são seguidos dois diferentes tipos de critérios: a viabilidade e a otimalidade. A viabilidade leva em conta o custo benefício da solução final e a otimalidade refere-se ao nível de desempenho necessário para atingi-la.

Dentre os vários tipos de planejamento utilizados na robótica, Kröger (2010), cita os planejamentos de caminho, trajetória e movimento. O planejamento de caminho consiste em encontrar um caminho livre de colisão entre a posição inicial do robô e um determinado alvo. No planejamento de trajetória definem-se as velocidades e acelerações que devem ser utilizadas ao longo do caminho planejado. O planejamento de movimento é a junção dos planejamentos de caminho e trajetória. Ele pode ser definido de modo *on-line* ou *off-line*. No modo *off-line* o planejamento ocorre antes da execução, estaticamente, já no modo *on-line* o plano é calculado dinamicamente, ao longo de sua execução.

A atuação baseia-se em funções que geram comandos para o controle dos atuadores. Segundo Ge e Lewis (2006), para que um sistema robótico possa ser considerado autônomo e inteligente, é fundamental que as camadas de baixo nível respondam com sucesso as ações planejadas pelas camadas de alto nível. Para isto cada robô deve ser capaz de controlar de maneira efetiva o hardware que possui. Um dos principais desafios sobre a atuação, é realizar o controle preciso dos atuadores para que o robô consiga por em prática no mundo físico a movimentação planejada (MORIN; SAMSON, 2008).

Heinen (2002) cita as técnicas de controle *Open Loop*, *Feedback* e *Feedforward* para o controle dos atuadores. A técnica de controle *Open Loop* não faz uso de sensores em seus atuadores, fazendo o controle por meio de cálculos baseados em seu modelo físico. Na

abordagem de *Feedback* faz-se uso de sensores para monitoramento dos atuadores, que passam a ser ajustados de acordo com algum sinal de retorno destes. No método *Feedforward* utiliza-se sensores para monitoramento do ambiente, que retornam informações úteis para o controle de movimentação.

2.4. PARADIGMAS DE ARQUITETURAS PARA CONTROLE DE SISTEMAS ROBÓTICOS

Russell e Norvig (2004) definem que uma arquitetura de software consiste em uma metodologia para estruturar algoritmos que, além de incluir as linguagem e ferramentas utilizadas em seu desenvolvimento, segue uma filosofia global que define como o sistema é estruturado. Segundo estes autores as arquiteturas de software modernas para robótica devem definir o modo pela qual são combinados o controle reativo e o controle deliberativo baseado em modelos.

Arquiteturas de software para sistemas robóticos visam facilitar o desenvolvimento, concepção e implementação de sistemas robóticos, que separados em unidades modulares possuem um grau maior de inteligibilidade e reutilização, facilitando os testes e validação. A arquitetura deve descrever os módulos e subsistemas em que o sistema está dividido, a interação entre eles e os conceitos computacionais na qual ela se baseia (KORTENKAMP; SIMMONS, 2008).

Segundo Murphy (2000a) uma arquitetura é um método de implementar um paradigma, de modo a incorporar seus princípios de forma concreta. Um paradigma de arquitetura refere-se a como a arquitetura de software é estruturada. Sua correta aplicação possibilita que problemas sejam solucionados de maneira mais fácil. Tais paradigmas podem ser descritos de duas formas. A primeira baseia-se na relação entre três primitivas da robótica, Percepção, Planejamento e Atuação. O segunda refere-se a maneira que as informações percebidas são processadas e distribuídas pelos módulos do sistema. Os três paradigmas de arquiteturas de sistemas robóticos mais comuns são: o hierárquico, descrito na Seção 2.4.1, o reativo, descrito na Seção 2.4.2 e o híbrido, descrito na Seção 2.4.3.

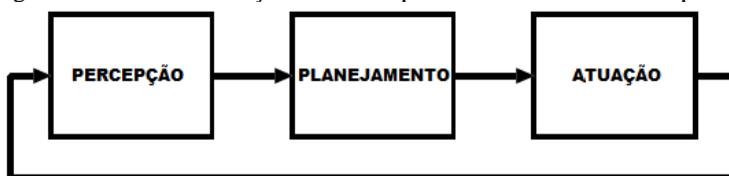
2.4.1. Paradigma Hierárquico

O paradigma hierárquico é o mais antigo dos métodos de organização de arquiteturas de sistemas robóticos. Nele as três primitivas da robótica são organizadas de maneira sequencial.

Primeiramente ocorre a percepção, onde o robô, por meio de seus sensores, capta informações do ambiente para que este módulo construa um modelo de mundo global. Este modelo de mundo consistirá em uma estrutura que irá representar o ambiente de atuação, contendo informações sobre o estado dos robôs presentes neste. A partir disto o módulo de planejamento define um plano com as ações necessárias para atingir o objetivo proposto e o módulo de atuação realiza essas ações (NAKHAEINIA et al., 2011).

A Figura 3 ilustra o fluxo de execução de sistemas que seguem o paradigma hierárquico onde cada módulo é executado sequencialmente. Tal fluxo de execução é estruturado de maneira horizontal.

Figura 3 - Fluxo de execução de uma arquitetura de software hierárquica.



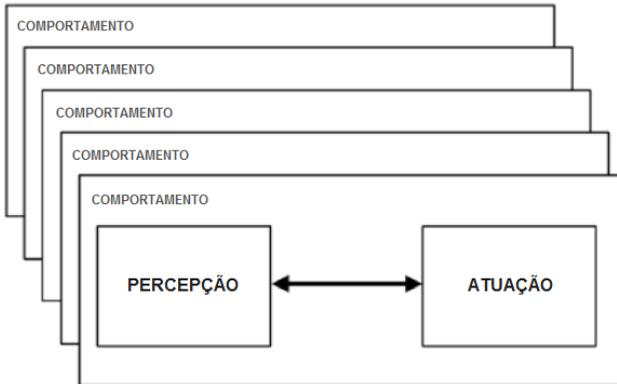
Adaptado de: Murphy (2000a).

Segundo Brooks (1986) esta abordagem pode ser desvantajosa para aplicações que realizem tarefas de navegações em ambientes dinâmicos que exijam um tempo de resposta rápido, pois, se o estado do ambiente é modificado constantemente durante o fluxo de execução e se houver atrasos consideráveis durante a percepção e o planejamento, a execução do plano de ações pode resultar em uma atuação inconsistente com a tarefa planejada.

2.4.2. Paradigma Reativo

Este paradigma foi proposto por Brooks (1986) para atender aplicações que exijam um tempo de resposta rápido em ambientes dinâmicos. Uma de suas características é a inexistência de um módulo de planejamento global. Conforme ilustrado na Figura 4 este paradigma baseia-se na estruturação vertical de comportamentos, que são dispostos em níveis e podem ser executados sequencial ou paralelamente. Cada comportamento consiste em ações de resposta a estímulos sensórios e possui acesso direto e independente aos sensores e atuadores, sendo ativado ao longo de interações com o ambiente (NAKHAEINIA et al, 2011).

Figura 4 - Fluxo de execução de uma arquitetura de software reativa.



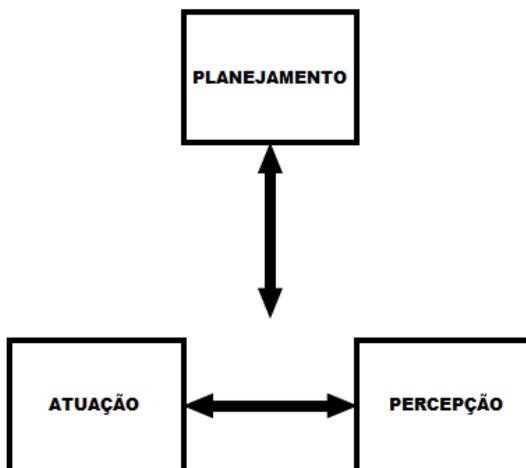
Adaptado de: Murphy (2000a).

A vantagem deste método está no tempo de processamento e resposta que tendem a ser menores, pois não há a necessidade de se estabelecer um modelo de mundo, os comportamentos em geral são simples e estão diretamente ligados aos módulos de percepção e atuação. Porém devido ao fato de inexistir um módulo de planejamento, a execução de tarefas mais complexas torna-se limitada (FARIA, 2006).

2.4.3. Paradigma Híbrido

Segundo Murphy (2000a), o paradigma híbrido surgiu a partir da necessidade de integrar um módulo de planejamento às arquiteturas de sistemas robóticos sem abdicar de características reativas, de modo a possibilitar o planejamento de tarefas mais elaboradas em ambientes complexos e dinâmicos sem prejudicar o tempo de resposta. A Figura 5 ilustra a organização do paradigma híbrido. Nele o Planejamento é dissociado do fluxo de execução de tempo real, e passa a atuar no planejamento de tarefas de longo prazo, enquanto comportamentos reativos são responsáveis por tarefas de tempo real.

Figura 5 - Fluxo de Execução de uma arquitetura de software híbrida.



Adaptado de: Murphy (2000a).

Existem quatro principais estratégias de implementação do paradigma híbrido. A primeira é a de seleção, onde o módulo de planejamento configura os comportamentos e os parâmetros utilizados por estes durante sua execução. A segunda é a de assessoria, na qual o planejamento sugere cursos de ações e o controle reativo determina o mais adequado. A terceira é a de adaptação, em que o planejador realiza alterações nos componentes reativos do robô de acordo com as mudanças no ambiente e as tarefas exigidas. A quarta estratégia é a de postergação, nela o planejador adia ao máximo a tomada de decisões visando proporcionar um curso de ação efetivo (ARKIN, 1999).

Os principais níveis que uma arquitetura híbrida possui são, o sequenciador, o gerente de recursos, o cartógrafo, o planejador de missão e o monitor de desempenho e resolução de problemas. O sequenciador é responsável por gerar um conjunto de comportamentos e determinar sua sequência e condições de sua ativação. O gerente de recursos atua na alocação de recursos aos comportamentos. O cartógrafo cria, armazena e gerencia o mapa de mundo. O planejador de missão, com base na tarefa a ser realizada, constrói um plano de ações e o monitor de desempenho e resolução de problemas percebe se o sistema está tendo êxito em sua tarefa (MURPHY, 2000a).

A utilização isolada de abordagens reativas ou hierárquicas podem não trazer resultados satisfatórios, pois cada uma delas atende a diferentes conjuntos de complexidades da robótica. Portanto para ter-se um sistema robusto, inteligente e flexível deve-se considerar as duas abordagens, sendo que o principal desafio de arquiteturas híbridas está na integração destas (ARKIN, 1999).

3. SISTEMAS MULTIRROBÓTICOS

Este capítulo aborda os principais fundamentos de sistemas multirrobóticos. Dentre estes são apresentadas as abordagens de arquiteturas de grupo, o conceito, os tipos e a problemática de alocação de tarefas em sistemas multirrobóticos assim como os principais domínios de aplicações destes sistemas. Também é descrito o futebol de robôs, um desafio que oferece um domínio de testes para a pesquisa de sistemas multirrobóticos.

3.1. FUNDAMENTOS DE SISTEMAS MULTIRROBÓTICOS

O interesse por pesquisas em sistema multirrobóticos surgiu por várias razões. Dentre elas, Parker (2008), destaca que algumas tarefas são complexas demais para serem realizadas por um único robô. Outras podem ser realizadas por um único robô, mas seriam executadas com maior eficiência por meio do paralelismo gerado pela utilização de múltiplos robôs. Além disso, o uso de uma equipe de robôs pode aumentar a robustez do sistema por meio da redundância, pois se um membro falha ou se sobrecarrega, outros podem compensá-lo devido ao balanceamento de carga.

Um Sistema Multirrobótico (SMR) é formado por uma coleção ou grupo de dois ou mais robôs autônomos que trabalham em conjunto para alcançar metas bem definidas. Neste tipo de sistema, os robôs são menos capazes como entidades individuais, mas atuando em cooperação ou de forma paralela podem realizar tarefas com maior eficiência em relação a um único robô (GAUTAM; MOAHN, 2012).

Mataric (2007) afirma que existem diferentes tipos de SMR que podem ser classificados com base nas características dos indivíduos que o compõem e no tipo de estratégia de coordenação que utilizam. Quanto às características de seus membros, um grupo pode ser classificado como homogêneo, caso todos os robôs sejam idênticos, possuindo a mesma forma e sendo capazes de realizar os mesmos comportamentos, ou heterogêneo, quando os membros possuírem formas ou comportamentos distintos. Quanto a estratégia de coordenação, um grupo pode apenas coexistir, ser fracamente acoplado ou ser fortemente acoplado.

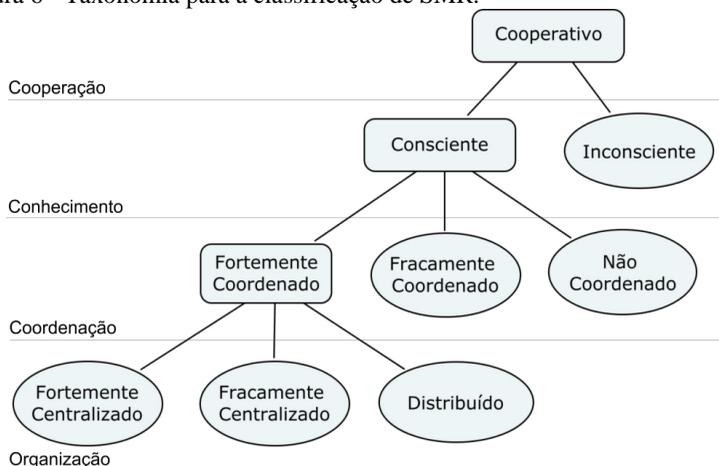
Em um grupo que apenas coexiste, apesar de seus membros estarem partilhando tarefas no mesmo ambiente, os robôs não se intercomunicam e tratam uns aos outros como obstáculos. Em grupos fracamente acoplados os robôs reconhecem o restante do grupo e podem

até seguir comportamentos coordenados, mas não dependem uns dos outros para completar as tarefas. Isto torna o sistema robusto, pois caso um membro do grupo seja removido ou falhe o comportamento de outros membros não será afetado. Na abordagem de coordenação fortemente acoplada os robôs se intercomunicam e se coordenam de modo preciso durante a realização das tarefas. Isto aumenta o desempenho da equipe, mas diminui a sua robustez, pois a falha ou remoção de um robô afeta diretamente seu desempenho.

Farinelli et al. (2004), conforme ilustrado na Figura 6, propõem uma taxonomia para a classificação de SMR. A taxonomia baseia-se em duas dimensões: Dimensões de Coordenação e Dimensões de Sistema, cada uma composta por 4 (quatro) níveis:

- **Dimensões de Coordenação:**
 - **Cooperação:** distingue sistemas cooperativos de sistemas não cooperativos.
 - **Conhecimento:** considera o conhecimento que um robô tem sobre o estado de seus companheiros.
 - **Coordenação:** leva em conta a estratégia de coordenação do grupo.
 - **Organização:** refere-se ao modo como são tomadas as decisões no grupo.
- **Dimensões de Sistema:**
 - **Comunicação:** refere-se as técnicas de comunicação utilizadas pelos robôs.
 - **Composição do time:** distingue grupos de robôs homogêneos de grupos de robôs heterogêneos.
 - **Arquitetura do sistema:** refere-se a arquitetura do grupo de robôs.
 - **Tamanho do time:** considera o número de robôs presentes no grupo.

Figura 6 - Taxonomia para a classificação de SMR.



Adaptado de Farinelli et al. (2004).

Segundo Parker (2008), SMR podem ser divididos em duas categorias: sistemas coletivos de enxame e sistemas intencionalmente cooperativos. Sistemas coletivos de enxame são compostos por um grande número de robôs homogêneos, que executam tarefas com pouca comunicação explícita e coordenação entre os membros do grupo. Cada membro possui conhecimentos mínimos sobre o restante do grupo, mas atua com base em leis de controle locais que geram comportamentos globalmente coerentes. Em sistemas intencionalmente cooperativos os robôs podem seguir estratégias de coordenação fracamente ou fortemente acopladas. Eles possuem o conhecimento do estado, ações e capacidades dos companheiros de grupo e agem em conjunto com base nestas informações, buscando o mesmo objetivo.

Dependendo da aplicação multirrobótica, existe a necessidade dos robôs se comunicarem entre si. Mataric (2007) afirma que a comunicação é necessária quando os robôs necessitam realizar ações sincronizadas, sendo esta importante para a coordenação e negociação durante a execução de tarefas e também para a modelagem do ambiente. Desta forma, um robô pode comunicar informações referentes ao estado individual, do ambiente, de execução de determinadas tarefas e ainda seus objetivos e intenções. Estas informações podem ser obtidas através de uma série de técnicas. Parker (2008) cita três delas:

- **Comunicação implícita:** também chamada de estigmergia, os robôs obtêm informações através da

percepção do efeito de ações realizadas por companheiros de grupo no ambiente de atuação.

- **Reconhecimento de ação passiva:** por meio de seus sensores os robôs observam diretamente as ações de seus companheiros.
- **Comunicação Explícita:** os robôs se comunicam de maneira intencional e diretamente trocando informações úteis através de meios ativos, como via rádio por exemplo.

A aplicação de SMR apresenta um desafio muito maior se comparado ao uso de um único robô. De acordo com Mataric (2007), um único robô tem de lidar com ruídos em seus sensores e efetadores, além de incertezas sobre seus conhecimentos. Porém robôs que atuam em equipe passam a lidar também com incertezas quanto a localização, comunicação, ações e intenções do restante dos robôs pertencentes ao grupo. Por conta desta característica o ambiente torna-se inerentemente dinâmico.

Dentre os desafios de SMR, Cai e Yang (2012) citam a alocação de tarefas, a fusão de múltiplos sensores, a localização dos robôs assim como o controle de formação e os processos de coordenação e cooperação. A alocação de tarefas consiste em dividir o problema em subtarefas e definir quantos e quais robôs serão responsáveis por cada uma delas. A fusão de múltiplos sensores consiste na junção de informações percebidas a nível local e global. A localização visa estimar a posição exata dos robôs no ambiente. O controle de formação preocupa-se em evitar que ocorram colisões entre os robôs e obstáculos, e os processos de coordenação e cooperação definem o modo como o grupo de robôs irá atuar para atingir a sua meta.

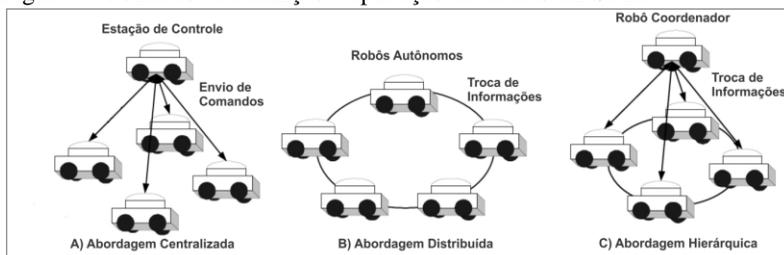
As arquiteturas de SMR serão descritas com mais detalhes na Seção 3.2, tipos de tarefas na Seção 3.3, alocação de tarefas na Seção 3.4 e domínios de aplicações multirrobóticas na Seção 3.5.

3.2. ARQUITETURAS DE SISTEMAS MULTIRROBÓTICOS

Para garantir a eficiência do sistema, é crucial especificar políticas de coordenação que determinam como ocorre a comunicação entre o grupo e como as tarefas são definidas e alocadas a cada robô. Dentre as três arquiteturas mais comuns estão as abordagens centralizada, distribuída e hierárquica. Na abordagem centralizada, um único robô do grupo, que pode ser uma estação de controle, possui o

conhecimento global do mundo e é responsável por calcular a solução e alocar as tarefas para os demais membros do grupo. Na arquitetura distribuída, cada indivíduo toma decisões locais com base em seu conhecimento de mundo local e com a troca de informações com o restante do grupo. O método hierárquico consiste na junção de características distribuídas e centralizadas. Nele define-se um coordenador que realiza o planejamento global e todos os integrantes podem perceber e trocar informações entre si (TRIGUI et al., 2012). A Figura 7 ilustra cada uma destas abordagens.

Figura 7 - Políticas de definição e aplicação de tarefas em SMR.



Adaptado de: Lei et al. (2010).

Cada uma das abordagens necessita de uma infraestrutura diferenciada e possui vantagens e desvantagens. A abordagem centralizada é a mais fácil de implementar, e por se ter conhecimento de todo o ambiente é possível chegar em soluções ótimas ou aproximadas. Porém ela exige que a estação central tenha o conhecimento completo do mundo, sendo que se esta falhar o sistema inteiro falha. O método distribuído é escalável e tolerante a falhas, mas seu desempenho global, previsibilidade, estabilidade e otimalidade, em geral, não são boas devido ao fato das decisões serem tomadas com informações de nível local.

O modelo hierárquico, une as vantagens das abordagens centralizada e distribuída, onde um coordenador central, por possuir informações globais, pode chegar a soluções ótimas e todos os robôs do grupo podem trocar informações. Isto torna o sistema robusto e tolerável a falhas, pois se o robô coordenador falhar, pode-se definir um novo coordenador (LEI; ZHU; FENG, 2010).

A precisão do modelo de mundo possui impacto direto no desempenho dos robôs. Em abordagens centralizadas o controlador central cria um mapa com as informações de todos os robôs, a partir de um sensor global ou de informações recebidas do grupo. Em métodos

descentralizados cada robô cria seu próprio submapa com base em informações captadas por seus sensores ou recebidas de outros integrantes (GAUTAM; MOHAN, 2012).

3.3. TAREFAS EM SISTEMAS MULTIRROBÓTICOS

Parker (2008), afirma que em muitas aplicações multirrobóticas a missão global da equipe é composta por conjuntos de tarefas. Cada tarefa pode ser executada por uma variedade diferente de robôs e cada robô pode executar uma variedade diferente de tarefas. Uma tarefa, segundo Gerkey e Mataric (2004), consiste em um sub-objetivo necessário para atingir uma determinada meta global. Tarefas podem ser classificadas como discretas, quando possuem início e fim reconhecíveis, e como contínuas, quando não possuem início e fim reconhecíveis. Elas podem variar em escalas de tempo, complexidade e especificidade.

Em SMR as tarefas podem também ser classificadas de acordo com a sua decomponibilidade e alocabilidade conforme explica Zlot (2006). A decomponibilidade descreve se a tarefa é passível de decomposição, isto é, se ela pode ou não ser decomposta em conjuntos de subtarefas. A alocabilidade define se é possível atribuir uma tarefa para um robô ou grupo de robôs. Uma tarefa só é alocável se existirem robôs, individuais ou em conjunto, que possuam capacidade e recursos necessários para executá-la.

Desta forma, as tarefas podem ser classificadas em simples, compostas ou complexas. Uma tarefa simples caracteriza-se por ser alocável a apenas um robô, podendo ser decomposta em tarefas atômicas, que consistem em uma única ação indivisível, ou ainda em subtarefas simples. As tarefas compostas consistem em um conjunto de subtarefas simples ou compostas. As tarefas complexas podem ser decompostas em subtarefas complexas, compostas ou simples, sendo que ao menos um de seus conjuntos de subtarefas é alocável a múltiplos robôs.

3.4. ALOCAÇÃO DE TAREFAS EM SISTEMAS MULTIRROBÓTICOS

Em um SMR a alocação de tarefas baseia-se na problemática de atribuir tarefas ao grupo de robôs com base em algum critério, que pode visar a otimização na distribuição de recursos, o aumento da eficiência e robustez da aplicação, e a redução de custo (CAI; YANG, 2012).

De acordo com Parker (2008), a alocação de tarefas pode seguir abordagens baseada em comportamento ou abordagens baseada em mercado. Abordagens baseadas em comportamento são comuns em arquiteturas de grupo distribuídas. Nelas os robôs não negociam a atribuição de tarefas de forma explícita. Cada robô define qual tarefa irá executar com base em seus conhecimentos do estado atual da missão e das capacidades e ações dos companheiros de equipe. Por outro lado, em abordagens baseadas em mercado o grupo negocia a atribuição de tarefas explicitamente, com base em funções de utilidade que estimam o desempenho a qual um robô individual terá aptidão para realizar determinada tarefa. Desta forma, esta abordagem busca atribuir uma tarefa ao robô que possui maior utilidade executá-la.

Gerkey e Mataric (2004), explicam que a utilidade baseia-se no conceito de que é possível quantificar o valor ou custo de um determinado robô executar uma determinada ação e descrevem um método genérico para estimar a utilidade, ilustrado na Equação 1. Nele, o valor de utilidade de um robô R para a execução de uma tarefa T é calculado com base na subtração entre Q e C, que são respectivamente, a qualidade esperada na execução da tarefa e o custo dos recursos utilizados. A qualidade pode ser estimada com base no método e equipamentos utilizados e o custo pode ser estimado considerando as exigências espaços-temporais da tarefa.

Equação 1 - Função genérica para estimar utilidade na execução de tarefas.

$$U_{RT} = \begin{cases} Q_{RT} - C_{RT} & \text{Se R for capaz de executar T e } Q_{RT} > C_{RT} \\ 0 & \text{Senão} \end{cases}$$

Adaptado de: Gerkey e Mataric (2004).

Gerkey e Mataric (2004) propõem uma taxonomia para classificação de problemas de alocação de tarefas para sistemas multirrobóticos (MRTA - *Multi-Robot Task Allocation*) baseada em três eixos:

- **Robôs Mono-tarefa (ST - *Single-task Robots*) vs Robôs Multi-tarefa (MT - *Multi-task Robots*):** ST são capazes de executar apenas uma tarefa por vez e MT executam múltiplas tarefas simultaneamente.

- **Tarefas Mono-robô (SR - *Single-robot Tasks*) vs Tarefas Multi-robô (MR *Multi-robot Tasks*):** SR são tarefas que exigem apenas um robô para concluí-las enquanto MR são tarefas que exigem múltiplos robôs para concluí-las.
- **Atribuição Instantânea (IA - *Instantaneous Assignment*) vs Atribuição de Tempo Estendida (TA - *Time-extended Assignment*):** na IA a informação disponível sobre os robôs, tarefas e ambiente permitem somente uma alocação instantânea de tarefas aos robôs, sem o planejamento para futuras alocações. Por outro lado, na TA existem mais informações disponíveis, como o conjunto de todas as tarefas que precisam ser atribuídas, ou um modelo de como as tarefas chegarão ao longo do tempo.

Um problema MRTA pode ser descrito por meio de uma tripla destas abreviaturas descritas, onde a primeira sigla refere-se a categoria do robô, a segunda indica o tipo de tarefa e a terceira o tipo de atribuição. Diferentes variações de problemas podem ser formadas, possuindo diferentes complexidades computacionais. O problema mais simples consiste na variante ST-SR-IA que pode ser resolvido em tempo polinomial. Outras variantes consistem em problemas NP-Difícil como por exemplo ST-MR-IA, ST-MR-AT, MT-SR-IA, e MT-SR-TA (PARKER, 2008).

3.5. DOMÍNIOS DE APLICAÇÕES DE SISTEMAS MULTIRROBÓTICOS

Segundo Parker (2008) muitas aplicações de mundo real podem beneficiar-se do uso de sistemas multirrobóticos. Porém devido a complexidade dos sistemas multirrobóticos em conjunto com a necessidade de evolução da tecnologia atual, poucos SMR foram efetivamente implementados em aplicações do mundo real. A pesquisa em SMR tem sido explorada através de domínios de testes de aplicações que, apesar de a maioria deles consistirem em experimentos de laboratórios, fornecem um meio para que pesquisadores possam desenvolver e testar estratégias de controle para SMR com grande relevância para aplicações de mundo real. Farinelli et al. (2004), descreve os domínios de aplicações mais comuns:

- **Forrageamento e Cobertura:** o forrageamento consiste na tarefa de múltiplos robôs capturarem objetos espalhados em um determinado local e a cobertura exige que os robôs processem todos os pontos livres de espaço em um ambiente. No domínio do forrageamento estão presentes aplicações de limpeza de resíduos tóxicos, limpeza de minas e robôs de serviço. Dentre as aplicações para cobertura estão a remoção de neve, pintura de carrocerias, cortes de gramados e remoção de minas explosivas. Os principais desafios destes domínios estão em definir técnicas eficazes para analisar todo o ambiente e evitar que haja interferência entre os robôs durante a execução destas tarefas.
- **Manipulação e Transporte Cooperativo de Objetos:** neste domínio de aplicação de SMR, considera-se que os robôs pertencentes ao grupo não são capazes de manipular ou transportar objetos individualmente, devendo fazer isto com base na coordenação de seus movimentos e controle de formação. Pertencem a este domínio aplicações de transporte de objetos pesados em ambientes industriais, montagens de estruturas em larga escala, estocagem, operações de carga e descarga de caminhões e construção. A Figura 8 ilustra um sistema multirrobótico autônomo, desenvolvido por pesquisadores da NASA, para aquisição, transporte e acoplamento de componentes com precisão em tarefas de construção.

Figura 8 - Tripulação Robótica para Construção (RCC - *Robotic Construction Crew*).



Extraído de: Stroupe et al. (2006).

- **Observação de Múltiplos Alvos:** requer a utilização de múltiplos robôs para detectar e monitorar múltiplos alvos. O objetivo é maximizar a quantidade de tempo em que os alvos são observados por pelo menos um dos robôs da equipe. Este domínio possui relação com atividades de segurança, vigilância e reconhecimento de problemas e também com redes de sensores.
- **Exploração e Rebanho:** em tarefas de exploração, o objetivo do SMR é coletar o máximo de informações úteis possíveis sobre uma determinada área. Em tarefas de rebanho o objetivo é mover o grupo de robôs pelo ambiente mantendo uma formação específica. A grande questão sobre estes domínios é a utilização da cooperação entre os robôs para a localização e fusão das informações percebidas do ambiente. Problemas de exploração e rebanho relacionam-se com aplicações de transbordo em portos, aeroportos e estações de triagem, coordenação de movimento em aplicações industriais, exploração de ambientes desconhecidos e busca e resgate em ambientes de desastres urbanos, conforme robô exemplo ilustrado na Figura 9.

Figura 9 - Grupo de robôs marsupiais para busca e resgate em ambientes urbanos.



Extraído de: Murphy (2000b).

- **Futebol de Robôs:** oferece um ambiente dinâmico repleto de incertezas para a pesquisa e desenvolvimento de sistemas multi-agente e cooperação multirrobótica. O futebol de robôs será descrito detalhadamente na Seção 3.5.1.

Cao et al. (1997), cita também o controle de tráfego, que visa minimizar as interferências entre robôs que atuam em um mesmo ambiente e ocupam grandes porções do ambiente navegável. Um exemplo são redes de estradas onde veículos partilham o espaço com outros veículos e cooperam inteligentemente para evitar impasses e gargalos.

3.5.1. Futebol de Robôs

O futebol de robôs foi citado pela primeira vez por Mackworth (1992), que descreveu a ideia de robôs jogarem futebol como um grande desafio na representação de planejamento e atuação devido a necessidade de robôs cooperarem entre si para atingir um objetivo comum, o ambiente de atuação é muito dinâmico necessitando de um agente árbitro que atue como mediador.

Noda (1995) afirma que o futebol de robôs fornece um ambiente de testes para sistemas de múltiplos robôs, que além de exigir conceitos de sistemas distribuídos, inteligência artificial e comunicação entre agentes inteligentes põe em prática as problemáticas de processamento

de tempo real, robustez contra ruído e processamento de informações incompletas e inconsistentes. Este desafio foi criado com o intuito de incentivar pesquisas na área de robótica autônoma multi-agente em ambiente acadêmico. Competindo com robôs adversários, em um ambiente dinâmico de tempo real¹, robôs devem agir em cooperação tomando decisões rápidas, que devem ser transparentes a atrasos e falhas. A Tabela 3 apresenta uma comparação entre os desafios de um jogo de Xadrez e um jogo de futebol de robôs.

Tabela 3 - Comparação entre o Xadrez e o Futebol de Robôs.

	Xadrez	Futebol de Robôs
Ambiente	Estático	Dinâmico
Troca de Estados	Em turnos	Constante
Informações do ambiente	Completas	Incompletas
Leitura de sensores	Simbólica	Não Simbólica
Controle	Centralizado	Distribuído

Adaptado de: Kitano et al. (1997).

Em um jogo de Xadrez o ambiente é estático, pois os agentes não mudam de estado durante o planejamento das jogadas, o que faz com que todos os estados possíveis após o término de cada jogada possam ser previstos facilmente. No futebol de robôs os agentes estão em constante movimento ao mesmo tempo em que ocorre o planejamento das jogadas. Devido ao fato de existirem muitos ruídos durante a leitura dos sensores, muitas vezes as informações obtidas sobre o estado do jogo são inconsistentes, o que faz com que o sistema gere estratégias erradas que possam gerar ações não previstas. As arquiteturas de controle de um time de futebol de robôs costumam ter módulos de processamento distribuídos, o que aumenta a complexidade do sistema.

Em 1992, no Japão, Hiroaki Kitano organizou um workshop chamado "*The Workshop on Grand Challenge AI Applications*", onde, com a participação de um grande número de pesquisadores japoneses, houve sérias discussões sobre usar o futebol de robôs para promover a ciência e tecnologia. Realizou-se pesquisas sobre a viabilidade tecnológica e financeira e avaliações sobre o impacto social da atividade. A partir disto foram estabelecidos os primeiros protótipos de robôs e sistemas de controle, onde foi possível concluir sua viabilidade.

¹O ambiente de uma partida de futebol de robôs é dinâmico devido as suas mudanças de estado constantes e também de tempo real por exigir prazos para a conclusão de tarefas. (BROWNING et al., 2004).

Em 1993, um grupo de pesquisadores japoneses que incluía Hiroaki Kitano, Asada Minoru e Yasuo Kuniyoshi realizaram no Japão a competição de robótica *J-League*, que gerou grande interesse de pesquisadores ao redor do mundo, que sugeriram expandir o projeto internacionalmente. Desta forma surgiu a *World Robot Cup* que, mais tarde, passou a chamar-se *RoboCup* (KITANO, 1997).

A *RoboCup* é uma competição mundial de robótica que ocorre anualmente ao redor do mundo com o objetivo de promover a Inteligência Artificial e Robótica inteligente. Ela propõe desafios que reúnem um vasto conjunto de problemas científicos, que além de incentivar à pesquisa e desenvolvimento atrai a atenção do público para estas áreas. A *RoboCup* possui como meta de longo prazo a criação de um time de futebol de robôs que possa ganhar uma partida jogada contra a seleção de futebol vencedora da copa do mundo de futebol de 2050 (Reis, 2003). A Figura 10 mostra uma partida de futebol de robôs da *RoboCup Small Size League* realizada na CBR 2011 (Competição Brasileira de Robótica).

Figura 10 - Partida de Futebol de Robôs realizada durante a CBR 2011.



A primeira *RoboCup* aconteceu em 1997 durante a XV *International Joint Conference on Artificial Intelligence (IJCAI-97)* em Nagoya, no Japão, que atraiu a presença de mais de 5 (cinco) mil espectadores. Inicialmente existiam duas ligas, a *Real-Robot-League* e a *Simulation-League*. A *Real Robot League* utilizava robôs físicos para disputar as partidas. Eles eram separados por duas categorias que limitavam as dimensões e o peso dos robôs e do campo: a *Small Size* e a

Middle Size. Na *Simulation League* as partidas eram realizadas em um servidor que simulava 11 (onze) jogadores para cada time, em um campo com dimensões proporcionais ao futebol. Durante as partidas, as equipes conectavam seu software de controle ao servidor de simulação. No total, participaram 10 equipes na *Real Robot League* e 29 equipes na *Simulation League* de cerca de 10 países diferentes. Com o sucesso desta edição a *RoboCup* continuou a ser disputada todos os anos, foram criadas novas categorias, o número de equipes participantes cresceu e o hardware e o software evoluíram (NODA et al., 1998).

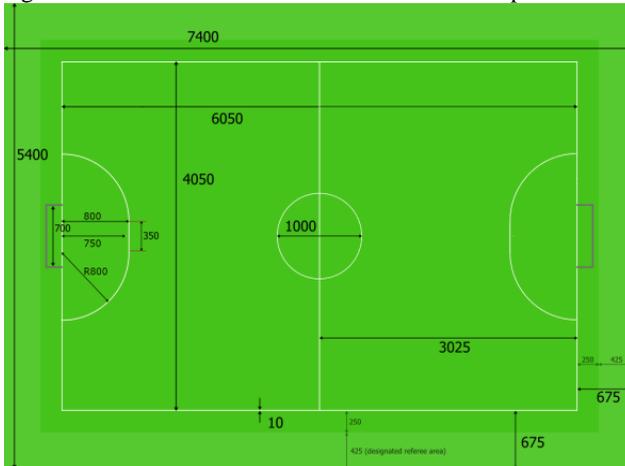
Atualmente a *RoboCup* está dividida em 4 grandes categorias: Futebol, Resgate, *Home* e Futebol Júnior. A categoria de resgate tem foco no uso de robôs para busca e salvamento em ambientes que representam desastres urbanos. A categoria *Home* lida com o desenvolvimento de tecnologias assistivas para aplicação em ambiente doméstico. A Junior está ligada a projetos educacionais que atendem jovens estudantes fornecendo a eles uma introdução a robótica. A categoria de Futebol é a principal delas, trabalha com sistemas cooperativos multi-agentes e autônomos, sendo organizada em 5 (cinco) sub-categorias que regulamentam as regras do jogo e as características dos robôs. Destas as mais conhecidas são a *Small Size*, *Middle size* e *Simulation*, presentes desde a primeira edição da *RoboCup* e a *Humanoid*, que exige que os robôs tenham corpo e sentidos semelhantes ao de seres humanos. Além da *RoboCup*, a FIRA² (*Federation of Internacional Robot-Soccer Associated*) também promove campeonatos com objetivos e categorias semelhantes, porém o ambiente de teste deste trabalho seguirá os padrões e regras da *RoboCup Small Size League*, que será detalhada na seção seguinte.

3.5.2. Small Size League

A *Small Size League* (SSL), também conhecida como F180 League, é uma das primeiras categorias da *RoboCup*. Atualmente ela limita as dimensões dos robôs em 180 x 150 mm, do campo em 6,5 x 4,5 m e do número de robôs para 6 (seis) por equipe. Presente desde a primeira edição, suas regras passam constantemente por mudanças e ajustes para fazer com que continue desafiadora, evitando que o desenvolvimento e o avanço tecnológico, proporcionados por ela, cheguem ao seu limite. A Figura 11 ilustra as dimensões do campo para a categoria F180.

² Site oficial: <http://www.fira.net/>

Figura 11 - Dimensões em centímetros de um campo oficial da RoboCup SSL.



Extraído de: RoboCup Rules (2012).

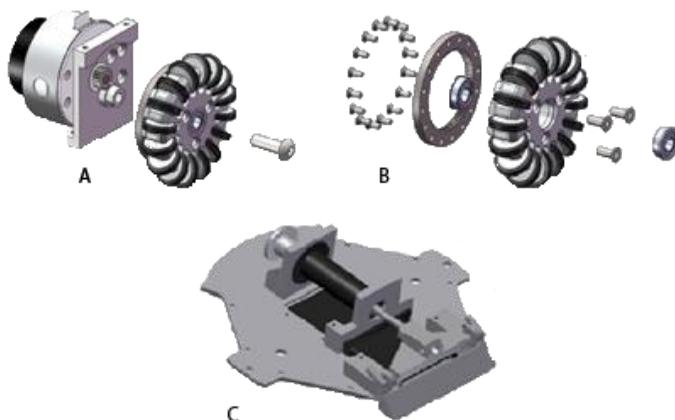
Um robô SSL, geralmente possui 6 (seis) atuadores, sendo quatro motores responsáveis pela locomoção, um motor responsável por acionar um mecanismo de drible e um solenóide ou dispositivo semelhante, que possui a função de disparar a bola, possibilitando a realização de chutes e passes. Cada atuador responsável pela locomoção é conectado a rodas holônicas que dispostas em um ângulo omnidirecional, permitem ao robô mover-se em todas as direções com agilidade e rapidez. Os robôs também possuem rádios para a comunicação e podem possuir vários sensores para a percepção do ambiente e controle de seus atuadores. A Figura 12 ilustra a mecânica de um robô SSL e a Figura 13 alguns de seus componentes.

Figura 12 - Mecânica de um robô da equipe RoboFEI, categoria RoboCup SSL.



Adaptado de: Gurzoni et al. (2010).

Figura 13 - A e B. Roda holonômica, motor e sistema de redução da equipe Parsian Robotics. C. Chassi com sistema de chute da equipe RoboFEI.



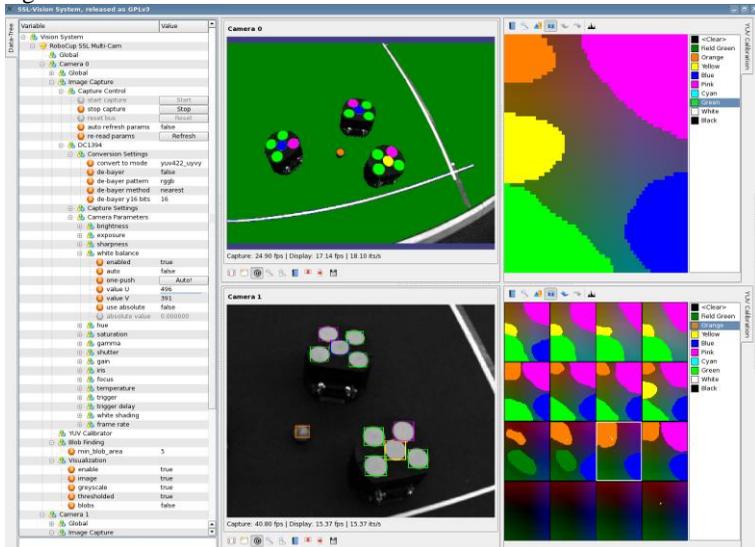
Adaptado de: Mehrabi et al. (2012), Gurzoni et al. (2010)

Normalmente, na SSL, o planejamento de cada equipe é centralizado. A percepção do ambiente se dá por um sistema de visão global. Cada equipe possui um computador com um sistema de estratégia responsável pela coordenação de seu time, tomada de decisões

de alto nível e transmissão via rádio de comandos aos seus robôs. Cada Robô atua recebendo as informações e pondo-as em prática por meio do acionamento e controle de seus atuadores. Eles possuem pouca percepção do ambiente e atuam reagindo aos comandos recebidos, que especificam quais atuadores devem ser acionados e com quais velocidades.

Segundo Zickler et al. (2009), durante as primeiras competições da SSL, a visão computacional era de responsabilidade de cada uma das equipes. Cada equipe tinha o seu próprio software e hardware de visão. Era permitido usar câmeras locais aos robôs ou globais, localizados acima do campo ou nas laterais. A maioria das equipes utilizava 2 (duas) câmeras globais cada, fixadas acima do campo. Isto era um grande problema, pois a cada jogo era necessário que cada equipe desmontasse e remontasse seu hardware de visão, tendo que calibrá-lo novamente, o que levava muito tempo.

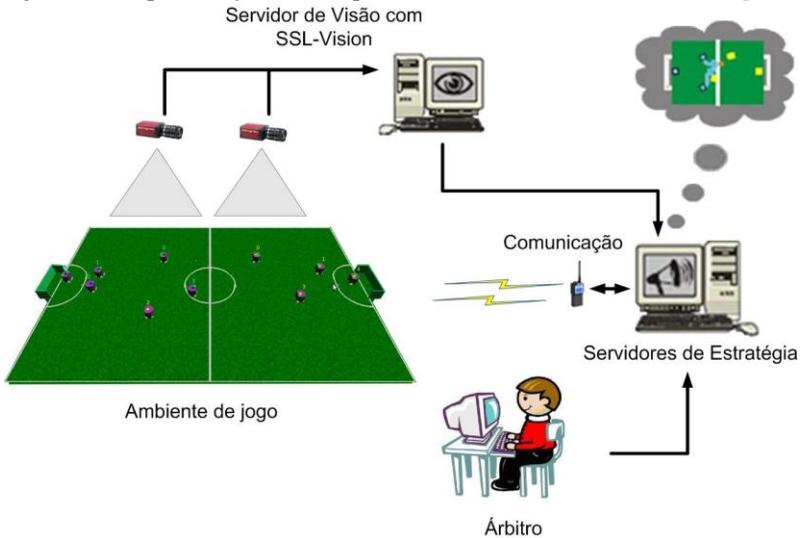
Para superar este problema a comissão da SSL optou por migrar para uma arquitetura de visão compartilhada, onde haveria apenas um conjunto de 2 (duas) câmeras localizadas acima do campo que seriam conectadas a um servidor compartilhado com as equipes. O software para este servidor precisaria ser flexível e escalável, para suportar futuras alterações e ser aberto a novas abordagens, não podendo restringir o desempenho das equipes. A partir destes critérios foi desenvolvido o *SSL-Vision* (ZICKLER et al., 2009).

Figura 14 - Interface Gráfica do *SSL-Vision*.

Extraído de: Zickler et al, (2009).

O *SSL-Vision* identifica a posição e orientação dos robôs e da bola por meio de suas cores. A bola é alaranjada e cada um dos robôs possui um padrão único de cores fixado em sua superfície. Os padrões de cores são especificados por 5 (cinco) círculos fixados sobre os robôs, onde o do centro, que pode ser azul ou amarelo, especifica o time ao qual o robô pertence e os 4 círculos restantes identificam o robô. A Figura 14 ilustra a interface do *SSL-Vision* em execução realizando a captura destas informações.

Figura 15 - Representação de uma partida de futebol de robôs da *RoboCup SSL*.



Durante uma partida, o SSL-Vision é executado em um servidor, de responsabilidade da organização da competição. Ele faz a aquisição das imagens por meio de duas câmeras padrão IEEE 1394. Cada câmera é responsável pela captura das imagens de uma metade do campo. As informações extraídas pelo SSL-Vision são serializadas em uma estrutura de dados e enviadas a servidores de estratégia de cada equipe via padrão *Ethernet UDP (User Datagram Protocol)*. Os servidores de estratégia transmitem via rádio comandos para seus robôs. Um árbitro apita o jogo e um árbitro auxiliar repassa as decisões para os servidores de estratégia das equipes operando um computador com um sistema de arbitragem. A Figura 15 ilustra o fluxo de execução de uma partida de futebol de robôs da categoria F180.

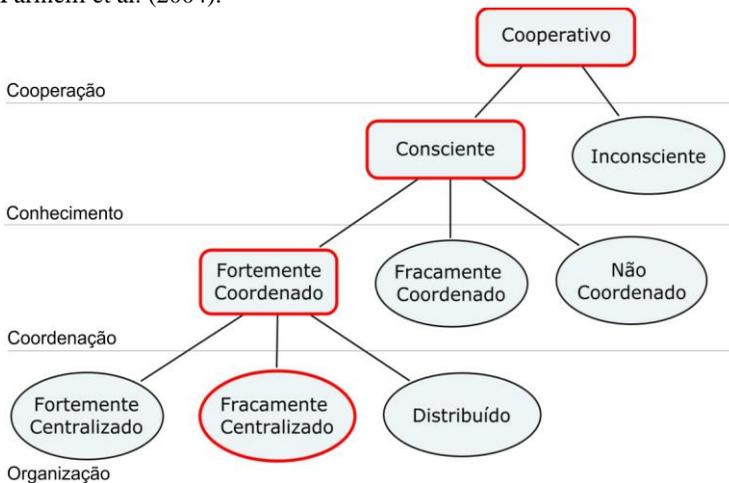
4. ARQUITETURA DE COORDENAÇÃO DE SISTEMAS MULTIRROBÓTICOS

Este capítulo descreve a Arquitetura de Coordenação de Sistemas Multirroboóticos (ACSM), decompondo-a em duas partes: arquitetura individual e arquitetura de grupo. A arquitetura de grupo refere-se às estratégias globais de coordenação, troca de informações e atribuições de tarefas entre os robôs; enquanto a arquitetura individual está focada na divisão dos módulos de responsabilidades individuais de cada robô.

4.1. DEFINIÇÕES PRELIMINARES

Na definição da arquitetura considera-se um grupo de robôs que atendem os seguintes pressupostos: são heterogêneos, com diferentes habilidades; devem ser capazes de realizar variados tipos de tarefas de modo cooperativo, podendo eventualmente sofrer falhas; todos os robôs podem comunicar-se, podendo haver perda nas mensagens; o Sistema Multirroboótico (SMR) deverá ser controlado de modo autônomo, realizando sequências coordenadas de tarefas; as tarefas são alocadas dinamicamente a cada membro do grupo e visam atingir um objetivo de longo prazo em um ambiente de atuação incerto e dinâmico.

Figura 16 - Classificação da arquitetura conforme a taxonomia proposta por Farinelli et al. (2004).



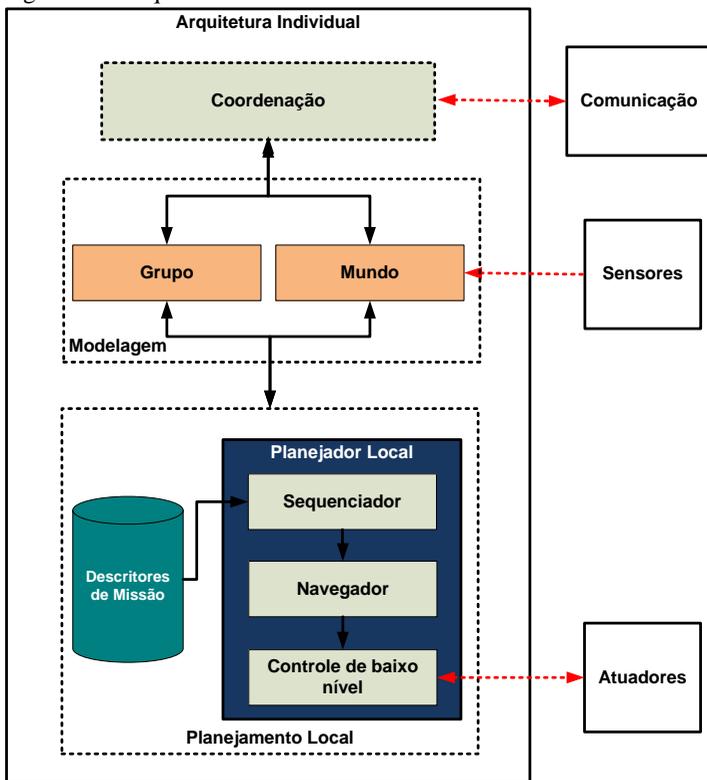
Com base nestes pressupostos e na taxonomia de classificação de SMR descrita no Capítulo 3, na Seção 3.1 delimita-se o problema conforme ilustra a Figura 16. A arquitetura proposta considera sistemas classificados como cooperativos, onde os robôs trabalham juntos para atingir uma meta global. Cada robô conhecerá o restante do grupo. O SMR seguirá uma abordagem de coordenação fortemente acoplada, o que obriga os robôs a serem capazes de intercomunicar-se e coordenar-se de modo preciso. A organização do SMR será fracamente centralizada, pois o grupo é organizado de modo hierárquico, o que permite chegar-se a soluções de nível global ótimas ou aproximadas, sem abdicar de características distribuídas, como tolerância a falhas e escalabilidade. A arquitetura proposta busca minimizar o uso de recursos, o tempo de resposta e conclusão das tarefas e a sobrecarga de comunicação.

Conforme Yan e Zehnmin (2009), uma arquitetura de software de um SMR pode ser dividida em duas partes: arquitetura individual e arquitetura de grupo. A arquitetura individual foca-se em organizar as responsabilidades individuais de um robô, tais como a percepção, planejamento individual e atuação. Por outro lado, o foco da arquitetura de grupo está nas questões globais do sistema, como a troca de informações entre os robôs e estratégias de controle e coordenação utilizadas pelo grupo. A arquitetura individual será descrita na Seção 4.2, e a arquitetura de grupo será detalhada na Seção 4.3.

4.2. ARQUITETURA INDIVIDUAL

A arquitetura individual está dividida em 3 (três) camadas, que são: coordenação, modelagem e planejamento local, conforme ilustra a Figura 17.

Figura 17 - Arquitetura Híbrida individual.



A camada de coordenação é responsável pela troca de informação do robô com o restante do grupo. Um dos robôs deve ser definido como líder, por meio de um operador que supervisiona a aplicação ou por meio de algoritmos de eleição, descritos por Tanenbaum (2008). Caso o robô torne-se líder a camada de coordenação passa a realizar o planejamento global da missão e alocação de tarefas.

Na camada de modelagem considera-se dois modelos: modelo de mundo e de grupo. O modelo de mundo é responsável pelo mapeamento do ambiente por meio da fusão sensorial entre as informações percebidas pelos sensores locais ao robô e as informações recebidas de outros robôs. O modelo de grupo mantém as informações dos robôs pertencentes ao grupo, para que estes conheçam as ações e intenções de seus companheiros.

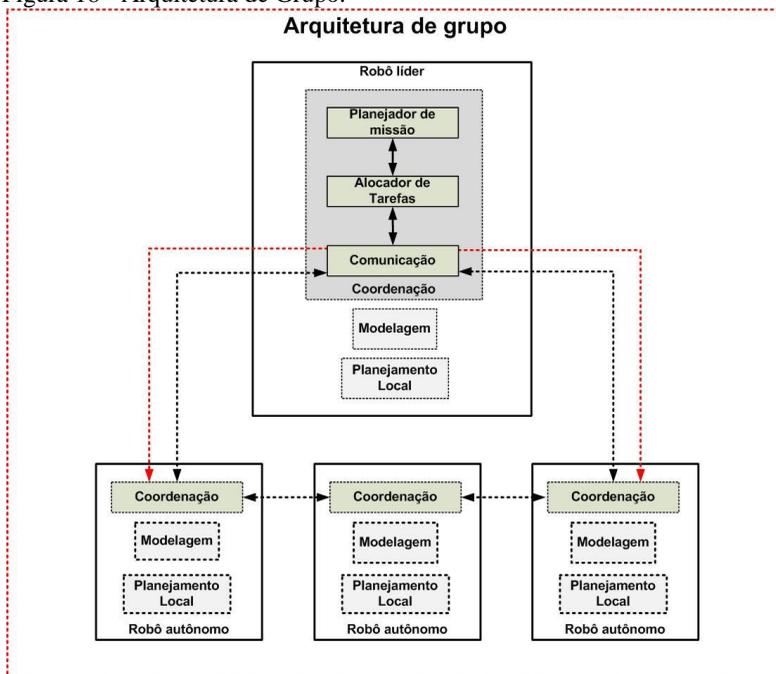
O fato de todos os robôs possuírem informações globais do grupo e do ambiente possibilita que cada robô possa realizar seu planejamento

local de forma completamente autônoma. A camada de planejamento local é composta pelos descritores de missão, pelo sequenciador, navegador e pelo controle de baixo nível. Os descritores de missão devem oferecer ao robô os conhecimentos necessários para realizar qualquer tarefa a ele alocada. Dado o objetivo global atribuído ao robô, o sequenciador determina e organiza a sequência de ações para atingi-lo. O navegador é responsável pelo planejamento de caminhos livres de colisões e de trajetórias com as velocidades e acelerações adequadas. O controle de baixo nível é responsável pelo controle dos atuadores de modo que o robô consiga por em prática a movimentação planejada.

4.3. VISÃO GERAL DA ARQUITETURA DE GRUPO

Ilustrada na Figura 18, a arquitetura de grupo está organizada no conceito hierárquico que é descrito no Capítulo 2, na Seção 2.4.1. Isto obriga que um líder seja definido para o planejamento global do grupo. Após ser definido, um líder passa a ser responsável pela execução do “planejador de missão” e pelo “alocador de tarefas”.

Figura 18 - Arquitetura de Grupo.



O planejador de missão define um objetivo global para a equipe de acordo com os descritores de missão e os modelos de grupo e de mundo. Este objetivo é decomposto em sub-tarefas. O Alocador de tarefas possui a função de atribuir cada tarefa aos robôs mais aptos para executá-la. Nesta arquitetura os robôs se comunicam de forma explícita, diversos tipos de mensagem. Serão descritos com mais detalhes, o planejador de missão na Seção 4.3.1, o alocador de tarefas na Seção 4.3.2 e o protocolo de comunicação na Seção 4.3.3.

4.3.1. Planejador de Missão

Para o planejamento de missão, buscou-se por uma abordagem de representação de tarefas robusta o bastante para manipular variados tipos de tarefa e suficientemente simples para o entendimento humano. Tal abordagem baseia-se em uma adaptação de Bowling et al. (2004). Considera-se uma estrutura hierárquica que pode ser decomposta em 3 (três) módulos: plano de equipe, comportamento individual e controle de baixo nível. Planos de equipe são utilizados para coordenar as ações

globais do grupo. Cada plano de equipe, conforme ilustra a Figura 19, é composto por um identificador da tarefa, condições de aplicação e rescisão do plano, assim como um tempo limite para sua execução e uma sequência de comportamentos que devem ser alocados a cada membro da equipe.

Figura 19 - Descrição de um plano de equipe.

Plano de Equipe	
Identificação do plano:	01.
Condição de aplicabilidade:	!Area-Mapeada.
Condição de rescisão:	Area-Mapeada, Plano-Abortado.
Tempo limite:	10 minutos.
Comportamento 1:	Mapeamento.
Comportamento 2:	Busca.
Comportamento 3:	Transporte de equipamentos 1.
Comportamento 4:	Transporte de equipamentos 2.

As condições de aplicabilidade e rescisão de um plano de equipe podem ser compostas por um ou mais predicados que refletem o estado global do ambiente de atuação, sendo que um plano pode ser selecionado ou abortado apenas quando seu respectivo predicado for verdadeiro. No caso da rescisão, um plano também pode ser abortado caso seu tempo limite de execução for ultrapassado ou então quando o planejador de missão decidir abortá-lo.

Os comportamentos individuais consistem em funções ou habilidades de alto nível que podem ser realizadas por um robô. Um plano de equipe especifica sequências de comportamentos individuais que devem ser alocados ao grupo de robôs. O planejador de missão poderá formar um plano de equipe a partir de uma base de planos de equipe ou ainda em tempo real por um operador humano que supervisiona a aplicação. Assim que um plano de equipe é escolhido, o alocador de tarefas define o robô mais apto para cada comportamento individual e atribui a estes seu respectivo comportamento. Cada robô

passa a executar seu comportamento até a sua conclusão com sucesso ou até a troca do plano de equipe ou realocação dos comportamentos.

Cada comportamento é formado por seqüências de primitivas de controle de baixo nível que formam uma máquina de estados. Tais primitivas consistem em processamento de informações sensoriais, geração de comandos para os atuadores através de um módulo de navegação e condições de transição para outras primitivas.

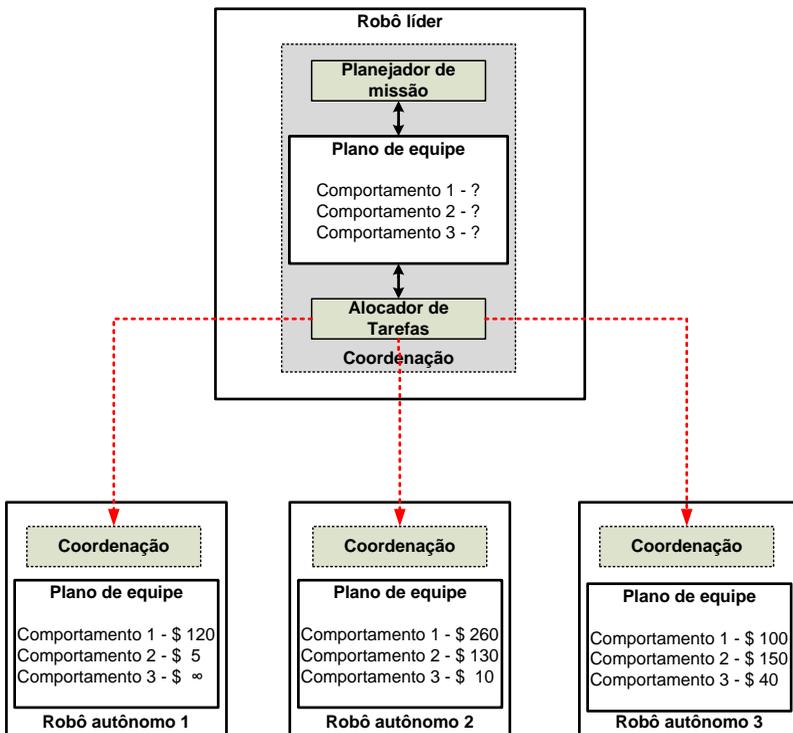
Desta forma o planejador de equipe tem a função de definir o plano de equipe mais adequado para o SMR e avaliar a sua execução, definindo novos planos quando necessário. Um plano de equipe define a seqüência de comportamentos realizados por cada robô. Cada comportamento encapsula primitivas de controle de baixo nível que são responsáveis pelo controle do robô. Esta organização modular hierárquica torna a arquitetura transparente a diversas aplicações.

4.3.2. Alocador de Tarefas

A alocação de tarefas será baseada em mercado e seguirá uma abordagem de leilão semelhante ao sistema de alocação de tarefas Murdoch, proposto por Gerkey e Mataric (2002). Cada robô deverá ser capaz de estimar seu valor de utilidade para a realização de cada tarefa. Dado um plano de equipe, ele deverá atribuir cada comportamento individual ao robô que possua maior valor de utilidade para a sua execução. Este processo ocorre em 5 (cinco) passos:

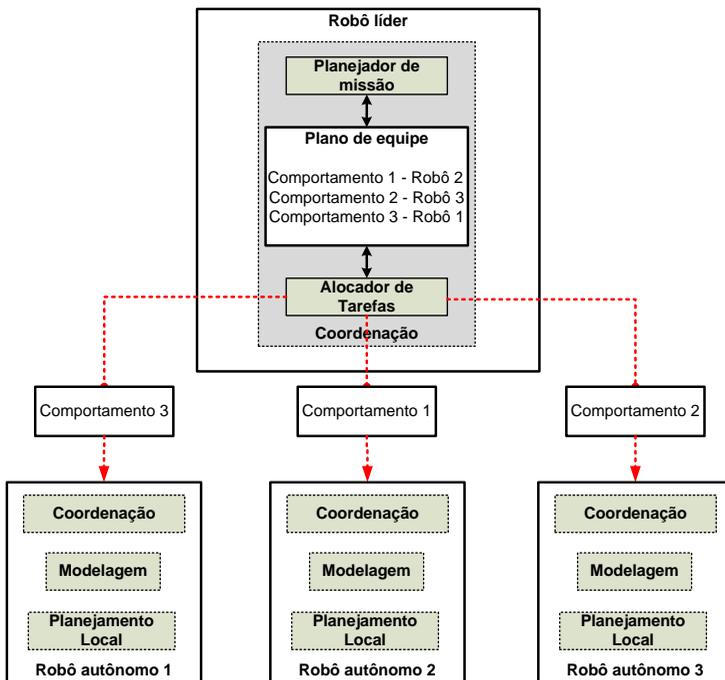
1. **Anúncio de um plano de equipe:** assim que o planejador de missão define um plano de equipe, o alocador de tarefas informa para todos os robôs cada um dos comportamentos individuais presentes no plano.

Figura 20 - Início do leilão.



- Quantificação de valores de utilidade:** ao receber a demanda de comportamentos individuais para a missão, cada robô irá quantificar seu valor de utilidade para cada um dos comportamentos. O método para estimar o valor de utilidade varia de acordo com a aplicação. A Figura 20 ilustra o processo de quantificação de valores de utilidade.
- Submissão de oferta:** após quantificar os valores, os robôs enviarão ao robô líder do grupo a relação de seus valores de utilidade para cada um dos comportamentos individuais.

Figura 21 - Fim do leilão e alocação de tarefas.



4. **Fechamento do leilão:** o alocador de tarefas irá escolher o robô mais apto para cada comportamento e realizar as atribuições, conforme ilustra a Figura 21. Cada robô passará a seguir o comportamento individual que lhe foi atribuído e o leilão será fechado.
5. **Monitoramento do progresso:** o alocador de tarefas passa a monitorar o planejador de missão. Quando um novo plano de equipe é gerado inicia-se um novo leilão.

Para garantir a robustez da aplicação, o planejador de missão irá monitorar o desempenho de cada robô na execução de seu comportamento e também o progresso do plano de equipe em atingir o objetivo de longo prazo. Quando necessário um novo plano de equipe poderá ser gerado ou o mesmo plano poderá ser realocado.

4.3.3. Protocolo de Comunicação

Em um SMR é comum que alguns robôs eventualmente percam o alcance do sinal de comunicação. Para garantir a robustez da aplicação, os robôs não poderão enviar mensagens para um destinatário específico, devendo comunicar-se de forma anônima na rede. As mensagens serão enviadas por *broadcast* seguindo o padrão de envio de mensagens *publish/subscribe* (publicar/assinar). Desta forma o remetente da mensagem, denominado publicador, irá enviá-la de forma anônima na rede e os destinatários chamados de assinantes serão os consumidores de dados, que irão considerar apenas as mensagens de seu interesse. Isto também garante economia de largura de banda, pois uma única mensagem poderá ser recebida por todos os robôs.

Figura 22 - Representação de Pacote de dados.



São necessários no mínimo 4 (quatro) tipos diferentes de mensagens: definição de líder, anúncio de missão, atribuição de tarefas e troca de dados. A Figura 22 ilustra os campos de dados padrões presentes em todas as mensagens: um identificador para a mensagem, um para o robô remetente, um para o tipo de mensagem e campos de conteúdo.

O conteúdo de cada mensagem irá variar de acordo com o seu tipo. Mensagens de anúncio de missão possuem um campo de descrição da missão e um vetor com as tarefas, mensagens de atribuição de tarefas possuem um vetor que especifica quais tarefas foram atribuídas a quais robôs, estes tipos de mensagens apenas podem ser enviadas pelo líder.

Mensagens de definição de líder devem conter a identificação do novo líder e somente pode ser enviada por meio de um sistema remoto, por um operador que supervisiona a aplicação ou pelos próprios robôs, caso seja utilizado algum algoritmo de eleição de líder. E finalmente,

mensagens de dados podem ser enviadas por qualquer robô, seja para responder as requisições do líder, informar valores de utilidade para as tarefas anunciadas ou ainda transmitir ao restante do grupo informações referentes ao seu estado ou estado do ambiente de atuação. Estas mensagens podem variar de acordo com o contexto da aplicação.

5. ANÁLISE DA ARQUITURA DE COORDENAÇÃO PARA COORDENAÇÃO DE SISTEMAS MULTIRROBÓTICOS

Este capítulo apresenta os resultados da aplicação da arquitetura de coordenação proposta no contexto do futebol de robôs. O ambiente de testes será o simulador *grSim*, que segue as regras e padrões da *RoboCup Small Size League*. Com a simulação de situações reais de uma partida será possível analisar o aproveitamento geral dos recursos e a alocação de tarefas entre os robôs da Arquitetura de Coordenação de Sistemas Multirrobóticos (ACSM), proposta no Capítulo 4.

5.1. DEFINIÇÕES PRELIMINARES

A arquitetura de grupo, proposta no Capítulo 4, será aplicada e testada no contexto do futebol de robôs, seguindo os padrões e regras impostos pela *RoboCup Small Size League (SSL)*. Os testes serão realizados em ambiente simulado, no simulador *grSim*.

O *grSim*, descrito por Monajjemi et al. (2012), é um software livre de código aberto criado para simular partidas da *RoboCup SSL*. Ele disponibiliza 2 (duas) equipes de 5 (cinco) robôs, informando posição e ângulo de cada um dos robôs e da bola através do mesmo protocolo utilizado pelo *SSL-Vision*. Como entrada, ele espera receber comandos de velocidade para os atuadores de cada robô. A Figura 23 ilustra a interface principal do simulador *grSim*.

Figura 23 – Interface principal do simulador *grSim*.



Na SSL é comum as equipes utilizarem arquiteturas de coordenação centralizadas, onde uma estação fixa é responsável por realizar todo o planejamento global e os robôs atuarem como escravos, apenas obedecendo aos comandos de velocidades que recebem. Com a aplicação da arquitetura proposta, que une elementos de coordenação centralizados e distribuídos, não haverá uma estação fixa de controle. Um robô da equipe será definido remotamente como líder e além de realizar seu próprio planejamento realizará a coordenação global de toda a equipe, considera-se que cada robô receberá as informações do *SSL-Vision* através de um meio sem fio, tendo capacidade de processamento embarcado suficiente para realizar seu próprio planejamento local e coordenação global, caso seja o líder de grupo. Qualquer robô suficientemente apto poderá tornar-se líder de grupo. Caso o robô líder falhar outro robô responsável pela coordenação global será definido.

Para os testes da arquitetura foram criados 5 (cinco) sistemas de controle que são interfaceados com o simulador, ou seja, recebem as informações de visão enviadas por ele. Cada sistema irá controlar um robô. A comunicação entre os sistemas utiliza a interface *Socket UDP (User Datagram Protocol)*. Ao se comunicar, cada robô envia mensagens em *broadcast*, ou seja, todos os outros robôs receberão e avaliarão seu conteúdo, descartando-as se forem desnecessárias.

Os dados serão serializados e enviados por meio do *Google Protocol Buffer (GPB)* (Google Developers, 2012). O GPB consiste em um mecanismo automatizado para a serialização de dados estruturados compatível com as linguagens Java, C++ e Python. Ele foi desenvolvido pela Google para ser utilizado em protocolos de comunicação e armazenamento de dados, baseando-se em simplicidade e desempenho. Os dados estruturados são definidos em um arquivo de extensão *proto* e seu tipo do dado é denominado *message*.

Cada *message*, é composta por uma série de atributos. Cada atributo tem um nome, um tipo e um identificador, pode-se definir modificadores que indicam se o atributo é opcional, obrigatório ou repetido e é possível também declarar *messages* dentro de outras *messages* de forma hierárquica. A partir do *proto* é gerado uma classe de interface para a manipulação dos dados.

Cada missão irá descrever uma formação tática, na qual os comportamentos individuais serão posições em campo. Foram definidas 4 (quatro) posições: goleiro, defensor, criador e atacante. O goleiro tem a função de defender a meta, afastando a bola de dentro da área. O defensor possui a função de defender a zona defensiva, impedindo que a bola entre na área. O criador busca roubar e dominar a bola, procurando

fazer um bom passe para o atacante. O atacante irá se posicionar no ataque buscando receber um passe para realizar a finalização. Foram definidas as seguintes missões: "1-2-1", "2-1-1", "1-1-2", onde o primeiro número define o número de defensores, o segundo o número de criadores e o terceiro o número de atacantes, respectivamente.

Para a alocação de tarefas foi utilizado o método húngaro de otimização para problemas de alocação de tarefas, descrito por Kuhn (1955). Seu objetivo é selecionar os robôs mais aptos para a realização das tarefas buscando o melhor aproveitamento global de recursos possíveis. Ao anunciar uma missão, o líder espera receber valores de utilidade de cada robô para cada tarefa. Utilizando tais valores, por meio do "alocador de tarefas", é definida uma matriz de utilidade.

No planejamento local, o sequenciador especifica máquinas de estado para cada um dos comportamentos. A Figura 24 ilustra a máquina de estados do goleiro, a Figura 25 a máquina de estados do defensor, a Figura 26 a do criador e a Figura 27 ilustra a máquina de estados do atacante.

Figura 24 - Máquina de estados do goleiro.

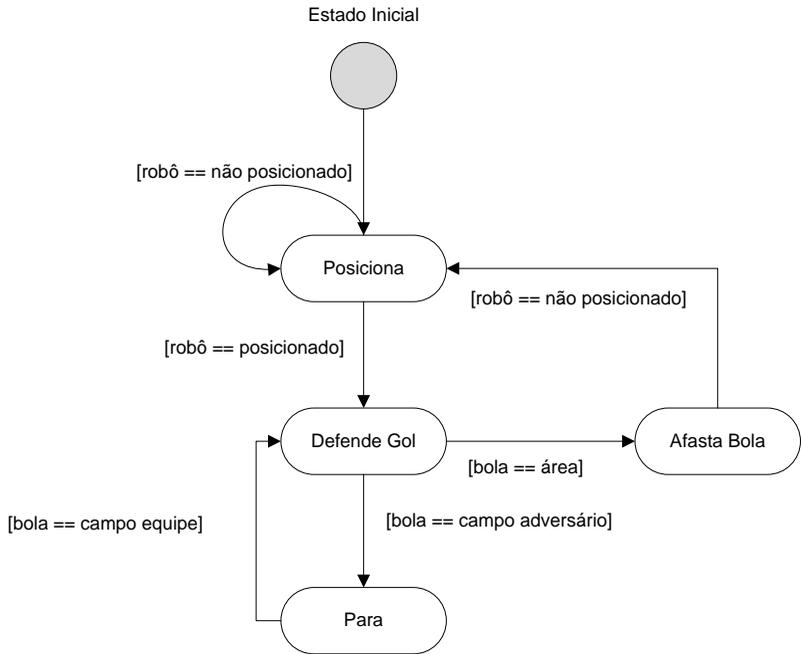


Figura 25 - Máquina de estados do defensor.

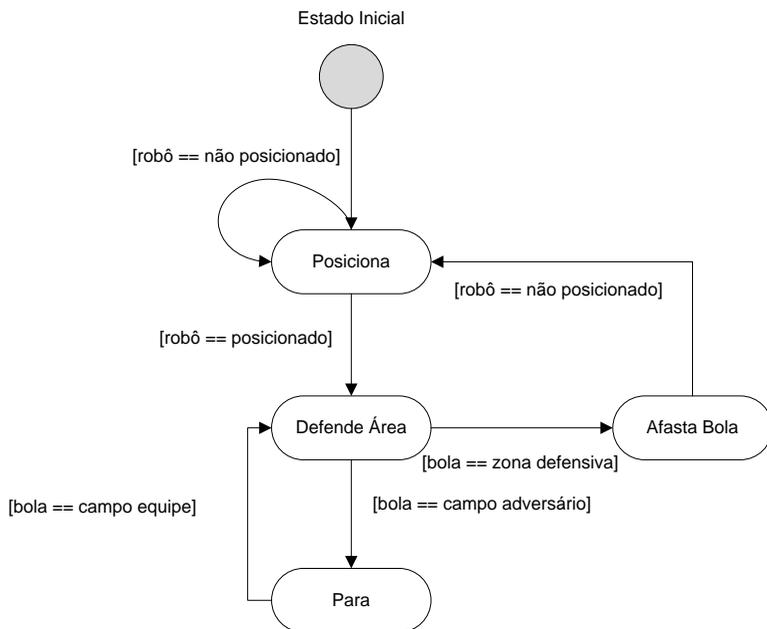


Figura 26 - Máquina de estados do criador.

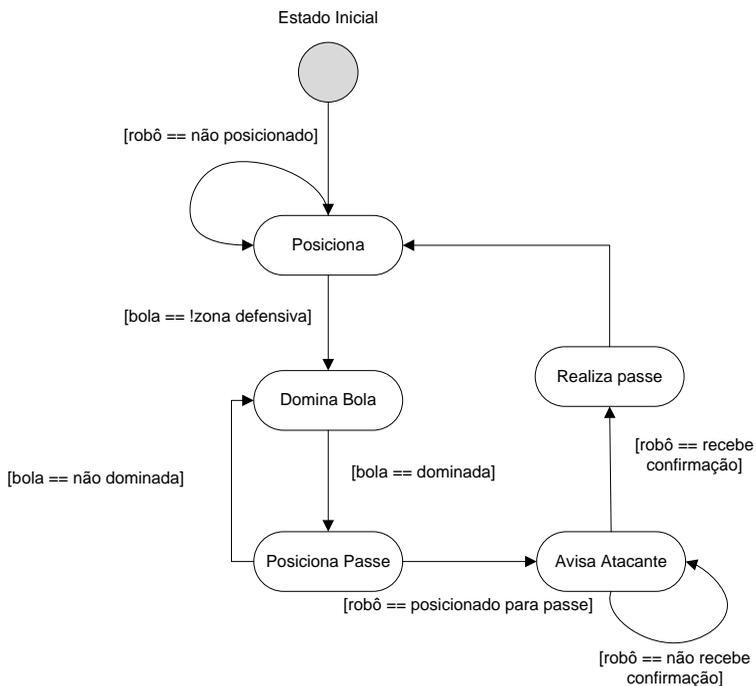
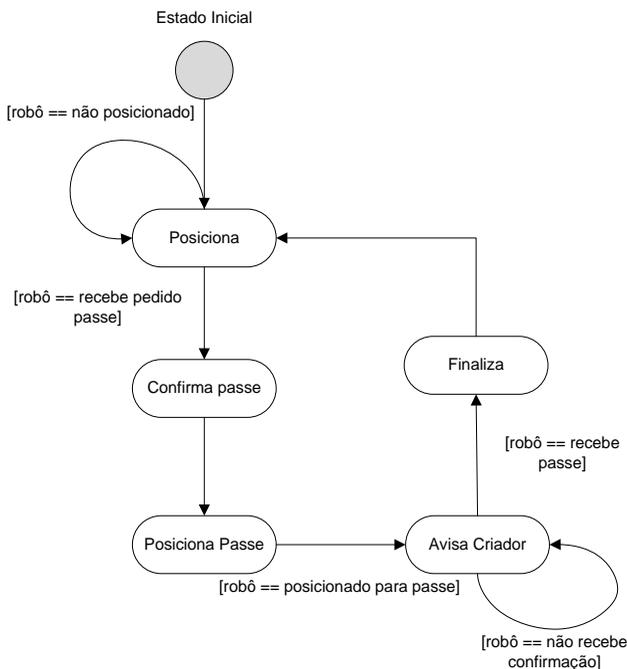


Figura 27 - Máquina de estados do atacante.



O navegador irá calcular rotas livres de colisão, para isso utilizou-se o algoritmo de busca A^* ³, descrito por Hart et al. (1968). Dado as coordenadas de posição dos robôs e da bola, cada robô cria uma matriz em escala reduzida, que representa o campo. Nela são definidas, em tempo de execução, as posições dos outros robôs como obstáculos e um alvo definido pelo sequenciador, que pode ser a posição da bola ou uma posição do campo. O algoritmo A^* irá calcular uma rota livre de colisão e o módulo de controle de baixo nível irá converter a rota em comandos de velocidades para cada um dos atuadores de maneira a controlar o robô de acordo com a rota definida.

³ O A^* é um algoritmo de busca muito utilizado para o planejamento de rotas. Visa encontrar um caminho livre de obstáculos entre dois pontos, com base em uma função de custo (denominada $f(x)$) que consiste na soma de outras duas funções: A função de custo de caminho passado, que representa a distancia do nó inicial até o nó atual (denominado $g(x)$) e a função de custo de caminho futuro, que é uma "estimativa heurística" da distância de x até o alvo (denominado $h(x)$) (RUSSELL; NORVIG, 2004).

Para que sejam atribuídos os comportamentos aos robôs mais aptos é necessário que cada robô quantifique e informe ao líder valores de utilidade para cada tarefa quando solicitado. Para isto, os fatores descritos na Tabela 4 foram considerados:

Tabela 4 - Fatores para a quantificação de utilidade.

Fator	Estado Mínimo	Estado Máximo
Hardware	0%	100%
Bateria	0%	100%
Velocidade	1 m/s	3 m/s
Chute	2 m/s	8 m/s
Domínio	1 (não possui)	2 (possui)
Líder	0 (não)	1000 (sim)

Hardware e Bateria indicam seu estado geral, Velocidade indica a velocidade máxima que o robô pode alcançar, o Chute corresponde a velocidade máxima que o robô consegue disparar a bola, o Domínio define se o robô possui um dispositivo para domínio de bola e o Líder indica se o robô é ou não líder. Com base nesses fatores, foram criadas as métricas para o cálculo de utilidade de acordo com cada comportamento, tais métricas são ilustradas na Tabela 5.

Tabela 5 - Métricas para quantificação de utilidade.

Posição	Métrica
Goleiro	<i>Goleiro = Líder</i>
Defensor	<i>Defensor = Hardware</i>
Criador	$\text{Criador} = \left(\frac{\text{Bateria} + \text{Hardware}}{2} \right) * \text{Domínio} + \text{Velocidade}$
Atacante	<i>Atacante = (Velocidade + Chute) * Domínio</i>

O goleiro é o robô que menos irá se movimentar, gastando menos bateria, por isso o mais adequado para esta posição é sempre o robô líder, que além do seu próprio planejamento local, irá realizar o planejamento global. O defensor caracteriza-se por estar em constante contato físico, tendo maior desgaste de seu hardware, desta forma o robô mais apto para ser defensor será aquele que possuir melhor estado de hardware. O criador terá de se movimentar com velocidade para a criação de jogadas, tendo maior gasto de bateria e apresentando um certo desgaste de hardware. O atacante pouco se movimentará, mas precisará fazê-lo com velocidade, sendo capaz de dominar a bola e finalizar para o gol com rapidez.

5.2. CENÁRIO DE APLICAÇÃO DA ARQUITETURA AO FUTEBOL DE ROBÔS

O "planejador de missão" define um plano de equipe de acordo com o placar do jogo. Inicialmente, com o jogo empatado, o planejador de missão definiu o plano de equipe ilustrado na Figura 28.

Figura 28 - Plano de equipe definido pelo "planejador de missão".

Plano de Equipe	
Identificação do plano:	01.
Descrição:	"1-2-1".
Condição de aplicabilidade:	Empate.
Condição de recisão:	Vencendo, Perdendo, Plano-Abortado.
Tempo limite:	2 minutos.
Comportamento 1:	Goleiro.
Comportamento 2:	Defensor.
Comportamento 3:	Criador.
Comportamento 4:	Criador.
Comportamento 5:	Atacante.

A partir da definição de um plano de equipe, o "alocador de tarefas" inicia um leilão anunciando a missão. A Tabela 6 lista o estado dos fatores de quantificação de cada um dos robôs em um determinado momento. Com base em seus fatores de quantificação de utilidade cada robô calcula um valor de utilidade para cada tarefa de acordo com sua respectiva métrica.

Tabela 6 - Estado dos fatores de quantificação de utilidade do grupo de robôs.

Fator/Robô	Robô 0	Robô 1	Robô 2	Robô 3	Robô 4
Hardware	70%	60%	90%	100%	30%
Bateria	100%	60%	50%	100%	80%
Velocidade	2 m/s	1 m/s	3 m/s	3 m/s	2 m/s
Chute	8 m/s	5 m/s	2 m/s	8 m/s	7 m/s
Domínio	Não	Sim	Sim	Não	Sim
Líder	Sim	Não	Não	Não	Não

Após calcular seu valor de utilidade, cada robô submete sua oferta ao líder. O líder elabora uma matriz de utilidade. Foram considerados dois casos, o primeiro onde a alocação ocorre normalmente, ilustrado na Tabela 7, e o segundo onde o líder aguardou o tempo limite para submissão de ofertas e não recebeu as ofertas dos robôs 2 e 4, ilustrado na Tabela 8.

Neste caso o líder atribuiu o valor “0” nas linhas correspondentes a estes robôs e continuou normalmente o processo de alocação. O fato de o líder eventualmente não receber ofertas, devido a perda de mensagens ou estouro do limite de tempo, não prejudicará a execução da aplicação, mas poderá acarretar em perdas de desempenho referentes a otimalidade da alocação global de recursos.

Tabela 7 - Matriz de utilidade.

Robô/Comportamento	Goleiro	Defensor	Criador	Criador	Atacante
Robô 0	5000	70	255	255	900
Robô 1	0	60	180	180	840
Robô 2	0	90	350	350	400
Robô 3	0	100	400	400	900
Robô 4	0	30	220	220	1440

Tabela 8 - Matriz de utilidade com alocação com perda de mensagens.

Robô/Comportamento	Goleiro	Defensor	Criador	Criador	Atacante
Robô 0	5000	70	255	255	900
Robô 1	0	60	180	180	840
Robô 2	0	0	0	0	0
Robô 3	0	100	400	400	900
Robô 4	0	0	0	0	0

A partir da matriz de utilidade o "alocador de tarefas" utiliza o método húngaro para obter os maiores valores seguindo o critério de melhor aproveitamento global de recursos. O método húngaro consiste em um algoritmo de cinco passos:

1. Subtrair o menor valor de cada linha de todos os valores dessa mesma linha;
2. Subtrair o menor valor de cada coluna de todos os valores de cada coluna;
3. Cobrir com traços linhas e colunas que possuem “0” utilizando um número mínimo de traços;
4. Se o número mínimo de traços utilizado para cobrir as linhas e colunas for igual a ordem da matriz a alocação ótima foi encontrada e o processo é encerrado, mas caso o número seja menor, deve-se pular para o passo 5;
5. Utilizando o menor valor não riscado, diminuir esse valor de todos os valores não riscados e somá-lo a todos os valores riscados. Retornar ao passo 3.

A partir dos maiores valores obtidos, o "alocador de tarefas" define quais tarefas serão alocadas a quais robôs e então envia uma mensagem de atribuição. Cada robô recebe a mensagem, envia uma mensagem de dados com a confirmação de seu estado atual e passa a executar seu respectivo comportamento. Neste momento o planejador de missão recebe todas as confirmações e passa a aguardar o tempo limite de execução do plano, para definir um novo plano, enquanto o alocador de tarefas passa a aguardar a definição de um novo plano de equipe para ser alocado.

6. CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS

Sistemas robóticos são utilizados para a execução de tarefas impossíveis de serem realizadas por seres humanos, podendo também efetuar tarefas perigosas, insalubres e maçantes. Eles apresentam diversas vantagens, tais como maior eficiência e rapidez e custo operacional mais baixo. Entretanto, existem tarefas que não podem ser realizadas por um único robô por serem complexas, exigirem maior efetividade ou necessitarem da robustez e a flexibilidade de sistemas distribuídos.

Estes sistemas multirrobóticos necessitam de uma arquitetura robusta e versátil que coordene os robôs de modo coerente durante a execução das tarefas. Esta arquitetura deve definir políticas de coordenação que determinam como as tarefas serão decompostas e alocadas a cada robô, o modo como o conhecimento e modelagem de ambiente serão representados, além do modo como ocorrerá a comunicação entre os robôs do grupo. Ela também deverá preocupar-se com o controle individual de cada robô, que é responsável, de modo geral, pela sua percepção, planejamento e atuação.

Este trabalho apresentou a Arquitetura de Coordenação de Sistemas Multirrobóticos (ACSM) que está estruturada em duas partes: arquitetura individual e arquitetura de grupo. A arquitetura de grupo refere-se as estratégias globais de coordenação, troca de informações e atribuições de tarefas entre os robôs. A arquitetura individual está focada na divisão dos módulos de responsabilidades individuais de cada robô. O principal objetivo da ACSM consiste na coordenação de sistemas multirrobóticos cooperativos que realizam tarefas fortemente acopladas em ambientes de tempo real.

A arquitetura ACSM visava um bom aproveitamento geral dos recursos, uma alocação efetiva de tarefas aos robôs considerados globalmente mais aptos e o controle e planejamento efetivo de cada robô. Para a realização de testes e avaliação da arquitetura, aplicou-se a ACSM no *grSim*, um ambiente simulado de futebol de robôs que segue os padrões da *RoboCup Small Size League*. Foi possível analisar o aproveitamento geral dos recursos e a alocação de tarefas entre os robôs. Nos casos onde o processo de alocação ocorre normalmente, chega-se a resultados ótimos. A ocorrência de perdas de mensagens podem prejudicar o desempenho referente a otimalidade da alocação global de recursos, mas não impede a aplicação de seguir sua execução normalmente.

Para aperfeiçoar a arquitetura é necessário que ela seja testada em robôs reais e avaliada em outros problemas da robótica. A Seção 6.1 apresenta algumas propostas para trabalhos futuros.

6.1. PROPOSTAS PARA TRABALHOS FUTUROS

Seguem listadas abaixo algumas sugestões para trabalhos futuros que visam aperfeiçoar a arquitetura proposta.

1. Implementar e testar a arquitetura proposta em robôs reais.
2. Avaliar a arquitetura proposta em outros problemas de robótica distribuída.
3. Implementar um algoritmo para eleição do líder.
4. Aperfeiçoar o protocolo de comunicação, tornando-o confiável para a alocação de tarefas.
5. Implementar uma linguagem para a descrição do plano de missão.

REFERÊNCIAS

ANGELO, J. A. **Robotics: a reference guide to the new technology**. Greenwood Publishing LGroup, Incorporated, 2007.

ARKIN, R. C. **Behavior – Based Robotics**. The MIT Press, Cambridge, Massachusetts, pp, 1998.

BERRY, C. A. **Mobile Robots: Inspirations for Multidisciplinary Study**. Morgan & Claypool Publishers, 2012.

BOWLING, M; BROWNING, B; VELOSO, M. **Plays as effective multiagent plans enabling opponent-adaptive play selection**. Proceedings of International Conference on Automated Planning and Scheduling (ICAPS'04), 2004.

BROOKS, R. A. **A robust layered control system for a mobile robot**. IEEE Journal of Robotics and Automation 2:14-23, 1986.

BROWNING, B; BRUCE, J; BOWLING, M; VELOSO, M. **STP: Skills, tactics and plays for multi-robot control in adversarial environments**. Carnegie Mellon University, Pittsburgh, PA 2004.

CAO, Y; FUKUNAGA, A; KAHNG, A. **Cooperative mobile robotics: Antecedents and directions**. Auton. Robot. 4, 1–23, 1997.

CAI, Y; YANG, S. X. **A Survey on multi-robot systems**. World Automation Congress (WAC), vol., no., pp.1,6, 24-28 June 2012.

CRUZ, R. C. **Visão Ativa Aplicada à Robótica: Um Estudo de Sistemas Inteligentes Utilizando Percepção**. Pós-Graduação Lato Senso Especialização em Mecatrônica. USJT, 2004. Disponível em: < <http://www2.dem.inpe.br/mcr/Orient/pdf/RicardoCruz.pdf>> Acesso em: 2013-03-18.

DE PIERI, E. R. **Curso de Robótica Móvel**. Programa de Pós Graduação em Engenharia Elétrica. UFSC, 2002. Disponível em: <ftp://ftp.unilins.edu.br/pacheco/RobosMoveis/p1/robotica_movvel.pdf> Acesso em: 2013-03-18.

FARIA, G. **Uma arquitetura de controle inteligente para múltiplos robôs**. 2006. Tese (Doutorado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2006. Disponível em:

<<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-28022007-100455/>>. Acesso em: 2013-01-29.

FARINELLI, A; IOCCHI, L; NARDI, D;. **Multirobot systems: a classification focused on coordination**. IEEE Transactions on Systems, Man, and Cybernetics, Part B 34(5): 2015-2028, 2004.

GAUTAM, A.; MOHAN, S. **A review of research in multi-robot systems**. *Industrial and Information Systems (ICIIS), 2012 7th IEEE International Conference on* , vol., no., pp.1,5, 6-9 Aug. 2012.

GE, S.S; LEWIS, F.L. (eds.): **Autonomous Mobile Robots: Sensing Control Decision Making and Applications**. CRC Press, Boca Raton, 2006.

GERKEY, B, P.; MATARIC´, Maja J. **A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems**. The International Journal Of Robotics Research, Thousand Oaks, p. 939 - 954. 6 set. 2004.

GERKEY, B.P.; MATARIC, M.J. **Sold!: auction methods for multirobot coordination**. Robotics and Automation, IEEE Transactions on , vol.18, no.5, pp.758,768, Oct 2002.

GHALLAB, M.; NAU, D.; TRAVERSO, P. **Automated Planning: Theory and Practice**. Morgan Kaufmann Publishers Inc., USA, 2004.

G-NIUS. Unmanned Ground Systems. Disponível em: <<http://g-nius.co.il/unmanned-ground-systems/guardium-mk-iii.html>>. Acesso em: 20 mai. 2013.

GOOGLE DEVELOPERS. **Protocol Buffers**. Disponível em: < <https://developers.google.com/protocol-buffers/>> Acesso em: 2013-02-23.

GROTZINGER, J., J. Crisp, et al. **Mars Science Laboratory Mission and Science Investigation**. *Space Science Reviews* 170(1-4): 5-56, 2012.

GURZONI Jr., J. A. ; NASCIMENTO, E. G.; MALHEIRO, D.; FRANCISCHINI, G.; CORTEZ, M.; TEBET, B.; MARTINEZ, S.; FELIZATTI, R.; BIANCHI, R. A. C.; TONIDANDEL F. **RoboFEI Small Size League 2010 Team Description Paper.**

Disponível em:

< <http://small-size.informatik.unibremen.de/tdp/2010/RoboFEI%20TDP%202010.pdf>>

Acesso em: 2013-03-18.

HART, P. E; NILSSON, N. J; RAPHAEL, B. **A Formal Basis for the Heuristic Determination of Minimum Cost Paths.** IEEE Trans. Systems Science and Cybernetics 4(2): 100-107, 1968.

HEINEN, Farlei José. **Sistema de Controle Híbrido para Robôs Móveis Autônomos.** 130 f. Dissertação (Mestrado) - Curso de Mestrado em Computação Aplicada, Centro de Ciências Exatas e Tecnológicas, Unisinos, São Leopoldo, 2002.

KITANO, H. **RoboCup: The Robot World Cup Initiative.** The First International Conference on Autonomous Agent (Agents-97)), Marina del Ray, The ACM Press, 1997.

KORTENKAMP, D.; SIMMONS, R. G. **Robotic Systems Architectures and Programming.** Springer Handbook of Robotics, pp. 187-206, 2008.

KRÖGER, T. **On-Line Trajectory Generation in Robotic Systems.** Springer Tracts in Advanced Robotics, Vol. 58, Springer, January 2010.

KUHN, H.W. **The hungarian method for the assignment problem.** Naval Research Logistics Quarterly 2(1), 83-97, 1955.

KUKA. Industrial Robots. Disponível em: <<http://www.kuka.com>>. Acesso em: 20 abr. 2013.

KUMAGAI, J. **Techno cops police robotic and electronic technology.** Spectrum, IEEE , vol.39, no.12, pp.34,39, Dec 2002.

KURFESS, T. R. **Robotics and Automation Handbook.** Taylor & Francis, 2010

LAVALLE, S. **Planning Algorithms**. Cambridge: Cambridge University Press. ISBN 0-521-86205-1, 2006.

LEI, Y; ZHU, Q; FENG, Z. **Research on architecture of multiple robots system based on the dynamic networks**. *Information and Automation (ICIA), 2010 IEEE International Conference on* , vol., no., pp.93,98, 20-23 June 2010.

MACKWORTH, A. K. **On Seeing Robots**. A. Basu and X. Li, editors, Computer Vision: Systems, Theory and Applications, pp. 1–13, World Scientific Press, Singapore, 1993.

MATARIĆ, M. J. **The robotics primer**, The MIT Press, 2007.

MEHRABI, V.; KOOCHAKZADEH, A.; POORJANDAGHI, S. S.; SAEIDI, A.; POUR, S. M. M.; SHEIKHI, E.; KAVIANI, P.; SAHARKHIZ, S.; RAZZAGHPANAH, A.; PAHLAVANI, A.; BAHMAND, A. **Team Description for Robocup 2012**. Parsian. Amirkabir Univ. Of Technology Robocup Small Size Team. Disponível em:
< http://robocupssl.cpe.ku.ac.th/_media/parsian_ssl_tdp_2012.pdf>
Acessado em: 2013-0-19.

MONAJJEMI V.; KOOCHAKZADEH A.; GHIDARY. S. S.; **grSim – RoboCup Small Size Robot Soccer Simulator**. Lecture Notes in Computer Science, Volume 7416, RoboCup 2011: Robot Soccer World Cup XV, Pages 450-460, 2012.

MORIN, P; SAMSON, C. **Motion control of wheeled mobile robots**. In B. Siciliano and O. Khatib (eds.), Handbook of Robotics, volume 1, 799-826. Springer, Germany, 1st edition, 2008.

MURPHY, R. **An Introduction to AI Robotics**. Intelligent Robotics and Autonomous Agents. MIT Press, 2000.

MURPHY, R. **Marsupial robots for urban search and rescue**, IEEE Intell. Syst. 15(2), 4–19, 2000.

NAKHAEGINIA, D.; TANG, S.H.; MOHD NOOR, S.B. & MOTLAGH, O. **A review of control architectures for autonomous navigation of**

mobile robots. International Journal of the Physical Sciences, Vol. 6, No.2, pp. 169-174, 18, ISSN 1992-1950, 2011.

NIKU, S. **Introduction to Robotics.** John Wiley & Sons, 2011.

NODA, I. **Soccer Server: A Simulator for RoboCup.** JSAI AI-Symposium 95: Special Session on RoboCup, 1995.

NODA, I; SUZUKI, S.; MATSUBARA, H.; ASADA, M.; KITANO, H. **RoboCup-97 — The First Robot Soccer Games and Conferences.** AI Magazine 19(3): 49–59, 1998.

PARKER, L. E.; **Multiple Mobile Robot Systems.** Springer Handbook of Robotics, B. Siciliano and O. Khatib, editors, 2008.

PEDERSEN, L; KORTENKAMP, D; WETTERGREEN, I; NOURBAKSH, I. **A Survey of Space Robotics.** Proceedings of the 7th International Symposium on Artificial Intelligent, Robotics and Automation in Space (i-SAIRAS-03), 2003.

REIS, L. P. **Coordination in Multi-Agent Systems:** Applications in University Management and Robotic Soccer. PhD Thesis, FEUP, July of 2003. Disponível em: < <http://paginas.fe.up.pt/~lpreis/Research.htm>> Acesso em: 2013-04-02.

ROBOCUP. Disponível em: < <http://www.robocup.org.br/>>. Acesso em: 03 jun. 2013

ROBOCUP RULES. **Laws of the RoboCup Small Size League 2012.** Disponível em: <http://small-size.informatik.uni-bremen.de/_media/rules:ssl-rules-2012.pdf> Acesso em: 2013-03-18.

RUSSELL, S; NORVIG, P. **Inteligência Artificial.** 2. Ed. Rio de Janeiro: Campos, 2004.

SICILIANO, B.; KHATIB, O. **Springer Handbook of Robotics.** Heidelberg: Springer-Verlag, 2008.

SIEGWART, R.; NOURBAKHSI, I. **Introduction to Autonomous Mobile Robots**. MIT Press, 2004.

SOLUTIONS, A. K. Robotics, Jones & Bartlett Learning, 2010

STONE, L. W; **The History of Robotics**. Robotics and Automation Handbook, Taylor & Francis. 2010.

STROUPE, A.; OKON, A.; ROBINSON, M.; HUNTSBERGER, T.; AGHAZARIAN, H.; Baumgartner, E. **Sustainable cooperative robotic technologies for human and robotic outpost infrastructure construction and maintenance**. *Auton. Robot.* 20(2), 113–123, 2006.

TANENBAUM, Andrew S.; STEEN, Maarten Van. **Sistemas Distribuídos: Princípios e Paradigmas**. 2. ed. São Paulo: Person Education do Brasil, 2008. 402 p.

TRIGUI, S.; KOUBAA, A.; BEN JAMAA, M.; CHAARI, I.; AL-SHALFAN, K. **Coordination in a multi-robot surveillance application using Wireless Sensor Networks**. *Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean* , vol., no., pp.989,992, 25-28 March 2012

YUAN, P.; MOALLEM, M.; PATEL, R.V. **A real-time task-oriented scheduling algorithm for distributed multi-robot systems**. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on* , vol.3, no., pp.2562,2567 Vol.3, 26 April-1 May 2004.

YAN, Y; ZHENMIN, Z. **Control Architecture for Autonomous Multi-Robot System: Survey and Analysis**. *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on* , vol.4, no., pp.376,379, 10-11 Oct. 2009.

ZICKLER S.; LAUE T.; BIRBACH O.; WONGPHATI M. **Veloso M.;SSL-Vision: The Shared Vision System for the RoboCup Small Size League**. RoboCup 2009: Robot Soccer World Cup XIII. Springer. pp. 425-436, 2009.

ZLOT, R. M. **An Auction-Based Approach to Complex Task Allocation for Multirobot Teams**. PhD thesis, Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave 2006.