

UNIVERSIDADE FEDERAL DE SANTA CATARINA
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

TELMO DOS SANTOS KLIPP

**PROPOSTA DE UMA ARQUITETURA PARA ALOCAÇÃO DE
TAREFAS EM GRUPOS DE ROBÔS MÓVEIS BASEADA EM
ACORDO BIZANTINO**

Araranguá, 28 de fevereiro de 2013

TELMO DOS SANTOS KLIPP

PROPOSTA DE UMA ARQUITETURA PARA ALOCAÇÃO DE TAREFAS EM GRUPOS
DE ROBÔS MÓVEIS BASEADA EM ACORDO BIZANTINO

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 de Informação e comunicação. Sob a orientação do Professor Anderson Luiz Fernandes Perez.

Araranguá, 2013

Telmo dos Santos Klipp

**PROPOSTA DE UMA ARQUITETURA PARA ALOCAÇÃO DE TAREFAS EM
GRUPOS DE ROBÔS MÓVEIS BASEADA EM ACORDO BIZANTINO**

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


Professor Anderson Luiz Fernandes Perez, Dr.
Presidente da Banca - Orientador


Professora Eliane Pozzebon, Dr.^a.
Membro


Professor Carlos André de Souza Rocha, MSc.
Membro

Araranguá, 28 de fevereiro de 2013

*Aos meus pais, Nauro José e Rosa
Honorata, e ao meu irmão Elton Davi.*

AGRADECIMENTOS

A Deus por me conceder sua centelha divina, a vida da qual desfruto todos os dias. Aos meus pais que sempre me apoiaram concedendo-me educação para a vida. Aos meus colegas e amigos pelo apoio e presença nessa caminhada de aprendizado, em especial ao Tiago Elias e ao Fernando Puntel. Aos membros do LARM (Laboratório de Automação e Robótica Móvel) e aos nobres colegas do Posto de Fiscalização Sanitária (BR-101, Araranguá) pelos auxílios concedidos. A Universidade Federal de Santa Catarina que abre tantos caminhos e em especial ao meu orientador, professor Anderson Luiz Fernandez Perez, por suas importantes contribuições ao desenvolvimento do trabalho e por ser razão de intensa motivação, paciência e inspiração.

Examinai tudo. Retende o bem.

Tessalonicenses 5-21

RESUMO

Robôs móveis vêm sendo usados cada vez mais para a execução de inúmeras tarefas, seja na indústria, na prestação de serviços ou em aplicações militares. Muitas tarefas são de natureza complexa e exigem a adoção de mais de um robô para sua completa execução. Sendo assim, a formação de grupos de robôs é fundamental e deve basear-se nas características da tarefa a ser executada, ou seja, das exigências funcionais do que é preciso para executá-la, mas também garantindo que, havendo uma falha de um ou mais robôs, a tarefa poderá ser concluída, mesmo em um tempo computacionalmente maior. Assim os robôs membros de um grupo devem ser escolhidos baseados em critérios físicos, comportamentais (habilidades) e de tolerância a falhas. Uma maneira de garantir a execução de uma tarefa por um grupo de robôs é adotar alguma técnica de tolerância a falhas, tal como o acordo bizantino. Acordo bizantino é baseado no princípio de que havendo falha de um robô no grupo existirão tantos outros que poderão dar continuidade a execução da tarefa. Este trabalho apresenta uma proposta de arquitetura para a alocação de tarefas em sistema multi-robôs baseada na taxonomia MRTA e que utiliza acordo bizantino para garantir o consenso e a resiliência entre membros de um grupo de robôs.

Palavras-chave: robótica móvel, sistemas multi-robôs, acordo bizantino, alocação de tarefas, algoritmo de eleição.

ABSTRACT

Mobile robots are increasingly being used to perform many tasks, such as in industry, service or military applications. Many tasks are complex and require the adoption of more than one robot to complete its execution. Thus, the formation of groups of robots is critical and must be based on the characteristics of the task to be performed, that is, the functional requirements it takes to carry it out, but also ensuring that there is a failure of one or more robots, the task can be completed, even in a longer computationally time. Thus the robot group members should be chosen based on physical criteria, behavioral (skills) and fault tolerance. One way to ensure the execution of a task by a group of robots is to adopt some fault tolerance technique, such as byzantine agreement. Byzantine Agreement is based on the principle that existing failure of one robot in the group there will be many others who will continue the task. This work presents a proposed architecture for task allocation in multirobot system based on MRTA taxonomy and byzantine agreement to ensure consensus and resilience among members of a group of robots.

Keywords: mobile robotics, multirobot systems, byzantine Agreement, task allocation, election algorithm.

LISTA DE ILUSTRAÇÕES

Figura 2.1 Braço manipulador de linha industrial.....	22
Figura 2.2. Possível classificação para máquinas.....	23
Figura 2.3 Robô <i>Curiosity</i>	23
Figura 2.4. Robô móvel <i>Asimo</i>	24
Figura 2.5. Ciclo percepção-ação.....	26
Figura 2.6 Ligação entre robô, tarefa e ambiente.....	26
Figura 2.7 Alguns sensores presentes no robô <i>PowerBot</i>	28
Figura 2.8 Alguns atuadores disponíveis no robô <i>PowerBot</i>	28
Figura 2.9 Classificação segundo a mobilidade.....	29
Figura 2.10 Robôs móveis terrestres.....	30
Figura 2.11 Veículo aéreo autônomo <i>Tiriba</i>	31
Figura 2.12 Veículo autônomo aquático <i>Argo Float</i>	32
Figura 2.13 Sistema de controle de robôs móveis.....	33
Figura 2.14 Arquitetura de controle reativa.....	34
Figura 2.15 Arquitetura de controle deliberativa.....	35
Figura 2.16 Arquitetura de controle híbrida.....	36
Figura 3.1 Robôs do projeto <i>Swarmanoid</i>	40
Figura 3.2 Representação de tarefas.....	42

Figura 3.3 Clássico problema bizantino.....	50
Figura 4.1 Visão geral da arquitetura MRTA-B.....	53
Figura 4.2 Anúncio das capacidades por parte dos robôs da população.	54
Figura 4.3 Exemplo de um bloco de descrição da tarefa.....	55
Figura 4.4 Exemplo do algoritmo do anel.....	58
Figura 4.5 Processo de eleição.	59
Figura 4.6 Arquitetura do sistema de controle de cada robô.....	60
Figura 4.7 Exemplo do processo de estabelecimento do acordo bizantino.....	62

LISTA DE TABELAS

Tabela 1 Taxonomia de problemas de MRTA.	45
Tabela 2 Abstração das informações do senso populacional	64
Tabela 3 Abstração dos requisitos das tarefas presentes nos BDT.	65
Tabela 4 Resultado da alocação de tarefas com os respectivos grupos formados.	67

LISTA DE ABREVIATURAS E SIGLAS

GPS	<i>Global Position System</i>
IA	<i>Inteligência Artificial</i>
IAD	<i>Inteligência Artificial Distribuída</i>
JPL	<i>Jet Propulsion Laboratory</i>
MR	<i>Multi Robots</i>
MRTA	<i>Multi Robot Task Allocation</i>
MT	<i>Multi Tasks</i>
NASA	<i>National Aeronautics and Space Administration</i>
RIA	<i>Robotic Industries Association</i>
SMR	<i>Sistema de Multi-robôs</i>
SR	<i>Single Robots Tasks</i>
ST	<i>Single Tasks Robots</i>

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 <i>Objetivos</i>	17
1.2 <i>Motivação e justificativa</i>	17
1.3 <i>Metodologia</i>	18
1.4 <i>Organização do trabalho</i>	19
2 ROBÓTICA MÓVEL	21
2.1 <i>Breve histórico e importância</i>	21
2.2 <i>Robôs móveis</i>	25
2.2.1 <i>Percepção e ação</i>	27
2.2.2 <i>Classificação de robôs móveis quanto a mobilidade</i>	29
2.3 <i>Sistemas de controle em robôs</i>	32
2.4 <i>Arquitetura de controle</i>	33
2.4.1 <i>Arquitetura de controle reativa</i>	34
2.4.2 <i>Arquitetura de controle deliberativa</i>	34
2.4.3 <i>Arquitetura de controle híbrida</i>	35
3 COMPARTILHAMENTO DE TAREFAS EM ROBÓTICA MÓVEL.....	37
3.1 <i>Sistemas distribuídos</i>	37
3.2 <i>Tarefas em robótica móvel</i>	38
3.2.1 <i>Utilidade na execução de tarefas</i>	40
3.2.2 <i>Tratamento de tarefas</i>	41
3.2.3 <i>Modelo e ambiente da tarefa</i>	43
3.3 <i>Métodos de alocação de tarefas entre múltiplos robôs</i>	44
3.4 <i>Acordos em sistemas com falhas</i>	46
3.4.1 <i>Acordo bizantino</i>	49
4 ARQUITETURA DE COMPARTILHAMENTO DE TAREFAS.....	52
4.1 <i>Definições preliminares</i>	52
4.2 <i>Visão geral da arquitetura proposta</i>	53
4.3 <i>Modelo de comunicação na arquitetura MRTA-B</i>	56
4.4 <i>Formação de grupos</i>	57
4.5 <i>Processo de eleição do coordenador e dos líderes de grupo</i>	58
4.6 <i>Arquitetura de controle dos robôs</i>	60
4.7 <i>Comunicação baseada em acordo bizantino</i>	61
5 ANÁLISE SOBRE A ARQUITETURA PROPOSTA.....	63
5.1 <i>Definições gerais</i>	63

6	CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS..	69
6.1	<i>Propostas para trabalhos futuros.....</i>	<i>70</i>
	REFERÊNCIAS.....	71

1 INTRODUÇÃO

Ao longo dos tempos a robótica móvel vem sendo aplicada a uma gama de tarefas (BRUMITT, 1998), tais como: exploração espacial (*JPL - NASA*), sondas de inspeção de dutos de petróleo e gás (robô *G.I.R.I.N.O - Petrobrás*), cirurgias (*Da Vinci Surgical System - Intuitive Surgical*), eletrodomésticos (robô *Roomba - iRobot*), entre outras. Em algumas aplicações o uso de apenas um único robô não é o suficiente para a solução do problema, neste caso faz-se necessário à utilização de um grupo de robôs móveis.

A partir do interesse na área de robótica móvel se faz necessário entender além do funcionamento de um robô as tecnologias que permitem atribuir comportamentos mais complexos, como o compartilhamento de tarefas entre robôs, que permite estender a capacidade individual de um robô conjuntamente com outros a problemas maiores ou mais elaborados.

O compartilhamento, seja de informações ou recursos, tem se apresentado como solução para diversos problemas de origem distribuída e assume um papel importante em diferentes tecnologias, permitir seu uso pode significar economia de tempo, esforço, capacidade de processamento, vida útil de recursos, entre outros fatores.

Não obstante, existem problemas que podem ser complexos o suficiente para exigir uma divisão em partes e distribuição entre entidades especializadas nessas partes para uma possível solução. Nessa divisão de tarefas, vê-se que, o termo distribuído permite uma série de benefícios no que tange a utilização de recursos, podendo-se aplicar vários conceitos computacionais que otimizam esse processo, como, paralelismo, multiprocessamento, divisão da carga de trabalho, entre outros.

O uso e o compartilhamento de tarefas entre robôs móveis pode significar uma vantagem na execução destas. Segundo Ribeiro et al.(2001), tarefas inerentemente distribuídas (espaço, tempo, funcionalidade), como busca e resgate, exploração, combate a incêndios são adequadas para o uso de um grupo de robôs.

Por mais que robôs possam apresentar cada vez mais um comportamento inteligente utilizando tecnologias que os aproximem da capacidade cognitiva e racional humana, o que mais diferencia e valoriza o seu uso é o compartilhamento de tarefas proporcionando uma variedade de possibilidades tecnológicas. Um sistema pode atribuir a capacidade de monitorar, explorar, avaliar locais objetos e operações em vastas regiões geográficas.

Destaca-se o aproveitamento de robôs móveis para o provimento de diversos serviços e que por meio de um sistema de ampla abrangência possa produzir dados para análise e gerência de amplos recursos dispostos geograficamente, a exemplo de um sistema de vigilância efetuado por robôs em locais e ambientes determinados (prédios, pátios, oceanos, florestas).

A adoção de um grupo de robôs para a realização de algumas tarefas exige que o sistema de controle seja bem projetado, neste caso, é importante que o sistema seja capaz de determinar o que cada robô fará, ou seja, qual a tarefa que ele executará. Para este tipo de aplicação faz-se necessário à adoção de um sistema de compartilhamento de tarefas, que visa dividir o problema em problemas menores e então associar cada problema menor a um tipo de robô, respeitando as características e potencialidade do robô.

Para que seja possível o compartilhamento de tarefas entre um grupo de robôs móveis é necessário a adoção de algum critério de alocação. A alocação de tarefas deve considerar que esta pode ser executada por um único robô, ou por mais de um robô, e ainda, que alguns robôs possam realizar mais de uma tarefa (GERKEY; MATARIC, 2004).

Devido à multiplicidade de atribuições entre robôs e tarefas pretende-se utilizar uma taxonomia que trata de problemas de alocação de tarefas, MRTA (*Multi Robot Task Allocation*) proposta por Gerkey e Mataric (2004), somadas à necessidade de gerência e alocação dessas tarefas entre o grupo.

Em sistemas multi-robôs a comunicação entre os robôs é essencial e deve incorporar confiabilidade e segurança na troca de mensagens, para que um grupo de robôs possa entrar em concordância sobre condições aplicadas ao mesmo (aceitação de uma tarefa). Em razão disso será proposta uma abordagem confiável através de acordo bizantino, para garantir que a comunicação entre o grupo seja tolerante a falhas eventuais, que não podem ser previstas.

A garantia de tolerância a falhas em grupos de robôs móveis está relacionada com o objetivo principal deste trabalho que culmina com a proposta de uma arquitetura capaz de alocar tarefas entre robôs de maneira confiável. A arquitetura proposta é chamada de MRTA-

B, pois faz uso da taxonomia MRTA com comunicação confiável baseada no acordo bizantino.

1.1 Objetivos

Esta sessão apresenta o objetivo geral e os objetivos específicos deste trabalho.

- **Geral**

O objetivo principal deste trabalho é o projeto de uma arquitetura de controle para grupos de robôs móveis que vise a alocação de tarefas em grupos de robôs móveis de maneira confiável com o uso de acordo bizantino.

- **Específicos**

Para se atingir o objetivo geral proposto neste trabalho, faz-se necessário o cumprimento dos seguintes objetivos específicos:

- Apresentar as principais técnicas de controle de robôs móveis;
- Apresentar os algoritmos de alocação de tarefas entre robôs móveis, sobretudo a taxonomia MRTA;
- Apresentar a abordagem de acordo bizantino;
- Propor uma arquitetura de controle que seja baseada no modelo MRTA e que utilize acordo bizantino, intitulada MRTA-B;
- Análise sobre a arquitetura MRTA-B.

1.2 Motivação e justificativa

O uso de robôs especializados no provimento de serviços é cada vez mais corrente em vários setores de atividades humanas, é uma tendência atual e futura, que se configura em atividades que são tediosas, repetitivas, insalubres, de risco, no entanto, alguns desses serviços não podem ser realizados por um único robô, sendo necessária a utilização de um grupo (ZLOT, 2006).

Como exemplo cita-se um grupo de robôs para serviços de busca e resgate ou combate a incêndio, onde cada robô tenha capacidades próprias para executar um papel no serviço geral que somadas às capacidades de outros robôs determinam a realização do serviço. Um robô pode ter o gatilho do disparo de água, outro ser capaz de realizar arrombamentos e perfu-

rações, outro ainda pode ter a capacidade de retirar escombros, entre outras especialidades necessárias.

Isso leva a questões inerentes ao trabalho em grupo, que é cooperativo. Quando cada robô deve iniciar sua tarefa? Quais informações um robô deve repassar para outros no grupo? Quando uma tarefa executada por um robô não é mais necessária? Nessas condições a realização de tarefas implica na proposição de soluções a problemas que influenciam diretamente no sucesso e qualidade do serviço geral.

Um compartilhamento de tarefas bem realizado têm implicações positivas em sistemas distribuídos. Um sistema de robôs móveis deve ter uma arquitetura bem construída, nele a comunicação e divisão de tarefas tem papéis fundamentais.

Cada robô, membro de um grupo, deve ter suas habilidades e modo de funcionamento mapeado e classificado em virtude da distribuição de tarefas, que devem ser atribuídas para os robôs certos aptos a executá-las.

A comunicação tem fator crucial e essencial em uma arquitetura distribuída exatamente como se configura no uso de um grupo de robôs móveis. Os robôs pertencentes ao grupo, além de manter o conhecimento sobre outros membros, devem ter um meio para provar a origem das tarefas que passam a receber, pois existem inúmeras possibilidades de corrupção das atividades desses robôs, seja por códigos maliciosos ou mecanismos de ataque externos, por interesses individuais de um robô, por um robô do grupo ficar inoperante, entre outros fatores.

O projeto de uma arquitetura que garanta a confiabilidade na realização de serviços por um grupo heterogêneo de robôs móveis é o principal objetivo deste trabalho.

1.3 Metodologia

Para a formulação da proposta apresentada neste trabalho foram estudados os conceitos e técnicas envolvidos na alocação de tarefas em SMR (Sistemas Multi-Robôs), bem como características, funcionamento e aplicação de robôs móveis. Ao longo do texto deste trabalho serão apresentados técnicas e algoritmos da computação distribuída, como, o algoritmo de eleição do anel e o consenso para tratamento de falhas bizantinas, sendo que, estes serão diretamente aplicados ao projeto da arquitetura MRTA-B.

A metodologia adotada levará em consideração uma arquitetura distribuída, onde um grupo heterogêneo de robôs será constituído. Cada grupo de robôs deverá solucionar um ou mais problemas, compartilhando entre si uma ou mais tarefas em busca de uma solução. No

projeto da arquitetura será levado em consideração, a estrutura, os componentes e suas ligações, as entidades integrantes, os processos necessários, e as condições dentro da arquitetura proposta para que ela forneça soluções na área de SMR.

Para que a alocação de tarefas seja possível será aplicado um ou mais algoritmos da arquitetura distribuída, sendo que, a divisão de tarefas será baseada na taxonomia MRTA, e no caso da comunicação será utilizado o acordo bizantino, sendo que os mesmos serão estudados e exemplificados no decorrer deste trabalho.

Os resultados obtidos serão exemplificados ao final deste trabalho e se basearam em um análise sobre a arquitetura de controle através de um estudo de caso. Serão feitas suposições sobre o funcionamento da arquitetura MRTA-B diante de uma população de robôs e uma lista de tarefas.

1.4 Organização do trabalho

Este documento, além desta introdução, está organizado em mais 5 (cinco) capítulos que versam sobre os seguintes conteúdos:

No Capítulo 2 serão descritas algumas definições em robótica móvel, modelos de arquiteturas, sua importância e aplicabilidade, bem como conceitos de projeto, arquitetura e implementação de robôs capazes de interagir com um ambiente real.

O Capítulo 3 aborda tarefas em robótica móvel e técnicas que permitam seu compartilhamento, tais como, decomposição e alocação, bem como a taxonomia MRTA (*Multi Robot Task Allocation*) para classificação de problemas de alocação de tarefas. O capítulo também aborda o conceito de acordos em sistemas com falhas bizantinas.

O Capítulo 4 descreve a arquitetura proposta para o controle de múltiplos robôs móveis. Apresenta uma descrição mais detalhada do problema, dos conceitos e definições que dão embasamento teórico a arquitetura MRTA-B que visa solucionar o problema de alocação de tarefas em grupos de robôs móveis de maneira confiável com a adoção de acordo bizantino.

O Capítulo 5 apresenta uma análise sistêmica do processo de alocação de tarefas a grupos de robôs com o uso da arquitetura MRTA-B. Para viabilizar a análise é descrito um estudo de caso.

O Capítulo 6 apresenta as considerações finais e lista um conjunto de tópicos que poderão ser realizados como trabalhos futuros.

2 ROBÓTICA MÓVEL

Neste capítulo serão descritas algumas definições em robótica móvel, modelos de arquiteturas, sua importância e aplicabilidade, bem como conceitos de projeto, arquitetura e implementação de robôs capazes de interagir com um ambiente real.

2.1 Breve histórico e importância

A robótica móvel possui um vasto campo de estudo, devido à necessidade de interdisciplinaridade de várias áreas do conhecimento como Mecatrônica, Tecnologias de Informação, Inteligência Artificial, Sistemas de Comunicação dentre outras (Rosário, 2005).

As pesquisas em robótica visam o projeto de peças mecânicas como rodas ou esteiras, bem como o desenvolvimento de hardware e circuitos eletrônicos juntamente com softwares de controle que podem ser desde um simples programa até um sistema operacional embarcado.

A história da humanidade revela sempre um interesse ligado à pesquisa e prática na produção de máquinas e dispositivos capazes de aliviar, substituir e ultrapassar a capacidade humana em tarefas cotidianas, como autômatos (ancestrais dos robôs atuais), robôs industriais (braços manipuladores), e mais atualmente robôs móveis.

A aplicabilidade de robôs se expande com a convergência de novas tecnologias e novas necessidades do ser humano. A evolução das ciências e economia, com destaque para circuitos integrados em microcomputadores, possibilitou o desenvolvimento e uso dos robôs sobre os mais diferentes aspectos na indústria, entretenimento e pesquisas científicas (ROSÁRIO, 2005).

A partir da revolução industrial, após a segunda guerra mundial, os robôs começaram a ser empregados de forma mais ostensiva em indústrias, área onde encontraram maior di-

mensão de aplicabilidade, com a substituição do trabalho braçal por estruturas mecânicas sobre bases fixas, braços mecânicos ou braços manipuladores.

Os braços manipuladores em linhas de montagem realizam movimentos padronizados geralmente simples e repetitivos. Atualmente sistemas automatizados são amplamente utilizados nas indústrias, e em seguimentos como, exploração de ambientes e pesquisas científicas, na medicina por meio de próteses e equipamentos cirúrgicos, na área militar e de segurança com a substituição do ser humano pelo uso de robôs manipuladores em situações de risco (ROSARIO, 2005). A Figura 2.1 ilustra um braço manipulador fabricada pela KUKA Roboter.



Figura 2.1 Braço manipulador de linha industrial.
Extraído de Kuka (2012).

A *Robotic Industries Association* (RIA) considera um robô como sendo um manipulador programável multi-funcional capaz de mover materiais, partes, ferramentas ou dispositivos específicos através de movimentos variáveis programados para realizar uma variedade de tarefas RIA apud (NEHMZOW, 2003).

Segundo Silva (2003), esta definição descreve toda uma categoria de máquinas, sendo possível considerar qualquer equipamento capaz de ser programado como um robô. A utilização de robôs (na condição de máquinas-ferramentas), de um modo geral, grandes, fixos, desprovidos de inteligência de alto nível e autonomia acontece a décadas nos cenários industriais, pela substituição da mão de obra humana, em especial onde a riscos envolvidos. Uma possível classificação é ilustrada na Figura 2.2:

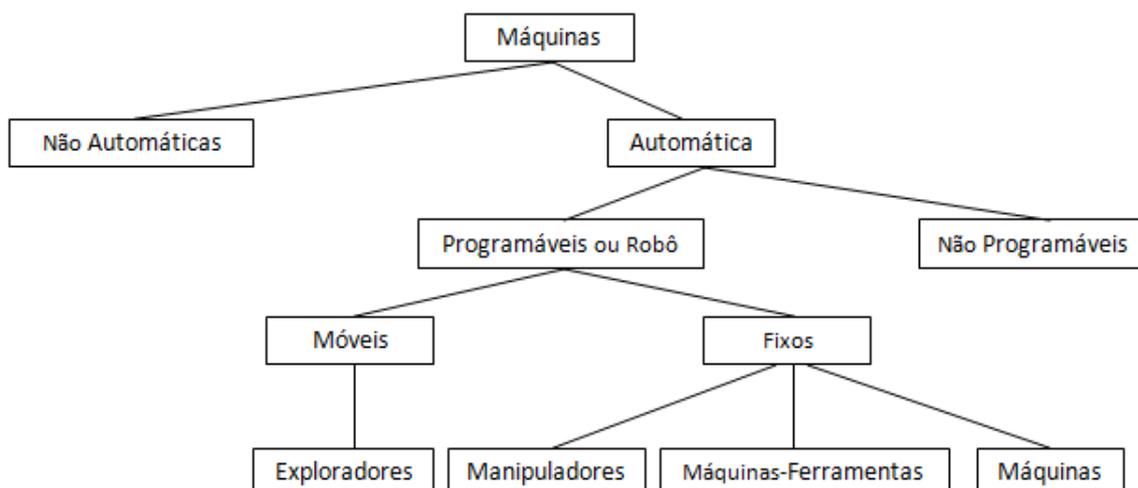


Figura 2.2. Possível classificação para máquinas.
Extraído de Silva (2003).

Os robôs móveis, pela inerente capacidade de locomoção e aplicabilidade em tarefas onde não existam limites geográficos, programáveis em alto nível e capazes de interagir com o ambiente via sensores e atuadores, são muito mais versáteis, independentemente da capacidade de autonomia (PEREIRA, 2003).

O desenvolvimento de robôs móveis tem aumentado, porém com um uso mais ostensivo e restrito, geralmente em indústrias, laboratórios e projetos de pesquisa de universidades, em instituições e agências de pesquisa. Um exemplo é o robô Curiosity desenvolvido no JPL (*Jet Propulsion Laboratory*) da *California Institute of Technology* em parceria com a NASA (*National Aeronautics and Space Administration*), que é considerado um laboratório de ciência móvel em solo marciano. A Figura 2.3 ilustra o robô Curiosity.



Figura 2.3 Robô *Curiosity*.
Extraído de Nasa (2012).

Segundo Marchi (2001) um robô móvel é um dispositivo mecânico montado sobre uma base não fixa que age sob o controle de um sistema computacional, equipado com sensores e atuadores que o permitem interagir com o ambiente.

O foco na construção de robôs móveis, dotados de locomoção e capacidade de interagir com o ambiente expande as possibilidades do uso da robótica, mas insere um nível de complexidade exponencialmente maior à medida que a autonomia desses robôs e o nível de entropia nos ambientes em que estão imersos crescem.

Em diversas áreas (ciência, exploração aeroespacial e terrestre, infraestrutura, indústria, automatização de processos), esta complexidade é confrontada com a utilidade para as necessidades e progressos humanos presentes e futuros, denotando a importância da robótica móvel, seja para garantir a segurança do ser humano, seja pela economia de tempo e esforço.

O produto dessa área são robôs capazes de se moverem em um ambiente e apresentar comportamentos frente ao ambiente em que estão inseridos. A Figura 2.4 ilustra o robô móvel Asimo projetado pela Honda.

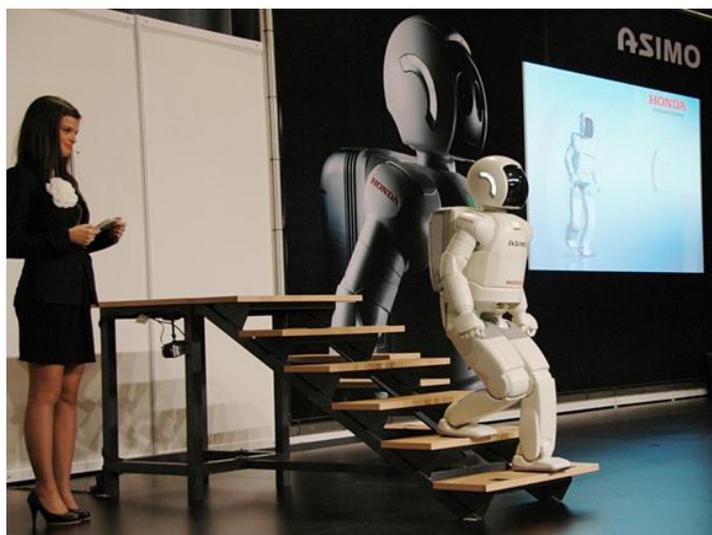


Figura 2.4. Robô móvel Asimo.
Extraído de *Honda Motors* (2012).

Robôs móveis podem ser de grande utilidade sobre vários aspectos e avaliações, sobretudo pelo potencial de realizar tarefas de complexidade acima da capacidade humana.

Na literatura encontra-se algumas classificações para robôs móveis segundo suas funcionalidades dentre as quais citam-se:

a) **Robôs Industriais** – São robôs que atuam em ambientes estruturados realizando tarefas em sequência, possuem conhecimento do ambiente em que estão imersos através de

informações pertinentes a suas atividades como, posição própria, posição de objetos. Estes robôs, geralmente são plataformas móveis que executam tarefas pesadas como carregar grande quantidade de materiais, ferramentas, entre outros. Navegam em ambientes industriais seguindo linhas no chão e são denominados AGVs (*Automated Guided Vehicles*) (DE PIERI 2002);

b) **Robôs de serviço** – São robôs imersos em ambientes estruturados (residências, pátios, etc), e reconhecidos por meio de modelos internos. Estes robôs realizam atividades com certa autonomia frente a eventos inesperados como, desviar de pessoas e objetos, sendo que para isso processam a informação de sensores. São aplicáveis a uma gama de serviços gerais como, limpeza, transporte, vigilância, manipulação de objetos (DE PIERI 2002).

c) **Robôs de campo** – São robôs que executam tarefas em ambientes desestruturados, pouco conhecidos, muitas vezes perigosos. Podem ser usados em exploração espacial, mineração, busca e resgate, limpeza de acidentes nucleares, entre outras aplicações, (DE PIERI 2002). Devido aos ambientes em que atuam, necessitam de alto grau de autonomia, possuem relativo hardware e executam algoritmos de alto nível para mapeamento, localização e locomoção, muitas vezes construindo modelos de mundo em tempo real.

d) **Para pesquisa** – Robôs usados para estudos e pesquisas, geralmente na área acadêmica, muitos destes são de baixo custo.

e) **Entretenimento** - Robôs associados a diversão voltados a públicos variados. São construídos para apresentarem uma gama de atividades ou comportamentos ligados a entretenimento, como interagir com o usuário, desempenhar papel de brinquedo inteligente, fornecer jogos ao usuário.

2.2 Robôs móveis

Basicamente um robô móvel é composto de uma arquitetura de hardware projetada para integrar e exercer funções sobre partes mecânicas e um sistema de controle. Segundo Bräunl (2008), um robô móvel é um sistema autônomo dotado de sensores e atuadores capaz de interagir com o mundo real.

A interação com o ambiente se dá através de ciclos percepção-ação que consistem em três passos fundamentais (DE PIERI 2002):

- Obtenção de informações sobre o ambiente por meio de sensores;

- Processamento das informações e seleção de ações a serem executadas;
- Execução das ações selecionadas por meio de atuadores.

Um robô realiza diversos ciclos de percepção-ação, modificando o estado do ambiente em busca da realização da tarefa (RIBEIRO et al., 2001). A Figura 2.5 ilustra o ciclo percepção-ação.

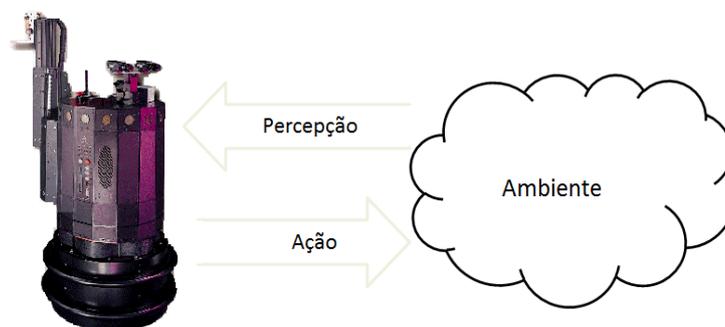


Figura 2.5. Ciclo percepção-ação.
Adaptado de Perez (2006).

Para que um robô apresente uma ou várias funcionalidades ele deve ser projetado sobre a perspectiva de como solucionar um problema do ponto de vista do ambiente em que está inserido, interagindo de forma a receber informações e agir nesse ambiente real.

Robô, tarefa e ambiente estão fortemente ligados. O comportamento geral de um robô resulta da interação desses três fatores, que são interdependentes e influenciam um ao outro. A funcionalidade e operabilidade de um robô estão relacionadas a execução de uma ou mais tarefas em um ambiente determinado (NEHMZOW, 2003). A Figura 2.6 ilustra essa interdependência.

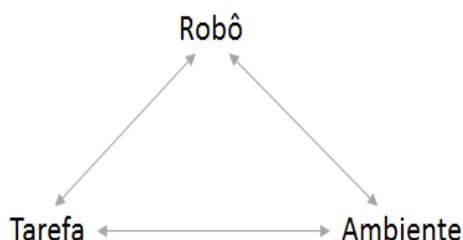


Figura 2.6 Ligação entre robô, tarefa e ambiente.
Adaptado de Nehmzow (2003).

Nesse contexto deve-se compreender os meios de que um robô dispõe para interagir diretamente com o ambiente, ou seja, os sensores e atuadores. Isso envolve diretamente qual tipo de problema estará apto a resolver.

2.2.1 Percepção e ação

Uma característica importante em robôs é a capacidade de extrair informações pertinentes do ambiente, resultando na sua percepção de mundo. Essa percepção é realizada através de sensores.

Um sensor é um dispositivo que pode sentir e mensurar certa propriedade física de um ambiente, como, temperatura, luminosidade, resistência ao toque, peso, tamanho, etc. São informações de baixo nível sobre o ambiente em que o robô atua, porém nenhum dispositivo é perfeito e estas informações podem ser imprecisas, contraditórias e ambíguas (NEHMZOW, 2003).

Existe uma grande variedade de sensores, que utilizam diferentes técnicas de medição e diferentes tipos de interface de controle (modo de acesso ao dispositivo). É de suma importância encontrar o sensor mais adequado a uma aplicação, com a técnica de aferição correta, e também características como, peso e tamanho, consumo de energia e preço adequados (BRÜANL, 2008).

Do ponto de vista de robôs móveis é mais interessante classificar sensores da seguinte maneira (BRÜANL, 2008) e (NEHMZOW, 2003):

- **Proprioceptivo** – monitoram o estado interno do robô, ex: *encoders*, monitor de nível de bateria, monitor de temperatura, potenciômetros, etc.;
- **Exteroceptivo** – monitoram e adquirem informação do ambiente em que o robô está imerso ex: lasers, sonares, sensores de toque, câmeras, bússola, GPS, etc.;
- **Passivos** – monitoram o ambiente sem afetá-lo e interferi-lo, ex: câmeras, microfones, sensores de toque, bússola, sensor de temperatura, entre outros;
- **Ativos** – interagem com o ambiente emitindo energia sobre o mesmo e então adquirindo informações, laser, sonares, *encoders*, GPS, entre outros.

Os sensores trabalham com uma faixa de leitura de uma propriedade física, e fornecem ao robô condições necessárias para se localizar no ambiente, reconhecer obstáculos, a velocidade que se desloca, o consumo de energia, entre outras. Caberá ao software de controle ou hardware lidar com imprecisões e erros nessas leituras. A Figura 2.7 ilustra alguns sensores de que dispõem o robô *PowerBot* da empresa *Adept Mobile Robots*.



Figura 2.7 Alguns sensores presentes no robô *PowerBot*.
Extraído e adaptado de *Adept Technology* (2012)

A maioria dos sensores permitem a coleta de informações básicas do ambiente, salvo alguns que realizam o pré-processamento dessas informações. Cabe a um sistema de controle, através de algoritmos especializados, desenvolver modelos elaborados do ambiente, junto com informações de alto nível que deem suporte a execução de tarefas pelo robô móvel.

As ações de um robô por sua vez são desempenhadas diretamente pelos atuadores, que entram em contato com o ambiente. Na robótica móvel é mais comum o uso de motores elétricos (motores de corrente contínua, motores de passo e servo motores), atuadores pneumáticos com válvulas e hidráulicos (BRÜANL, 2008) (NEHMZOW, 2003).

Motores de corrente contínua que movem rodas são bons para serem sensorizados, isto é permitem que seja feita a leitura do movimento executado determinando a distância percorrida por um robô. Os motores de passo desenvolvem movimentos precisos e sutis. Estes motores elétricos atuam com pouco ruído e geram energia suficiente para uma grande variedade de tarefas (NEHMZOW, 2003).

Os atuadores pneumáticos e hidráulicos podem ser usados em partes mecânicas como garras, braços, pernas, entre muitas outras possibilidades. A Figura 2.8 ilustra alguns atuadores de que dispõem o robô *PowerBot* da empresa *Adept Mobile Robots*.



Figura 2.8 Alguns atuadores disponíveis no robô *PowerBot*.
Extraído e adaptado de *Adept Technology* (2012)

2.2.2 Classificação de robôs móveis quanto a mobilidade

Os robôs móveis podem ser encontrados na literatura sob algumas classificações, conforme, De Pieri (2002), Siegwart (2004) e Brüanl (2008). Quanto a mobilidade os robôs podem ser classificados em três grandes grupos, aéreos, aquáticos e terrestres (DE PIERI 2002). A Figura 2.9 ilustra esta classificação:

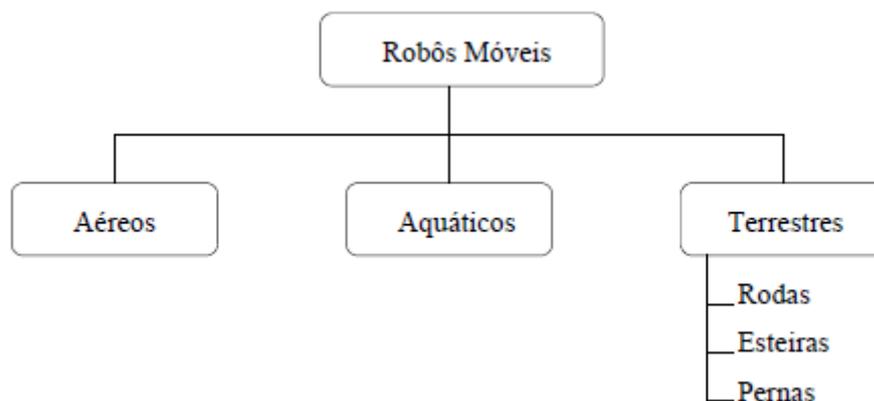


Figura 2.9 Classificação segundo a mobilidade.
Extraído de De Pieri (2002).

A definição dos tipos de robôs móveis segundo a classificação da Figura 2.9 pode ser, segundo autores da área:

Robôs com rodas – são robôs que utilizam uma ou mais rodas para se locomoverem, podendo ser de projeto e construção simples devido a questões de custo e engenharia. Isso pode ser provado pelo simples fato de não ser necessário maior preocupação com equilíbrio nesses robôs, já que passam todo tempo em contato com o chão. O foco no estudo e projeto de robôs com rodas passa para problemas como, tração, estabilidade, manobrabilidade e controle (SIEGWART, 2004). O robô móvel Seekur Jr. (ver Figura 2.10 a) da empresa *Adept MobilesRobots* é um exemplo de robô que se locomove por rodas e segundo o fabricante pode ser usado para pesquisa, segurança, exploração, entre outros.

Robôs com esteira – robôs com direção diferencial, que lhes permitem boa manobrabilidade em terrenos irregulares. Durante a tração possui maior contato e fricção com o solo. São robôs mais robustos dotados de sensores, geralmente na parte de cima, que lhes permite navegarem com certa segurança ao atuarem sobre terrenos irregulares em atividades como busca e resgate, no desarmamento de bombas, etc. (BRÜANL, 2008).

Alguns robôs com esteira da empresa *iRobot* (ver Figura 2.10 b), o *iRobot Warrior*, foram usados no episódio recente de terremoto e tsunami que arrasaram o nordeste do Japão e

que provocaram a explosão e o vazamento na usina nuclear de Fukushima. O *iRobot Warrior* é capaz de subir escadas e enfrentar terrenos íngremes e irregulares, possui uma garra que pode levantar até 52 quilos e bateria com autonomia de 5 horas, além de atuarem em ambientes hostis, sendo por isso usados na manutenção da usina de Fukushima.

Robôs com pernas – são robôs dotados de um conjunto de pernas para se locomoverem, biologicamente inspirados em animais e caracterizados por pontos de contato com o chão. Não são necessárias todas as pernas para manter o robô estável, sendo que por este fato se adaptam melhor em terrenos desestruturados e rudes, desdobrando-se nas habilidades de adaptabilidade, manobrabilidade e manipulação de objetos com destreza. Devido a locomoção por pernas são de complexidade maior, em vista de necessitarem partes articuladas que requerem certo nível de liberdade para prover movimentos variados (SIEGWART, 2004).

A empresa *Boston Dynamics* é um expoente na produção de robôs com locomoção por pernas, o robô *Cheetah* (ver Figura 2.10 c), por exemplo, é o mais rápido robô com pernas existente. O *Cheetah* possui quatro pernas flexíveis e foi inspirado no animal do qual carrega o nome, que é o mais rápido animal terrestre. A capacidade deste robô já lhe rendeu um recorde de velocidade em relação ao velocista recordista mundial, Usain Bolt.



Figura 2.10 Robôs móveis terrestres.

a) *SeeKur Jr.* Extraído de *Adept Mobilerobots* (2012). b) *iRobot Warrior*. Extraído de *Irobot* (2012) c) *Cheetah*. Extraído de *Boston Dynamics* (2012)

Robôs Aéreos – geralmente são aeronaves, principalmente aeromodelos ou LTAs (*Lighter-Than-Air*) (De Pieri, 2002), controladas remotamente, porém podem se tornar autênticos robôs móveis autônomos, as chamadas UAVs (*Unmanned Aerial Vehicles*). Para isso recebem sensores e atuadores (GPS, servo-motores, etc), um hardware controlador embarcado e um sistema de controle próprio. Segundo Brüanl (2008), aeronaves autônomas são mais difíceis na concepção de projeto e desenvolvimento do que outros tipos de robôs móveis, devido a exigência de alto grau de segurança, em virtude dos equipamentos que são de elevado custo e sobretudo para as pessoas, que podem ser expostas a situações de risco.



Figura 2.11 Veículo aéreo autônomo Tiriba.
Extraído de Inct-sec (2012).

Um exemplo de robô aéreo é o Tiriba (Figura 2.11) que é um VANT (Veículo Aéreo Não Tripulado) produzido pela empresa AGX tecnologia em parceria com o INCT-SEC (Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos). Essa aeronave elétrica utiliza um sistema de controle e navegação totalmente autônomo, conta com uma unidade de orientação automática baseada em sensores de pressão barométrica e inerciais e GPS (Sistema de Posicionamento Global), com a aplicação voltada na área de defesa civil, monitoramento ambiental e agrícola (INCT-SEC, 2012).

Robôs Aquáticos – são veículos que navegam acima da água (modelos de embarcações) ou abaixo dela (robôs-submarinos) controlados remotamente ou com autonomia própria por meio de um sistema de controle. São equipados com sensores (medidores de pressão) e atuadores (propulsores, balões de ar) que os permitem interagir e sobreviver em ambientes aquáticos.

Segundo Brüanl (2008), especialmente os AUVs (*Autonomous Underwater Vehicles*) requerem um nível de habilidade adicional em relação a outros tipos de robôs, devido a pressão no fundo do mar que aumenta conforme a profundidade e a necessidade de manter

conexões externas com os sensores e atuadores. Os AUVs são muito promissores com relação ao custo de missões teleoperadas ou tripuladas, porém um grande problema é manter comunicação com esses veículos, uma vez que, em imersão na água, métodos atuais de comunicação como WLAN ou Bluetooth não funcionam além de 50 cm, sendo o sonar o único método possível, mas com transmissão a taxas relativamente baixas.



Figura 2.12 Veículo autônomo aquático *Argo Float*.
Extraído de *Global Ocean Observing System* (2012).

O robô aquático *Argo Float* (Figura 2.12), por exemplo, é utilizado para medir e registrar a temperatura e o nível de salinidade das águas nos oceanos podendo submergir até aproximadamente 2000 metros, através de um sistema de flutuabilidade por motor de fluídos e bexiga externa. Este robô tem um formato cilíndrico e possui motor, bateria, sensores, sistema hidráulico, antena de transmissão via satélite, um sistema de controle embarcado entre outras peças. O sistema embarcado é responsável por transmitir os dados coletados para algum satélite, sendo que para isso o Argo tem de emergir a superfície, depois de determinado tempo sob a água. Esse robô faz parte do projeto *Integrated Global Observation Strategy*, que espalhou pelos oceanos do globo terrestre mais de 3000 unidades de *Argos* para formar um sistema global de coleta de dados dos oceanos. O projeto já tem uma enorme contribuição em pesquisas e estudos, destacando-se as pesquisas em aquecimento global.

2.3 Sistemas de controle em robôs

Um sistema de controle de um robô é um software que comanda suas funcionalidades. Segundo Heinen (2002), existem duas partes interagindo com um sistema de controle: o sistema controlado (robô), sistema do qual se deseja controlar, e o ambiente, que não pode ser totalmente controlado diretamente pelo sistema de controle, mas interfere no seu funcionamento. A Figura 2.13 ilustra um modelo de sistema de controle.

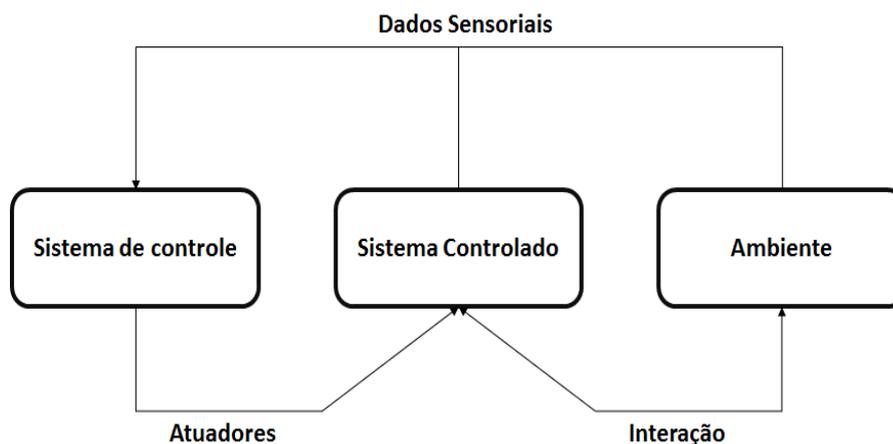


Figura 2.13 Sistema de controle de robôs móveis.
Extraído de (HEINEN, 2002).

Um sistema de controle reage de acordo com as mudanças no ambiente, no sistema controlado ou da interação de ambos, sendo necessário o monitoramento dessas mudanças para agir quando necessário. Para auxiliar o sistema de controle podem ser utilizados sensores no robô ou no ambiente (HEINEN, 2002).

Em um ambiente em que um robô deve simplesmente arrastar um objeto de um ponto a outro, qualquer situação ou condição que leve o robô a ser impedido ou a ignorar o comportamento de arrastar a caixa, deve induzir o sistema de controle a agir e conduzir novamente o robô ao comportamento desejado e conseqüentemente ao estado esperado para todo o sistema.

Os sistemas são projetados de acordo com alguma arquitetura. Os tipos de arquiteturas representam modelos de projeto que podem ser de complexidade variável, como um simples programa, até um sistema operacional embarcado.

2.4 Arquitetura de controle

Uma arquitetura de software é composta por um conjunto de ações que são executadas conforme o ciclo percepção-ação, desta forma existe a interpretação de informações do ambiente fornecidas pelos sensores e sua relação pelos atuadores (PEREZ, 2006).

Um robô que é controlado por um sistema computacional, tem o conhecimento dos dispositivos de hardware que dispõem para perceber e agir sobre o ambiente, nele algumas estruturas de software são responsáveis pela interpretação de como se comportar frente a este ambiente (ALVAREZ et al., 2001).

A arquitetura de controle determinará a forma de comportamento do robô. Dentre as arquiteturas destacam-se como principais: Reativa, Deliberativa e Híbrida.

2.4.1 Arquitetura de controle reativa

Nessa arquitetura o robô é controlado por um sistema baseado em ação-reação, onde por meio de sensores o robô capta dados que traduzidos em informações designam uma ou mais ações por meio de atuadores. O sistema de controle fica a cargo de atribuir a cada informação uma ação conveniente.

Na reatividade ocorrem mapeamentos simples ligando estímulos sensoriais a respostas (ações) (RIBEIRO et al., 2001). Como exemplo cita-se um robô móvel que faz uso de sensores de toque que ao se chocar em uma parede recebe o sinal do sensor e induz uma reação contrária, como ir na direção oposta ao choque. A Figura 2.14 ilustra a arquitetura de controle reativa.

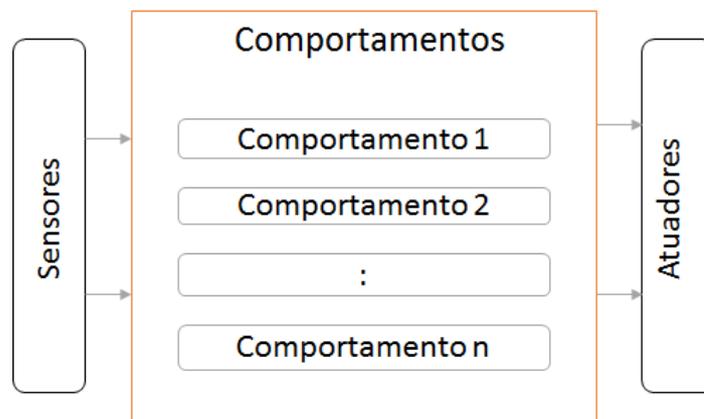


Figura 2.14 Arquitetura de controle reativa.
Adaptado de De Pieri (2002)

Não é necessário manter um histórico de estados do robô e ambiente, nem mapeamento do ambiente no qual o robô está inserido, mas uma contínua captação do ambiente em tempo real suficiente para responder com ações do robô que lhe permitam a execução de uma tarefa.

2.4.2 Arquitetura de controle deliberativa

Na arquitetura deliberativa o robô faz uma análise de longo prazo onde ações serão alicerçadas para chegar a um objetivo. É realizado um planejamento sobre as ações.

A deliberação implica um estágio intermediário de planejamento, baseado em um modelo do mundo real onde o robô interage (RIBEIRO et al., 2001). Para isso é fundamental manter um histórico do ambiente, ações já tomadas, e dependendo da política de decisões até estados anteriores em que o robô se encontrava. Como exemplos podem ser citados os algoritmos de mapeamento, localização e navegação, fundamentados na leitura do ambiente.

Dessa forma esse planejamento pode acontecer por meio de mapas do ambiente (pré-processados ou construídos em tempo real), bases de dados (estados anteriores do ambiente ou robô, ações tomadas) internas ou externas em relação ao robô, pré-processamento de sensores por rotinas e algoritmos de alto nível, suporte a processamento de informações por núcleos externos, entre outras formas. A

Figura 2.15 ilustra um modelo de arquitetura deliberativa simplificado.

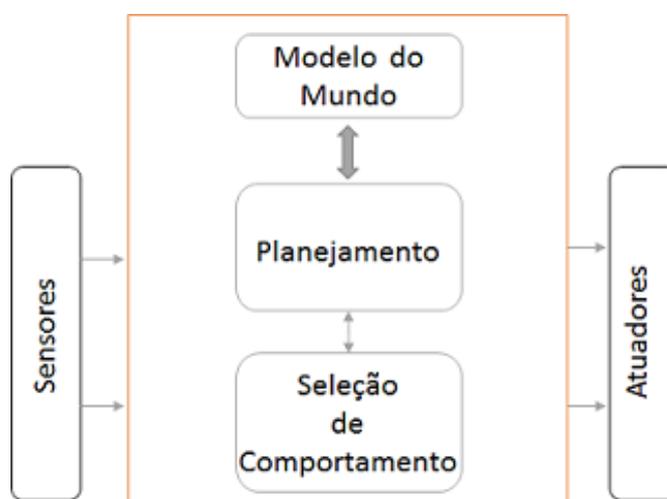


Figura 2.15 Arquitetura de controle deliberativa.

Na arquitetura deliberativa subentende-se a adição de inteligência ao robô que pode aumentar conforme a necessidade e complexidade exigida pelas tarefas e o ambiente.

2.4.3 Arquitetura de controle híbrida

A arquitetura híbrida integra às arquiteturas reativa e deliberativa. Sendo que a arquitetura reativa fica responsável pelas respostas de curto prazo que são imediatas, e a arquitetura deliberativa produz um planejamento de longo prazo nas ações de um robô.

Nesse tipo de arquitetura se realiza um planejamento sobre a definição e seleção de comportamentos reativos individuais (RIBEIRO et al., 2001).

Um terceiro módulo geralmente é necessário para gerenciar a comunicação entre o módulo reativo e o planejador. Assim tem-se uma arquitetura chamada de três camadas onde o comportamento reativo individual é controlado por meio de um planejamento (Mataric, 2001) apud (PEREZ, 2006). O raciocínio é realizado sobre modelos internos do mundo.

Nessa arquitetura a união intrínseca entre as arquiteturas anteriores (reativa e deliberativa) ocorre por situações em que a resposta ao estímulo deve ser imediata, sacrificando-se o planejamento em um primeiro momento, mas mantendo registro de informações para planejar futuras ações, ou em condições que exijam comportamentos simples pelo robô. Um modelo da arquitetura híbrida é ilustrado na Figura 2.16.

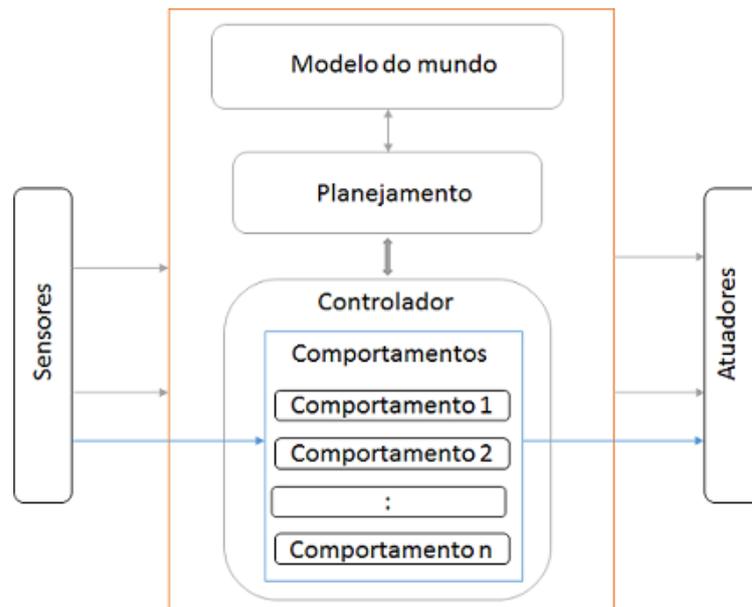


Figura 2.16 Arquitetura de controle híbrida.

Uma comparação simples pode relacionar esta arquitetura ao funcionamento de organismos mais desenvolvidos. Pode-se observar, por exemplo, na natureza animais que agem por instinto, a exemplo, de situações emergenciais como reagir a situações de perigo, mas aprendem com a situação para se posicionar em relação a acontecimentos futuros.

3 COMPARTILHAMENTO DE TAREFAS EM ROBÓTICA MÓVEL

Em sistemas onde a cooperação entre robôs deve existir o compartilhamento de tarefas torna-se um aspecto fundamental. Neste capítulo serão definidos os conceitos de tarefas em robótica móvel e das técnicas que permitam seu compartilhamento, tais como, decomposição e alocação, bem como a taxonomia MRTA (*Multi Robot Task Allocation*) para classificação de problemas de alocação de tarefas. O capítulo também aborda o conceito de acordos em sistemas com falhas bizantinas.

3.1 Sistemas distribuídos

Na realização de tarefas por um grupo de robôs móveis deve ser considerada uma arquitetura distribuída, para que este grupo possa trabalhar conjuntamente. A distribuição está inerente à necessidade de o robô trabalhar em conjunto com outros robôs na solução de um problema. Segundo García et al. (2013), neste caso os robôs devem cooperar de forma explícita e o sistema deve prover condições para que esses robôs apresentem comportamentos sociais.

Segundo Brumitt (1998) através da computação distribuída uma equipe cooperativa de robôs é mais capaz do que a simples soma de suas partes, esta equipe apresentará maior confiabilidade e a capacidade para completar tarefas distribuídas fisicamente.

Lidar com sistemas de múltiplos robôs significa explorar problemas complexos em ambientes dinâmicos e imprevisíveis utilizando recursos escassos (ZLOT, 2006). A Computação distribuída possui diversos estudos e técnicas que fundamentam os sistemas de múltiplos robôs.

3.2 Tarefas em robótica móvel

Uma forma de descrever uma tarefa em robótica móvel e defini-la como uma atividade a ser iniciada com requisitos mínimos e um objetivo específico, que pode ser realizada por uma ou mais entidades. Os requisitos mínimos de uma tarefa podem ser descritos como habilidades necessárias a sua execução, tal como, a detenção de uma ou mais ferramentas específicas, a capacidade de atender a um limite de tempo, um estado específico da entidade executora, etc.

Algumas habilidades básicas de robôs móveis são mapeamento, navegação e localização (SIEGWART, 2004). Estas capacidades são largamente estudadas em robótica móvel sobre diferentes abordagens e técnicas. A navegação condiz com a capacidade de um robô de se locomover em um ambiente seja ele fechado (indoor – salas, prédios, etc.) ou aberto (outdoor – campos, oceano, cidade, etc.). A localização é a descoberta pelo robô de sua posição no ambiente, seja local (parte do ambiente visível pelo robô) ou global (ambiente total em que o robô está imerso). Já o mapeamento refere-se a construção de mapas do ambiente e sua relação com o que o robô percebe a sua volta, um mapa pode ser conhecido a priori ou construído pelo robô a medida que navega e visualiza o ambiente. Para uma abordagem mais completa, essas três habilidades são combinadas.

Além das três habilidades básicas citadas pode ser adicionado a robôs uma grande variação de capacidades para o cumprimento de atividades específicas. Robôs podem ter a capacidade de reconhecer objetos, manipular objetos (carregar, arrastar, agarrar, arremessar, absorver, etc.), subir escadas, voar, aquaplanar, mergulhar, perfurar materiais. Enfim, a uma enorme variação devido aos componentes computacionais, as ferramentas (sensores e atuadores), o sistema de controle, e os propósitos e problemas que o robô deve resolver.

Dessa forma, como mencionado por Nehmzow (2003), há uma interdependência entre robô, tarefa e ambiente. A tarefa de pegar um objeto situado a certa distância necessitará de um robô as capacidades de locomoção, reconhecimento e manipulação do objeto. A tarefa poderá ser dividida na sequência, localizar o objeto, se locomover até o objeto e pegá-lo. Se um robô não possuir essas três habilidades ele poderá solicitar auxílio para outros robôs que contemplam esses requisitos, isso pode ser entendido como um caso em que as capacidades estão distribuídas entre mais de um robô, mas nenhum possui as três ao mesmo tempo, sendo necessário cooperarem de alguma forma para a realização da tarefa.

As três habilidades para realizar a tarefa de pegar o objeto seriam suficientes para um robô em um ambiente estático, no entanto, se no percurso há movimentação de pessoas ou objetos ou até mesmo outros robôs, é necessário a habilidade de desvio de obstáculos. Assim pode-se reconhecer uma tarefa básica como uma única ação (locomoção de um ponto a outro), mas que pode exigir mais de uma capacidade.

Existem capacidades que permitem a realização de uma enormidade de tarefas, assim como existem tarefas que só são realizáveis por mais de uma ação, como, por exemplo, pegar um objeto a certa distância ou arrastar uma caixa por uma determinada distância. Para o problema da caixa seria necessário localizar e reconhecer a caixa, ir até a caixa, e arrastar a caixa por uma distância determinada. A simples tarefa de reconhecer e localizar objetos em um ambiente compreende mais de uma ação. A tarefa o robô e o ambiente são determinantes para solução de um problema.

Robôs geralmente são construídos com propósitos específicos, sendo praticamente impossível construir um robô de propósito geral. Robôs móveis vêm sendo construídos para incluírem e superarem níveis de habilidades que os predisponham a realização de mais tarefas, apesar disso as tecnologias disponíveis e os custos ainda remetem os robôs a tarefas específicas e bem definidas. Como citado neste trabalho algumas tarefas são mais complexas e requerem múltiplas habilidades e ações, sendo impossível sua execução por um único robô.

A utilização de SMR (Sistemas de Múltiplos Robôs) incide sobre tarefas complexas que necessitam o uso de mais de um robô. Nesses sistemas o principal desafio é gerenciar o uso dos diversos recursos disponíveis através do grupo de robôs, que devem se comunicar e coordenar suas ações para a execução de uma tarefa (CAO et al., 2010).

Em SMR uma tarefa pode ser definida como um objetivo parcial que pode ser alcançado independentemente do objetivo global de todo o sistema. Uma tarefa pode ser discreta (locomoção de um ponto a outro) ou contínua (monitoramento de um objeto), e variar em escala de tempo, complexidade e especificidade, entre outras formas (GERKEY; MATARIC, 2004).

Um exemplo do uso de múltiplos robôs é o projeto científico *Swarmanoid* fundado pela Comissão Européia. O principal objetivo do projeto foi de criar um sistema de múltiplos robôs heterogêneos, autônomos e interligados dinamicamente para desenvolverem um comportamento de enxame (*swarm*) (SWARMANOID.ORG, 2012). *Swarm* é um problema muito estudado em IA (inteligência Artificial) e geralmente é estendido para a robótica móvel.

Segundo Swarmanoid.org (2012), o projeto fez uso de três tipos de robôs: *eye-bots* (ver Figura 3.1 c), *hand-bots* (ver Figura 3.1 b), e *foot-bots* (ver Figura 3.1 a) em problemas reais. O projeto ganhou o prêmio de melhor vídeo na competição, “AAAI 2011 video competition”. Os três robôs possuem capacidades específicas que combinadas podem solucionar certos tipos de problemas.

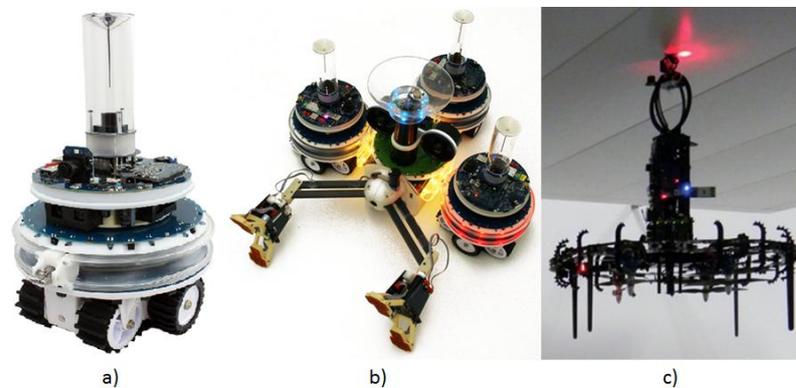


Figura 3.1 Robôs do projeto *Swarmanoid*.
Extraído e adaptado de Swarmanoid.org (2012).

Tarefas complexas podem ser decompostas em tarefas menores e independentes sempre que possível, desta forma elas podem ser distribuídas para um grupo de robôs visando melhor aproveitamento dos recursos disponíveis.

3.2.1 Utilidade na execução de tarefas

A utilidade na realização de uma tarefa qualquer é um valor que quantifica a predisposição para sua execução. A utilidade pode ser calculada através de modelos de custos, grandezas que mensuram propriedades na execução de uma tarefa, tais como, o *custo* na utilização de recursos, o *fitness* (aptidão para a execução da tarefa), uma *recompensa* para o desempenho de um robô, ou *prioridade* de execução da tarefa (MOSTEO; MONTANO, 2010).

O valor de utilidade pode surgir da combinação desses modelos de custos, comumente listados na forma de (MOSTEO; MONTANO, 2010):

$$Utilidade = Recompensa - Custo$$

$$Utilidade = Fitness - Custo$$

Em sistemas de múltiplos robôs a medição de utilidade pode ser usada para otimizar a realização de tarefas e conseqüentemente o desempenho do sistema. Conceitos comuns de utilidade propõem um valor único não negativo atribuído a um robô na execução de uma tarefa específica (GERKEY; MATARIC, 2004).

Gerkey e Mataric (2004) propõem uma inter-relação entre qualidade na execução da tarefa com os equipamentos disponíveis (ex: precisão na manipulação de um objeto), e o custo no uso de recursos (ex: consumo de energia). Assim, é calculado o valor da utilidade na execução de uma tarefa por um robô supondo que este saiba sua aptidão.

Formalmente essa inter-relação é definida, como: dado um robô R capaz de executar uma tarefa T , com uma qualidade esperada Q_{RT} e um custo C_{RT} , tem-se uma utilidade U_{RT} como um valor resultante não negativo (GERKEY; MATARIC, 2004):

Equação 1 Utilidade na Execução de uma Tarefa.

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

Sob o ponto de vista da otimização a mensuração de utilidade tem por objetivo o alcance do melhor desempenho para um sistema fazendo a correta relação entre as capacidades e habilidades dos robôs, as condições e estados do robô e ambiente, os requisitos das tarefas e regras de alocação de tarefas impostas para o sistema.

3.2.2 Tratamento de tarefas

Existe uma diversidade de tarefas que podem ser executadas por robôs. Essa diversidade leva a uma variedade de formas, entre os requisitos necessários e execução das tarefas.

Segundo Dias et al. (2005), a forma mais comum entre as abordagens de atribuição de tarefas para robôs é delegar uma lista com tarefas em sua forma primitiva, ou seja, tarefas simples. Porém em sistemas de múltiplos robôs é mais natural o uso do termo missão, que é uma descrição de alto nível para uma tarefa complexa, que pode ser decomposta em sub-tarefas.

O tratamento de tarefas é realizado na forma de um planejamento sobre a questão do que deve ser feito. Geralmente isso envolve a decomposição de tarefas em partes. Para um melhor entendimento desta decomposição as seguintes terminologias para representação de tarefas são apresentadas (Zlot, 2006):

- **Decomposição e decomponibilidade** – uma tarefa t pode ser decomposta se puder ser representada por um conjunto de sub-tarefas σ_t , que satisfazem alguma combinação (relação) específica (ρ_t) de sub-tarefas. O conjunto de rela-

ções ρ pode definir restrições ou regras entre tarefas. O par (σ_t, ρ_t) é chamado decomposição de t ;

- **Múltiplas decomponibilidades** – uma tarefa t que apresenta mais de uma forma de ser decomposta;
- **Tarefas Simples** – pode ser uma tarefa atômica ou uma tarefa simples decomponível;
 - **Tarefa atômica** – é uma tarefa que não pode ser decomposta, geralmente referida por uma única ação;
 - **Tarefa simples decomponível** – uma tarefa que pode ser dividida em passos e executada por uma única entidade;
- **Tarefa composta** – é uma tarefa que pode ser decomposta em um conjunto formado por tarefas simples ou compostas, sendo que deve existir ao menos uma tarefa simples decomponível;
- **Tarefas complexas** – uma tarefa que possui múltiplas decomponibilidades, em que ao menos existe um conjunto de tarefas alocadas entre várias entidades. Cada tarefa no conjunto pode ser simples, composta ou complexa.

A Figura 3.2 ilustra a representação dos tipos de tarefas (KORSAH; DIAS; STENTZ, 2012).

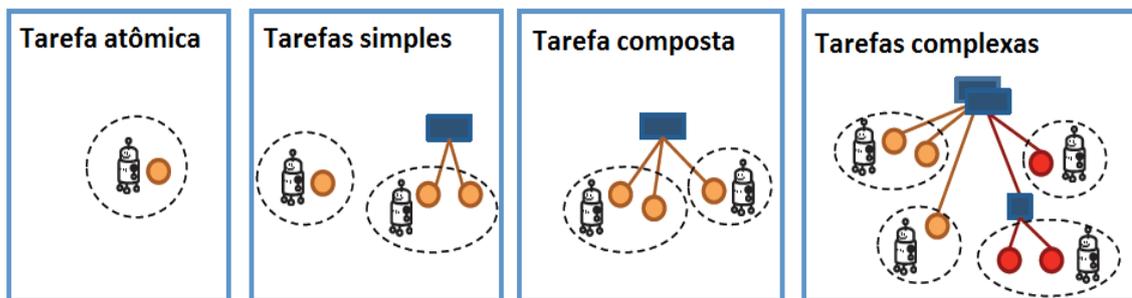


Figura 3.2 Representação de tarefas.
Adaptado de Korsah, Dias e Stentz (2012).

Na Figura 3.2 os círculos pontilhados representam possíveis alocações de tarefas para robôs, os círculos preenchidos são tarefas atômicas, os retângulos são tarefas decompostas que formam uma estrutura de árvore. No primeiro quadro é ilustrada uma tarefa atômica sendo atribuída a um robô; o segundo quadro ilustra uma tarefa atômica e uma tarefa simples decomponível cada uma sendo atribuída a um robô; o terceiro quadro ilustra uma tarefa composta formada por uma tarefa simples decomponível e sendo atribuída a dois robôs; o quarto quadro ilustra uma tarefa complexa alocada a múltiplos robôs e com várias possíveis decomposições (KORSAH; DIAS; STENTZ, 2012).

Segundo Cao et al.(2010), a diferenciação entre os tipos de tarefas é necessária, pois no desenvolvimento de um algoritmo de alocação, tarefas complexas requerem uma modelagem de abstração em vários níveis, isso ocorre pois uma tarefa atômica não pode ser delegada

a múltiplos robôs e uma tarefa composta formada por uma única decomposição é predeterminada e não alterável, sem a flexibilidade requerida por uma tarefa complexa.

Um algoritmo de alocação para tarefas complexas deve lidar com atribuições dinâmicas para produzir resultados de execução melhores, nele a decomposição (planejamento) de tarefas deve ser flexível, pois alterações podem ser necessárias (ZLOT, 2006). Isso ocorre devido a dinamicidade no ambiente dessas tarefas, a quantidade de caminhos para soluções, as capacidades dos robôs e a quantidade de atribuições entre robôs e tarefas. Soluções melhores não são predeterminadas, mas ocorrem dinamicamente em resposta a demanda do sistema.

3.2.3 Modelo e ambiente da tarefa

A descrição de uma tarefa no domínio de uma aplicação pode ser dada como um conjunto de variáveis (duração, início e término, habilidades necessárias), quando trata-se de tarefas simples a tendência é uma execução sequencial, em contrapartida tarefas complexas requerem mais flexibilidade na ordem de execução.

A realização de uma tarefa está fortemente ligada a capacidade de um robô e as condições do ambiente que permeiam a tarefa.

O ambiente de uma tarefa é um conjunto que inclui tudo que envolve e interage com um sistema de controle. Segundo Russel e Norvig (2004), o ambiente, os sensores e atuadores de um agente (robô) e medidas de desempenho que qualifiquem a execução de uma tarefa por um agente podem ser agrupados no ambiente da tarefa.

Ainda segundo Russel e Norvig (2004), algumas propriedades desses ambientes são úteis para compreensão da tarefa, tais como:

- **Completamente observável vs parcialmente observável** – se todos os aspectos relevantes para a execução de uma ação são captados pelos sensores em cada instante de tempo o ambiente é completamente observável, porém se existe imprecisões de leitura ou estados do ambiente inalcançáveis o ambiente torna-se parcialmente observável.
- **Determinístico vs estocástico** – se um estado futuro é determinável pelo estado e ação atual do agente, o ambiente é determinístico, do contrário ele passa a ser estocástico, ou seja, sem qualquer previsibilidade diante da dinâmica de um ambiente.
- **Episódico vs sequencial** – no ambiente episódico ações são atômicas e independem de ações anteriores. No ambiente sequencial ações atuais influenciam nas posteriores afetando o resultado final.

- **Discreto vs contínuo** – ações no ambiente discreto são ponderadas em intervalos de tempo com uma sequência de fim determinado, já no ambiente contínuo as ações variam em intervalos distintos e contínuos.
- **Único agente vs multi-agente** – a presença de apenas um agente ou duas ou mais entidades convivem no ambiente e agem de forma cooperativa ou competitiva.

Um projeto de sistema de controle deve levar em consideração o ambiente e suas propriedades. Sistemas de múltiplos robôs geralmente envolvem ambientes parcialmente observáveis, estocásticos e sequenciais com níveis exponenciais de complexidade.

3.3 Métodos de alocação de tarefas entre múltiplos robôs

Segundo Gerkey e Mataric (2004), um aspecto importante em sistemas de múltiplos robôs é a coordenação, em especial Problemas de Alocação de Tarefas entre Múltiplos Robôs (MRTA - do inglês *Multi-Robot Task Allocation*).

Sistemas de múltiplos robôs geralmente envolvem soluções para tarefas de caráter complexo e nível exponencial caracterizado como *NP-difícil* (ZLOT, 2006). São sistemas que envolvem uma multiplicidade de formas de atribuição entre robôs e tarefas. O problema de alocação de tarefas refere-se a determinação de qual robô deve executar qual tarefa para alcançar um objetivo global (GERKEY; MATARIC, 2004).

A alocação é necessária para tarefas que exijam um grupo de robôs detentor de recursos limitados e dispersos entre seus membros. A alocação pode ser descrita como um método que melhor faça uso desses recursos alcançando a execução da tarefa, levando em consideração a interação dos membros do grupo e as alterações no ambiente.

Gerkey e Mataric (2004) propõem uma taxonomia para classificar problemas comuns de MRTA facilitando pesquisas e trabalhos na área de Sistemas de Multi-Robôs. As seguintes definições são dadas:

- **Robôs de única tarefa (*ST - Single-task robots*) vs Robôs de múltiplas tarefas (*MT - Multi-task robots*):**
 - **ST:** cada robô pode executar apenas uma tarefa em um intervalo de tempo.
 - **MT:** cada robô pode executar mais de uma tarefa no mesmo intervalo de tempo.

- **Tarefas de único robô (*SR - Single-Robot tasks*) vs Tarefas de múltiplos robôs (*MR - Multi-Robot tasks*):**
 - **SR:** cada tarefa requer apenas um robô para ser completada
 - **MR:** cada tarefa requer mais de um robô para ser completada.
- **Atribuição instantânea (*IA - Instantaneous Assignment*) vs Atribuição de tempo estendido (*TA Time-Extended Assignment*):**
 - **IA:** a informação disponível sobre os robôs, as tarefas, e o ambiente somente permite a alocação instantânea de tarefas para cada robô, não há planejamento para futuras alocações.
 - **TA:** existe mais disponibilidade de informação, como conhecimento de todas as tarefas que precisam ser atribuídas, ou um modelo de como às tarefas iram chegar ao longo do tempo. Há a possibilidade de um planejamento sobre alocações de tarefas.

A combinação dos itens acima descritos forma uma taxonomia onde os problemas de MRTA podem ser incluídos permitindo a classificação de seus domínios, facilitando assim, o desenvolvimento de uma abordagem de solução, seja a aplicação de algoritmos ou compreensão do nível de complexidade (GERKEY; MATARIC, 2004). A Tabela 1 ilustra as combinações que podem representar campos de problemas.

Tabela 1 Taxonomia de problemas de MRTA.

	ST-SR	ST-MR	MT-SR	MT-MR
IA	ST-SR-IA	ST-MR-IA	MT-SR-IA	MT-MR-IA
TA	ST-SR-TA	ST-MR-TA	MT-SR-TA	MT-MR-TA

Para cada campo da taxonomia podem ser descritos problemas comuns em MRTA.

Dado um problema onde a atribuição é de um para um, seja uma condição entre robôs e tarefas ($R^1 \times T^1$). As tarefas são procuradas por robôs de forma preferencial e cada robô possui habilidades específicas. Para cada robô postulante, aplica-se uma medida de nível de habilidade na execução de cada tarefa (GERKEY; MATARIC, 2004).

Segundo Gerkey e Mataric (2004), o problema anterior pode ser descrito em termos da tripla taxonômica ST-SR-IA (*Single-task robots-Single-Robot tasks- Instantaneous Assignment*). O objetivo de um sistema para essa tripla é atribuir para cada robô uma tarefa levando

em consideração as prioridades de cada tarefa e as habilidades de cada robô de modo que se alcance um desempenho geral otimizado. Este problema simples pode ser solucionado por um programa linear, que a partir do nível de habilidade resolva a atribuição de robôs e tarefas por meio de busca.

3.4 Acordos em sistemas com falhas

Acordos entre processos em sistemas distribuídos é um requisito fundamental em um grande número de aplicações. Existem formas de coordenação entre processos que requerem a troca de informações para uma negociação em busca de entendimento em comum ou acordo (KSHEMKALYANI; SINGHAL, 2008).

A tolerância a falhas indica que um sistema é capaz de prover seus serviços mesmo na presença de falhas. As falhas são ignoradas enquanto o sistema funciona normalmente (TANENBAUM; STEEN, 2008).

Um processo de acordo é executado para se chegar a uma concordância entre um grupo de processos sobre um valor específico ou um conjunto de valores, que deve ser equivalente para todos os processos confiáveis. Essa concordância deve anteceder a realização de ações em um sistema (KSHEMKALYANI; SINGHAL, 2008).

A utilização de acordos entre processos em sistemas distribuídos pode aumentar a tolerância a falhas, ao introduzir-se a capacidade de identificação de componentes faltosos e dos componentes capazes de dar continuidade ao sistema.

Em contraste, geralmente com sistemas presentes em uma única máquina onde uma falha pode levar a uma parada total, sistemas distribuídos podem evitar danos maiores por falhas isolando e ignorando um componente faltoso (TANENBAUM; STEEN, 2008).

Para uma melhor compreensão das falhas que tem importância em determinados sistemas são necessários modelos de falha. Existem classes de modelos de falha massivamente estudados, que especificam maneiras em que componentes de um sistema podem falhar (KSHEMKALYANI; SINGHAL, 2008).

Alguns modelos de falhas podem ser (TANENBAUM; STEEN, 2008):

- **Falha por queda** – parada abrupta e prematura de um componente em um sistema.

- **Falha por omissão** – omissão no envio ou recebimento de mensagens independente de suas funcionalidades.
- **Falha de temporização** – um desvio do intervalo de tempo real especificado para prover um serviço corretamente.
- **Falha de resposta** – o nó envia uma resposta com valor incorreto.
- **Falha bizantina** – ocorre de maneira arbitrária em um componente que produz respostas inadequadas, muitas vezes de maneira maliciosa.

A melhor forma para tornar um sistema tolerante a falhas é utilizando redundância, que pode ser (TANENBAUM; STEEN, 2008):

- **Redundância de informação:** são adicionados bits extras com a intenção de permitir a recuperação de bits de dados deteriorados por meio de alguma técnica, a exemplo do código de *Hamming*.
- **Redundância de tempo:** se necessário, uma ação realizada é executada novamente.
- **Redundância física:** pode ser tanto de hardware como de software, sendo que são adicionados equipamentos ou processos extras para que um sistema tolere a falha em alguns componentes.

Um objetivo ao se utilizar modelos de erros é tornar um sistema resiliente. A resiliência é descrita como a propriedade em que um objeto abstrato ou físico retorna a forma original ou posição após sofrer alterações ou perturbações. Representa a capacidade de regresso a forma original de equilíbrio após recuperação e superação contra adversidades.

Segundo Tanenbaum e Steen (2008) uma boa forma de tornar um processo resiliente é o uso de processos redundantes (redundância física). É importante ter-se ciência de quanta redundância é necessário, para tonar um sistema tolerante a falhas o suficiente para ser considerado K -tolerante.

Um sistema é K -tolerante a falhas se este puder sobreviver a falhas em K componentes cumprindo suas especificações, como entendido para alguns casos (TANENBAUM; STEEN, 2008):

- **Falhas silenciosas:** K componentes param sem propagar mensagens com erros para outros componentes, nesse caso é necessário $K+1$ de redundância para manter o sistema tolerante a falhas.
- **Falhas bizantinas:** é necessário $2K + 1$ de redundância de componentes que funcionam corretamente, para manter o sistema tolerante a falhas de um total de $3K + 1$ componentes.

Um aspecto chave para contornar falhas de processos é organizar vários processos idênticos em um grupo no qual todos os processos recebam mensagens enviadas por qualquer membro do grupo (TANENBAUM; STEEN, 2008).

Por diversos fatores relativos a sistemas distribuídos o alcance de um acordo só é possível levando-se em conta premissas que envolvem o sistema e suas características, a fim de garantir uma terminação ao algoritmo de acordo (FISCHER, 1985).

Premissas diferentes sobre o sistema requerem soluções diferentes: se o atraso de comunicação é limitado ou não; se a entrega de mensagens é ordenada ou não; se a transmissão de mensagens é feita em *unicast* ou *multicast* (TANENBAUM; STEEN, 2008).

Se em um sistema distribuído a comunicação não é confiável, mesmo que seus processos não falhem, a concordância entre estes é impossível, pois não há limite para chegar a uma decisão comum devido a possível perda de mensagens.

Algumas premissas em sistemas distribuídos são elencadas:

- **Comunicação síncrona/assíncrona** - em uma comunicação assíncrona o alcance de um acordo é impossível, pois a garantia de entrega de mensagens entre processos não é alcançada tornando um processo de negociação interminável. Na comunicação síncrona, o não recebimento de uma mensagem pode ser ignorado pelo uso de uma mensagem padrão pelo processo destinatário, permitindo que a negociação continue e eventualmente chegue ao fim (KSHMKALYANI; SINGHAL, 2008).
- **Rede de comunicação** - todos os processos em um sistema estão aptos a participarem de uma rede e trocarem mensagens (KSHMKALYANI; SINGHAL, 2008).
- **Canal confiável** - considera-se que o canal de comunicação é confiável, ou seja, somente os processos podem falhar. Neste caso, um acordo pode ser impossível ou solucionável de maneira complexa (KSHMKALYANI; SINGHAL, 2008).
- **Identificação de emissor** - o processo que recebe uma mensagem pode identificar o emissor. Essa premissa é importante em falhas bizantinas, pois mes-

mo em virtude de um pacote de dados enviado com erros por um processo pode-se descobrir a identidade do emissor (KSHEMKALYANI; SINGHAL, 2008).

- **Mensagem autenticada vs não autenticada** - em mensagens não autenticadas ou sem assinatura, um processo faltoso pode alterar a identificação do real emissor de uma mensagem, antes de reenviá-lá a outros processos. Em mensagens autenticadas por técnicas como assinatura digital, uma mensagem com identificação alterada e reenviada por um processo malicioso pode ser detectada por outros processos (KSHEMKALYANI; SINGHAL, 2008).

3.4.1 Acordo bizantino

Sistemas com vários nós processadores, dentre esses os sistemas multi-robôs, podem falhar de maneira arbitrária, sendo que, uma falha é vista como um comportamento incomum ao especificado para um sistema, e é oriunda de um ou mais erros no estado do sistema (KSHEMKALYANI; SINGHAL, 2008).

As falhas podem ser oriundas de defeitos de software ou hardware, de erros de projeto, ou ainda por motivos alheios (ex. queima de componente eletrônico, incêndio, invasões etc.) (LUIZ, 2009).

Por sua vez uma falha é caracterizada como a causa que originou um erro em um sistema, que pode ser de razão física (hardware) ou algorítmica (fluxo de execução de um software) (TANENBAUM; STEEN, 2008).

A falha arbitrária (ou bizantina) é o tipo de falha mais severa e o menos restritivo, pois encapsula todos os modelos de falhas conhecidos (LUIZ, 2009).

Portanto, segundo Honorato (2009), a falha bizantina engloba qualquer comportamento diferente do considerado correto para um sistema por parte de um componente, e técnicas que toleram esse tipo de falha não assumem modelos sobre comportamentos do componente, tolerando qualquer comportamento fora do correto.

Segundo Luiz (2009), cresce o número de pesquisas para desenvolvimento de soluções práticas de suporte a aplicações distribuídas tolerantes a faltas bizantinas nos últimos anos como desafio a criação de protocolos que forneçam robustez, confiabilidade e segurança em sistemas distribuídos.

O clássico problema de falha bizantina foi inspirado pelas longas batalhas pelo império de Bizantino na idade média. Considera-se que quatro exércitos dispostos para batalha ao redor da cidade de Bizâncio, cada qual controlado por um general devem coordenar um ata-

que. O ataque só terá sucesso se for simultâneo pelas quatro armadas, dessa forma o horário exato do ataque tem de ser negociado por meio de mensagens (KSHEMKALYANI; SINGHAL, 2008). A Figura 3.3 ilustra o clássico problema de falha bizantina, na qual, quatro generais apresentam comportamento arbitrário e enviam mensagens confusas.

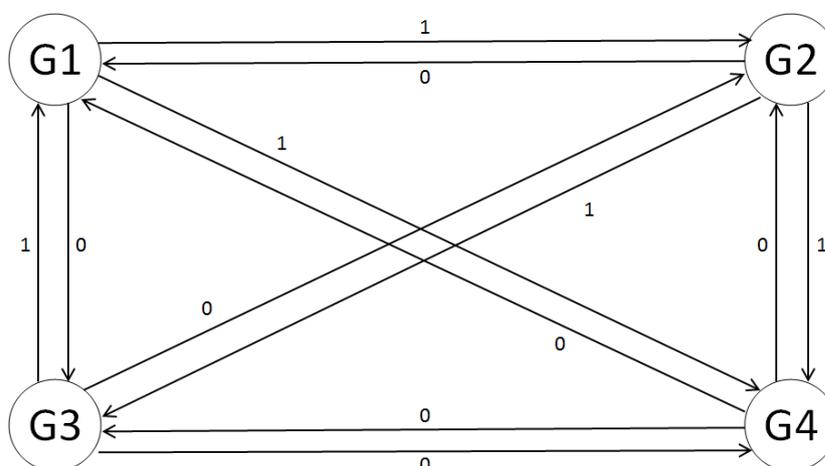


Figura 3.3 Clássico problema bizantino.
Extraído de Kshemkalyani; Singhal, (2008).

Três modelos de falha para o problema são apresentados por Kshemkalyani; Singhal, (2008):

- um modelo assíncrono indica um tempo ilimitado para a transmissão de mensagens;
- um modelo onde ocorre perda de mensagens é descrito como um mensageiro sendo capturado;
- um modelo bizantino é descrito como um general traidor, que tenta corromper o processo de acordo enviando informações erradas.

Considerando o modelo assíncrono no simples cenário do ataque a Bizâncio, se um general enviar um mensageiro a outro esperando uma resposta de confirmação da chegada da mensagem, uma vez que o tempo é indeterminado, a troca de mensagens não terá fim e consequentemente não haverá acordo.

Para os demais modelos um acordo será possível em se tratando de um sistema síncrono. Uma negociação deve ser iniciada por um general (processo), por meio de um valor inicial que deve ser acordado entre os demais, sempre que satisfeitas as condições descritas por Kshemkalyani; Singhal, (2008):

- **Acordo** - todo processo confiável deve concordar com o mesmo valor.
- **Validação** - se o processo que inicia um acordo é confiável todos os demais processos devem concordar com o mesmo valor inicial dado pelo processo que iniciou a negociação.

- **Terminação** - eventualmente todos os processos não faltosos devem concordar sobre um valor.

Para garantir a correção de falhas e a manutenção das funções de um sistema distribuído onde nós comunicam-se trocando informações através de mensagens estabelecendo um acordo bizantino, alguns algoritmos podem ser utilizados.

Os algoritmos para BFT (*Byzantine Fault Tolerance*) visam viabilizar a implementação de serviços capazes de atender aos requisitos de confiabilidade, integridade e disponibilidade através da adaptação de técnicas de redundância em nível de aplicação e de ambiente execução (LUIZ, 2009).

Partindo da premissa de um sistema distribuído síncrono para atingir a validação entre os valores corretos entre processos e tomada de uma mesma decisão. A ordenação das mensagens deve ser mantida, e o atraso de comunicação ser limitado. São elencados três protocolos para garantir a prevenção de falhas:

Consensus Algorithm (CA): todos os processos não defeituosos usam o mesmo valor $v(i)$ para um processo p ; se o processo que envia (i) não é defeituoso então todo processo não defeituoso usa o valor $v(i)$ enviado por p . Cada processo p replica a mensagem de um processo p a ser válida para os outros processos. São necessários $f + 1$ rodadas para validar o acordo, onde f representa o número de processos falhos. Deve haver dois terços de processos não defeituosos de um total de K processos.

Interactive Consistency Algorithm (ICA): deve haver dois terços de nós não defeituosos de um total de K nós. A cada rodada $n - 1$ mensagens são transmitidas de um nó transmissor para os demais nós receptores. Ao final os nós possuirão vetores de valores para todos os nós relativos a cada nó transmissor e chegarão a um senso comum sobre valores coincidentes que serão adotados como valores corretos (KSHEMKALYANI; SINGHAL, 2008).

Signed Messages (SM): os nós podem corromper mensagens de outros nós, encobrindo o real emissor quando repassam a mensagem. Para que isso seja solucionado cada nó assina sua mensagem, através de alguma técnica como assinaturas digitais, de maneira que se esta for modificada possa ser tomado conhecimento. Os nós defeituosos são ignorados (KSHEMKALYANI; SINGHAL, 2008).

4 ARQUITETURA DE COMPARTILHAMENTO DE TAREFAS

Este capítulo descreve a arquitetura proposta para o controle de múltiplos robôs móveis. Apresenta uma descrição mais detalhada do problema, dos conceitos e definições que dão embasamento teórico a arquitetura MRTA-B que visa solucionar o problema de alocação de tarefas em grupos de robôs móveis de maneira confiável com a adoção de acordo bizantino.

4.1 Definições preliminares

Para o projeto de uma arquitetura que vise à solução de problemas de forma cooperativa por um grupo de robôs móveis foram elencadas algumas pré-suposições. Sistemas de múltiplos robôs são sistemas distribuídos, portanto será feito o uso de várias técnicas que remetem a computação distribuída, como escalonamento de tarefas, algoritmo de eleição, acordo bizantino, além dos conceitos de paralelismo, divisão da carga de trabalho e tolerância a falhas.

Primeiramente tem-se que o problema para o qual se pretende fazer uso de um grupo de robôs possui características de complexidade de tal modo que apenas um único robô não é suficiente para solucioná-lo. Problemas típicos que podem ser elencados para este trabalho entre alguns já citados podem ser, combate a incêndio, busca e resgate, vigilância patrimonial, limpeza e higienização de cidades, forrageamento, entre outros.

Desta forma acrescenta-se como fator na solução de alguns destes problemas, além da necessidade de múltiplos robôs, uma heterogeneidade entre eles, que implica em características e habilidades diferentes. Essa heterogeneidade pode advir de diferenças na composição físico-mecânica (hardware) de cada robô ou de funcionalidades (habilidades) individuais advindas de abstrações do comportamento através do software de controle de cada robô.

4.2 Visão geral da arquitetura proposta

A arquitetura proposta é baseada na taxonomia MRTA (ver Capítulo 3; Seção 3.3) para alocação de tarefas em sistemas com múltiplos robôs. Dessa forma, como se pretende estabelecer acordo bizantino nas comunicações entre os robôs, a arquitetura proposta foi intitulada MRTA-B.

A taxonomia MRTA visa a classificação de problemas de alocação de tarefas em sistemas de multi-robôs, sendo a sigla MT-MR-TA (*Mult task robots – Mult robot tasks – Time extended assignment*) a classificação de problemas que a arquitetura MRTA-B pretende lidar, ou seja, a alocação de tarefas complexas entre multi-robôs. Problemas estabelecidos na MT-MR-TA são aqueles onde há uma multiplicidade de atribuições entre robôs e tarefas, ou seja, as tarefas podem exigir vários robôs para serem executadas e os robôs podem ter capacidades que os permitam atuar em mais de uma tarefa.

A arquitetura MRTA-B visa a alocação de tarefas entre grupos de robôs aptos a executá-las, porém fazendo uso de técnicas que possibilitem a tolerância de falhas eventuais e inesperadas no SMR. Dessa forma, isso envolve o estabelecimento de acordo entre as partes envolvidas na execução de uma dada tarefa. A arquitetura proposta é ilustrada na Figura 4.1.

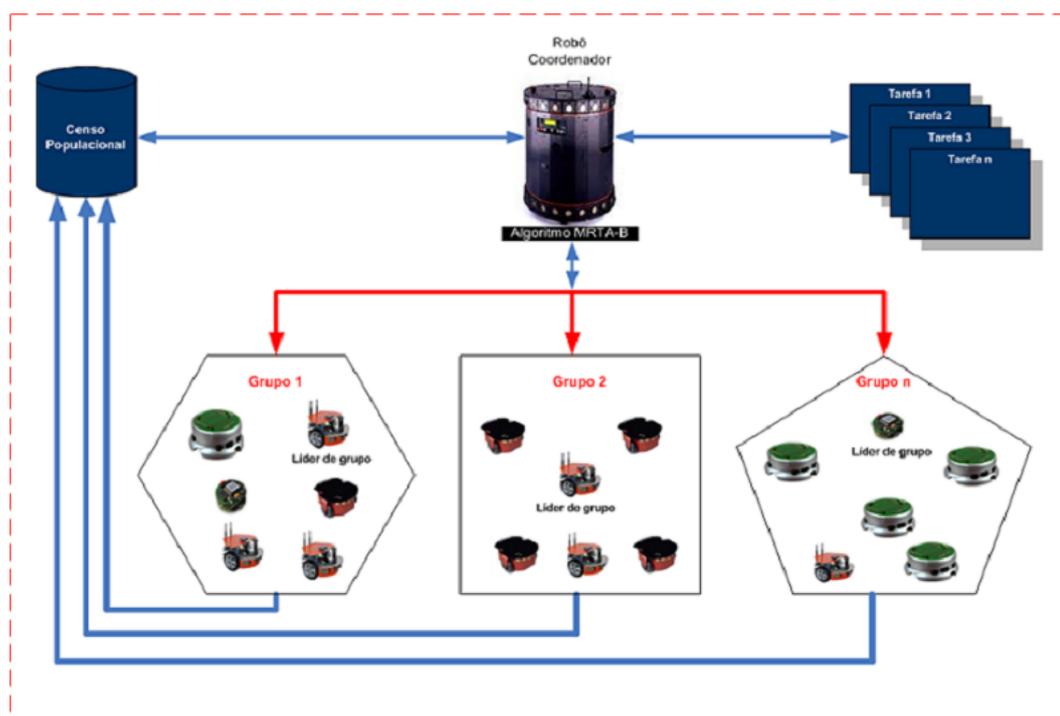


Figura 4.1 Visão geral da arquitetura MRTA-B.

Os componentes presentes na estrutura da arquitetura MRTA-B são condicionados pelas interações que mantêm entre si e processos pelos quais devem passar, assim é formada

uma hierarquia com diferentes fases na comunicação, sendo que, tudo é iniciado com uma população de robôs.

Uma população será formada por vários robôs presentes em um sistema de múltiplos robôs heterogêneos. Inicialmente cada membro da população de robôs não terá qualquer vínculo. Robôs que passam a integrar a população devem anunciar suas características e habilidades na base de dados intitulada “Censo Populacional”.

O censo populacional consiste em uma base de dados com informações sobre cada membro da população de robôs. Estas informações são as características anatômicas (físicas) e as abstrações de habilidades advindas do software chamadas de capacidades. O Censo Populacional terá o perfil de cada robô. A Figura 4.2 ilustra o processo de anúncio das capacidades pelos robôs da população.

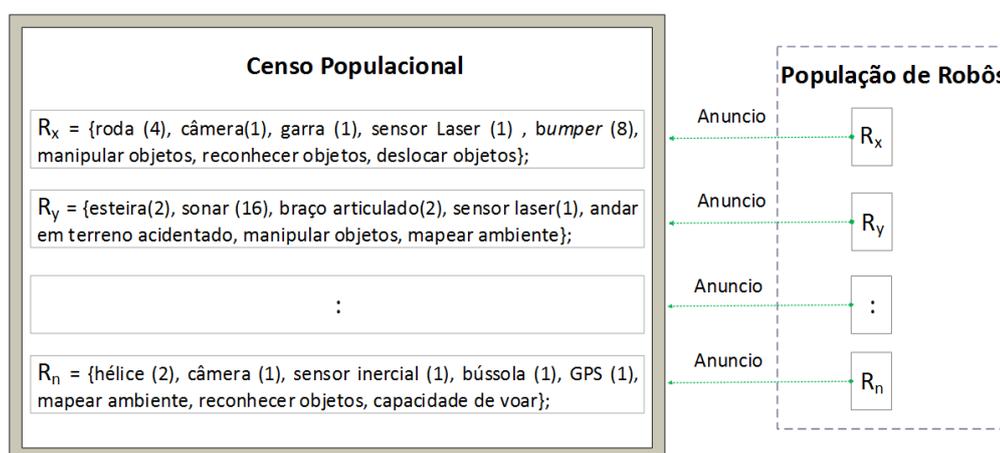


Figura 4.2 Anúncio das capacidades por parte dos robôs da população.

Robôs possuem habilidades que os permitem atuar em uma ou mais atividades, sendo necessário descobrir quais robôs podem cooperar e formar um grupo. Dessa forma tarefas de transporte, vigilância patrimonial ou limpeza, devem ser atribuídas a grupos específicos de robôs.

Uma tarefa será descrita por um BDT (Bloco de Descrição da Tarefa) com os requisitos necessários que constituem uma demanda de trabalho para sua execução. A Figura 4.3 ilustra um exemplo de um BDT.

Bloco de Descrição da Tarefa	
IDENTIFICAÇÃO DA TAREFA:	3
TIPO TAREFA:	Transporte-de-Carga
AMBIENTE DE EXECUÇÃO:	Fechado (<i>Indoor</i>)
COMPONENTES FÍSICOS NECESSÁRIOS:	[Garra, Câmera, Esteira, Braço Articulado, <i>Bumper</i>]
HABILIDADES:	[Carregar Carga(peso x), Locomoção em Terreno Acidentado, Reconhecer Objeto, Manipular Objeto]
TEMPO DE DURAÇÃO:	20 minutos

Figura 4.3 Exemplo de um bloco de descrição da tarefa.

Os itens que compõem um BDT são:

- **Tipo da tarefa:** uma pequena descrição sobre a tarefa.
- **Identificação:** um número que identifica a tarefa, para efeito de organização, manipulação e controle das tarefas presentes no sistema de múltiplos robôs.
- **Tempo de duração:** o tempo de duração de uma tarefa pode ser determinado ou indeterminado. Dessa forma uma tarefa pode ter um fim conhecido ou transcorrer infinitamente.
- **Ambiente de execução:** robôs podem estar aptos a trabalharem em determinados tipos de ambientes, e mesmo contendo as capacidades necessárias devem ser descartados se não atuam no ambiente exigido pela tarefa. O ambiente de execução pode ser em locais fechados (*indoor*) ou abertos (*outdoor*).
- **Componentes físicos:** identifica os componentes necessários que os robôs devem possuir para a realização da tarefa.
- **Habilidades:** as habilidades dão maior aporte sobre as reais capacidades de um robô. Estas podem definir métricas e tolerância sobre propriedades físicas, podendo estabelecer a capacidade de carga que um robô pode transportar, objetos que pode reconhecer, terrenos que pode se locomover, etc.

O BDT elenca requisitos necessários para que uma dada tarefa seja cumprida. Por exemplo, uma tarefa de transporte de carga deve definir o que e quanto deve ser carregado. No BDT não é necessário o registro de quais robôs devem realizar a tarefa, mas apenas elencar quais são os componentes físicos que esses robôs devem ter, deste modo a seleção dos robôs que farão parte de um grupo fica a cargo do coordenador.

A demanda de trabalho visa a associação dos robôs com capacidades condizentes aos requisitos da tarefa, desta forma, a tarefa tem informações suficientes para que a mesma seja

atribuída a robôs com condições de executá-la. Para que isso aconteça, esses robôs devem estar corretamente organizados em um grupo.

4.3 Modelo de comunicação na arquitetura MRTA-B

Todos os robôs da arquitetura proposta devem estar aptos a se comunicarem. Desta forma os robôs devem ter canais de comunicação entre si. A comunicação entre robôs da arquitetura MRTA-B inicia-se com a necessidade de robôs apresentarem comportamentos sociais, como o da eleição de coordenador e líderes de grupo. Para que isso aconteça um robô deve ter conhecimento do destinatário de uma mensagem, e da mesma forma, reconhecer o remetente.

Como os robôs anunciam suas capacidades no censo populacional, suas identificações vão sendo indexadas. Se um robô qualquer resolve iniciar um processo de eleição, este pode criar uma vetor com a identificação de cada robô que participará do processo e enviá-lo em uma mensagem a cada um dos participantes de forma síncrona. Assim, todos receberão a mensagem e poderão formar canais de comunicação.

Há duas principais atividades que podem acontecer com a população de robôs:

- Eleição de coordenador e líder – os robôs enviam mensagens em conformidade com o algoritmo do anel, ou seja, enviam somente para o primeiro sucessor ativo. A única mensagem transmitida em *broadcast* é do aviso do vencedor.
- Algoritmo do consenso – os robôs enviam as mensagens em *broadcast*, ou seja, para todos os robôs, a menos, que seja uma mensagem replicada, neste caso o remetente não recebe sua mensagem novamente.

As principais comunicações são:

- Entre robôs que elegem um coordenador;
- De coordenador para robôs avulsos (que formaram um grupo ou novos robôs recentemente cadastrados no censo populacional);
- De coordenador para líderes de grupo;
- De líderes de grupo para membros de grupo;
- Entre membros de grupo.

Como fica demonstrado um robô pode se utilizar de mais de uma forma de comunicação, enviando uma mensagem a um robô específico (sucessor ou não), ou para vários robôs, isso depende do nível de comportamento social que o robô estará envolvido.

4.4 Formação de grupos

Um robô da população de robôs será eleito coordenador de acordo com o algoritmo descrito na Seção 4.5. O processo de eleição do robô coordenador exige um número mínimo de cinco robôs na população. Esse número visa respeitar o valor mínimo de robôs para que um acordo bizantino seja possível mantendo a mesma condição para a posterior formação de ao menos um grupo de robôs na população, isso ocorre devido ao fato de o robô coordenador não participar de demandas no grupo.

É necessário $2k + 1$ robôs de redundância (robôs confiáveis) para validar um acordo, ou seja, o mínimo possível são 3 (três) robôs não faltosos, pois dessa forma há critério de desempate sobre a validação de um valor. No entanto é necessário um número de $3k + 1$ robôs para que um processo de acordo seja possível, ou seja, no mínimo 4 (quatro) participantes no acordo, pois, se houver três robôs negociando e um for faltoso, os outros dois não poderão decidir sobre a validade de um valor por não haver critério de desempate.

Dessa forma, o quinto robô que anuncia suas capacidades no censo populacional inicia o processo de eleição do robô coordenador com os robôs que anteriormente já haviam se anunciado, assim, é importante que o robô que iniciou a eleição envie uma mensagem aos demais robôs contendo todas as identificações presentes no censo populacional. O coordenador eleito terá o papel de criar grupos e atribuir tarefas a estes.

Tendo conhecimento da demanda de uma tarefa o robô coordenador consultará o censo populacional para saber quais robôs atendem os requisitos necessários para a execução da tarefa. Uma mensagem será enviada para cada robô escolhido informando que este fará parte de um grupo específico. A formação de um grupo levará em consideração um número mínimo de quatro robôs para que seja possível o estabelecimento de acordos bizantinos. Os robôs do grupo possuem habilidades suficientes para solucionar uma tarefa em questão.

O robô coordenador ficará aguardando a confirmação de resposta dos robôs escolhidos para formarem um grupo. Quando todos os robôs responderem ao coordenador, este enviará outra mensagem a todos os robôs do grupo solicitando que iniciem o processo de eleição do líder. A mensagem enviada pelo coordenador conterá um vetor com as identificações de cada robô. A partir deste vetor os robôs poderão iniciar a escolha do líder baseada no algoritmo descrito na Sessão 4.5.

O líder de um grupo de robôs terá o papel de decidir sobre a saída ou entrada de um novo robô no grupo, pelo término ou não de uma tarefa, e pela formação de equipes no grupo.

O grupo será fechado, dessa forma as mensagens somente serão trocadas internamente ao grupo e o contato externo será feito por intermédio do robô líder do grupo, desta forma, após a criação de um grupo o robô coordenador envia uma mensagem para o líder com a identificação da tarefa a ser realizada, este então negociará com os demais membros do grupo a execução da tarefa. As mensagens trocadas entre o líder e os demais robôs do grupo deverão seguir os critérios estabelecidos pelo acordo bizantino. O acordo bizantino é descrito na Seção 4.7.

Cada grupo terá como característica uma “força de trabalho” que é o somatório das capacidades de todos os robôs R_i de um grupo Cg . A força de trabalho deve representar os requisitos da tarefa, sendo representada pela Equação 2:

$$Cg = \sum_{i=0}^{n-1} CRi$$

Equação 2 Soma das capacidades dos robôs de um grupo.

Dentro de um grupo de robôs o acordo bizantino será estabelecido para atestar que há robôs confiáveis o suficiente para manter a força de trabalho exigida para a uma dada tarefa.

4.5 Processo de eleição do coordenador e dos líderes de grupo

O algoritmo escolhido para o processo de eleição do robô coordenador e do líder de grupo será o algoritmo do anel (*ring algorithm*) Chang e Roberts (1979) apud (COULOURIS; DOLLIMORE; KINDBERG, 2007). A Figura 4.4 ilustra o funcionamento do algoritmo do anel.

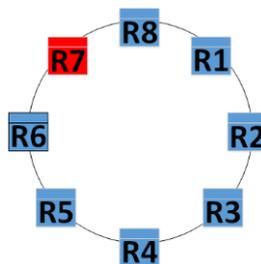


Figura 4.4 Exemplo do algoritmo do anel.

No algoritmo do anel os robôs no grupo estão dispostos em uma sequência em formato de anel (Figura 4.4). O quadrado destacado em vermelho representa o líder do grupo. Cada robô conhece seu antecessor e seu sucessor. Os passos para a eleição de um líder pelo algoritmo do anel são (COULOURIS; DOLLIMORE; KINDBERG, 2007):

1. O processo de eleição é iniciado por um robô ao tomar conhecimento da falta de líder no grupo ou perceber que o líder está inativo, este envia uma mensagem invocando uma eleição com sua identificação ao seu sucessor.
2. Cada robô reenvia a mensagem anexando sua identificação ao respectivo sucessor.
3. Eventualmente a mensagem retorna a um robô que anexou sua identificação. O robô checa o vencedor, que deverá ser o robô de identificação mais alta e envia uma mensagem a todos os robôs do grupo anunciando o novo líder.

A Figura 4.5 ilustra um exemplo do processo de eleição iniciado por não existir coordenador ou líder de grupo, sendo que, no quadro a) o robô R_2 inicia um processo de eleição; no quadro b) todos os robôs ativos participam da eleição enviando a mensagem original com sua própria identificação; e no quadro c) o robô R_2 anuncia aos demais que o robô R_4 foi eleito como novo coordenador.

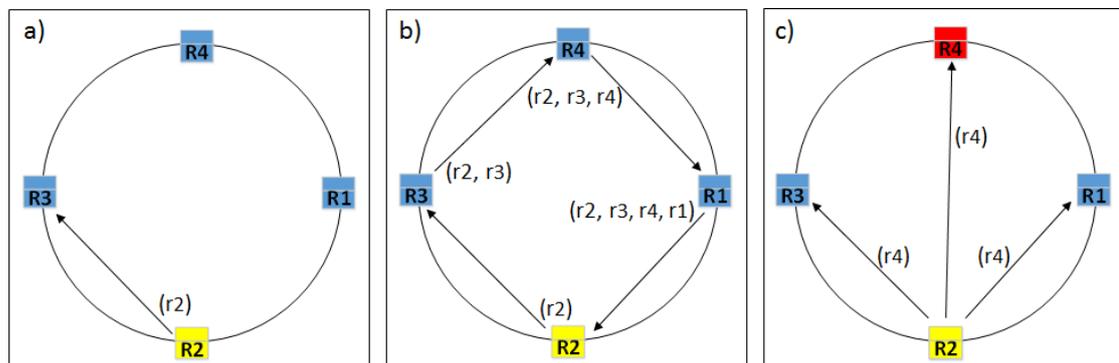


Figura 4.5 Processo de eleição.

Tendo em vista estas definições atenta-se para dois aspectos importantes para a construção da arquitetura, o controle das tarefas e a confiabilidade na comunicação entre o robô líder e os robôs membros de um grupo. Falhas arbitrárias podem advir de membros participantes do grupo ou do próprio líder, em virtude disso cada grupo será constituído de no mínimo quatro robôs para que exista a possibilidade de alcance de um acordo bizantino em negociações para a execução de tarefas.

Para simular processos falhosos e um comportamento arbitrário, cada robô terá um módulo em sua arquitetura de controle individual intitulado “módulo de má fé”. Esse comportamento é responsável por fazer com que o robô tenha comportamentos anômalos durante sua permanência em um grupo de robôs.

4.6 Arquitetura de controle dos robôs

Cada robô da população possui um sistema de controle baseado em uma arquitetura híbrida. A Figura 4.6 ilustra a arquitetura do sistema de controle de cada robô. É possível perceber que a arquitetura é composta por módulos reativos que exigem reações imediatas e módulos deliberativos que necessitam tomar decisões que podem envolver a visualização do modelo de mundo e o planejamento sobre o comportamento que se quer executar.

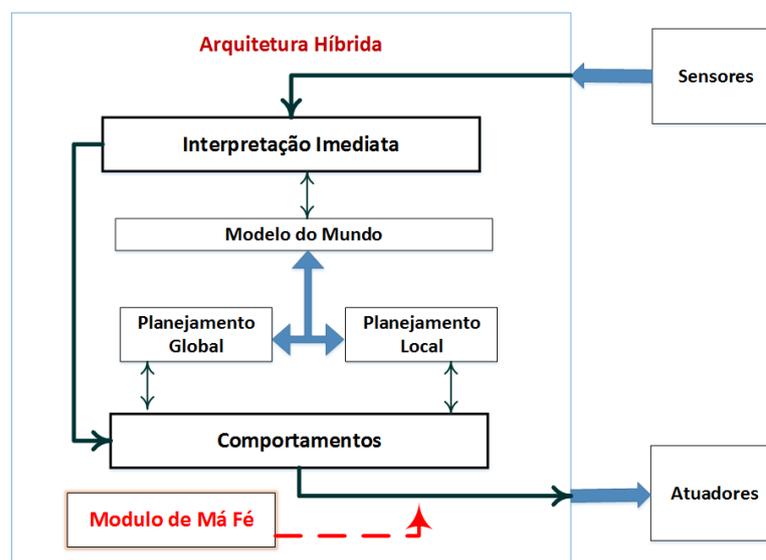


Figura 4.6 Arquitetura do sistema de controle de cada robô.

O planejamento local remete a objetivos de curto prazo, geralmente evocando uma ou mais ações executadas em período curto de tempo, e também ao ambiente parcialmente observável pelo robô, ou seja, informações limitadas ao alcance e capacidade dos sensores. O planejamento global refere-se a algoritmos de alto nível, para mapeamento e localização de um ambiente maior do que a visibilidade do robô, objetivos de longo prazo, onde geralmente ações são ordenadas. Participação de processos de eleição e consenso, e exercer funções de líder ou coordenador.

Os comportamentos acionados pelos robôs podem indicar tanto ações simples e relativas a execução de tarefas, como andar, reconhecer objetos até comportamentos sociais de eleição de coordenador e líder, processo de acordo, e executar papel de coordenador ou líder.

O “módulo de má fé” representa a execução de comportamentos anômalos para um ou mais robôs. Este módulo suprime comportamentos em execução, substituindo-os por outros anormais. Dessa forma se pretende simular situações de falha para o sistema pois qualquer robô pode apresentar comportamentos que prejudiquem o sistema como um todo.

Algumas tarefas poderão ser realizadas por uma equipe de robôs formada pelo líder. As equipes serão dinâmicas, ou seja, são formadas por tempo limitado, sendo este tempo delimitado a partir do início da execução até o término da tarefa.

Uma vez que há probabilidades de falhas, o acordo bizantino é parte essencial da arquitetura, correlacionando aspectos de robustez, confiabilidade, segurança, sincronismo, e resiliência do sistema.

4.7 Comunicação baseada em acordo bizantino

As mensagens enviadas pelo líder para os membros do grupo, que podem ser de atribuição de tarefas, entrada e saída de membros no grupo, encerramento de uma tarefa, entre outras, deverão ser confiáveis.

Qualquer robô que receba uma mensagem sobre a alocação de uma tarefa deve se certificar que esta mensagem partiu do líder e se encontra íntegra, para tanto ele fará uso do acordo bizantino.

O acordo se dará entre os membros do grupo que validaram ou não uma mensagem do líder. Um membro do grupo que deseja confirmar se a mensagem é mesmo do líder iniciará um acordo bizantino que incluirá todos os membros do grupo.

Um acordo só é possível, a partir de um grupo mínimo de quatro robôs (ver Capítulo 3 Sessão 3.4.1). No entanto, para que o processo de acordo seja confiável, são necessários $2k + 1$ robôs totalmente seguros de um total de $3k + 1$ robôs membros do grupo. A equação $3f + 1$, representa o número mínimo de robôs em um grupo para que um acordo seja possível, onde f representa o número de robôs faltosos. Caso existam três robôs faltosos é necessário que um grupo seja composto de um número mínimo de 10 robôs.

Será utilizado o algoritmo de Consenso (*Consensus Algorithm*), para o tratamento de falhas bizantinas especificado por Kshemkalyani; Singhal, (2008). Dessa forma, o exemplo de uma tarefa anunciada por um líder será utilizado para demonstrar o funcionamento do algoritmo de consenso, e conseqüentemente, como é possível identificar de modo seguro se a mensagem é do líder.

Quando o líder anuncia uma tarefa aos demais robôs do grupo um processo de acordo é estabelecido. A Figura 4.7 ilustra parte do processo de acordo bizantino com o algoritmo de consenso, na qual, no quadro a) o líder anuncia a chegada de uma tarefa ao grupo; no quadro b) o robô R_2 participa de uma rodada na negociação e assim sucessivamente até o R_8 .

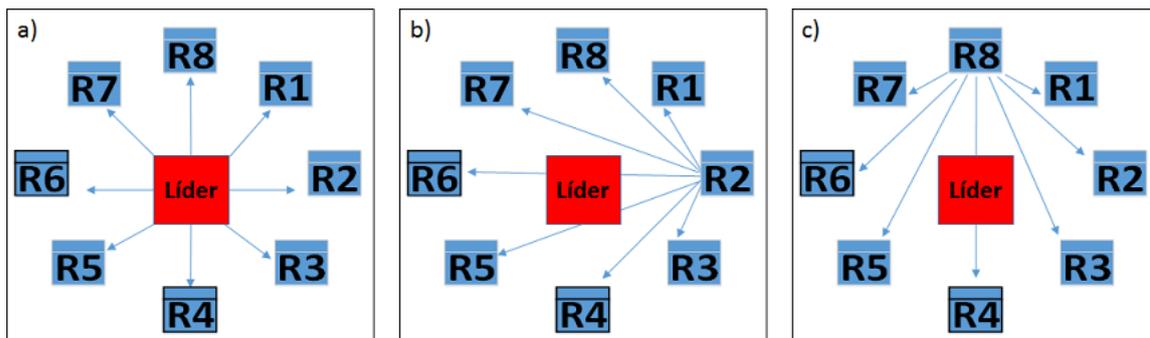


Figura 4.7 Exemplo do processo de estabelecimento do acordo bizantino.

Os robôs do grupo precisam decidir sobre um valor $T(x)$ que representa a tarefa. Cada robô replica o valor $T(x)$ recebido do líder do grupo em um total de $f + 1$ rodadas, na qual f representa o número máximo de robôs faltosos que podem ser tolerados para o alcance de um acordo. Em cada rodada um robô participa do processo de negociação enviando $n - 1$ mensagens aos demais robôs no grupo. As rodadas se repetem até o alcance de um acordo ou a não validação da tarefa por insuficiência de robôs confiáveis ou por um líder faltoso.

5 ANÁLISE SOBRE A ARQUITETURA PROPOSTA

Este capítulo apresenta uma análise sistêmica do processo de alocação de tarefas a grupos de robôs com o uso da arquitetura MRTA-B. Para viabilizar a análise é descrito um estudo de caso.

5.1 Definições gerais

A metodologia utilizada para avaliação da arquitetura MRTA-B foi de estudo de caso, ou seja, dada uma população composta por n robôs e uma lista de tarefas é feita a análise do funcionamento da arquitetura proposta.

A arquitetura MRTA-B tem como principal característica a alocação de tarefas complexas entre grupos de robôs de forma confiável. Ela dá suporte para que robôs aptos a executarem determinada tarefa formem um grupo. Para que isso ocorra, dentro da arquitetura existe uma hierarquia de comando, que é composta pela figura do coordenador, líder e membros de grupos. Essa hierarquia não é pré-existente, nem fixa, mas é formada dinamicamente na arquitetura MRT-B.

O algoritmo do anel e consenso garantirão, respectivamente, a eleição de coordenador e líderes de grupo e um número mínimo de robôs confiáveis para atuarem em uma tarefa. Isso implica que a arquitetura MRTA-B não leva em consideração certas condições para execução de uma tarefa, como, fracionamento e atribuição a robôs específicos ou ordem na execução de sub-tarefas, mas apenas a atribuição das tarefas supondo que os robôs tenham autonomia para executá-las.

No sistema de múltiplos robôs baseado na arquitetura MRTA-B a população de robôs inicialmente deve anunciar suas capacidades na base de dados “Censo Populacional”. Este sistema terá uma lista de tarefas que devem ser executadas por um grupo de robôs. A lista de tarefas contém os BDTs de cada tarefa.

A Tabela 2 ilustra o Censo Populacional onde são registrados 22 (vinte) robôs na população.

Tabela 2 Abstração das informações do senso populacional

Identificação Robô	Componentes Físicos	Habilidades
R ₁	Esteira(2), sonar (16), braço articulado(1), sensor Laser(1), perfuradora, câmera	Locomoção em terreno acidentado, reconhecimento de objetos, manipulação de objetos
R ₂	Esteira(2), sonar (16), braço articulado(1), sensor laser(1), perfuradora, câmera	Locomoção em terreno acidentado, reconhecimento de objetos, manipulação de objetos
R ₃	Esteira(2), sonar (16), braço articulado(1), sensor laser(1), perfuradora, câmera	Locomoção em terreno acidentado, reconhecimento de objetos, manipulação de objetos
R ₄	Rodas(2), sensor laser(1), garra, câmera	Reconhecimento de objetos, Deslocamento de Objetos
R ₅	Rodas(2), sensor <i>laser</i> (1), garra, câmera	Reconhecimento de objetos, deslocamento de objetos
R ₆	Hélice(2), sensor inercial, bússola, câmera	Reconhecimento de objetos, mapeamento de ambiente, capacidade de voar
R ₇	Hélice(2), Sensor inercial, bússola, câmera	Reconhecimento de objetos, mapeamento de ambiente, capacidade de voar
R ₈	Rodas(2), sonar(8), <i>bumper</i> (8), ventilador, saco poroso	Sucção de objetos
R ₉	Rodas(2), sonar(8), <i>bumper</i> (8) ventilador, saco Poroso	Sucção de objetos
R ₁₀	Rodas(2), sonar(8), <i>bumper</i> (8), ventilador, saco poroso	Sucção de objetos
R ₁₁	Rodas(2), sonar(8), <i>bumper</i> (8) ventilador, saco poroso	Sucção de objetos
R ₁₂	Rodas(4), sensor <i>laser</i> (1), sonar(16), caçamba	Transporte de carga
R ₁₃	Rodas(4), sensor <i>laser</i> (1), sonar(16), caçamba	Transporte de carga
R ₁₄	Rodas(4), sensor <i>lazer</i> (1), sonar(8), caçamba	Transporte de carga
R ₁₅	Rodas(4), sensor <i>lazer</i> (1), sonar(8), caçamba	Transporte de carga
R ₁₆	Rodas(4), sensor <i>laser</i> (1), sonar(16)	Reconhecimento de objetos
R ₁₇	Rodas(4), sensor <i>laser</i> (1), sonar(16)	Reconhecimento de objetos
R ₁₈	Braço articulado(2), garra(2), câmera, sensor <i>laser</i> (3)	Manipulação de objetos, escalar
R ₁₉	Braço articulado(2), garra(2), câmera, sensor <i>laser</i> (3)	Manipulação de objetos, escalar

R₂₀	Perna articulada (6), sensor <i>laser</i> (1), sonar(16), garra	Manipulação de objetos, subir degraus, reconhecimento de objetos
R₂₁	Perna articulada (6), sensor <i>laser</i> (1), sonar(16), garra	Manipulação de objetos, subir degraus, reconhecimento de objetos
R₂₂	Perna articulada (6), sensor <i>laser</i> (1), sonar(16), garra	Manipulação de objetos, subir degraus, reconhecimento de objetos

A Tabela 3 lista um conjunto de tarefas cada qual com seu respectivo BDT. Para a análise realizada neste capítulo são listadas 5 (cinco) tarefas.

Tabela 3 Abstração dos requisitos das tarefas presentes nos BDT.

Identificação da Tarefa	Descrição	Ambiente de execução	Componentes físicos necessários	Habilidades	Tempo de duração
T₁	Limpeza	Fechado (andar)	Sonar, <i>bumper</i>	Sucção de objetos	20 min.
T₂	Transporte de carga	Fechado (Garagem)	Caçamba, garra, braço articulado, rodas, câmera, sensor <i>laser</i>	Reconhecimento de objetos, manipulação de objetos, deslocamento de objetos, transporte de carga	10 min.
T₃	Vigilância	Aberto (pátio)	Câmera, sensor <i>Laser</i> , <i>bumper</i>	Mapeamento de ambiente, reconhecimento de objetos, capacidade de voar.	Contínuo
T₄	Empilhar caixas	Fechado (sala)	Braço articulado, garra, <i>laser</i> , sonar	Reconhecimento de objetos Manipulação de objetos	30 min.
T₅	Coleta de amostra	Aberto (pátio)	Esteira, sensor <i>laser</i> , câmera, perfuradora, braço articulado	Locomoção em terreno acidentado, mapeamento do ambiente, reconhecimento de objetos, manipulação de objetos	50 min.

Após anunciar suas capacidades no Censo Populacional o quinto robô inicia um processo de eleição com os outros robôs anteriormente registrados no censo visando a eleição do

coordenador através do algoritmo do anel. O robô R_5 envia uma mensagem requisitando uma eleição aos demais robôs. Nesse momento os robôs entram em consenso para garantir que a eleição para coordenador seja confiável, ou seja, determinando se há robôs confiáveis o suficiente para iniciar o processo de eleição do coordenador.

O processo de eleição do coordenador exige um número mínimo de cinco robôs confiáveis, pois uma vez que o coordenador é eleito, deverá haver um número mínimo de quatro robôs na população para manter futuros processos que exijam acordos bizantinos, como o da formação de um grupo. O processo de eleição do coordenador elegerá o robô com a maior identificação.

Uma vez eleito o coordenador avisará todos os demais robôs no grupo da sua condição como coordenador, enviando uma mensagem a cada robô com o vetor de identificação dos robôs registrados no censo. Os robôs da população entrarão em processo de consenso para confirmar a veracidade da mensagem, atestando que o robô que enviou a mensagem informando ser o coordenador realmente é o coordenador.

Supondo que o robô R_5 tenha sido eleito coordenador, este poderá consultar a lista de tarefas e ter acesso ao censo populacional. O robô R_5 analisará a primeira tarefa, T_1 , e iniciará o processo de seleção dos robôs que irão formar o grupo para a execução desta tarefa. O processo de seleção dos robôs é baseado na força de trabalho, ou seja, cada robô que apresentar uma capacidade (componente físico ou habilidade) correspondente com o requisito da tarefa receberá 1 (um) ponto.

O processo de seleção e do cálculo da força de trabalho deve obedecer os seguintes critérios:

- **Ambiente de execução:** robôs que não atuam no ambiente exigido pela tarefa são desconsiderados.
- **Habilidades:** cada habilidade presente em um robô selecionado deve corresponder com a habilidade requerida pela tarefa.
- **Capacidades físicas:** os componentes de hardware presentes em cada robô que tenham correspondência com as exigências da tarefa.

A capacidade de ambiente de execução pode restringir de início um robô para determinada tarefa. O robô coordenador procurara inicialmente atribuir um total de 4 (quatro) robôs a partir da pontuação para a habilidade, devendo alocar ao menos um robô para cada habilidade presente como requisito na tarefa. Atendendo cada requisito de habilidade da tarefa, mas tendo um grupo incompleto, o robô coordenador passará para a pontuação dos compo-

nentes físicos, se não houver na população robôs suficientes para alocar a tarefa, o robô coordenador prosseguirá para a próxima tarefa da lista.

Para a tarefa T_1 os robôs R_8 , R_9 , R_{10} e R_{11} que são do mesmo tipo foram selecionados para cumprirem os requisitos de habilidades da tarefa T_1 e formarão o Grupo $_1$. Os robôs selecionados receberão uma mensagem informando que ele fará parte do Grupo $_1$. O robô coordenador aguardará a resposta de confirmação da mensagem, para cada um destes robôs. Posteriormente enviará uma mensagem com o vetor dos robôs do Grupo $_1$ requerendo que estes iniciem a eleição de um líder.

Para a tarefa T_2 os robôs R_4 e R_5 , R_{12} , R_{13} serão listados para fazerem parte do Grupo $_2$, embora os robôs R_1 , R_2 e R_3 apresentem duas habilidades para a tarefa em questão eles são mais apropriados a ambientes externos.

Para tarefa T_3 , que é de vigilância, os robôs R_6 , R_7 , R_{20} e R_{21} serão selecionados formando o Grupo $_3$, cada um destes robôs pode se locomover tanto em ambientes fechados quanto em ambientes abertos.

Para a tarefa T_4 serão selecionados os robôs R_{16} , R_{17} , R_{18} e R_{19} que possuem habilidades próprias para empilhar caixas, como reconhecimento e malipulação de objetos, estes formaram o Grupo $_4$.

Para a tarefa T_5 será atribuída aos robôs R_1 , R_2 , R_3 e R_{22} , sendo que, estes são propícios para ambientes externos, além de cumprirem outros requisitos. Estes robôs formaram o Grupo $_5$.

A Tabela 4 lista todas as tarefas com os respectivos grupos responsáveis pela execução das mesmas.

Tabela 4 Resultado da alocação de tarefas com os respectivos grupos formados.

Tarefa	Grupo	Robôs	Líder
T₁	Grupo $_1$	R_8, R_9, R_{10}, R_{11}	R_{11}
T₂	Grupo $_2$	R_4, R_5, R_{12}, R_{13}	R_{13}
T₃	Grupo $_3$	R_6, R_7, R_{20}, R_{21}	R_{21}
T₄	Grupo $_4$	$R_{16}, R_{17}, R_{18}, R_{19}$	R_{19}

T_5	Grupo ₅	R ₁ , R ₂ , R ₃ , R ₂₂	R ₂₂
-------	--------------------	--	-----------------

A eleição dos líderes ocorre internamente aos grupos após o robô coordenador enviar uma mensagem a todos os robôs de cada grupo solicitando que estes iniciem o processo de escolha do líder. No Grupo₁, por exemplo, os robôs iniciaram um processo de eleição para eleger um líder através do algoritmo de anel. Supondo que o robô R₁₁ seja eleito o líder do Grupo₁ este mandará uma mensagem para o robô coordenador R₅. A partir desse momento as mensagens trocadas entre Grupo₁ e o coordenador somente acontecerão via robô líder.

O coordenador R₅ enviará uma mensagem para o líder R₁₁ para que a tarefa T₁ seja iniciada. O líder enviará a indicação para execução da tarefa a todos os robôs do Grupo₁, que iniciarão um processo de consenso para confirmar que há robôs confiáveis o suficiente para executar a tarefa T₁, sendo que se este processo for validado caberá aos robôs no grupo garantir a sua execução.

É importante ressaltar que a arquitetura proposta não leva em consideração a forma com que o robô líder irá gerenciar o processo de execução da tarefa pelos robôs liderados, aqueles pertencentes ao grupo do líder. A arquitetura proposta baseia-se no princípio de que tarefas precisam ser executadas por um grupo de robôs, logo a finalidade é criar os grupos e posteriormente fazer a alocação das tarefas a estes grupos.

6 CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS

Este trabalho apresentou uma proposta de arquitetura para alocação de tarefas em sistemas multi-robôs baseada na taxonomia MRTA. A arquitetura proposta, intitulada MRTA-B, visa a formação de grupos de robôs e a alocação de tarefas de maneira confiável, isto é, garantindo que mesmo na presença de falhas arbitrárias por parte de algum robô do grupo a tarefa consiga ser executada.

A arquitetura MRTA-B contempla problemas de múltiplas tarefas e múltiplos robôs (MT-MR) proposto na taxonomia MRTA. Na MRTA-B as tarefas são separadas conforme seus requisitos e associadas a grupos de robôs aptos a executá-las. Os grupos são formados por robôs que possuem características físicas e habilidades necessárias para o cumprimento da tarefa. O uso de acordo bizantino garante que a formação de grupos e a comunicação entre o robô coordenador e os líderes de grupos e entre o líder de um grupo e os robôs membros desse grupo sejam feitas de maneira confiável.

O uso de acordo bizantino conduz a um número mínimo de 4 (quatro) robôs por grupo corroborando com os problemas aplicáveis a MT-MR em MRTA, ou seja, garantindo que uma tarefa complexa será seja distribuída entre vários robôs, assim, um grupo formado por n robôs pode explicitamente cooperar na execução da tarefa.

Para a compreensão da arquitetura MRTA-B foi utilizado um estudo de caso, sendo que foram listadas algumas tarefas para serem distribuídas em uma população de robôs. O estudo de caso demonstrou que a arquitetura MRTA-B tem potencial de fornecer os objetivos de alocação e confiabilidade na execução de tarefas.

A arquitetura MRTA-B apresenta uma solução para a alocação de tarefas entre grupos de robôs, desta forma, o robô líder de cada grupo deve gerenciar e dividir a tarefa entre os

robôs membros do grupo. O gerenciamento da execução da tarefa intra-grupo foge aos propósitos deste trabalho, podendo ser objeto de estudo em trabalhos futuros.

Para melhor avaliar a arquitetura MRTA-B é necessário que a mesma seja implementada em ambiente simulado e em robôs reais. Ao longo da elaboração deste trabalho buscou-se implementar a arquitetura MRTA-B, entretanto por diversas razões isso não foi possível.

6.1 Propostas para trabalhos futuros

Abaixo são listadas algumas propostas de trabalhos que visam melhorar e/ou estender a arquitetura descrita neste trabalho.

1. Implementar a arquitetura MRTA-B e avaliá-la em ambiente simulado e também em robôs reais;
2. Adotar alguma técnica de representação de dados, como por exemplo ontologias, para representar as características físicas e comportamentais de cada robô da população de robôs;
3. Definir uma metodologia para a divisão de uma tarefa alocada a um grupo de robôs em sub-tarefas, que suporte regras e restrições entre estas;
4. Adotar algum protocolo de comunicação de tempo real que permita que as mensagens sejam encaminhadas no menor tempo possível.

REFERÊNCIAS

ADEPT MOBILEROBOTS. **Seekur Jr.** Disponível em: <<http://www.mobilerobots.com>>. Acesso em: 5 maio 2012.

ADEPT TECHNOLOGY. **PowerBot Datasheet.** Disponível em: <<http://www.mobilerobots.com/ResearchRobots/PowerBot.aspx>>. Acesso em: 9 nov. 2012.

ALVAREZ, B., Iborra, A., Pastor, J. A., Fernández, C., Alonso, A., e de la Puenta, J. A.(2001). **Software Architecture For Development of Mechatronic Systems: Service robots.** Dedicated Systems Magazine, Vol. 4pp. 17–22.

BRÄUNL, Thomas. **Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems.** 3. ed. Crawley: Springer, 2008. 560 p.

BRUMITT, Barry Lowell. **A Mission Planning System for Multiple Mobile Robots in Unknown, Unstructured, and Changing Environments.** 1998. 127 f. Tese (Doutorado) Carnegie Mellon University, Pittsburgh.

BOSTON DYNAMICS. **Cheetah: Fastest Legged Robot.** Disponível em: <<http://www.bostondynamics.com>>. Acesso em: 15 set. 2012.

CAO, Hung et al. **Complex Tasks Allocation for Multi Robot Teams under Communication Constraints.** In: NATIONAL CONFERENCE ON “CONTROL ARCHITECTURES OF ROBOTS”, 5, 2010, Douai.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas Distribuídos: Conceitos e Projeto.** 4. ed. Porto Alegre: Artmed Editora, 2007. 784 p.

DE PIERI, E. R. (2002). **Curso de robótica móvel.** Ufsc, Florianópolis, 2002. 133 p.

DIAS, M. Bernardine et al. **Market-Based Multirobot Coordination: A Survey and Analysis.** Pittsburgh: Carnegie Mellon Technical Report CMU-RI-TR-05-13, 2005. 25 p.
FISCHER, M. J., LYNCH, N. A., PATERSON, M. S., “**Impossibility of Distributed Consensus with One Faulty Process**”, Journal of the ACM, v. 32, n. 2, p. 374–382, April.1985.

GARCÍA, Paula et al. **Scalable Task Assignment for Heterogeneous Multi-Robot Teams.** International Journal of Advanced Robotic Systems, INTECH, New York, 2013. Vol. 10, 10 p, 07 fev 2013.

GERKEY, Brian 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. Disponível em: <<http://ijr.sagepub.com/content/23/9/939>>. Acesso em: 10 abr. 2012.

GLOBAL OCEAN OBSERVING SYSTEM. **Argo**: part of the integrated global observation strategy. Disponível em: <<http://www.argo.ucsd.edu>>. Acesso em: 2 out. 2012.

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.

HONDA MOTOR. **Asimo**: The world's most advanced humanoid robot. Disponível em: <<http://asimo.honda.com>>. Acesso em: 11 maio 2012.

HONORATO, Cristóvão Tavares. **Bizantium**: Replicação Bizantina de Bases de Dados. 2009. 57 f. Dissertação (Mestrado) - Instituto Superior Técnico, Lisboa, 2009.

INCT-SEC. **Tiriba**. Disponível em: <<http://www.inct-sec.org/br/aplicacoes/vant-tiriba>>. Acesso em: 10 out. 2012.

IROBOT. **IRobot Warrior**. Disponível em: <<http://www.irobot.com>>. Acesso em: 21 set. 2012.

KORSAH, G. Ayorkor; DIAS, M. Bernardine; STENTZ, Anthony. **A Comprehensive Taxonomy for Multi-Robot Task Allocation**. Disponível em: <http://www.cs.cmu.edu/afs/cs/Web/People/gertrude/documents/Korsah_IJRR_Taxonomy.pdf>. Acesso em: 20 nov. 2012.

KSHEMKALYANI, Ajay D.; SINGHAL, Mukesh. **Distributed Computing: Principles, Algorithms, and Systems**. New York: Cambridge University Press, 2008. 736 p.

KUKA. **Industrial Robots**. Disponível em: <<http://www.kuka.com>>. Acesso em: 20 abr. 2012.

LUIZ, Aldelir Fernando. **Arquitetura para Replicação de Serviços Tolerantes a Falhas Bizantinas Baseada em Espaço de Tuplas**. 2009. 131 f. Dissertação (Mestrado) - Pontifícia Universidade Católica do Paraná, Curitiba, 2009.

MARCHI, J. **Navegação de Robôs Móveis Autônomos: Estudo e implementação de abordagens**. Master's thesis, UFSC - Universidade Federal de Santa Catarina, 2001.

MOSTEO, Alejandro R.; MONTANO, Luis. **A survey of multi-robot task allocation**. Zaragoza: Aragon Institute Of Engineering Research, 2010. 27 p.

NASA. **Curiosity**. Disponível em: <<http://www.nasa.gov>>. Acesso em: 20 ago. 2012.

NEHMZOW, Ulrich. **Mobile Robotics: A Practical Introduction**. 2. ed. London: Springer, 2003.

PEREIRA, Jonas. **Avaliação e Correção do Modelo Cinemático de Robôs Móveis Visando a Redução de Erros no Seguimento de Trajetórias**. 2003. 141 f. Dissertação (Mestrado) - Curso de Mestrado Em Automação Industrial, Centro de Ciências Tecnológicas, Udesc, Joinville, 2003.

PEREZ, Anderson Luiz Fernandes. **Um Sistema Evolutivo Embarcado para Controlar uma População de Robôs Móveis usando Programação Genética**. 2006. 104 f. Qualificação (Doutorado) - Ufsc, Florianópolis, 2006.

RIBEIRO, C., Costa, A., e Romero, R. (2001). **Robôs Móveis Inteligentes: Princípios e Técnicas**. Proc. Anais do XXI Congresso da Sociedade Brasileira de Computação -SBC-2001, pp. 258–306.

ROSÁRIO, João Mauricio. **Princípios de Mecatrônica**. São Paulo: Pearson, 2005. 356 p. Disponível em: <<http://ufsc.bvirtual.com.br/editions/986-principios-de-mecatronica.dp>>. Acesso em: 20 agost. 2012.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência Artificial**. 2. ed. Rio de Janeiro: Elsevier, 2004. 1021 p.

SIEGWART, Roland; NOURBAKHS, Illah R.. **Introduction to Autonomous Mobile Robots**. Massachusetts: The Mit Press, 2004. 335 p.

SILVA, Luciano Rottava Da. **Análise e Programação de Robôs Móveis Autônomos da Plataforma Eyebot**. 2003. 109 f. Dissertação (Mestrado) - Curso de Mestrado Em Engenharia Elétrica, Departamento de Automação e Sistemas, Ufsc, Florianópolis, 2003.

SWARMANOID.ORG. **Swarmanoid**. Disponível em: <<http://www.swarmanoid.org>>. Acesso em: 10 nov. 2012.

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

ZLOT, Robert Michael. **An Auction-Based Approach to Complex Task Allocation for Multirobot Teams**. 2006. 187 f. Tese (Doutorado) - Robotics Institute, Carnegie Mellon University, Pittsburgh 2006.