

# TOLERÂNCIA A FALHAS COOPERATIVA EM SISTEMAS ROBÓTICOS MULTIAGENTES

Plano de pesquisa de mestrado em  
ciência da computação.

Márcio G. Morais

# Outline

---

- Introdução
  - Falhas em Robótica
  - Sistema de múltiplos robôs
  - Software em robótica
- Estado da Arte
  - Situação dos Frameworks c/ tolerância a falhas
  - Abstração em Programação de Robôs
- Plano de Pesquisa
  - Motivação
  - Objetivo e Escopo
  - Problema de Pesquisa
    - Tolerância a Falhas Cooperativa
- Cronograma e Atividades

# Introdução

---

Laboratórios – ambiente controlado



Mundo real – dinâmico e imprevisível

# Introdução

---

- ◉ Linhas de produção da indústria
- ◉ Agricultura de precisão
- ◉ Pecuária
- ◉ Entretenimento
- ◉ Pesquisa espacial
- ◉ Logística de tropas e bombardeios

# Introdução - Falhas em Robótica

---

Mundo real – dinâmico e imprevisível

Interação com humanos e outros robôs

Interação com o ambiente

Fenômenos naturais

Uso contínuo – desgaste natural

Suscetibilidade à interferências

**Causam defeitos!!!**

# Introdução - Falhas em Robótica

---

## Falhas podem ocasionar

Manutenção não programada - Prejuízos na produção

Resultado diferente do esperado – Prejuízo na produção

Interação com humanos – risco à integridade

Robôs – projetos customizados – custo ao usuário

## Tolerância a Falhas

Confiança do usuário no sistema

# Introdução

## Sistemas com Múltiplos Robôs

---

### ● Vantagens e aplicações

- Menor tempo para atingir objetivos
  - Paralelismo na execução de tarefas
- Resolver tarefas complexas com custo mais baixo
  - Redes de diferentes robôs especializados
- Exemplo de aplicação
  - Sistemas de busca e coleta
    - Qual robô está disponível?
    - Qual robô está mais próximo?
    - Qual robô é capaz de coletar um determinado item?

# Introdução

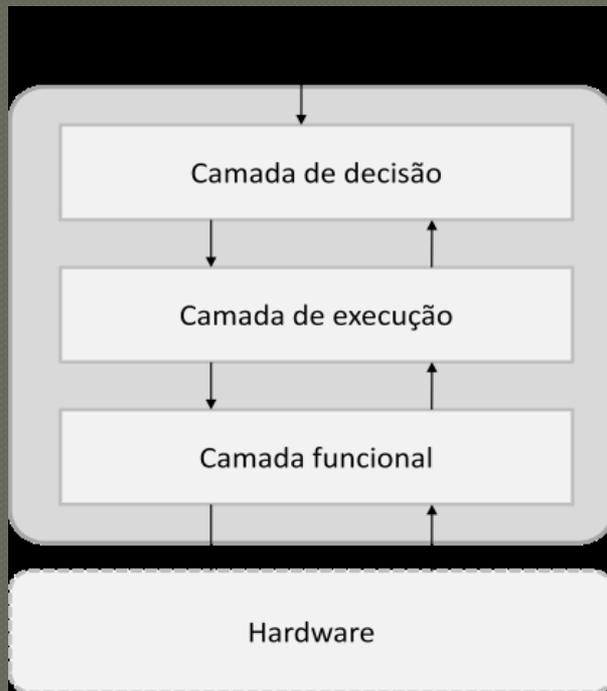
## Software em Robótica

---

- ◉ Uso do framework devido ao alto grau de complexidade do software
  - Sistema distribuído
  - Modular
  - Comprometido com restrições de tempo real
- ◉ Arranjo em camadas
  - Separação de responsabilidades
  - Baixo acoplamento
  - Alta coesão

# Introdução - Software em robótica

## Framework – arranjo em camadas



**Camada de decisão** - definição dos objetivos, execução de ações e avaliação de sucesso ou falha

**Camada de execução** - infraestrutura de software - troca de informações entre camadas ou módulos

**Camada funcional** – interface entre o hardware e as demais camadas

# Estado da Arte

---

# Frameworks e tolerância a falhas

---

Grande parcela dos frameworks não possui implementação de tolerância a falhas

Uma parcela menor possui de forma implícita

Pequena parte possui sistema tolerante a falhas

# Frameworks e tolerância a falhas

---

- Rotinas de controle tolerante a falhas
  - Eficientes enquanto atuam sobre sensores e atuadores
  - Não estão aptos a considerar falhas relacionadas ao conhecimento de alto nível
- Mecanismos de recuperação de falhas eficientes
  - Soluções não são flexíveis para gerenciar situações experimentadas durante missões complexas.

# Tolerância a falhas cooperativa

---

- Redes homogêneas de robôs
- Cooperação entre agentes na forma de redundância de sensores
- Detecção ocorre quando dois robôs se encontram
- Isolamento ocorre quando três robôs se encontram
- Falhas de alto nível são consideradas “falso positivo”

R. A. Carrasco, F. Núñez e A. Cipriano, “Fault detection and isolation in cooperative mobile robots using multilayer architecture and dynamic observers,” *Robotica*, pp. 555-562, 2011.

# Abstração em Robótica

---

- Middleware + linguagem de agente
  - Maioria dos sistemas robóticos multiagentes
    - Comunicação entre os agentes desenvolvendo software customizado
  - Uso de sistemas multiagentes
    - Software resultante é mais reusável, escalável e flexível
    - Mantém a robutez e os requisitos de modularidade
    - Oferece padrão de comunicação entre agentes
    - Permite sistema tolerante a falhas de alto nível
  - Abstração dos detalhes de hardware

P. Iñigo-Blasco, F. Diaz-del-Rio, M. C. Romero-Ternero, D. Cagigas-Muñiz e S. Vicente-Diaz, "Robotics software frameworks for multi-agent robotic systems development," *Robotics and Autonomous Systems*, vol. 60, pp. 803-821, 2012.

# Plano de Pesquisa

---

# Motivação

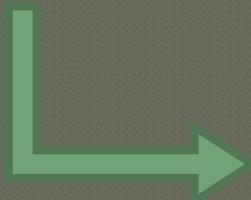
---

Aumento do uso de robôs

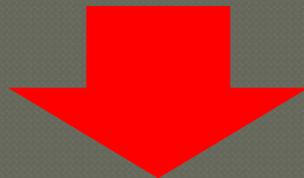


Aumento da expectativa  
quanto ao funcionamento

Sistemas múltiplos robôs



Potencial Robustez via  
redundância  
Assumir tarefas



Necessidade de maior abstração para descrever  
aplicação de tolerância a falhas

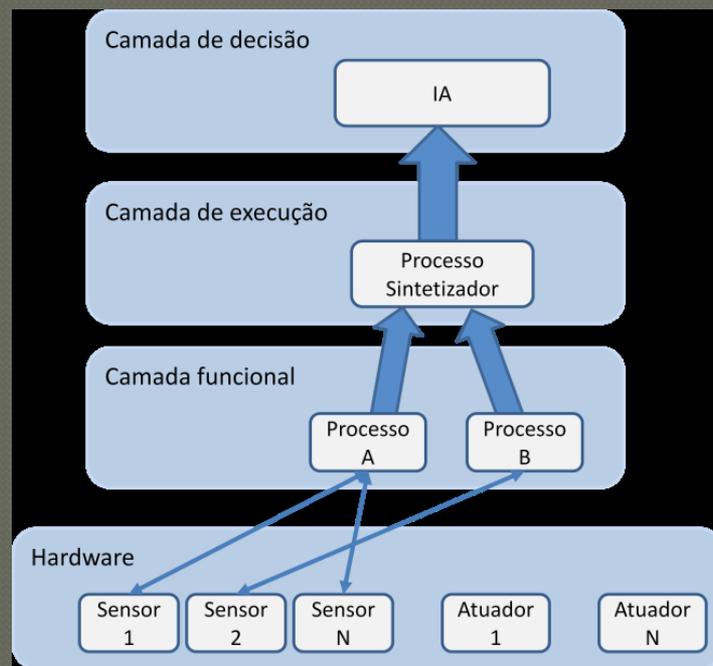
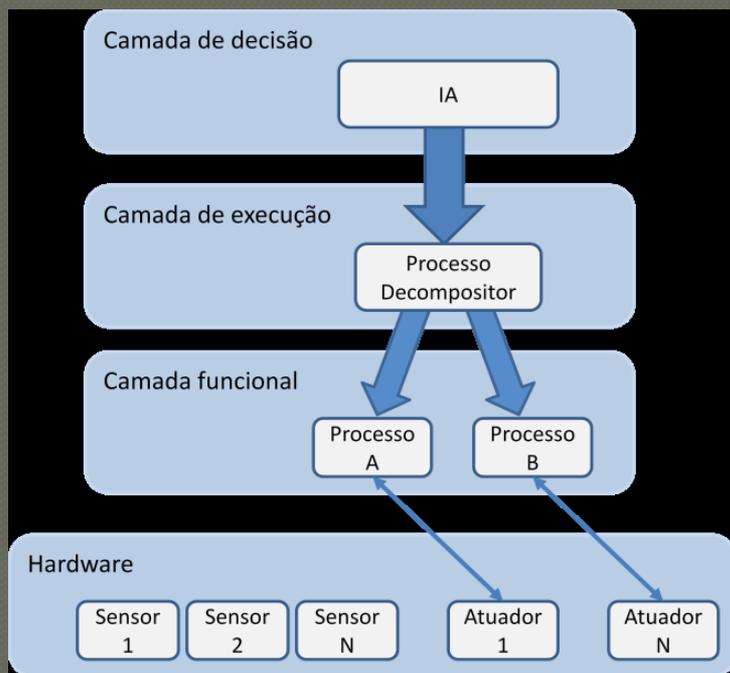
# Proposta

---

- ◉ Sistema de tolerância a falhas descrito em linguagem de alto nível
  - Abstração de detalhes de hardware
  - Cooperação entre múltiplos robôs

# Objetivos

Como? Integração do framework e sistema multiagente  
Propor processos de software que possibilitem o trânsito de informação entre as camadas de software



# Objetivos

---

Infraestrutura para permitir a comunicação entre agentes

Troca de mensagens entre a camada de alto nível dos robôs

Integrar middleware e sistema multiagente

# Objetivos

---

Técnicas de detecção e recuperação de falhas  
utilizando sistema multiagente

Múltiplos robôs

Falhas físicas  
Falhas de interação

# Cronograma e Atividades

---

## 1. Estudo do middleware

- Estudo das principais características e recursos de comunicação oferecidos pelo middleware
- Ao final desta etapa espera-se ter conhecimento detalhado sobre o middleware e os pontos estratégicos a serem trabalhados para viabilizar o método proposto

# Cronograma e Atividades

---

## 2. Estudo do sistema multiagentes

- Estudo das características do sistema multiagente, da sua infraestrutura e sistema de troca de mensagens entre agentes focando na integração com o middleware
- Ao final desta etapa espera-se ter o conhecimento necessário para efetuar as mudanças necessárias à integração;

# Cronograma e Atividades

---

3. Integração do middleware com sistema multiagente
  - Baseado nos estudos das duas atividades anteriores, realizar a integração middleware com o sistema multiagente
  - Desenvolver uma pequena aplicação multiagente para validar a implementação e avaliação de desempenho.

# Cronograma e Atividades

---

## 4. Definição de situações de falhas

- Análise das características dos diferentes tipos de robôs disponíveis para este estudo
- Definir um conjunto de possíveis falhas, sejam elas físicas ou de interação
- Desenvolver um método cooperativo de detecção, isolamento, e recuperação das falhas entre múltiplos robôs

# Cronograma e Atividades

---

## 5. Desenvolvimento de processos sintetizadores

- Desenvolver os processos da camada de execução capazes que receber informações de sensores por meio de mensagens oriundas de um ou mais processos da camada funcional
- *Sintetizar* tais informações gerando um percepção sobre si mesmo ou sobre o ambiente e enviar para o processo na camada de decisão no qual o sistema multiagente está sendo executado

# Cronograma e Atividades

---

## 6. Desenvolvimento de processos decompositores

- Desenvolver os processos da camada de execução capazes de receber ações por meio de mensagens oriundas da camada de decisão e decompor tais ações em um conjunto ou sequência de comandos, enviados para os processos da camada funcional

# Cronograma e Atividades

---

7. Desenvolvimento de estudo de caso
  - Validação e avaliação dos métodos de tolerância a falhas em múltiplos robôs
  
8. Revisão do Estado da Arte
  - Revisão, de forma contínua, de publicações científicas relacionadas com este trabalho

# Cronograma e Atividades

---

9. Seminário de Andamento
  - Revisão das atividades listadas
  - Apresentação sobre o andamento de tais atividades e pesquisas relacionadas
10. Escrita de Artigos
  - Escrita de artigos relacionados aos métodos, algoritmos e técnicas empregados no trabalho
11. Escrita da Dissertação
12. Defesa da Dissertação

