

# **ARQUITECTURA DE AGENTES REACTIVOS**

Luís Morgado

2024

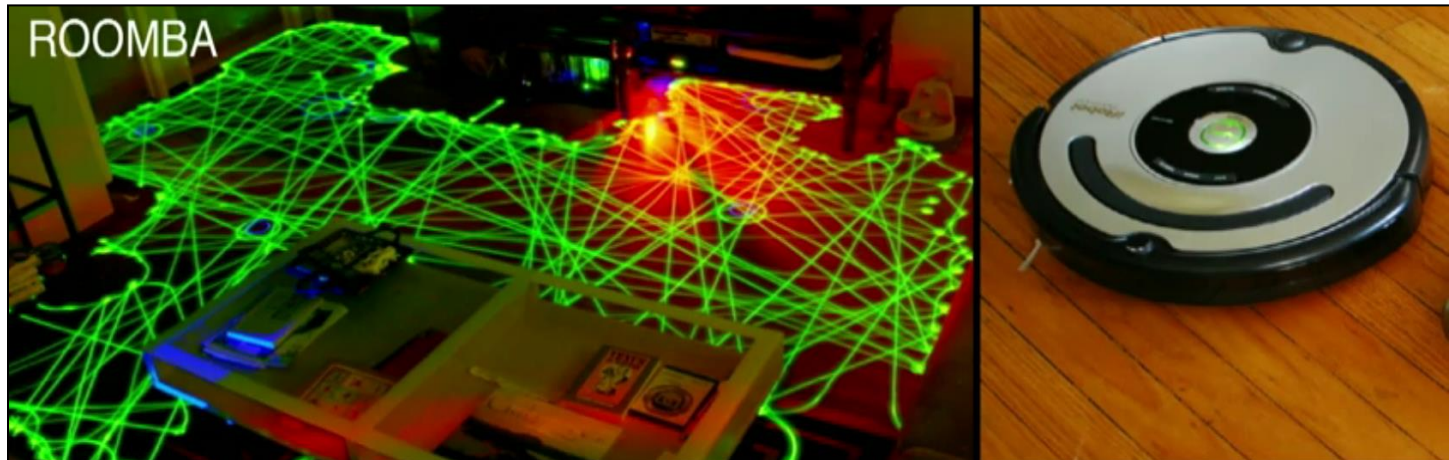
# AGENTES REACTIVOS SEM MEMÓRIA

## ACOPLAMENTO PERCEPÇÃO – ACÇÃO

- Depende fortemente das **capacidades sensoriais**
- Depende das **características do ambiente**

## Exemplo

- Comportamentos de **exploração**
  - Como evitar localizações já exploradas?

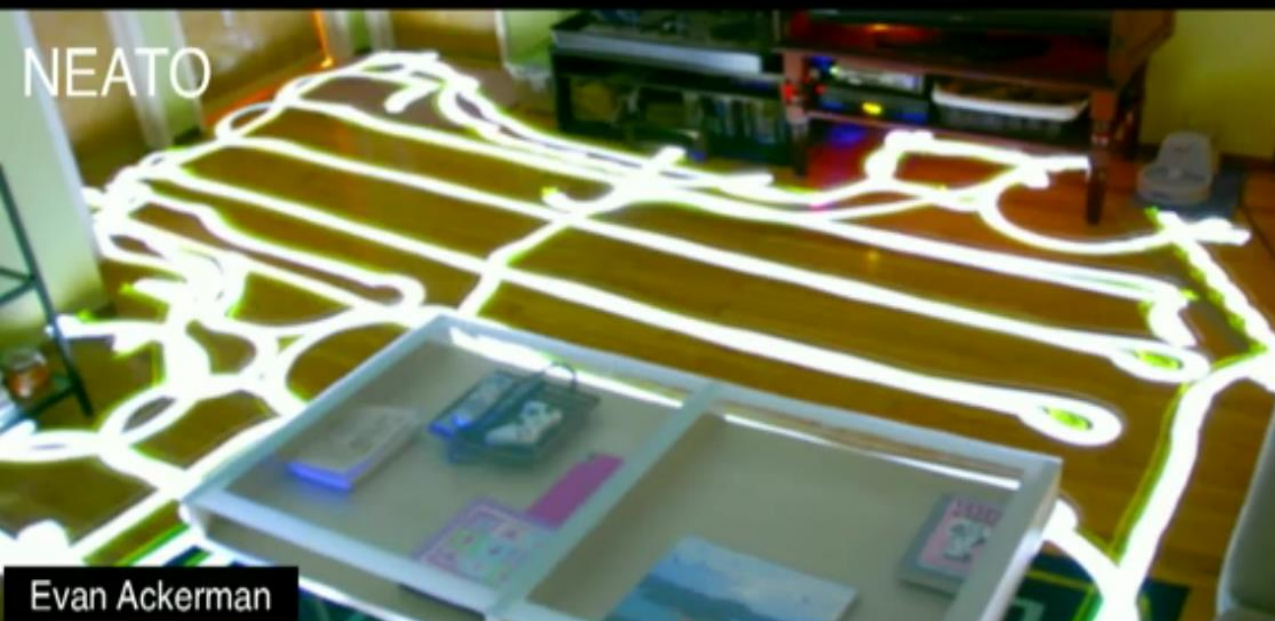


# ARQUITECTURAS DE AGENTES REACTIVOS

- **Problemas** na implementação de comportamentos **sem memória**
  - **Exploração**
    - Necessidade de evitar o passado
  - **Óptimos locais**
    - Por exemplo, os veículos de Braitenberg ficam presos nos cantos, incapazes de dar a volta
  - **Comportamentos cíclicos**
    - Por exemplo, os Veículos de Braitenberg ficam a movimentar-se ciclicamente perante determinadas configurações de alvos e obstáculos
- Necessidade de **manutenção de estado**



Diferença de comportamento entre um sistema que não utiliza memória para navegar num espaço, sendo a exploração do espaço realizada de forma aleatória e menos eficiente, e um sistema que utiliza memória, em que a exploração do espaço é sistemática e mais eficiente



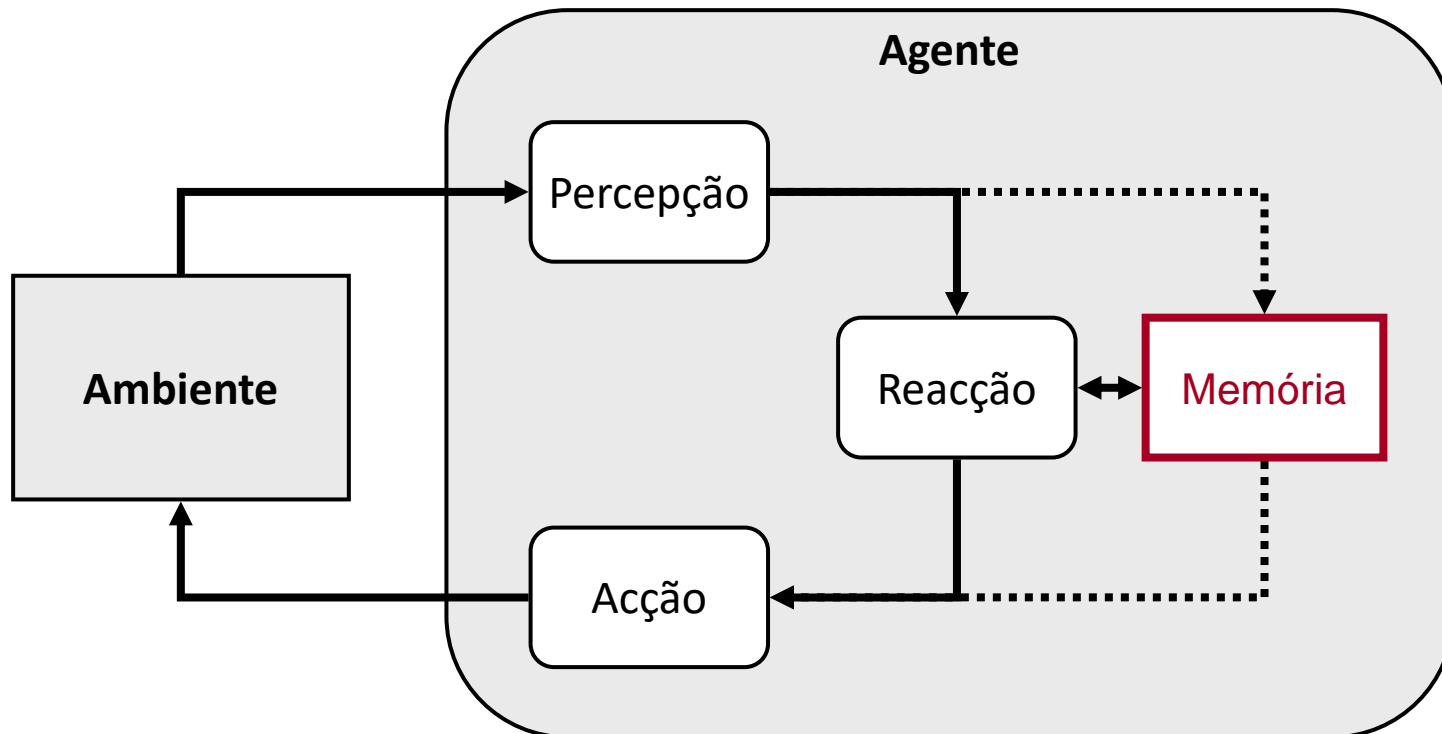
Evan Ackerman

# ARQUITECTURAS DE AGENTES REACTIVOS

## ARQUITECTURA REACTIVA COM MEMÓRIA

Numa arquitectura reactiva com memória, as reacções dependem não só das percepções, mas também da memória de percepções anteriores (ou de informação delas derivada) para gerar as acções.

Para esse efeito é necessário manter internamente memória, a qual é actualizada a partir das percepções e das reacções activadas, influenciando essas mesmas reacções, bem como as acções geradas.



# AGENTES REACTIVOS COM ESTADO

- Exemplo

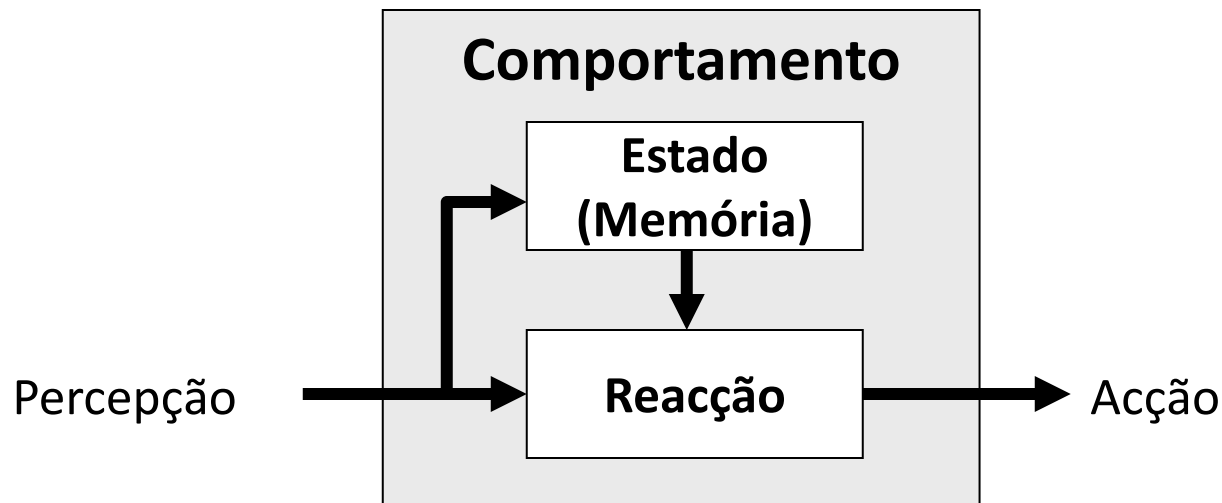
## Comportamento “**Evitar o Passado**”

- Representação interna de **percepções anteriores**
- Evitar situações conhecidas
- Gerar acções de afastamento de posições anteriormente percepcionadas e registadas em memória



# AGENTES REACTIVOS COM ESTADO

- **Reacções** podem envolver não apenas percepções mas também **estado interno (memória)**
- Manipulação de estado
  - Regras e acções para **alteração do estado interno**
- **Comportamentos com memória**



# COMPORTAMENTOS COM ESTADO

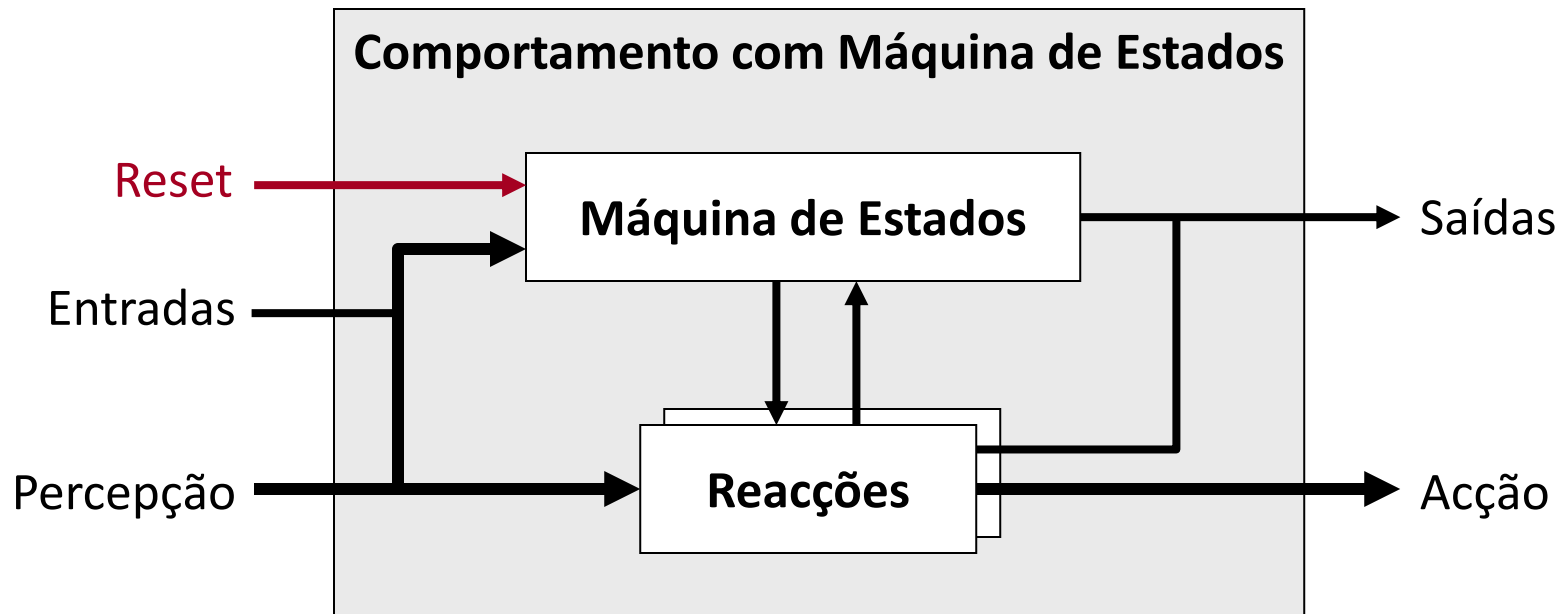
- Máquina de estados interna

- Reinicialização (*Reset*)

- Entradas

- Saídas

Interligação entre comportamentos





# AGENTES REACTIVOS COM ESTADO

- **Vantagens da manutenção de estado**
  - Uma arquitectura reactiva com estado **pode produzir todo o tipo de comportamento**
  - Possibilidade de **representar dinâmicas temporais**
    - **Evolução do estado** ao longo do tempo
    - **Resposta** não apenas em função das **percepções actuais**, mas também em função de **memórias de percepções anteriores**
  - Possibilidade de comportamentos mais complexos **baseados na evolução de estado**
    - Com continuidade no tempo
    - Agir devido a ausência de mudança
  - Capacidade de lidar com **situações de falha** por **exploração** de acções não realizadas anteriormente

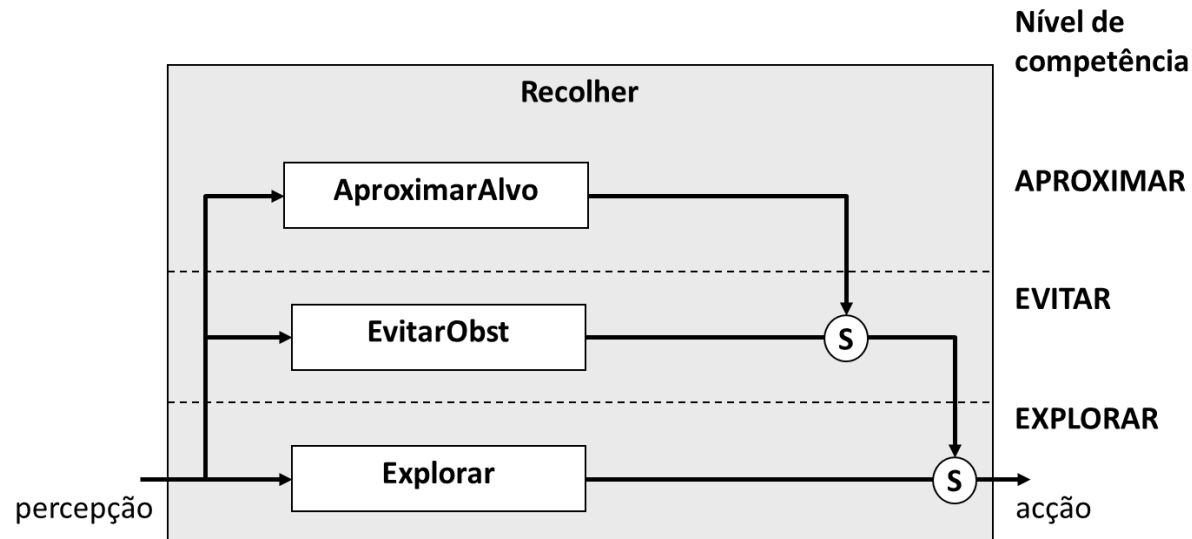
# AGENTES REACTIVOS COM ESTADO

- **Desvantagens da manutenção de estado**
  - Necessário **memória** (espaço)
    - **Aumento da complexidade espacial**
  - Necessário **manter** as representações de estado
    - **Aumento da complexidade computacional**
  - Mesmo com a manutenção de estado, **as arquitecturas reactivas não suportam representações complexas, nem exploram planos alternativos de acção**

# ARQUITECTURA DE SUBSUNÇÃO

[Brooks, 1985]

- Comportamentos **organizados em camadas (níveis de competência)** e responsáveis pela concretização independente de um objectivo
- **Resultado do comportamento** pode ser a **entrada de outro comportamento**
- Possibilidade de **comportamentos das camadas superiores assumirem o controlo** sobre comportamentos das camadas inferiores
- Camadas inferiores **não têm conhecimento** das camadas superiores
  - **Hierarquia de comportamentos**



# ARQUITECTURA DE SUBSUNÇÃO

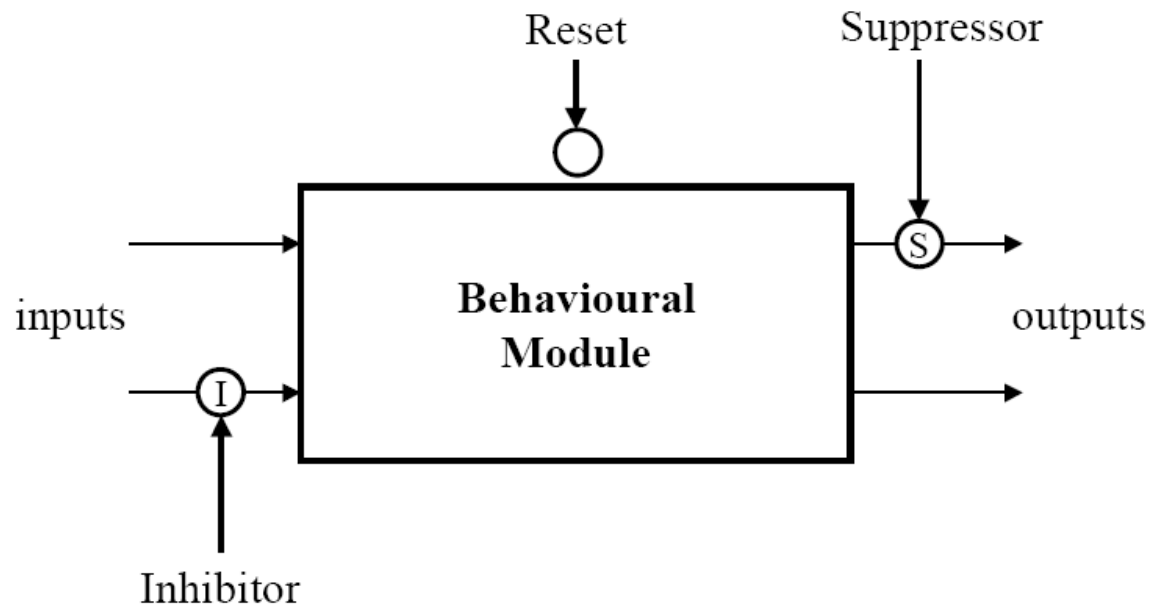
- Saídas das camadas inferiores podem ser utilizadas por camadas superiores
- Camadas superiores controlam as camadas inferiores
  - **Inibição**
    - Desactivação de comunicação entre módulos
  - **Supressão**
    - Desactivação de comportamento
  - **Reinício (*Reset*)**
    - Reposição do estado inicial de um comportamento

# ARQUITECTURA DE SUBSUNÇÃO

## MÓDULOS COMPORTAMENTAIS

Principais tipos de mecanismos de controlo associados:

- **Inibição** de entradas (a informação de entrada não é considerada)
- **Supressão** das saídas (a informação de saída não é considerada)
- **Reinício** do estado interno (*reset*)



[Brooks, 1991]

# ARQUITECTURA DE SUBSUNÇÃO

## IMPLEMENTAÇÃO DE MÓDULOS COMPORTAMENTAIS

- Implementação com base em **seqüências de activação fixa** (procedimentos)
- Implementação com base em **regras estímulo – resposta**
- Implementação com base em **máquinas de estado aumentadas** (AFSM - *Augmented Finite State Machines*)
  - Temporizadores
  - Cada AFSM realiza um comportamento e é responsável pela sua própria percepção do mundo

# ARQUITECTURA DE SUBSUNÇÃO

## Exemplo: Tarefa de prospecção

- Tarefa consiste na procura de elementos do ambiente com características específicas (*alvos*)
- Quando o agente detecta um alvo, dirige-se até ele, pega no alvo e transporta-o até uma base
- Estas acções são repetidas até todos os alvos terem sido recolhidos para a base



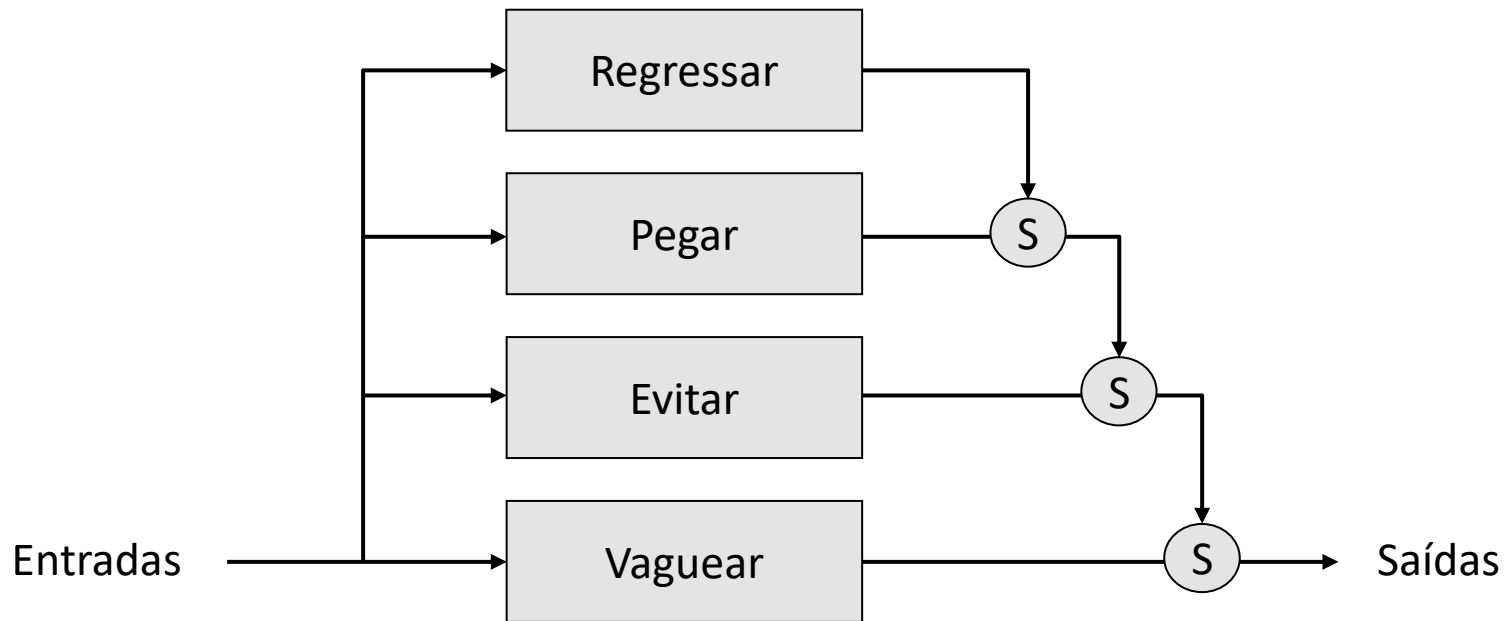
# ARQUITECTURA DE SUBSUNÇÃO

## Exemplo: Tarefa de prospecção

- Implementação com base em quatro comportamentos distintos
  - **Vaguear**
    - Movimentação em direcções aleatórias
  - **Evitar**
    - Virar para a esquerda (direita) caso seja detectado um obstáculo à direita (esquerda) e de seguida avançar
    - Após três tentativas sem sucesso recuar
  - **Pegar**
    - Tomar a direcção do alvo e avançar até ele; após o alvo alcançado fechar a pega
  - **Regressar**
    - Tomar a direcção da base e avançar até a atingir; após atingir a base parar

# ARQUITECTURA DE SUBSUNÇÃO

## Exemplo: Tarefa de prospecção

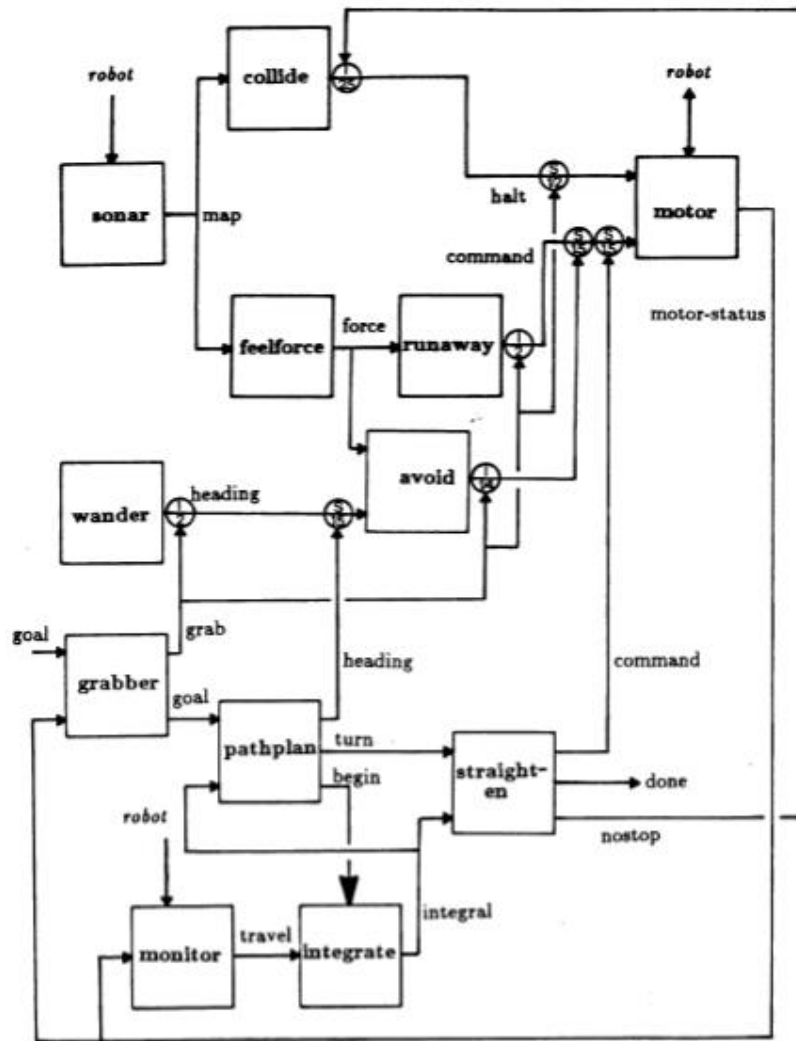


# ARQUITECTURA DE SUBSUNÇÃO

- Proposta como **alternativa a abordagens simbólicas**
- Arquitectura definida por **conjuntos de comportamentos**
- Comportamentos organizados em camadas (**níveis de competência**)
- Desenvolvimento **incremental**
- **Robustez**
- **Simplicidade** relativa
  - Problemas de escala
    - Para tarefas complexas, o número de reacções e comportamentos pode ser muito elevado, podendo ser difícil de modelar as múltiplas relações entre comportamentos de forma robusta, ou seja, garantido que o sistema cumpre corretamente a sua função

# ARQUITECTURA DE SUBSUNÇÃO

Exemplo:



Quando o número de comportamentos aumenta, o aumento de complexidade resultante das múltiplas dependências entre comportamentos pode tornar difícil a modelação de um sistema, pelo que uma adequada organização modular, que maximize a coesão e minimize o acoplamento é importante.

# BIBLIOGRAFIA

[Russel & Norvig, 2003]

S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd Edition, Prentice Hall, 2003

[Murphy, 2000]

R. Murphy, *An Introduction to AI Robotics*, MIT Press, 2000

[Wooldridge, 2002]

M. Wooldridge, *An Introduction to Multi-Agent Systems*, John Wiley & Sons, 2002

[Pfeifer & Scheier, 2002]

R. Pfeifer, C. Scheier, *Understanding Intelligence*, MIT Press, 2000

[Brooks, 1985]

R. Brooks, *A Robust Layered Control System for a Mobile Robot*, A. I. Memo 864, MIT AI-Lab, 1985

[Hoagland *et al.*, 2001]

M. Hoagland, B. Dodson, J. Hauck, *Exploring The Way Life Works: The Science of Biology*, Jones & Bartlett Learning, 2001

[J. Staddon, 2001]

J. Staddon, *Adaptive Dynamics: The Theoretical Analysis of Behavior*, MIT Press, 2001

[Logan, 2001]

B. Logan, *Designing Intelligent Agents*, School of Computer Science, University of Nottingham, 2001