

Robô Autônomo para Coleta de Cubos



Luis Felipe Sena

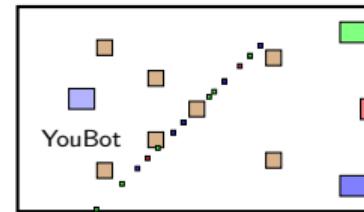


Agenda

Tempo	Conteúdo
0–3 min	Problema e Desafios – Definição da tarefa e restrições
3–6 min	Percepção – LIDAR, Câmera RGB e Rede Neural (MobileNetV3)
6–9 min	Navegação Fuzzy – Regras linguísticas e funções de pertinência
9–12 min	Planejamento A* – Grade de ocupação e busca de caminhos
12–15 min	Arquitetura e Demo – Máquina de estados e demonstração

O Problema

- **Objetivo:** Coletar 15 cubos coloridos (vermelho, verde, azul)
- **Ação:** Depositar cada cubo na caixa de cor correspondente
- **Plataforma:** KUKA YouBot com rodas Mecanum (omnidirecional)
- **Ambiente:** Arena 7m × 4m com obstáculos fixos



Restrições do Projeto

Obrigatórios:

- Usar **Rede Neural** (MLP ou CNN) para classificação
- Usar **Lógica Fuzzy** para controle de navegação
- Sensores: **LIDAR + Câmera RGB**

Proibidos:

- **GPS** – Robô deve usar odometria
- **Teleoperação** – Completely autônomo
- **Informações privilegiadas** – Sem acesso direto à posição dos cubos

Desafio Principal: Navegação autônoma em ambiente com obstáculos dinâmicos (cubos podem ser empurrados) usando apenas sensores locais

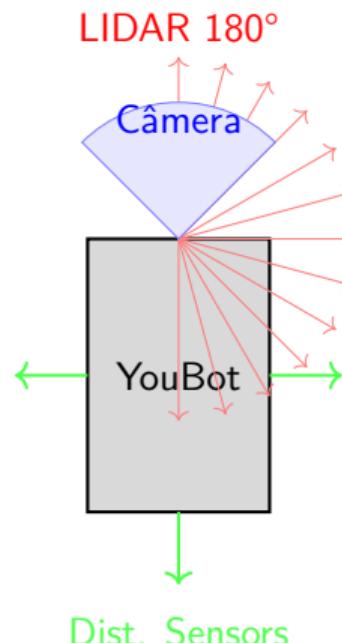
Sensores do YouBot

LIDAR (180° FOV)

- 180 feixes de laser
- Alcance: 0.1m – 5.0m
- Atualização: 32Hz
- **Uso:** Detecção de obstáculos

Câmera RGB

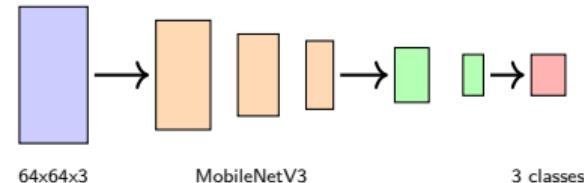
- Resolução: 320×240
- API de Reconhecimento Webots
- **Uso:** Detecção e classificação de cubos



Rede Neural – MobileNetV3-Small

Arquitetura CNN

- Backbone: MobileNetV3-Small (pré-treinado ImageNet)
- Classificador customizado: 256 → 3 classes
- Entrada: 64×64 RGB normalizada
- Saída: Probabilidades (vermelho/verde/azul)



Fallback: Heurísticas HSV
quando confiança < 0.5

Treinamento (Transfer Learning)

- Fase 1: Congelar backbone, treinar apenas classificador
- Fase 2: Fine-tuning com learning rate reduzido
- Acurácia validação: **99.4%**

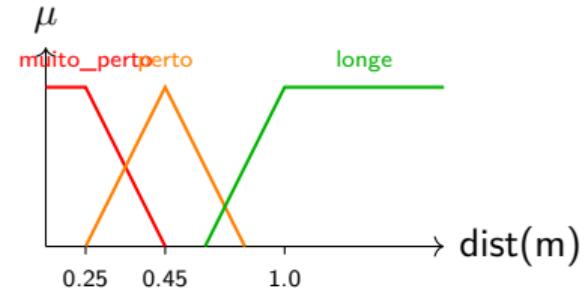
Lógica Fuzzy – Conceito

Por que Fuzzy?

- Controle suave e contínuo (sem transições bruscas)
- Lida com incerteza dos sensores
- Regras linguísticas intuitivas

Variáveis Linguísticas

- **Distância:** muito_perto, perto, longe
- **Ângulo:** pequeno, médio, grande
- **Velocidade:** reverso, lento, normal, rápido

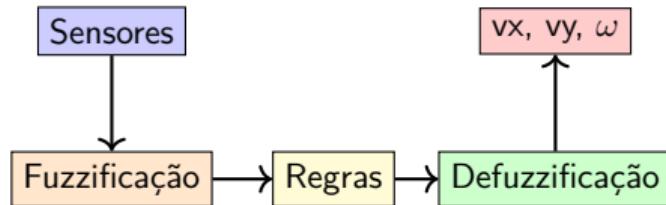


Funções de pertinência para distância ao obstáculo

Regras Fuzzy de Navegação

Regras de Desvio

- SE obstáculo_muito_perto ENTÃO reverso + strafe
- SE obstáculo_perto ENTÃO reduzir_velocidade
- SE lateral_bloqueado ENTÃO strafe_oposto



Defuzzificação: Centróide ponderado

Regras de Alinhamento

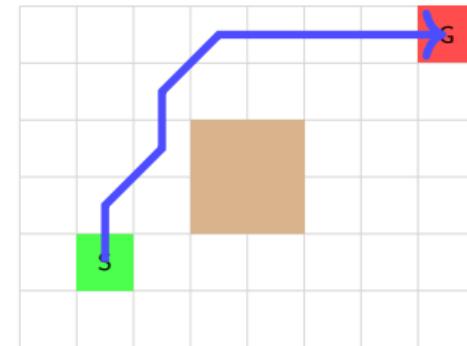
- SE ângulo_grande ENTÃO apenas_rotacionar
- SE ângulo_médio ENTÃO avançar_lento + rotacionar
- SE ângulo_pequeno ENTÃO avançar_rápido

Algoritmo A* – Visão Geral

Função de Custo

$$f(n) = g(n) + h(n)$$

- $g(n)$: Custo do início até n
- $h(n)$: Heurística (Manhattan) até objetivo
- $f(n)$: Custo total estimado



Caminho A*

Grade de Ocupação

- Célula: 12cm × 12cm
- Arena: 58 × 33 células
- Atualizada por raycasting (LIDAR)

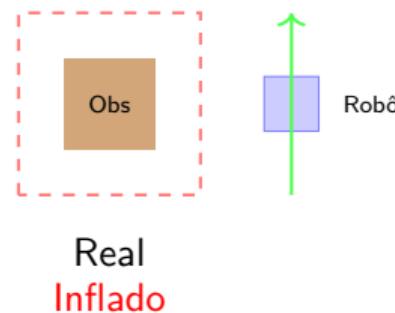
Inflação de Obstáculos

Por que inflar?

- Robô não é um ponto
- YouBot: 58cm × 38cm
- Margem de segurança: **30cm**

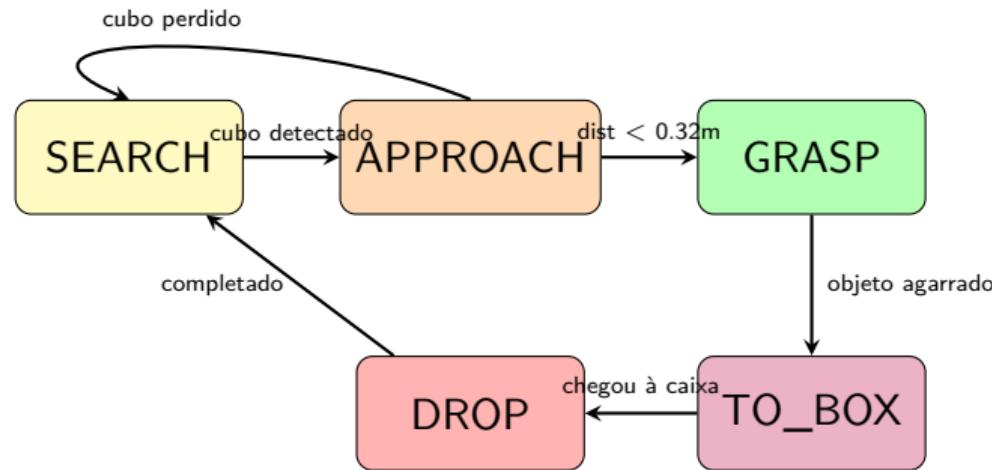
Obstáculos Conhecidos

- 7 caixas de madeira (fixas)
- 3 caixas de depósito (destinos)
- Paredes da arena



A inflação garante que o centro do robô nunca colida

Máquina de Estados Finita

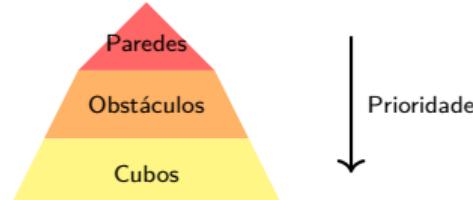


Fluxo principal de execução

Hierarquia de Prioridades

Regras de Segurança (Top → Bottom)

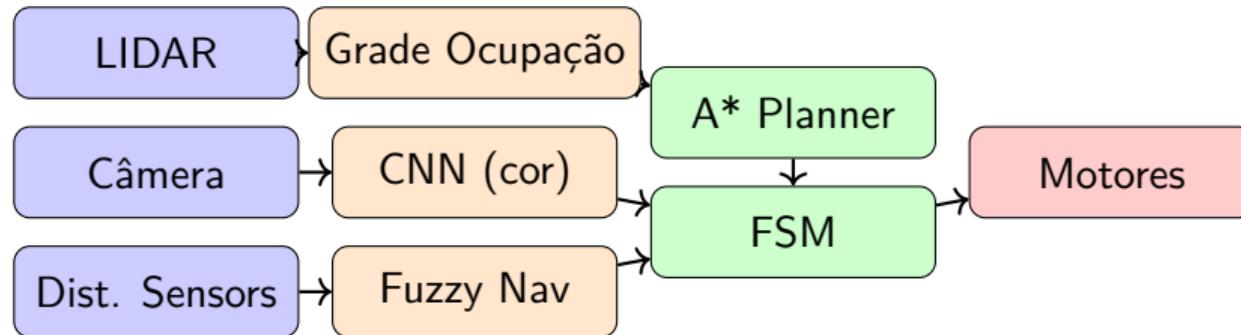
1. **NUNCA** colidir com paredes ou obstáculos de madeira
2. Evitar empurrar cubos pequenos (podem ficar inacessíveis)
3. Manter progresso em direção ao objetivo



Recuperação de Travamento

- Detecção de oscilação (posição fixa por 12s)
- Escape em 4 fases: strafe puro, diagonal
- Rota via corredores seguros (bordas da arena)

Integração dos Componentes



Pipeline completo: Sensores → Processamento → Planejamento → Atuação

Resultados

Métricas da Rede Neural

- Acurácia: 99.4%
- Formato: ONNX (portável)
- Inferência: < 10ms por frame

Navegação

- A* encontra caminhos válidos consistentemente
- Fuzzy evita colisões em tempo real
- Recuperação de travamento funcional

Desafios Encontrados

- Drift de odometria ao longo do tempo
- Cubos empurrados bloqueando caminhos
- Coordenação braço-navegação

Soluções

- Sincronização periódica com ground truth
- Rotas via corredores seguros
- Máquina de estados bem definida

Demonstração

DEMO

Execução do robô coletando cubos no Webots

Conclusão

Contribuições

- Integração bem-sucedida de CNN + Fuzzy + A*
- Sistema de navegação robusto e reativo
- Arquitetura modular e extensível

Trabalhos Futuros

- SLAM para melhor localização
- Planejamento com obstáculos dinâmicos
- Otimização da ordem de coleta

Obrigado!

Perguntas?

Referências

- Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*
- Zadeh, L. A. (1965). *Fuzzy Sets – Information and Control*
- Howard, A., et al. (2019). *Searching for MobileNetV3*
- Elfes, A. (1989). *Using Occupancy Grids for Mobile Robot Perception and Navigation*