



Robótica Móvel Inteligente

Localização

Anna Helena Reali Costa

Laboratório de Técnicas Inteligentes

Escola Politécnica da USP

Robótica Móvel: evolução histórica

Tarefa fundamental:
NAVEGAÇÃO

1950



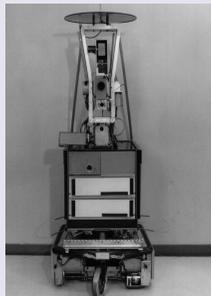
1997



1998



1969



1997



2001



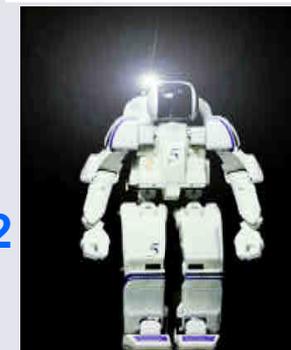
1977



1995



2002



1989



1992



1993



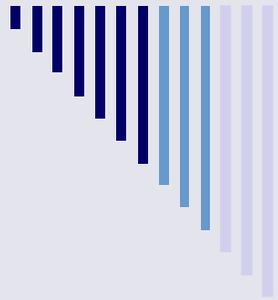
2011



Navegação

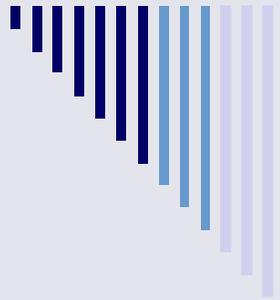
- Onde estou?
- Onde já estive?
- Para onde vou?





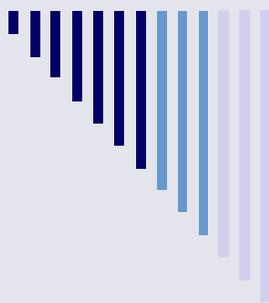
Navegação

- Para responder a estas questões, o robô precisa:
 - Ter um modelo do ambiente
 - Perceber (sensoriar) e analisar o ambiente
 - Encontrar sua posição no ambiente
 - Planejar e executar o movimento

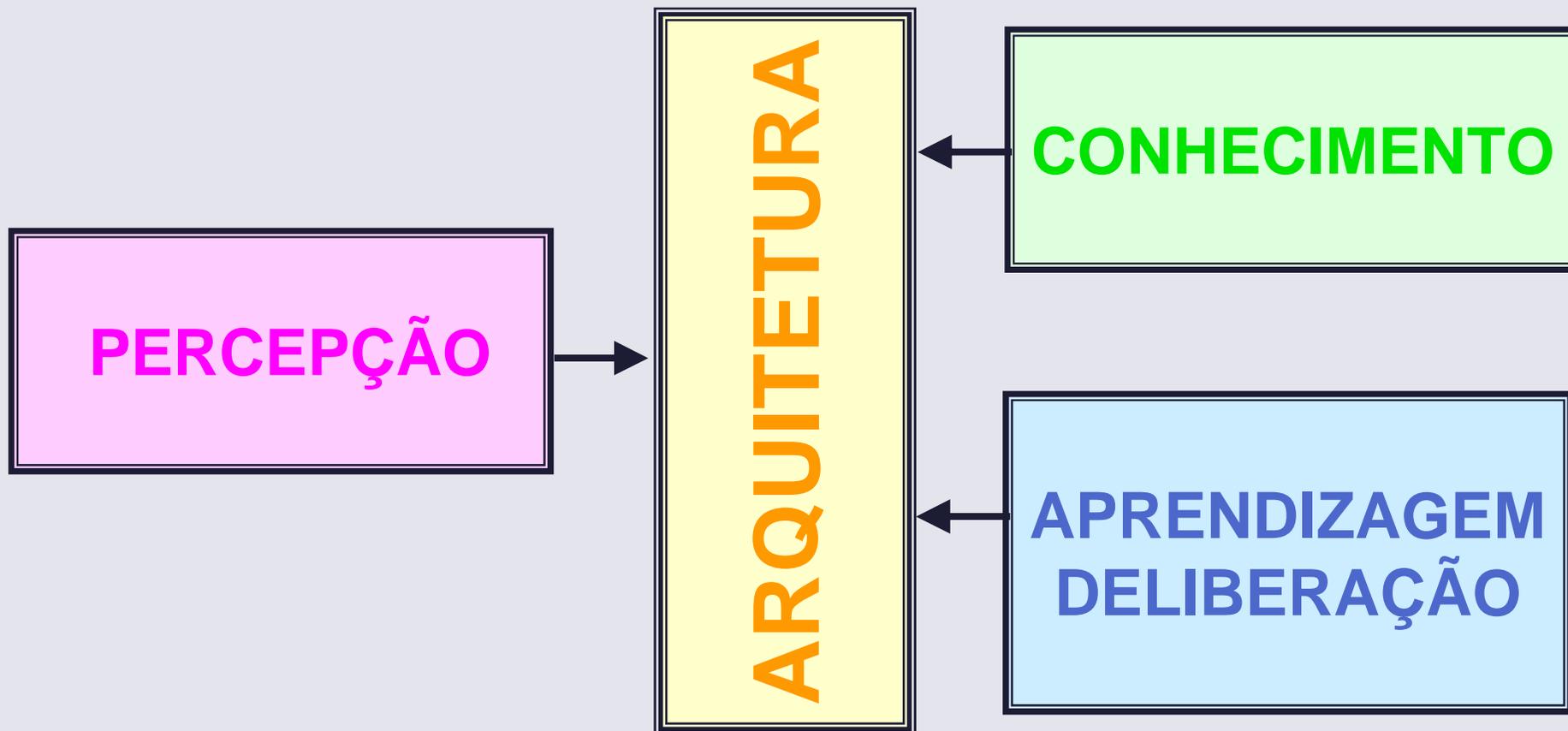


Projeto de Robôs Inteligentes

- Desafios de engenharia:
 - **Arquitetura flexível**
 - **Percepção poderosa**
 - Aquisição e representação adequada de **conhecimento**
 - **Deliberação e aprendizagem** para decidir sob incerteza



Máquina Inteligente: Módulos Essenciais





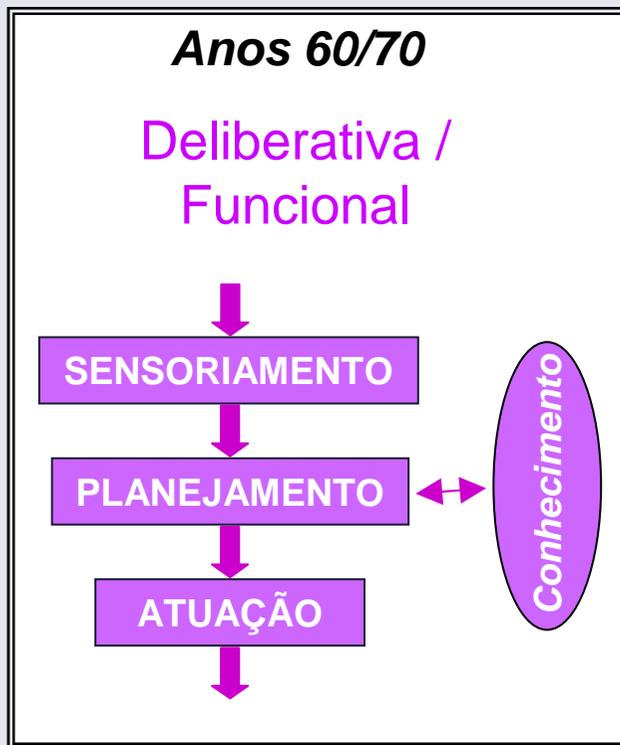
ARQUITETURA

ARQUITETURA

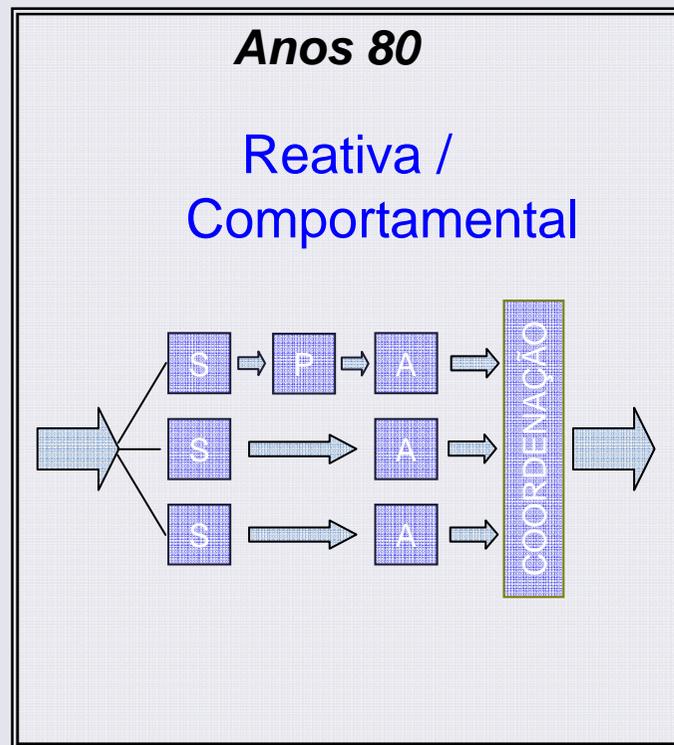
Quais são os módulos básicos e como organizá-los?

Como deve ser a interação entre eles?

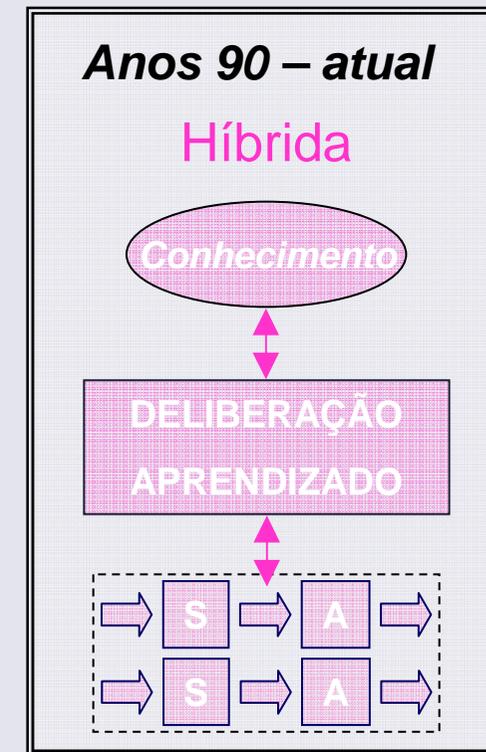
Arquiteturas – evolução



Fikes & Nilsson (1971)

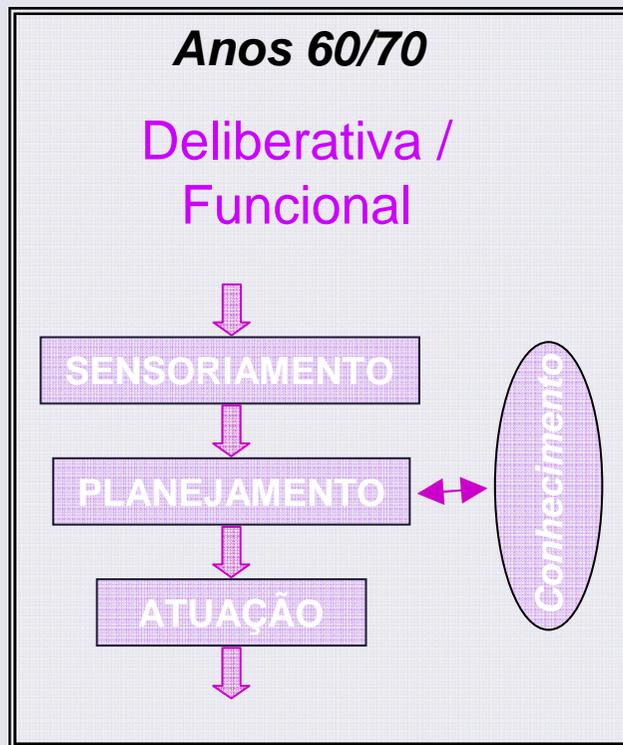


Brooks (1986)

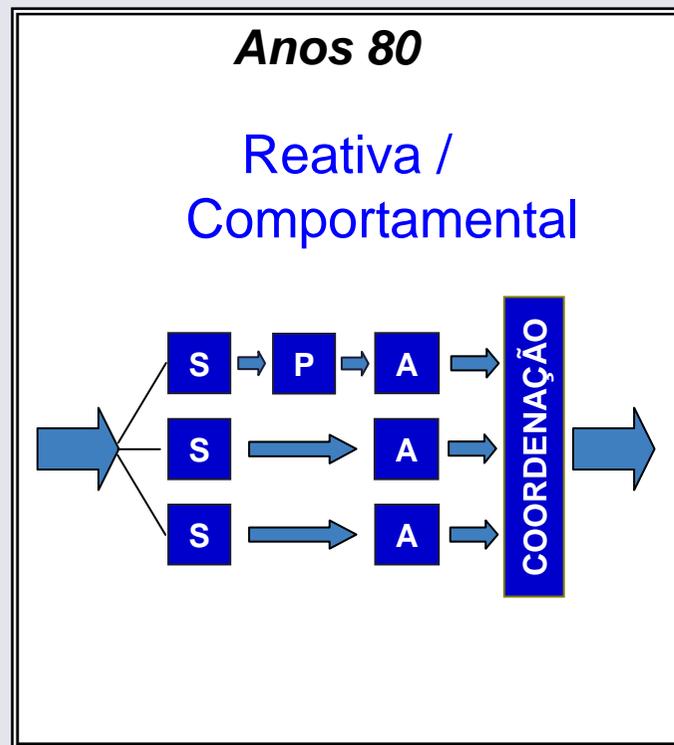


Murphy (2000)

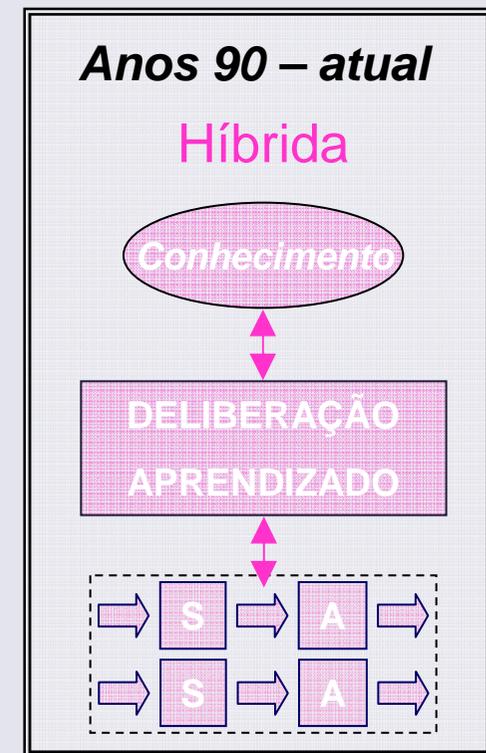
Arquiteturas – evolução



Fikes & Nilsson (1971)

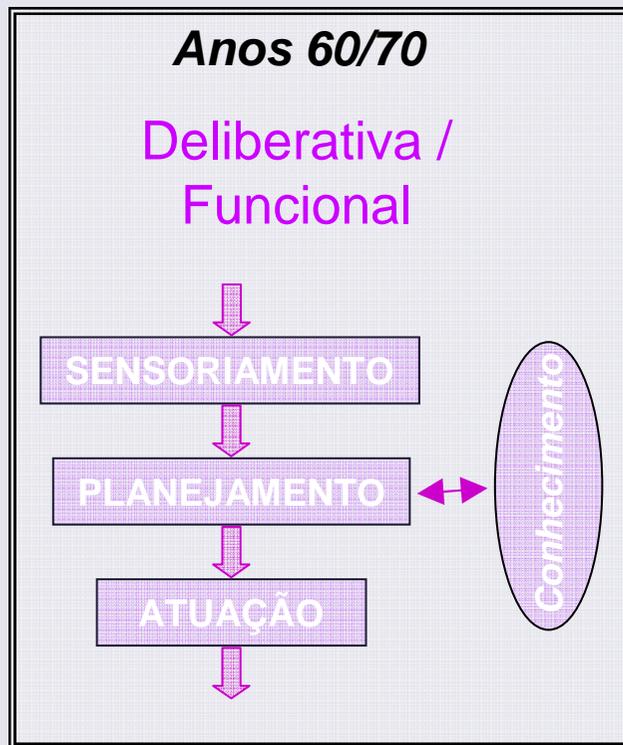


Brooks (1986)

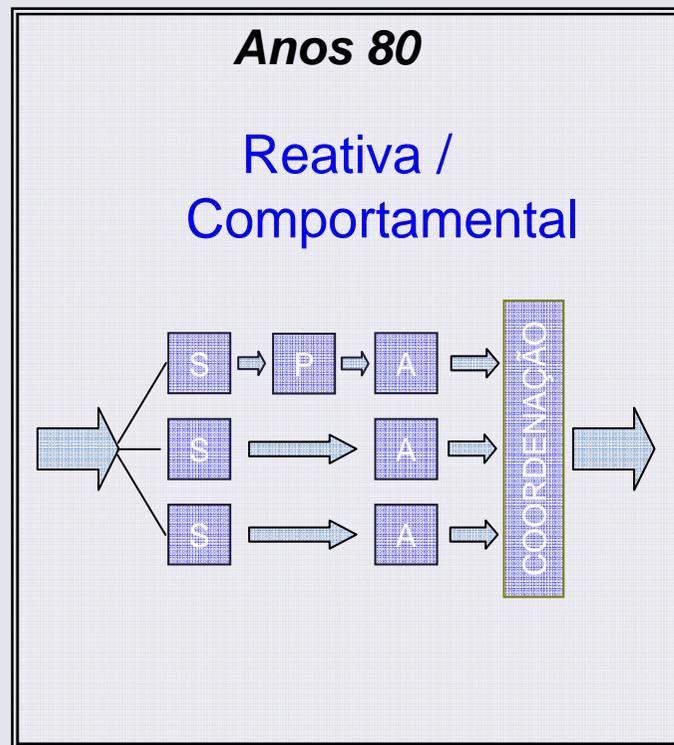


Murphy (2000)

Arquiteturas – evolução



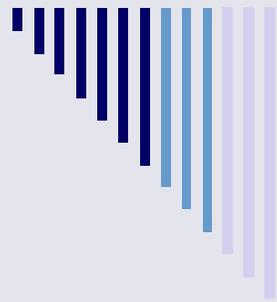
Fikes & Nilsson (1971)



Brooks (1986)



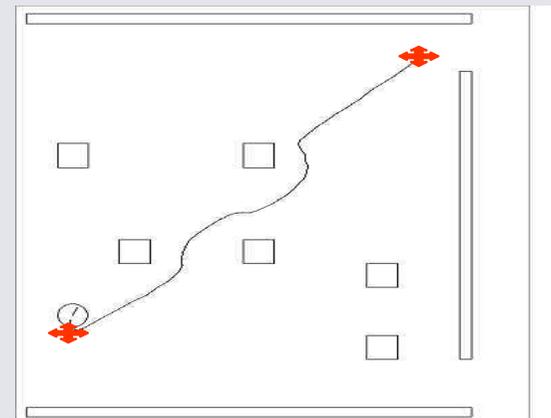
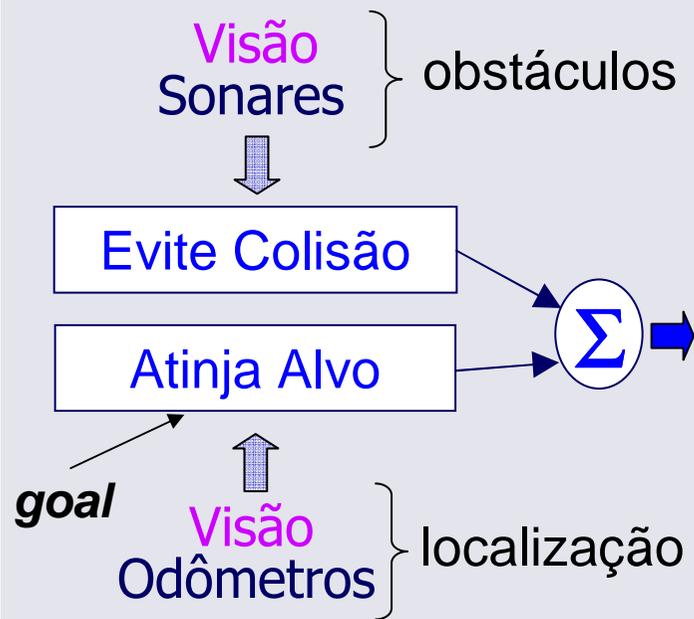
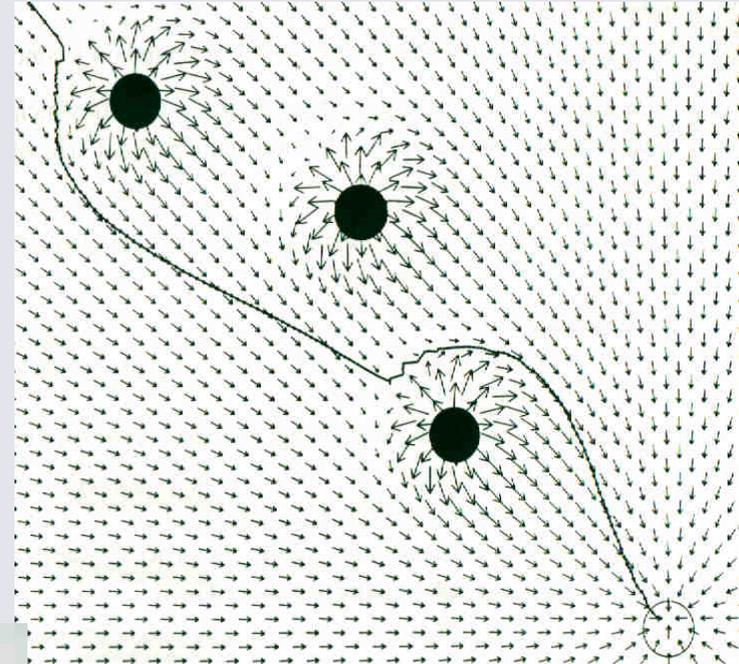
Murphy (2000)



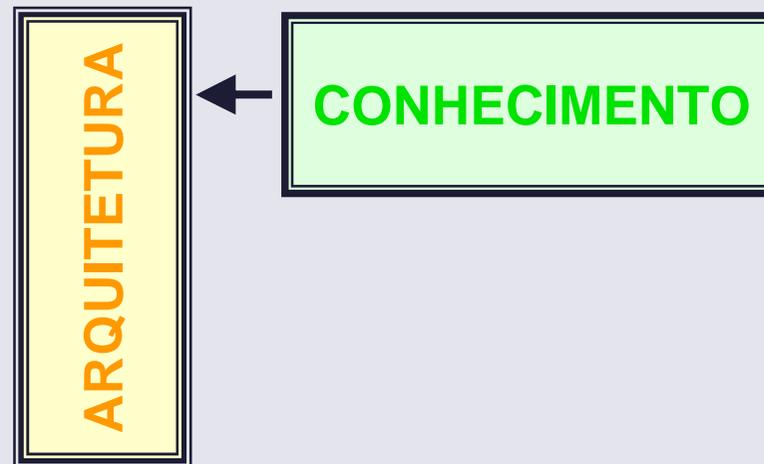
Exemplo: REACT

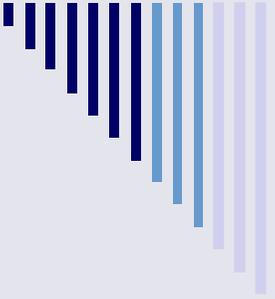
Arquitetura Comportamental

- Comportamentos **reativos**
 - Codificação contínua
 - Campos potenciais
- Coordenação **cooperativa**



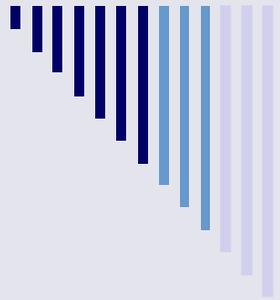
Conhecimento na Navegação





Que tipo de conhecimento?

- Informação de um contexto, organizado de forma a ser prontamente usado
- Entendimento, consciência ou familiaridade conseguida por ensino ou experiência
- Estruturas físicas que têm correlação com os aspectos do ambiente e que permitem predições para o sistema

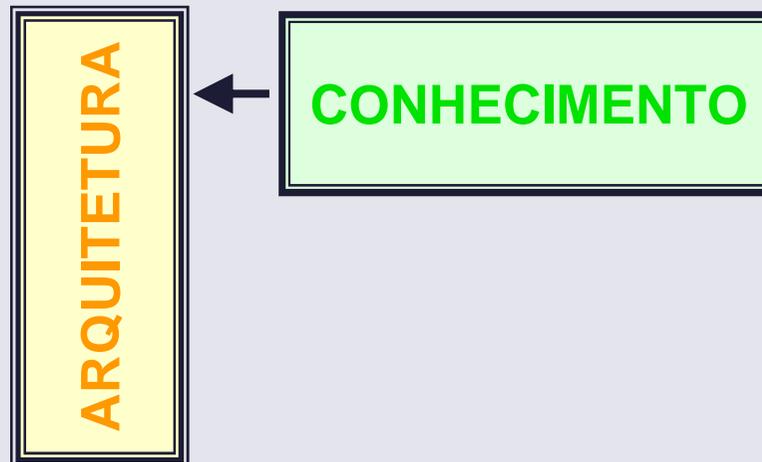


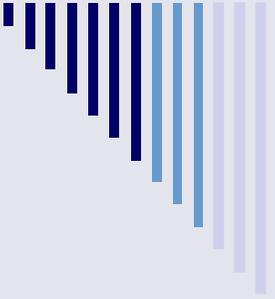
Representação e controle

- Representação é a forma como codificamos **conhecimento** no sistema de controle
- O papel de diferentes representações varia de acordo com a arquitetura de controle:
 - Reativa: nenhuma representação
 - Comportamental: nenhuma ou pouca / distribuída
 - Híbrida: usa (várias)

Conhecimento na Navegação

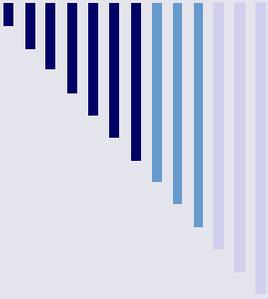
Onde estou? Como é o mundo?
Para onde devo ir?





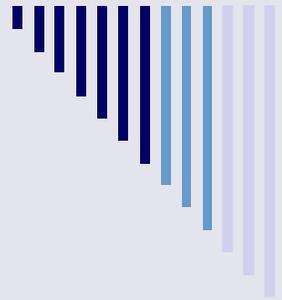
Por que usar mapas?

- Para um robô, nem sempre é possível atingir o objetivo apenas com navegação reativa (que só reage a estímulos do ambiente)
- A tarefa de planejamento leva a alguma representação do ambiente e da tarefa
- **Mapa:** representação do conhecimento sobre o mundo



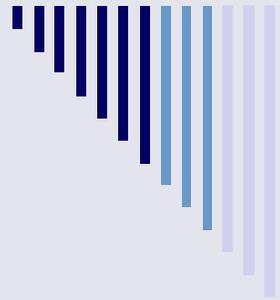
Como obter um mapa?

- Fornecido pelo Projetista
- Construído autonomamente pelo robô
 - Normalmente são necessários vários tipos de mapas (diferentes representações), em diferentes níveis de abstração



Por que construir mapas autonomamente?

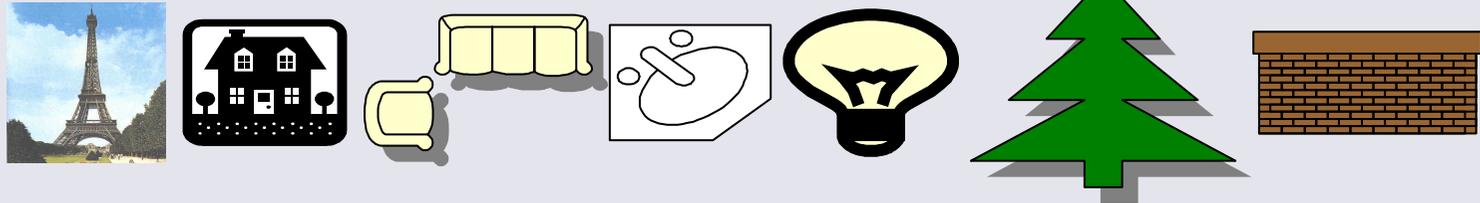
- Dificuldade de construir “na mão” mapas adequados às características sensoriais do robô
- Ambientes do mundo real são dinâmicos: necessidade de constantes atualizações
- “Mapear” na mão dá trabalho: deixe o robô fazer isso (cartografia)



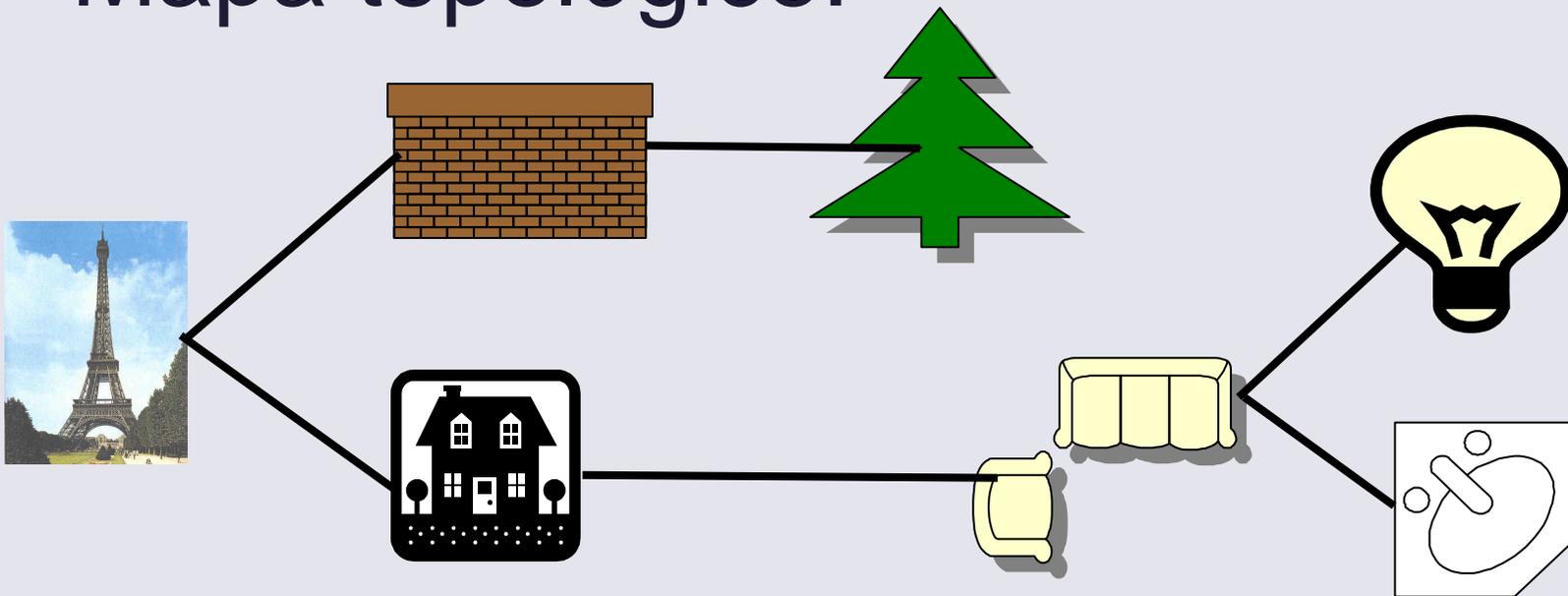
Tipos de mapas

- Escolha depende: do ambiente, da tarefa, dos sensores do robô e sua capacidade de armazenamento e manipulação
- Dois tipos básicos: métrico/geométrico e topológico

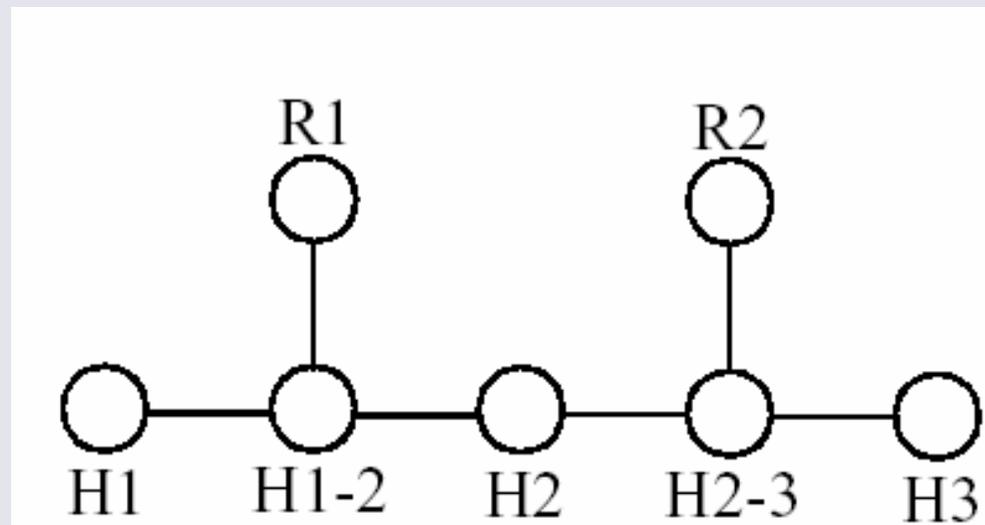
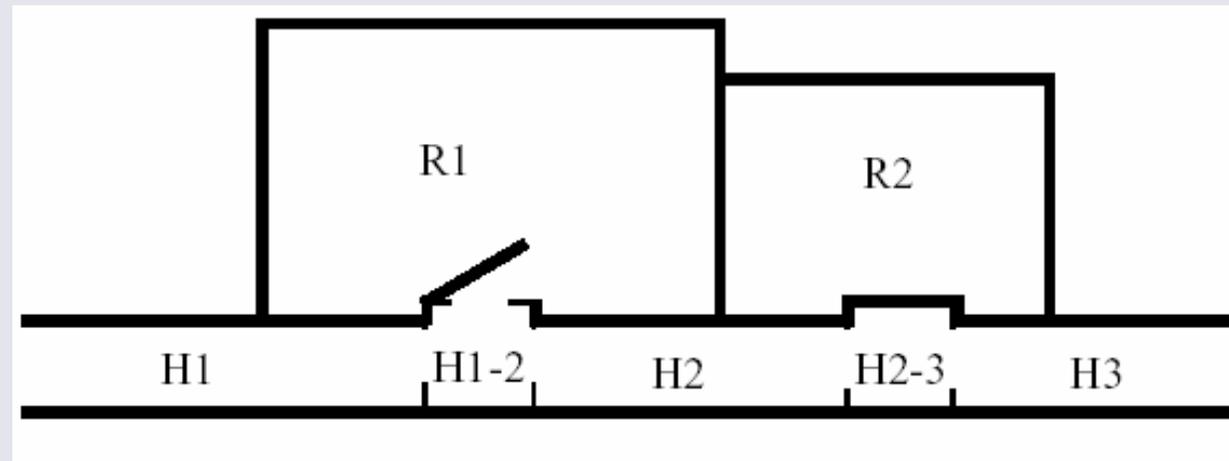
Marcos / locais reconhecíveis pelos sensores externos:



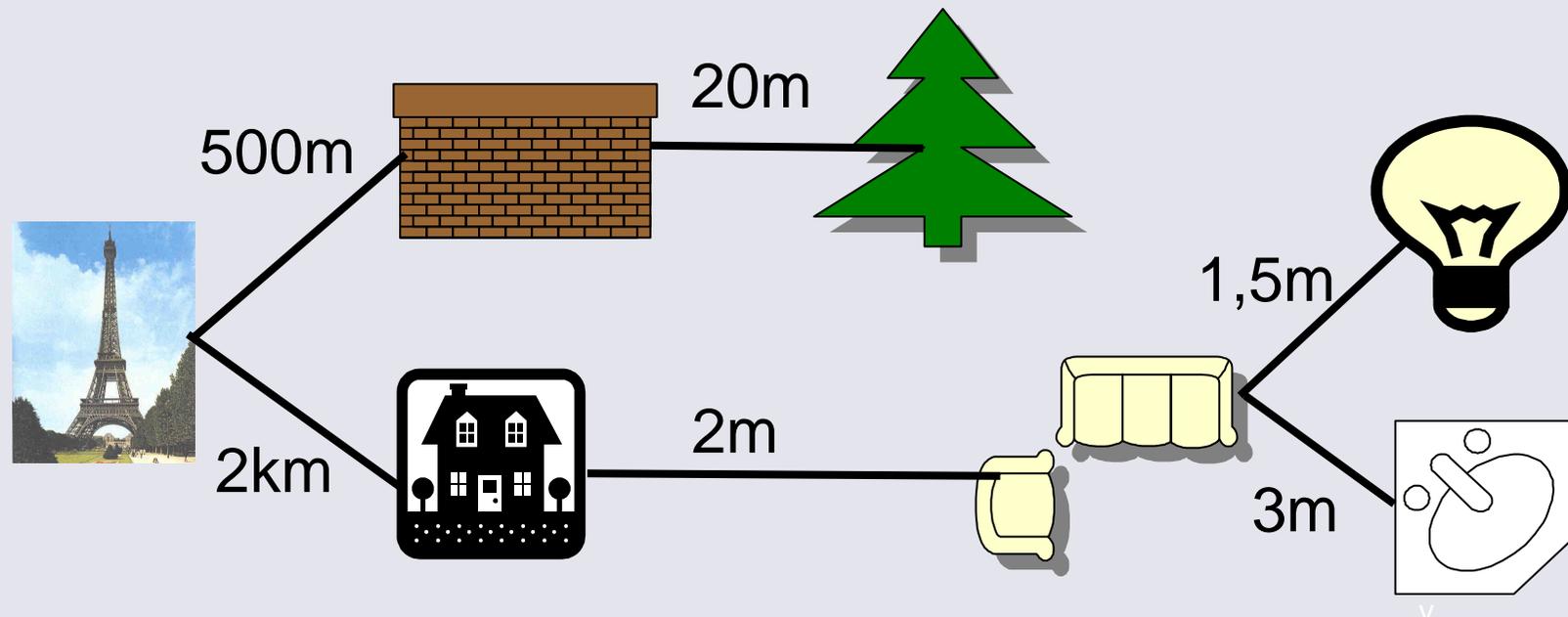
Mapa topológico:



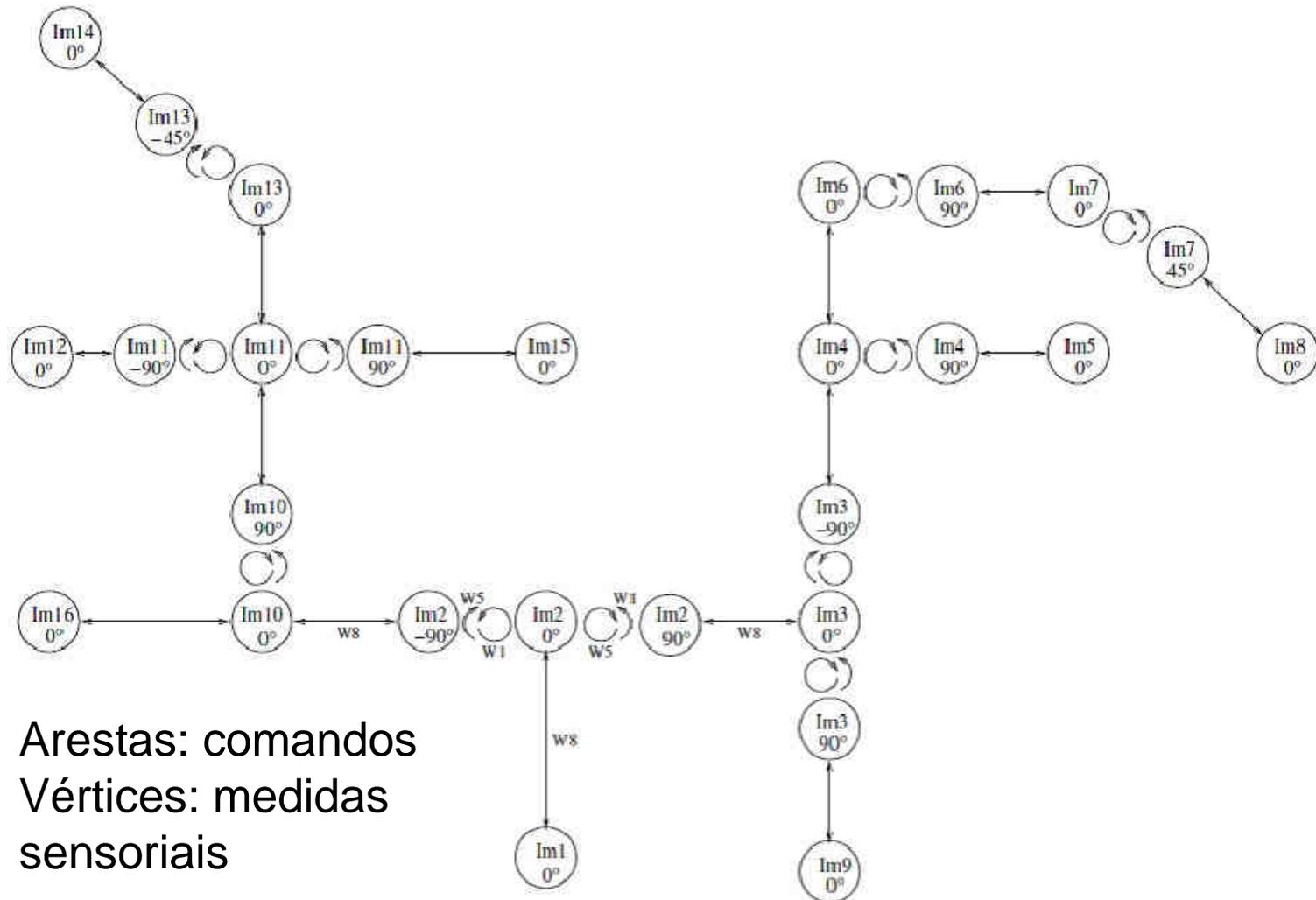
Mapa topológico: exemplo

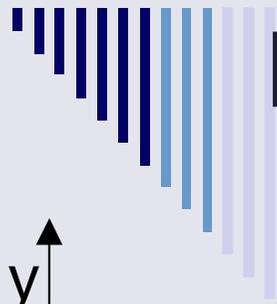


Mapa topológico híbrido:

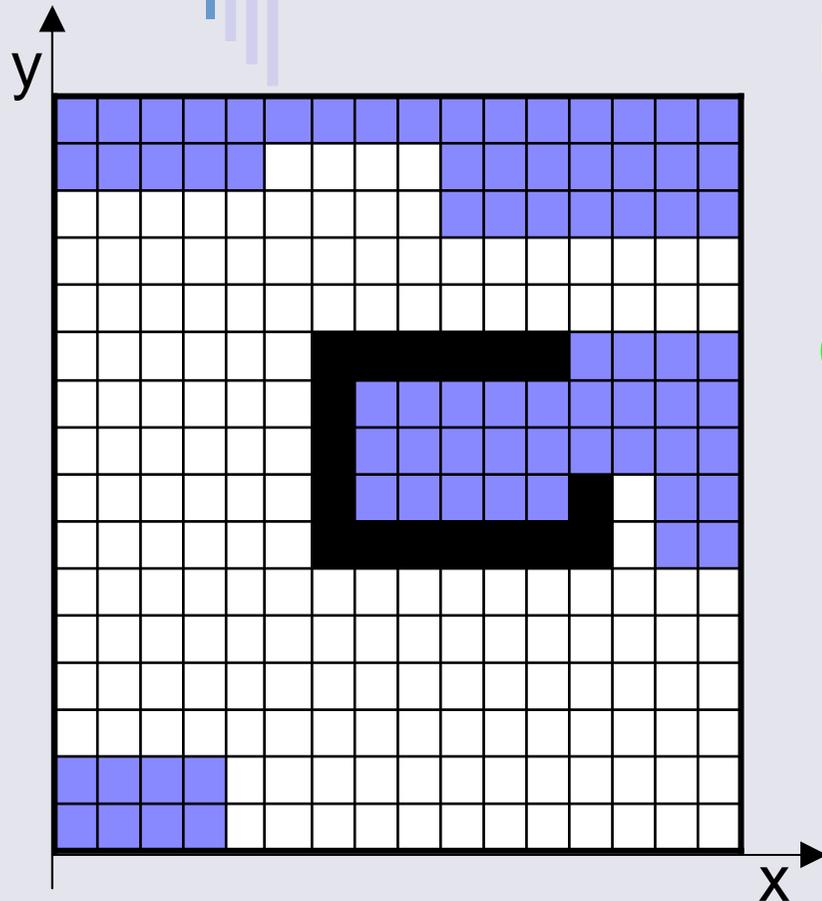


Mapa topológico híbrido: exemplo





Mapa métrico: grade de ocupação

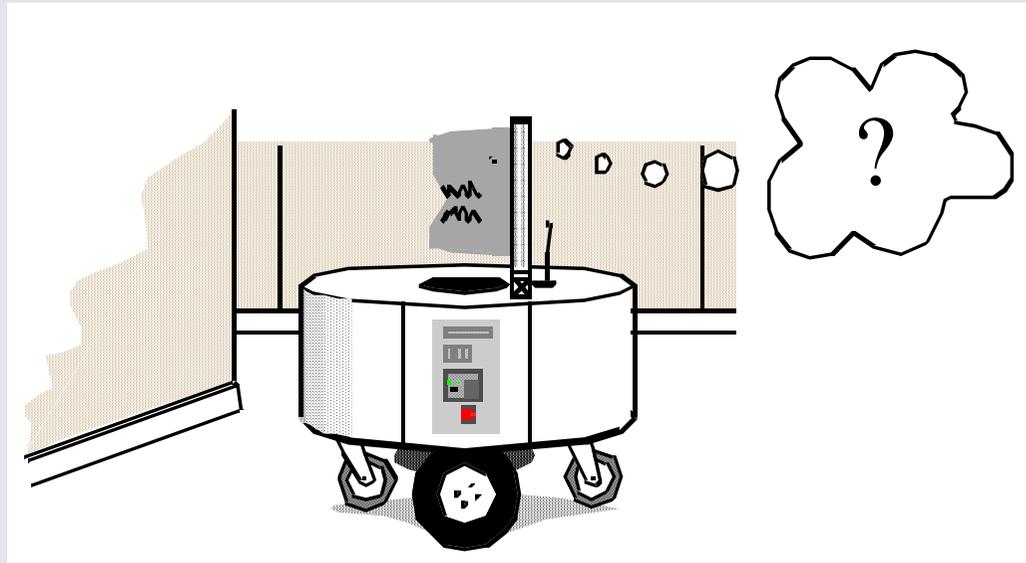


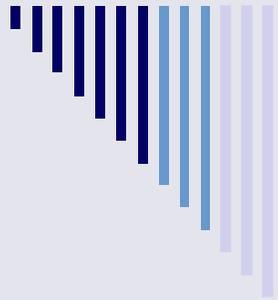
Branca: vazia Preta: ocupada
Cinza: desconhecida

- Grade de Ocupação: diferentes níveis de ocupação (prob. ocupação)
- 😊 Simples, generalidade de grade regular
- 😞 Fidelidade depende do tamanho da célula, requer muita memória

Localização

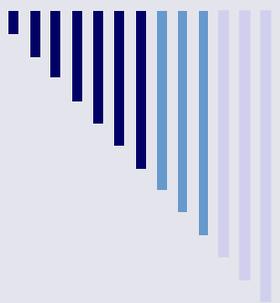
- A localização de robôs móveis está relacionada com a tarefa de responder a pergunta: **Onde o robô está?**





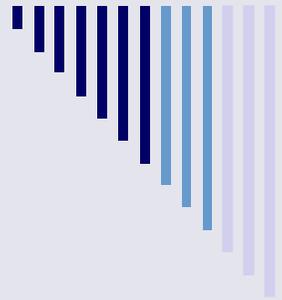
Tipos de Localização

- **Localização forte:** localizar o robô em relação a alguma representação global do ambiente.
- **Localização fraca:** determinar se regiões do ambiente foram vistas antes ou não.



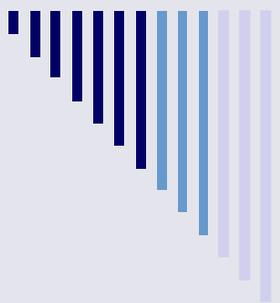
Tipos de Localização

- **Global:** estimar a posição do robô sem conhecer uma estimativa a priori de sua posição.
- **Local:** estimar a posição do robô conhecendo uma estimativa a priori de sua posição, ou seja, continuamente refinar a estimativa da posição.



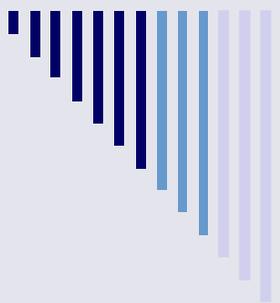
Tipos de Localização

- Estimar a posição do robô:
 1. **Dead reckoning** (local)
 2. **Baseado em sensores externos** (utiliza mapa)
 3. **Dead reckoning + sensor** → abordagem probabilística é mais efetiva e popular



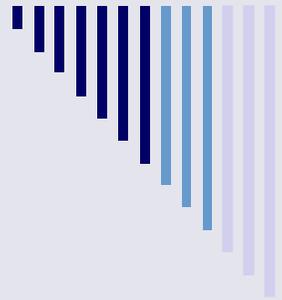
1. Dead Reckoning

- Determinação da localização do robô móvel através da observação e integração consecutiva do movimento das rodas – sem utilizar referência do mundo externo.
 - Muito usada em arquiteturas reativas
 - Abordagem local
- Odômetros - sensores que medem a distância percorrida através da contagem do número de rotação das rodas do robô.

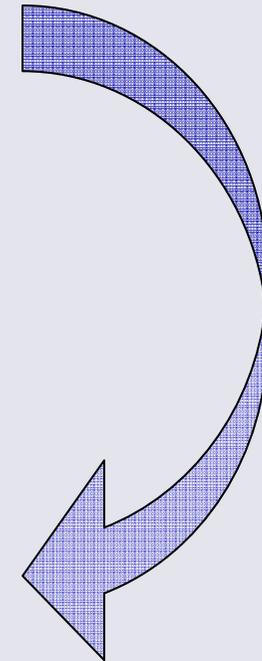
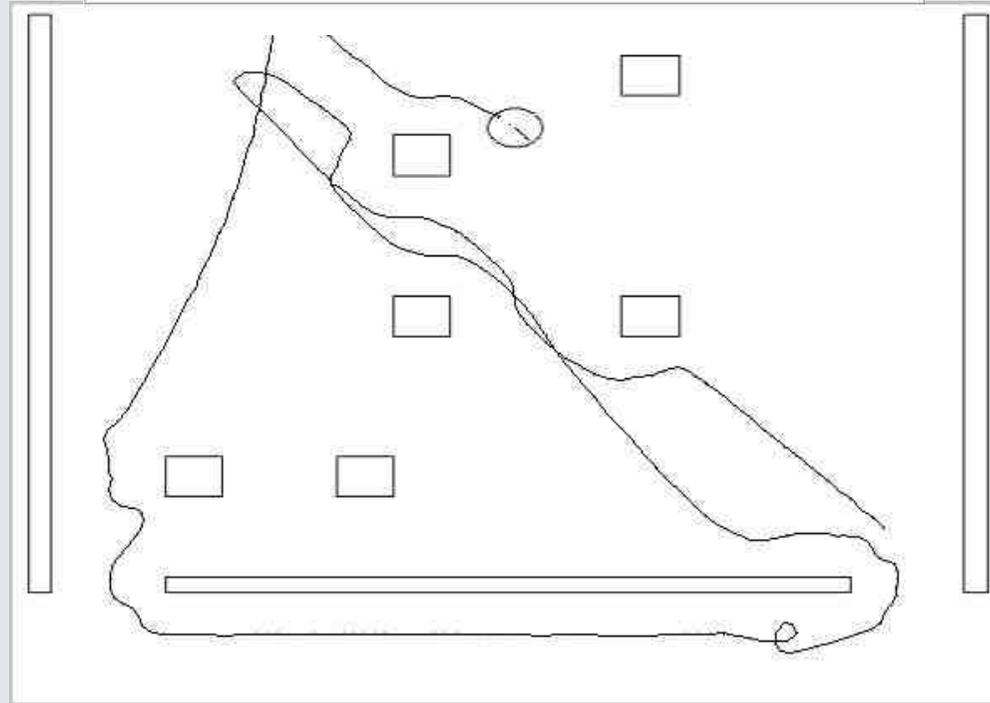
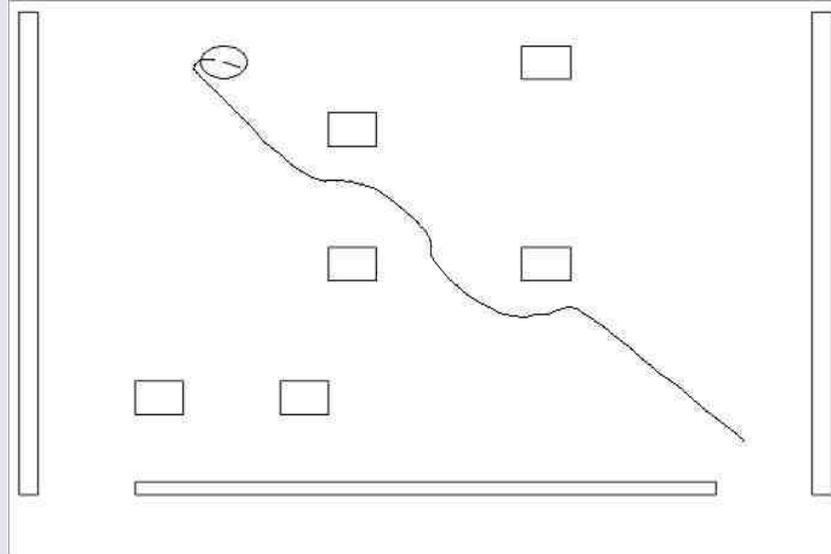


1. Dead Reckoning – erros

- Integração de informação sobre movimentos incrementais resulta em acumulação de erros.
- Erros devido a irregularidades no terreno, deslizamento das rodas, desalinhamento das rodas, diâmetro desigual das rodas, etc.
- Pode ser usado para curtas distâncias.



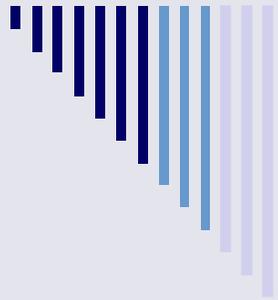
Erros de odometria com a REACT





2. Localização baseada em marcos usando sensores externos

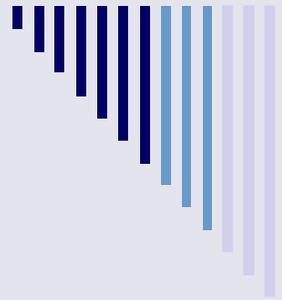
- Localização global
- Marcos são percepções sensoriais dependentes da posição, e servem como referência para a localização.
- Técnicas de localização baseadas em marcos utilizam-se de solução de problemas geométricos envolvendo restrições relacionadas às posições dos marcos no ambiente.



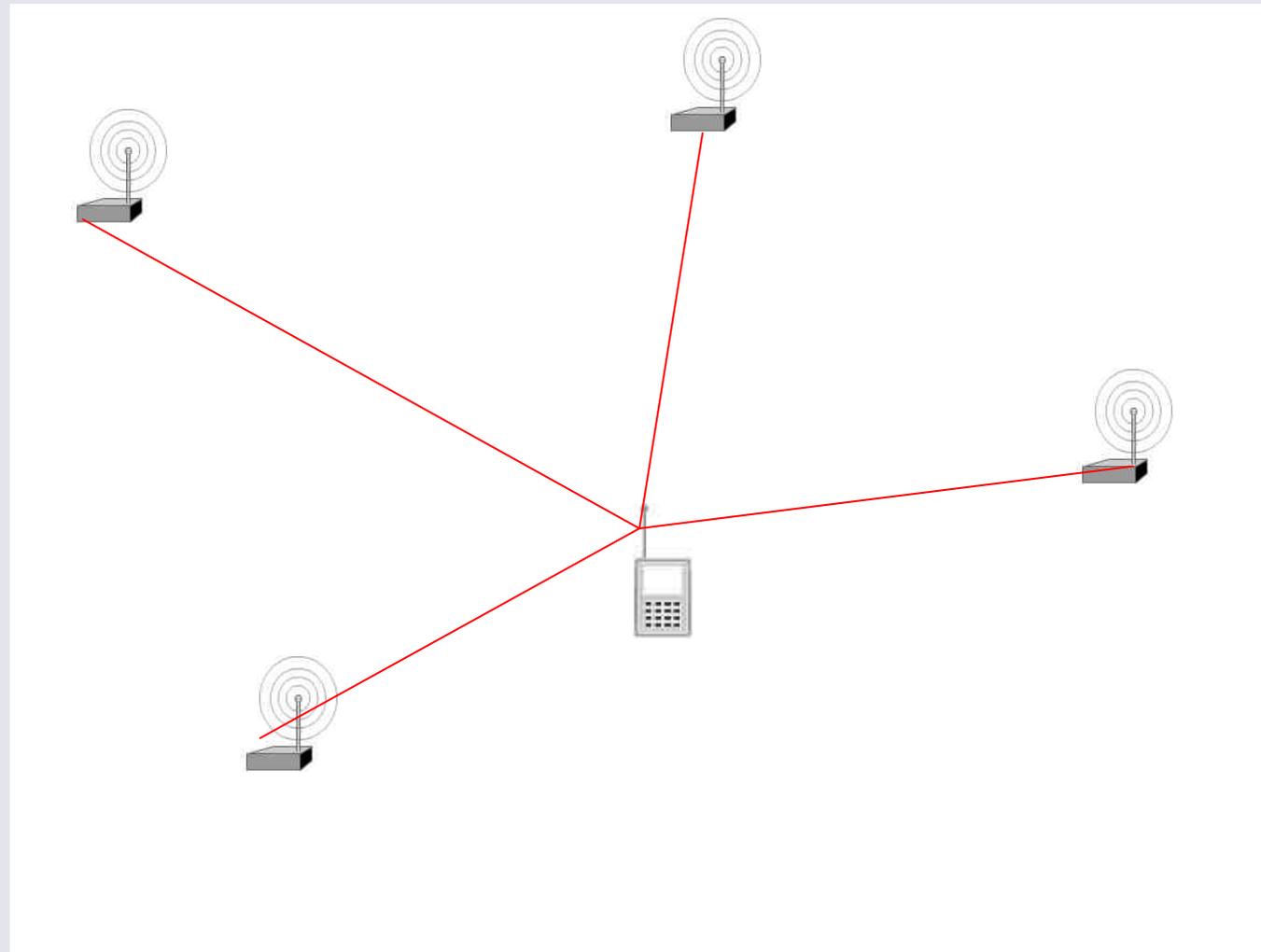
Tipos de Marcos

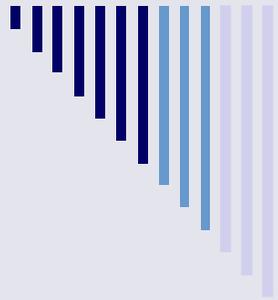
- **Ativos:** emitem algum tipo de sinal captado pelos sensores do robô.
- **Passivos:** não emitem sinal.

- **Naturais:** que pertencem ao ambiente original, sem alterações.
- **Artificiais:** marcos construídos com o propósito de serem utilizados pela localização.



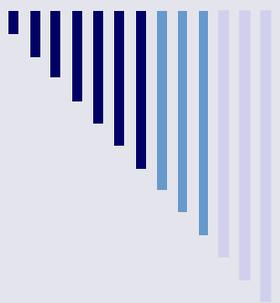
Triangulação





Triangulação

- Refere-se à solução de equações de restrições relacionando a posição de um observador com a posição de um conjunto de marcos conhecidos.
- Em geral utiliza medidas de ângulo e distância de dois ou mais marcos para encontrar a posição do robô.
 - Soluciona parâmetros de um triângulo dada uma combinação de lados e ângulos (LLL, AAL, etc).



3. Abordagem probabilística (dead reckoning + marcos)

- Bastante difundida atualmente
- Resultados muito bons

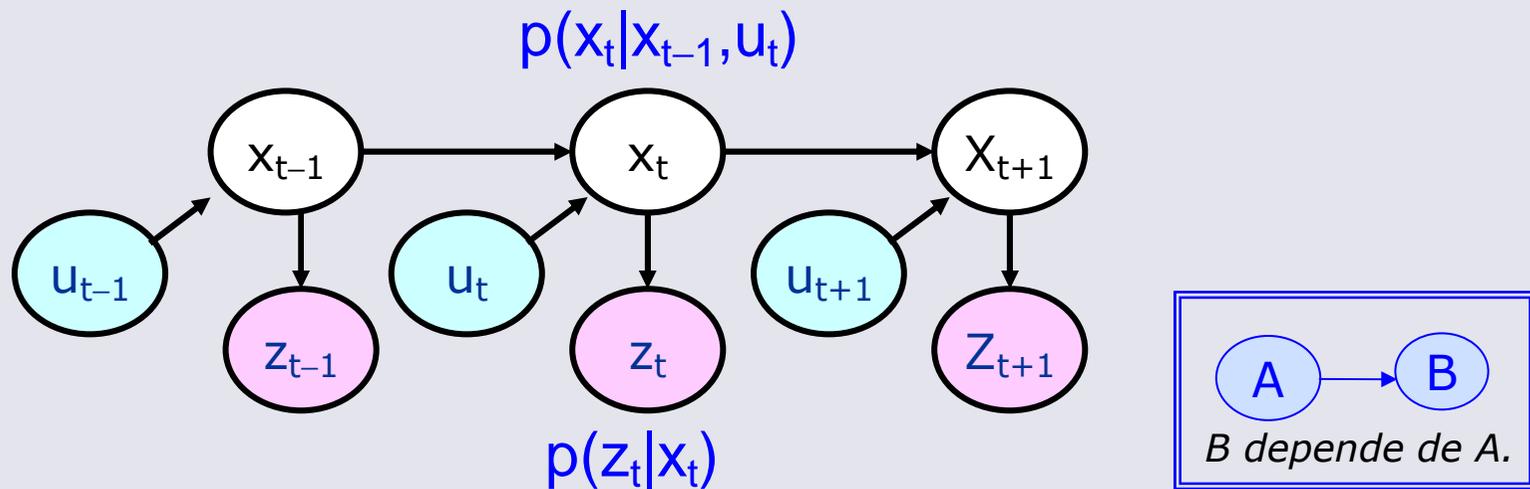
→ Lida explicitamente com a
incerteza envolvida no processo

Lidando com a Incerteza

Robótica Probabilística

Thrun & Burgard & Fox (2005)

- Representação de um sistema estocástico dinâmico como uma DBN (*Dynamic Bayes Network*):

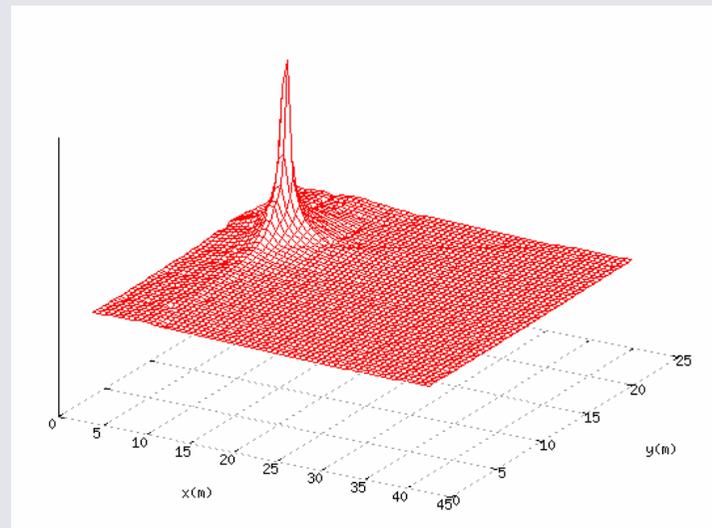


Objetivo: estimar a crença

Crença **bel**(x_t): distribuição de probabilidades sobre x_t e reflete o conhecimento do robô a respeito do estado

Exemplo de funcionamento

- Estimativa de localização (crença): global, discreta, multimodal

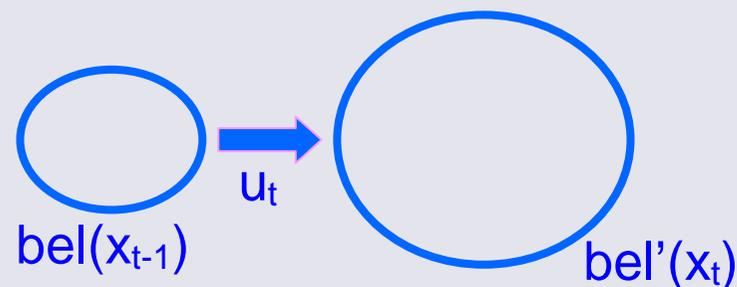
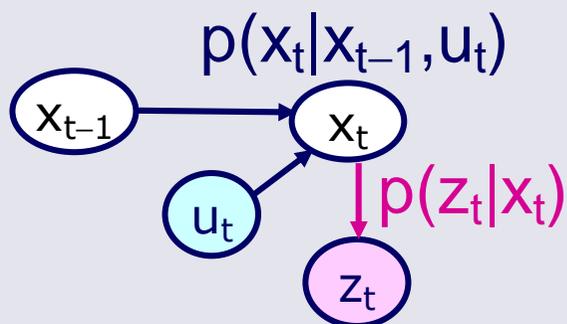


Principal algoritmo para estimação

Filtro de Bayes

1. Algoritmo filtro_Bayes ($bel(x_{t-1}), u_t, z_t$)
2. para todo x_t faça
 1. $bel'(x_t) = \int p(x_t|x_{t-1}, u_t) bel(x_{t-1}) \cdot dx_{t-1}$
 2. $bel(x_t) = \eta \cdot p(z_t|x_t) \cdot bel'(x_t)$
3. fim do para
4. retorne $bel(x_t)$

Passo 2.1: **predição**



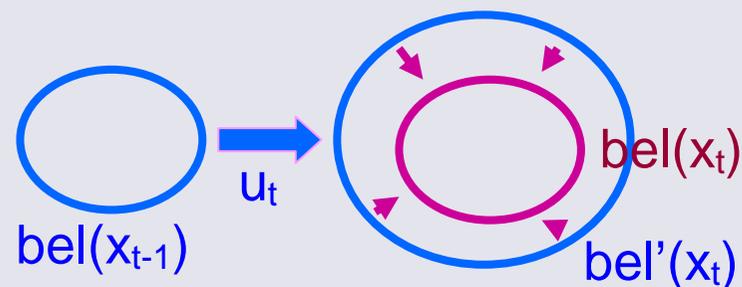
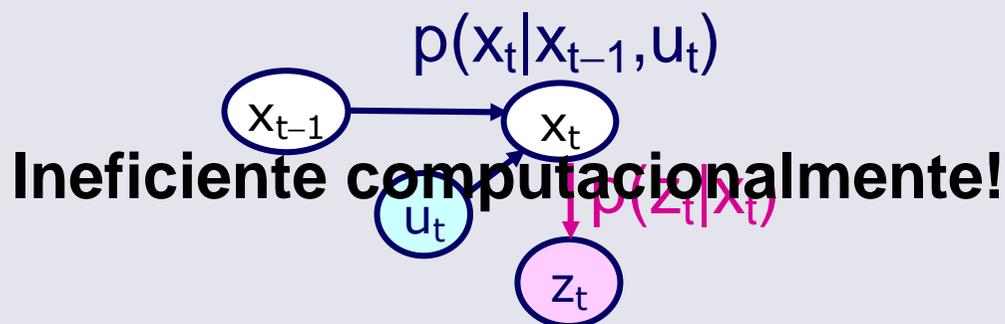
Principal algoritmo para estimação

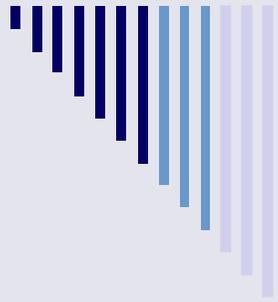
Filtro de Bayes

1. Algoritmo filtro_Bayes ($bel(x_{t-1}), u_t, z_t$)
2. para todo x_t faça
 1. $bel'(x_t) = \int p(x_t|x_{t-1}, u_t) bel(x_{t-1}) \cdot dx_{t-1}$
 2. $bel(x_t) = \eta \cdot p(z_t|x_t) \cdot bel'(x_t)$
3. fim do para
4. retorne $bel(x_t)$

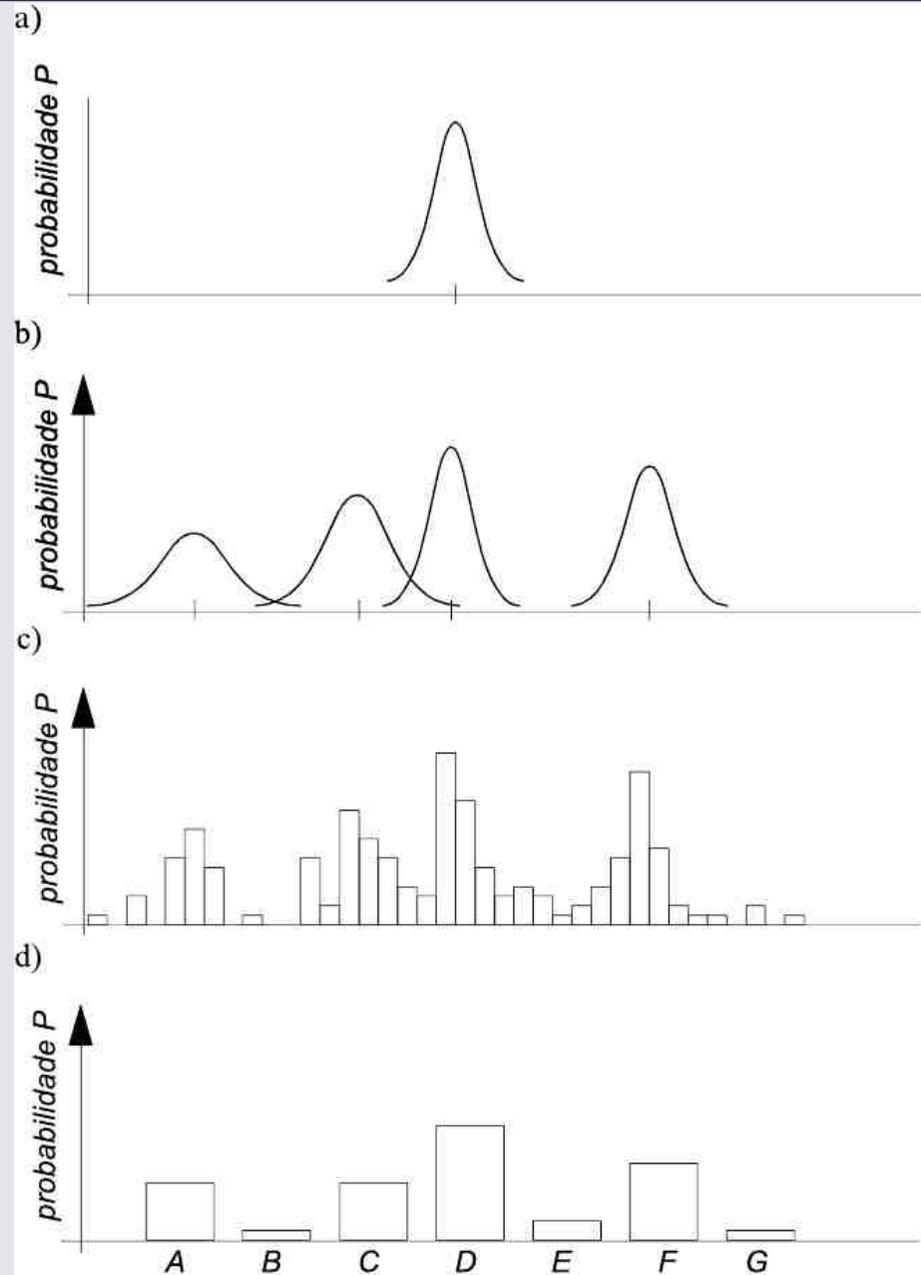
Passo 2.1: **predição**

Passo 2.2: **atualização**

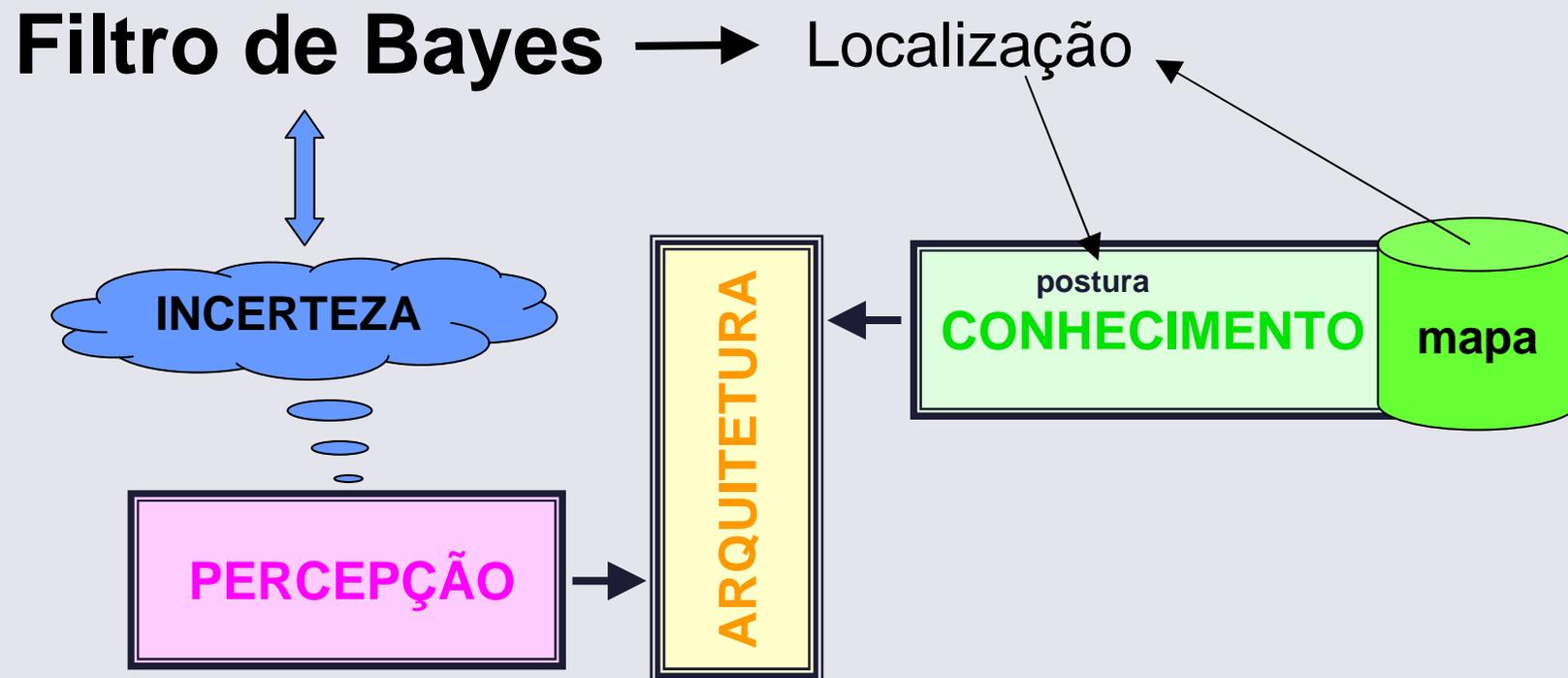


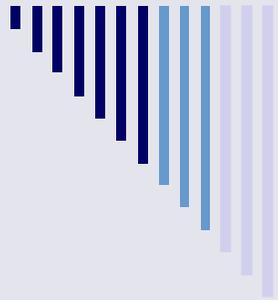


Representação da crença



Navegação: localização

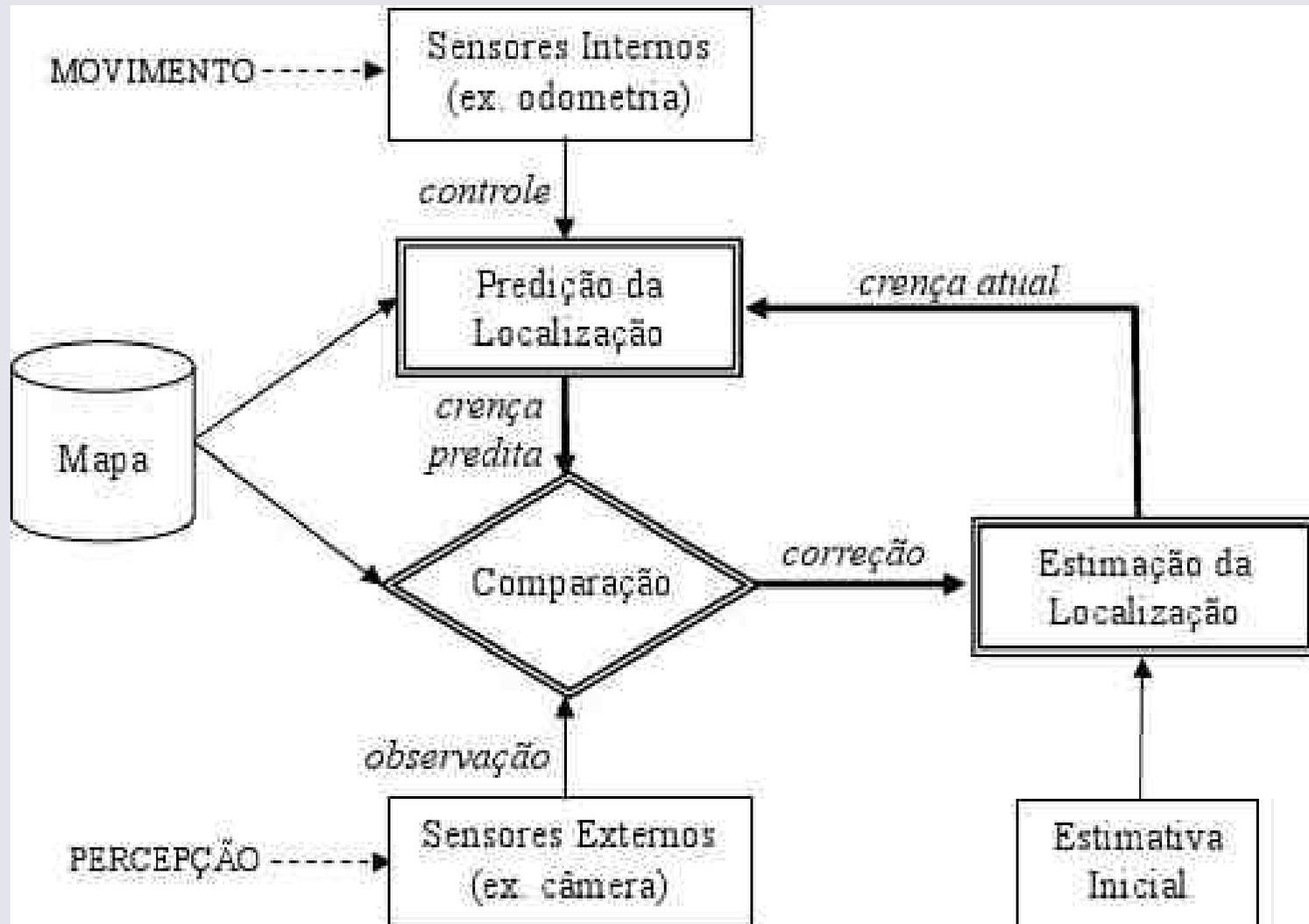




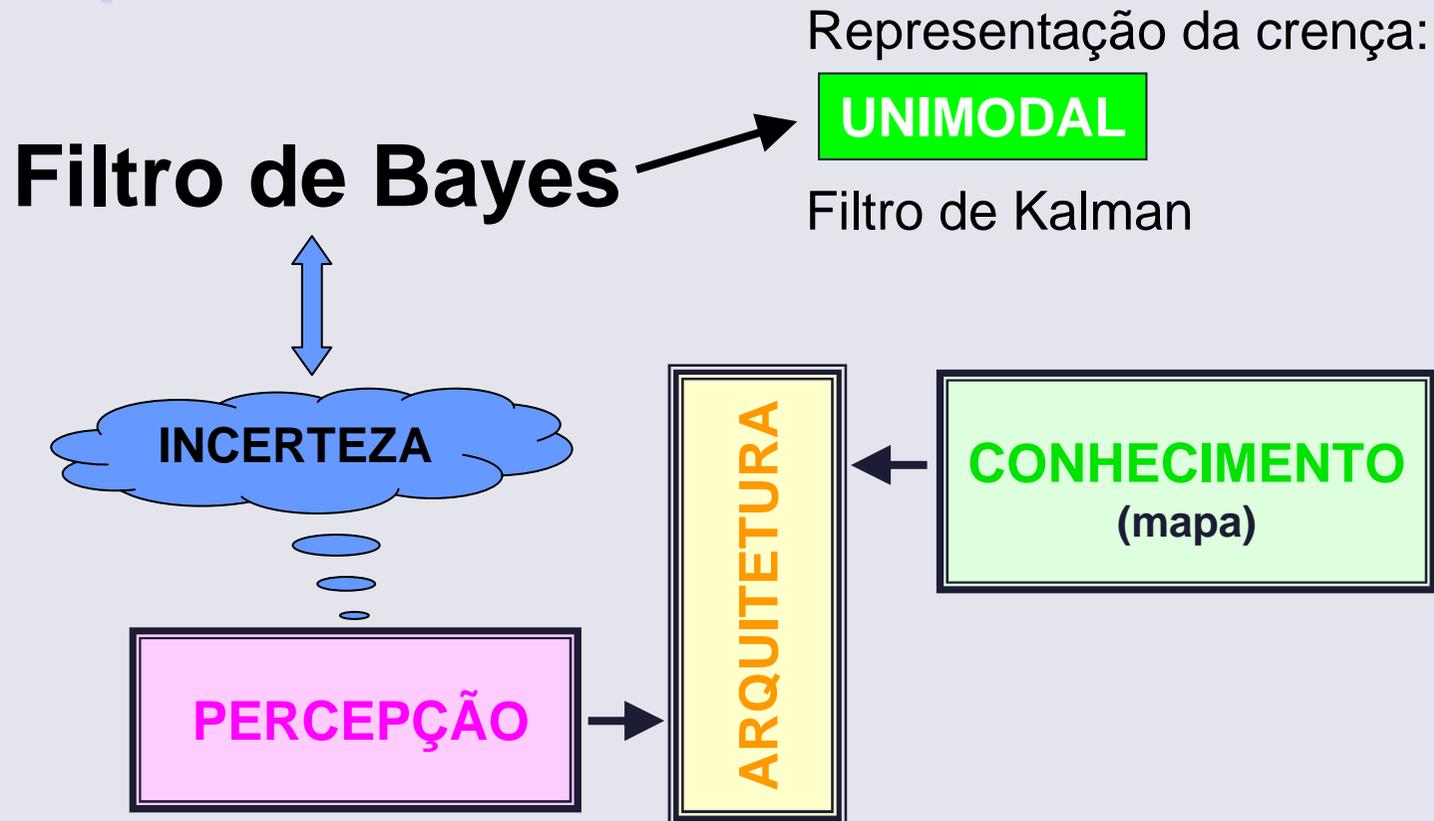
Localização probabilística

- Filtro de Kalman:
 - mantém apenas uma hipótese sobre a postura do robô – localização local.
 - vantagem de ser eficiente computacionalmente.
- Localização de Markov com grades de ocupação:
 - representa uma hipótese de postura para cada célula da grade – localização global.
 - tamanho da grade pode tornar computacionalmente inviável.
- Localização com Filtro de Partículas:
 - mantém várias hipótese sobre a postura do robô, uma para cada partícula – localização global.
 - mais eficiente computacionalmente que a localização de Markov.

Arquitetura básica para localização



Localização: local, contínua e unimodal

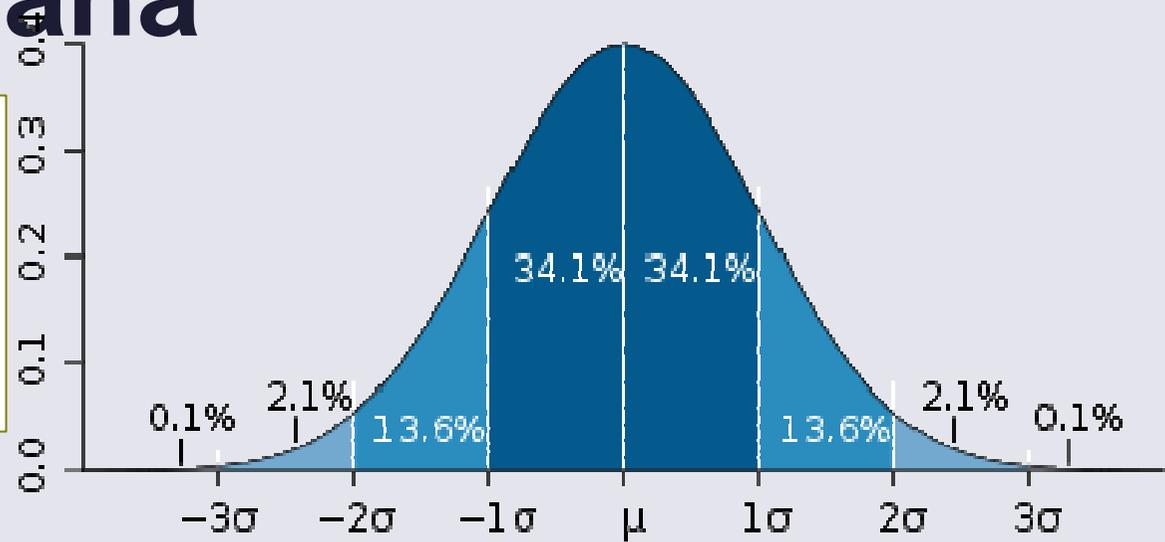


Gaussiana

$$p(x) \sim N(\mu, \sigma^2):$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}}$$

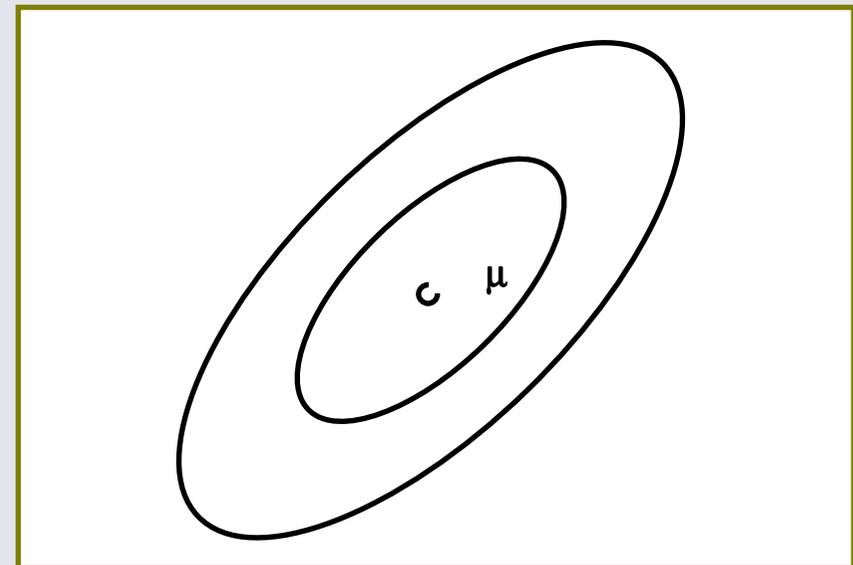
Univariável



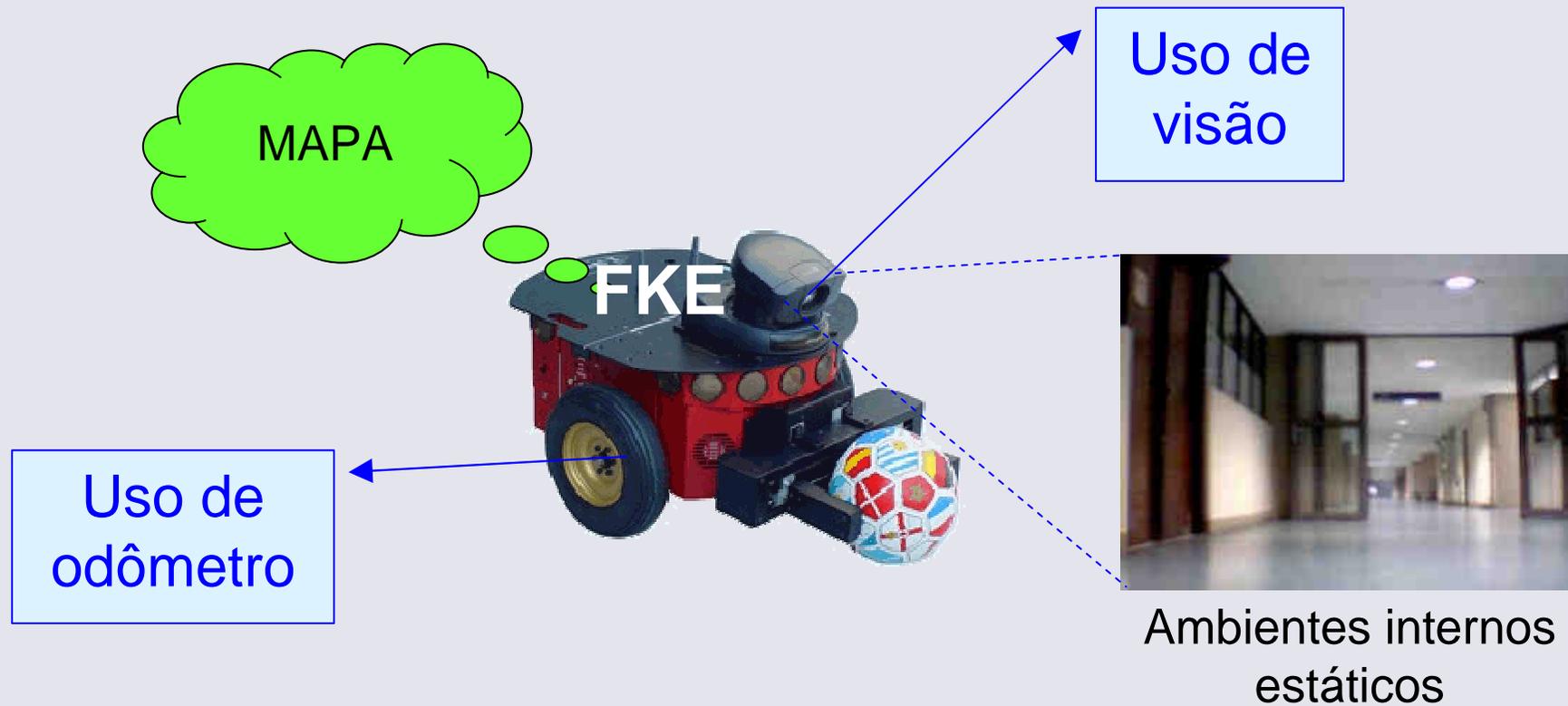
$$p(\mathbf{x}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma}):$$

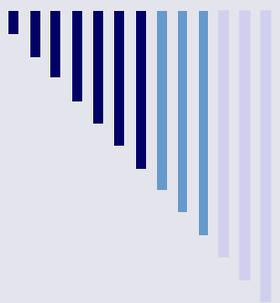
$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2} (\mathbf{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

Multivariável



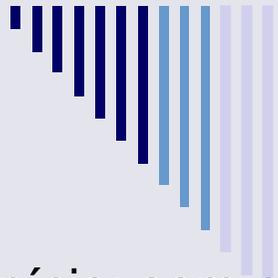
Localização





Localização de Markov

- Localização de Markov é uma técnica probabilística que mantém uma distribuição de probabilidade sobre o espaço de **todas** as hipóteses de onde o robô pode estar.
- Localização: global, discreta, multimodal (*video mostrado antes*)



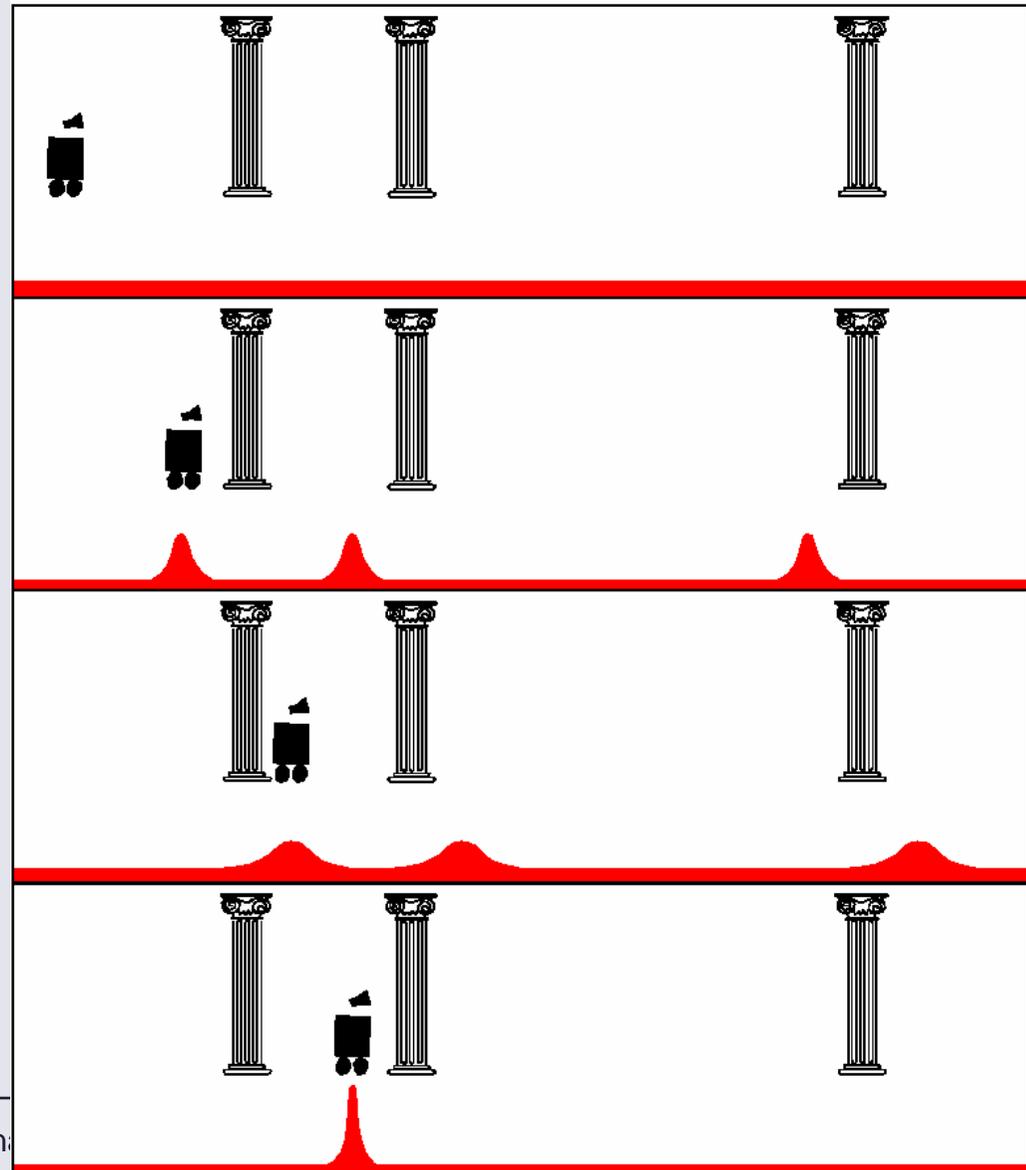
Localização de Markov

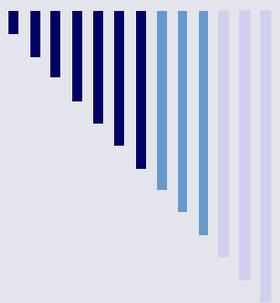
Início: sem conhecimento da posição inicial. Distribuição de probabilidade uniforme.

Robô percebe o primeiro pilar: ao ver só um pilar, possui a mesma probabilidade de estar nos pilares 1, 2 ou 3.

Robô anda: modelo de movimento possibilita estimar uma nova distribuição de probabilidade baseada no movimento.

Robô percebe o segundo pilar: baseado no conhecimento a priori, a probabilidade de estar no pilar 2 é dominante.

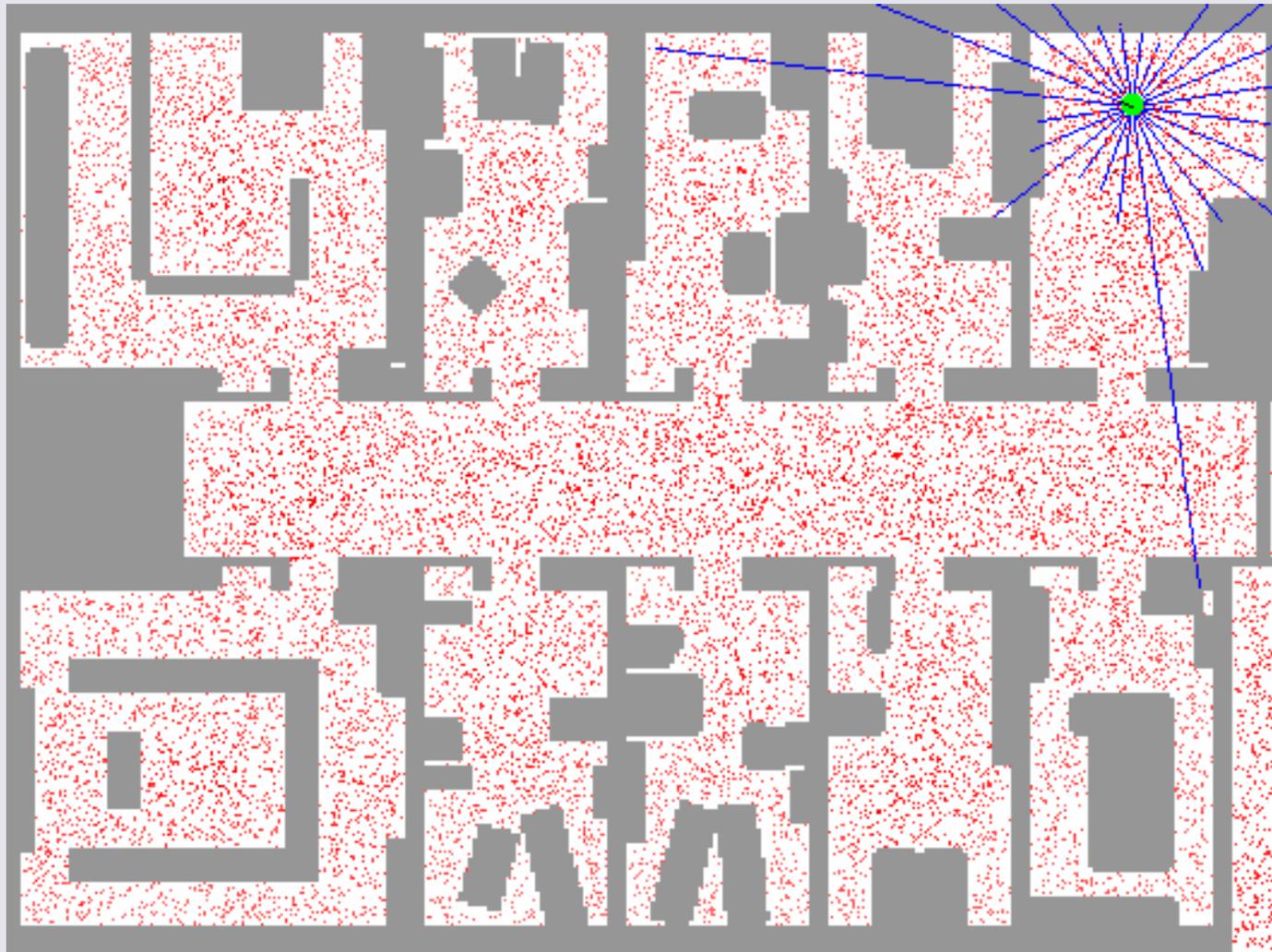




Filtro de partículas

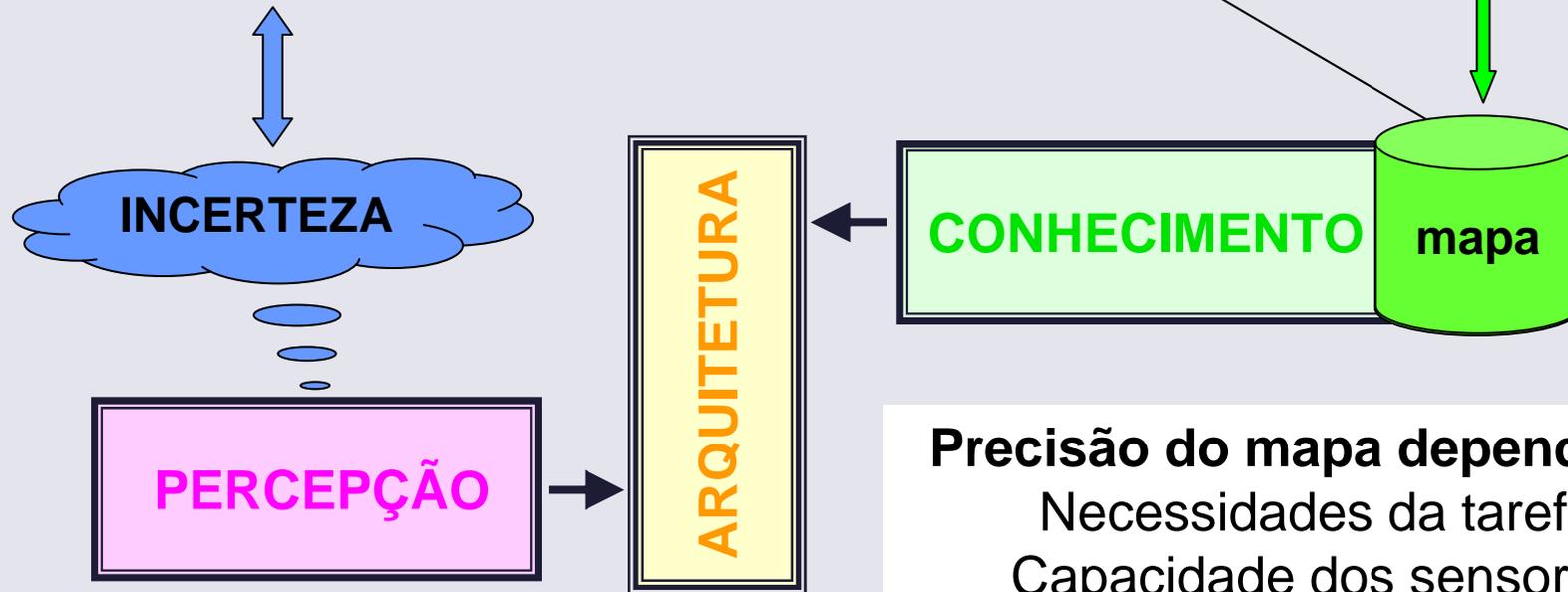
- Representa a crença por **amostras aleatórias**
- Estimação de processos **não-Gaussianos, não lineares**
- Denominações: filtro Monte Carlo, *Survival of the fittest*, *Condensation*, *Bootstrap filter*, Filtro de Partículas

Filtro de Partículas



Mapeamento

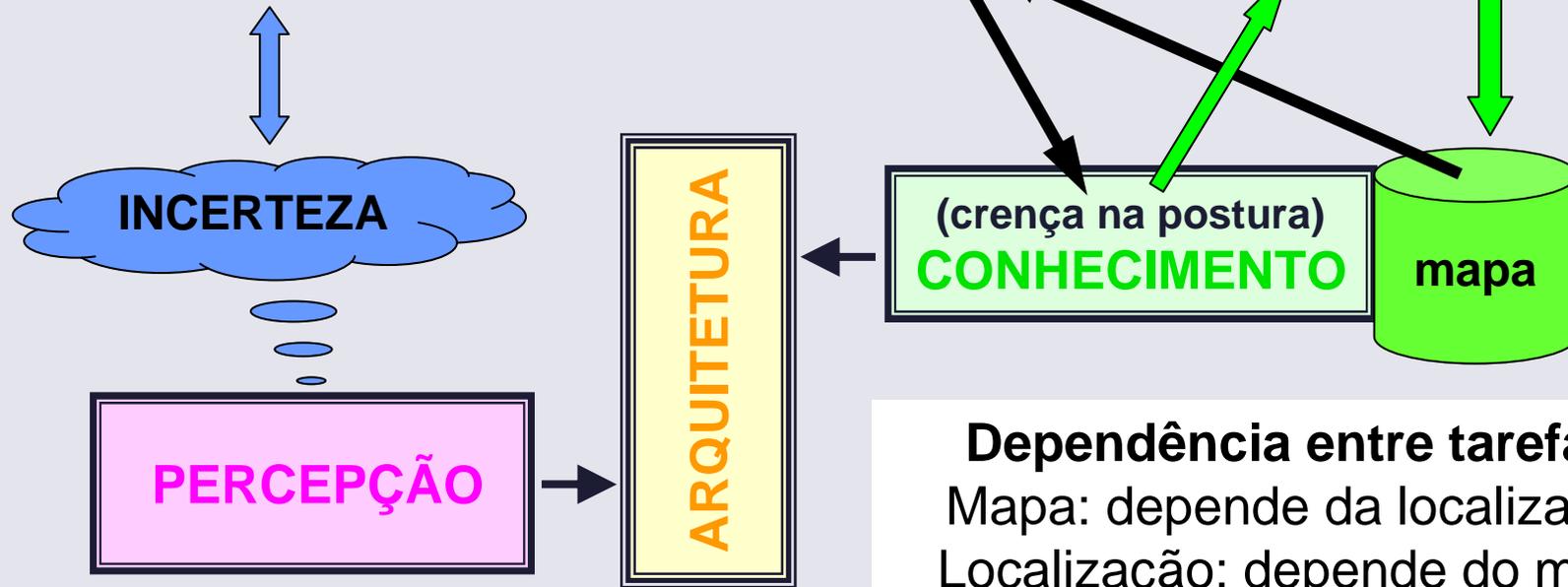
Filtro de Bayes → Localização Mapeamento



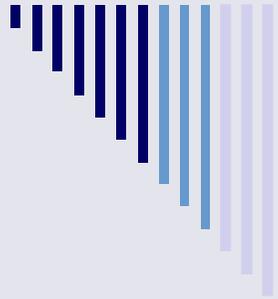
Precisão do mapa depende de:
Necessidades da tarefa
Capacidade dos sensores
Complexidade da representação:
Impacta deliberação e localização

Mapeamento

Filtro de Bayes → Localização ↔ **Mapeamento**

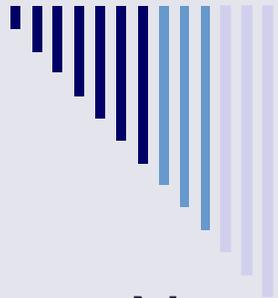


Dependência entre tarefas:
Mapa: depende da localização
Localização: depende do mapa



SLAM – *Simultaneous Localization and Mapping*

- Efetua a estimação conjunta do mapa e da localização



Resumo final

- Navegação: localização, mapeamento e planejamento de trajetória
- Arquiteturas: restringem o conhecimento
- Conhecimento na navegação: mapas
- Mapas: topológico \leftrightarrow métrico
- Localização: forte/fraca, global/local
- Tipos: só sensores internos, só sensores externos, sensores internos + externos
 - Int + ext: abordagem probabilística: KF, ML, PF
- SLAM