

Robôs Móveis Inteligentes: Princípios e Técnicas

Profa. Roseli Aparecida Francelin Romero

Instituto de Ciências Matemáticas e de Computação - USP São Carlos

`rafrance@icmc.sc.usp.br` `www.icmc.sc.usp.br/~rafrance`

- **Hastes articuladas**

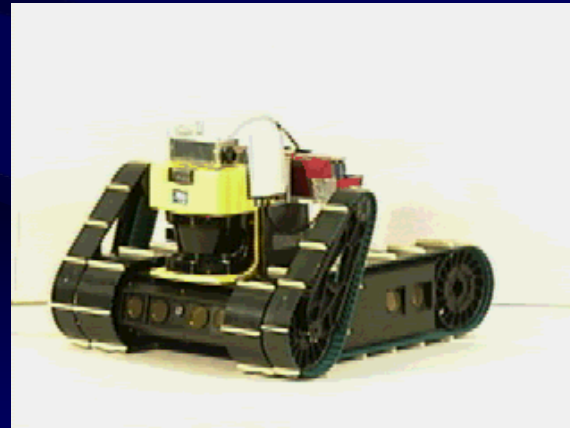
- Cada haste simula uma “perna” ou “pata”.
- Mínimo de dois graus de liberdade por haste.
- Projeto complexo (problemas de sincronismo).
- Custo alto: dois graus de liberdade = dois motores por haste.
- Pode ser interessante para robôs que se movimentam em terrenos muito irregulares ou com escadas.



Attila
Brooks

- **Lagartas**

- Similar ao processo de locomoção em alguns tanques de guerra e tratores.
- Pouco eficiente: fricção de rodas com esteiras e deslizamentos durante o movimento de giro são grandes dissipadores de energia.
- Pode ser interessante para robôs que se movimentam em terrenos muito irregulares.



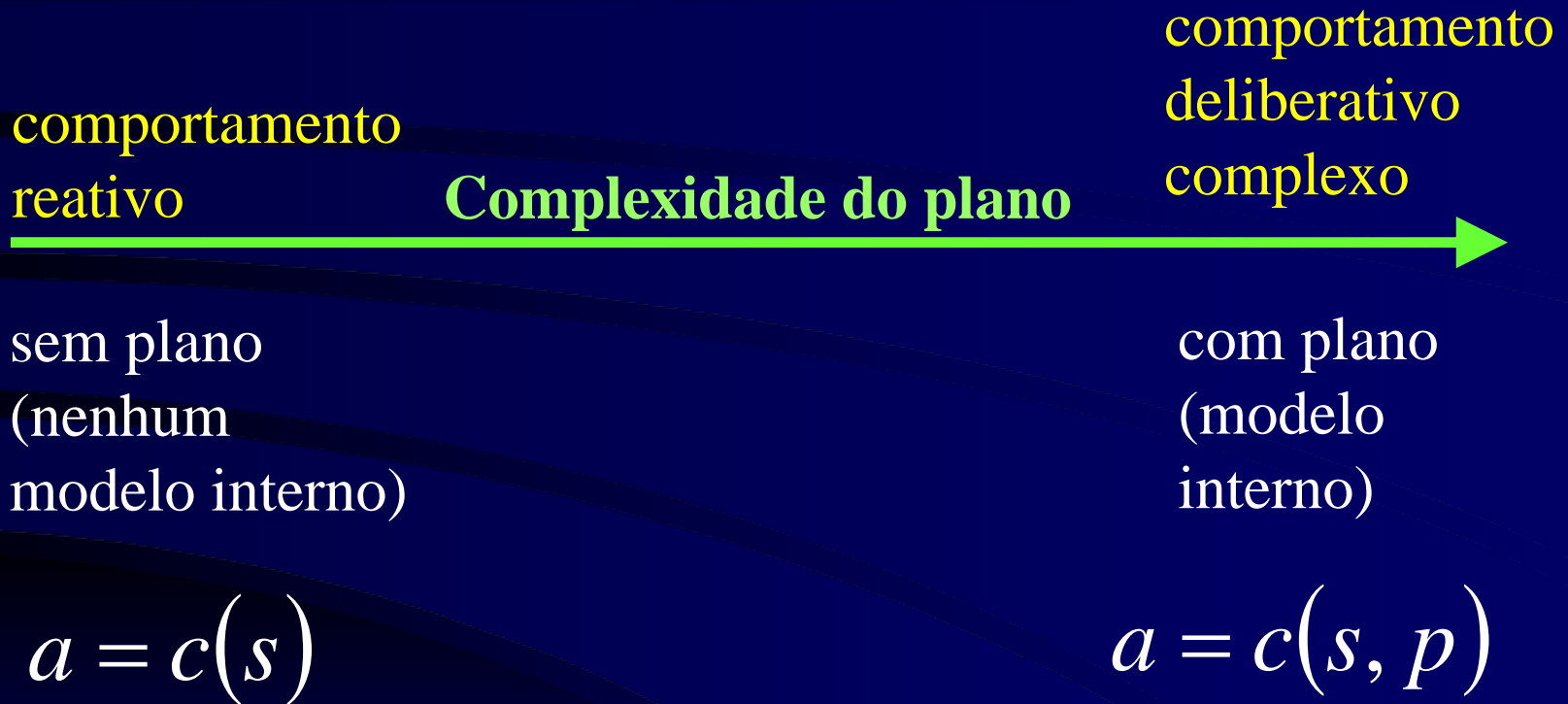
Littejohn
IRobot

2.4 Comportamentos: Definição

Um comportamento é uma função que produz **ações a partir de percepções** (obtidas pelos sensores) e **plano de ação** (obtido a partir de um modelo interno do ambiente):

$$a = c(s, p)$$

Comportamentos: Escala de Complexidade



Exemplo de Comportamento Reativo

Evitar Obstáculos

SE (leitura(algum sensor do lado dir.) > L)

ENTÃO ($v < 0$, roda esq. e $v > 0$, roda dir.)

SENÃO

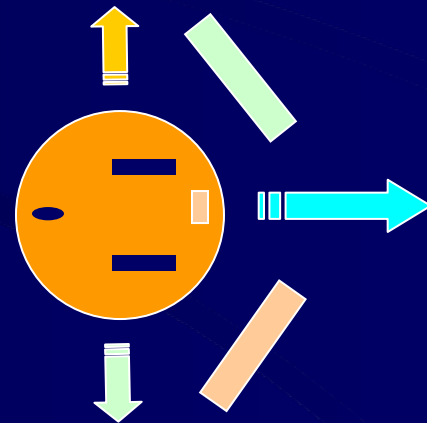
(SE (leitura(algum sensor do lado esq.) > L)

ENTÃO ($v > 0$, roda esq. e $v < 0$, roda dir.)

SENÃO ($v > 0$, nas duas rodas))

Evitar Obstáculos

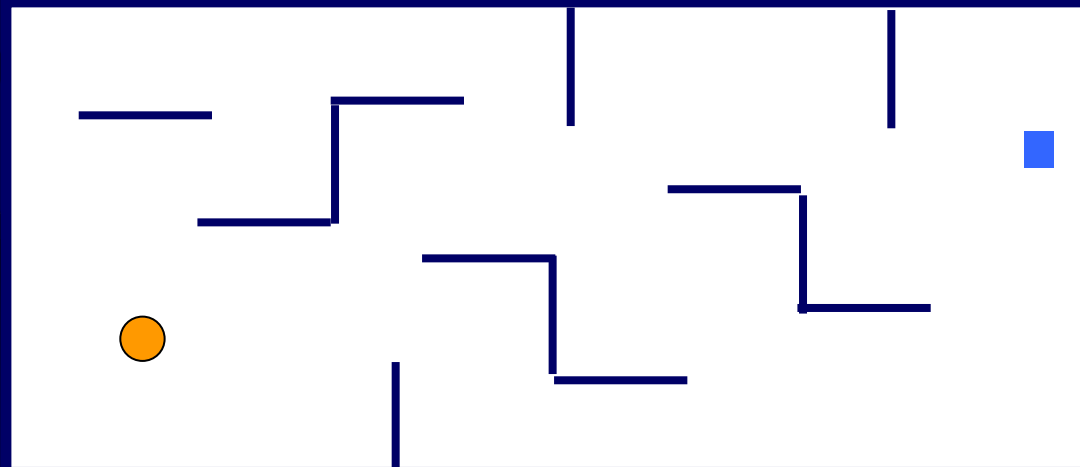
$$a = c(s) = \begin{cases} \text{girar à esquerda, se } s^d > L, \\ \text{senão girar à direita, se } s^e > L, \\ \text{senão avançar} \end{cases}$$



Exemplo de Comportamento Deliberativo

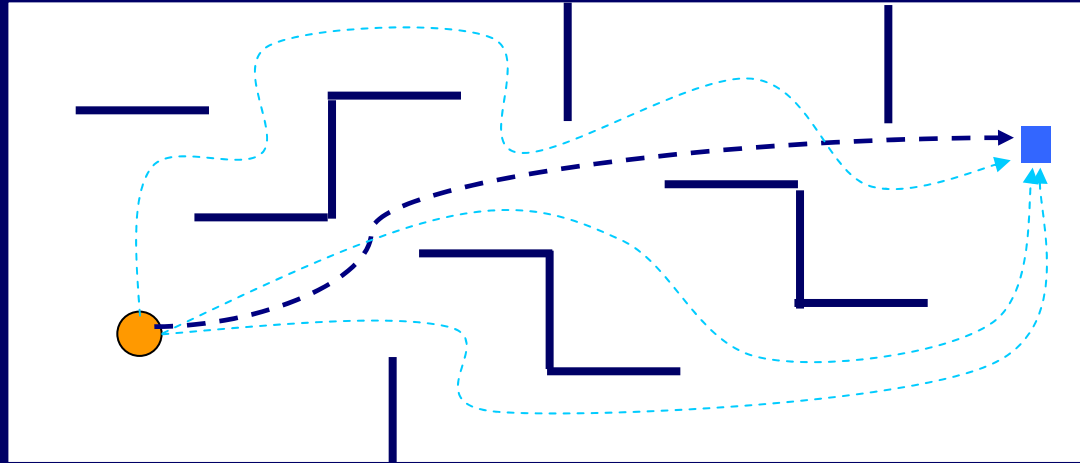
Aproximação de Alvo

- Robô em ambiente com salas, corredores e portas.

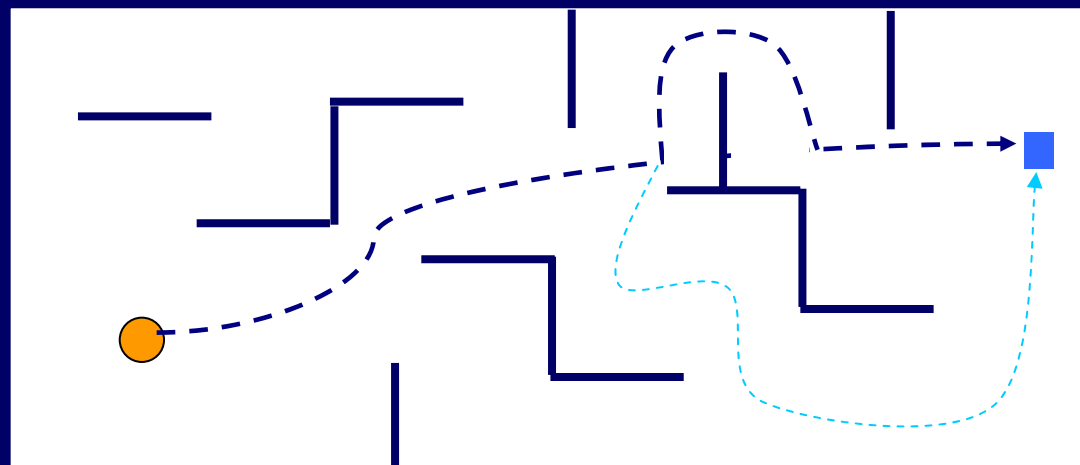


- Enfoque deliberativo: planejamento de trajetórias.
- Modelo interno:
 - Mapa do ambiente (pos. do alvo, obstáculos, etc.).
 - Modelo da dinâmica do robô (efeitos das ações).
- Algoritmo de **planejamento** para geração de trajetória sobre modelo interno (deliberação).
- Algoritmo de **replanejamento** que considera alterações no modelo interno (produzidas por novas observações).

planejamento



replanejamento



2.5 Métodos para Implementação de Comportamentos

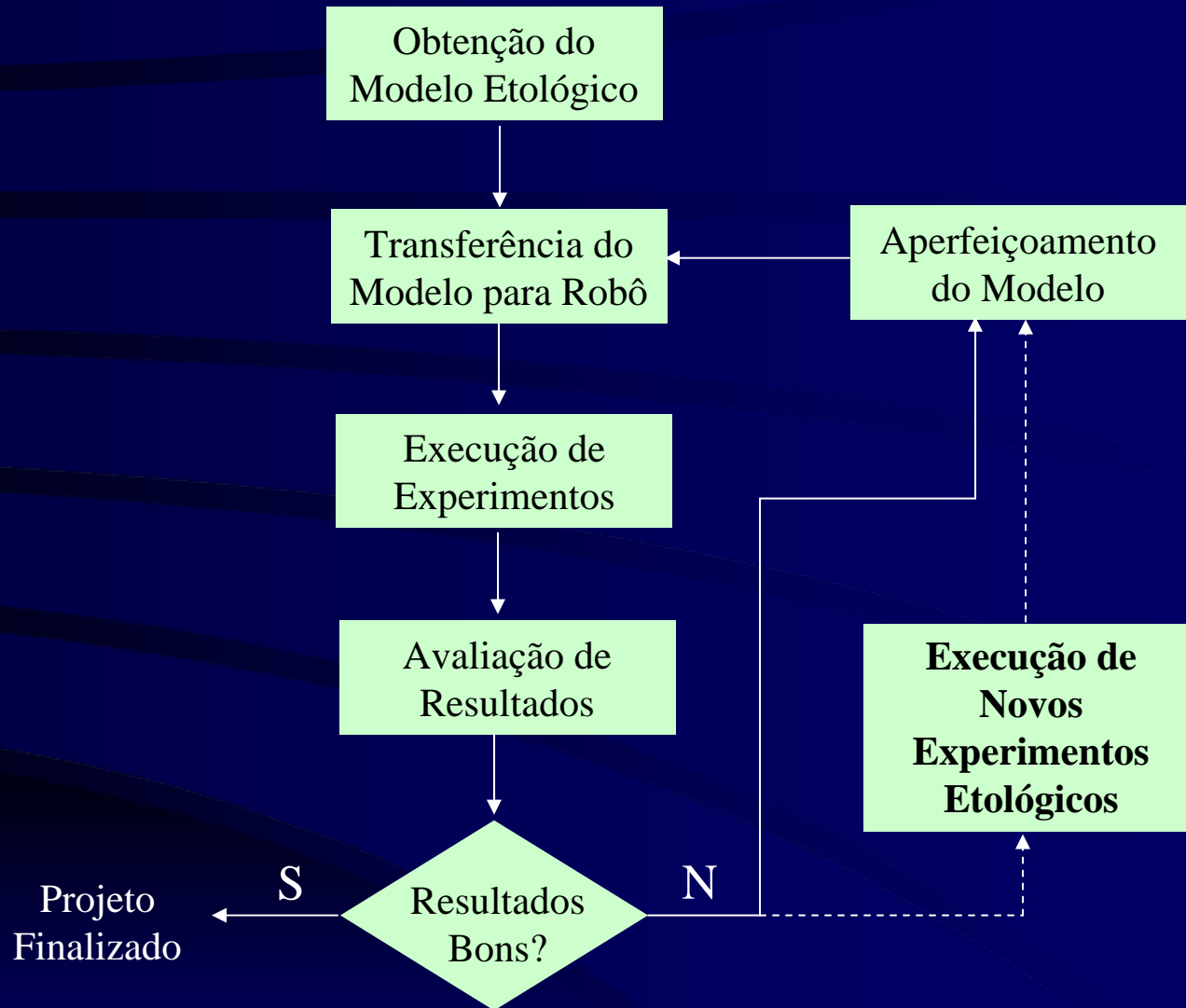
Implementar comportamentos em um robô móvel **normalmente** exige obediência à um conjunto bem definido de princípios e regras, facilitadores de um **projeto organizado**.

Um método = um modo de proceder.

2.5.1 Métodos Baseados em Etologia

Etologia estuda hábitos e acomodação de animais às condições ambientais.

- Comportamentos complexos frequentemente resultam da combinação de comportamentos simples (reativos).
- Informação sensorial é ignorada a menos que seja necessária para a execução do comportamento.
- Mecanismos de coordenação operam sempre que vários comportamentos estejam ativos.
- Indivíduos têm comportamentos adequados para o nicho em que vivem

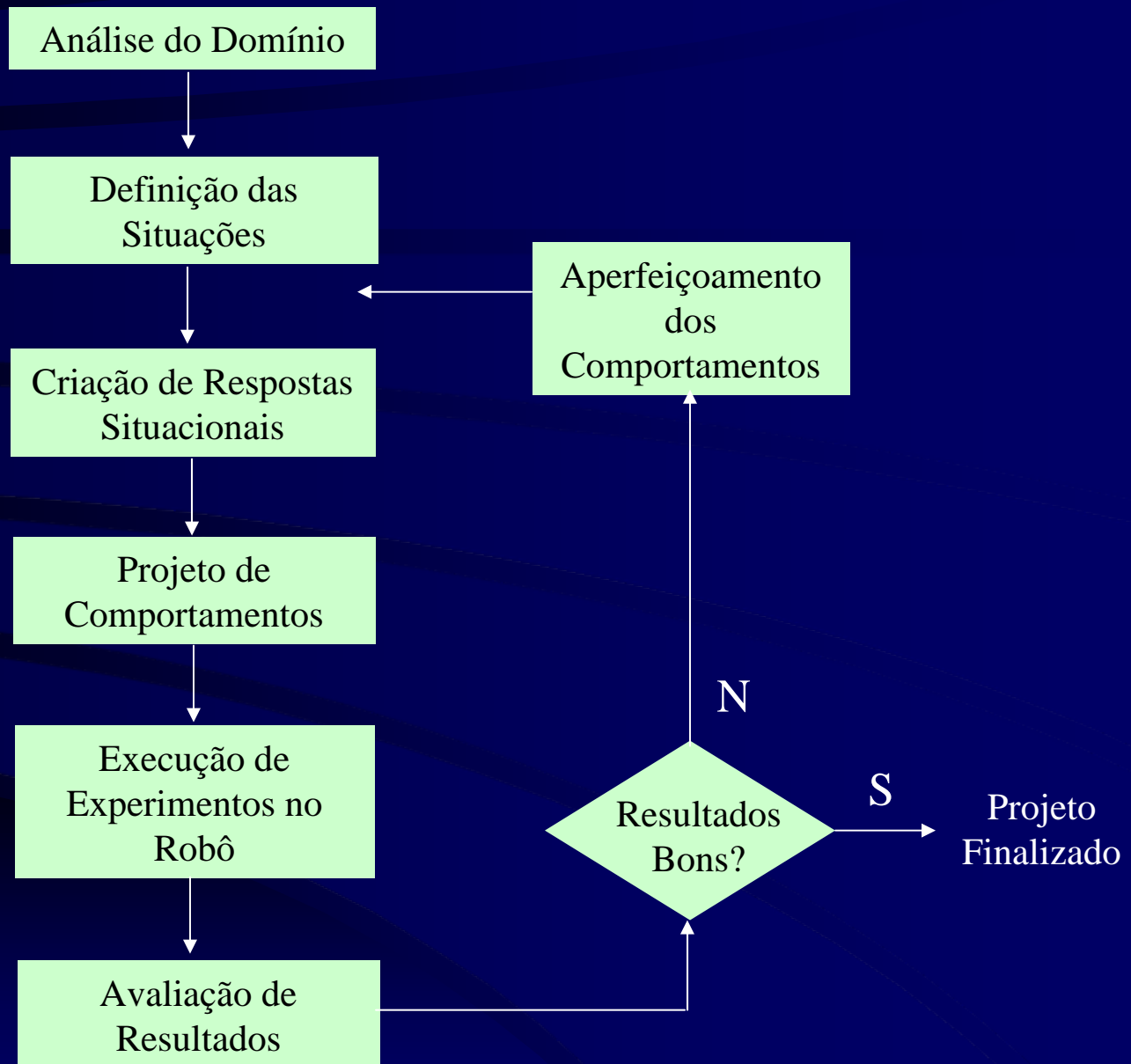


2.5.2 Métodos Baseados em Atividade Situada

Comportamentos são projetados para cada possível situação encontrada pelo robô.

Comportamentos projetados não têm necessariamente um equivalente etológico.

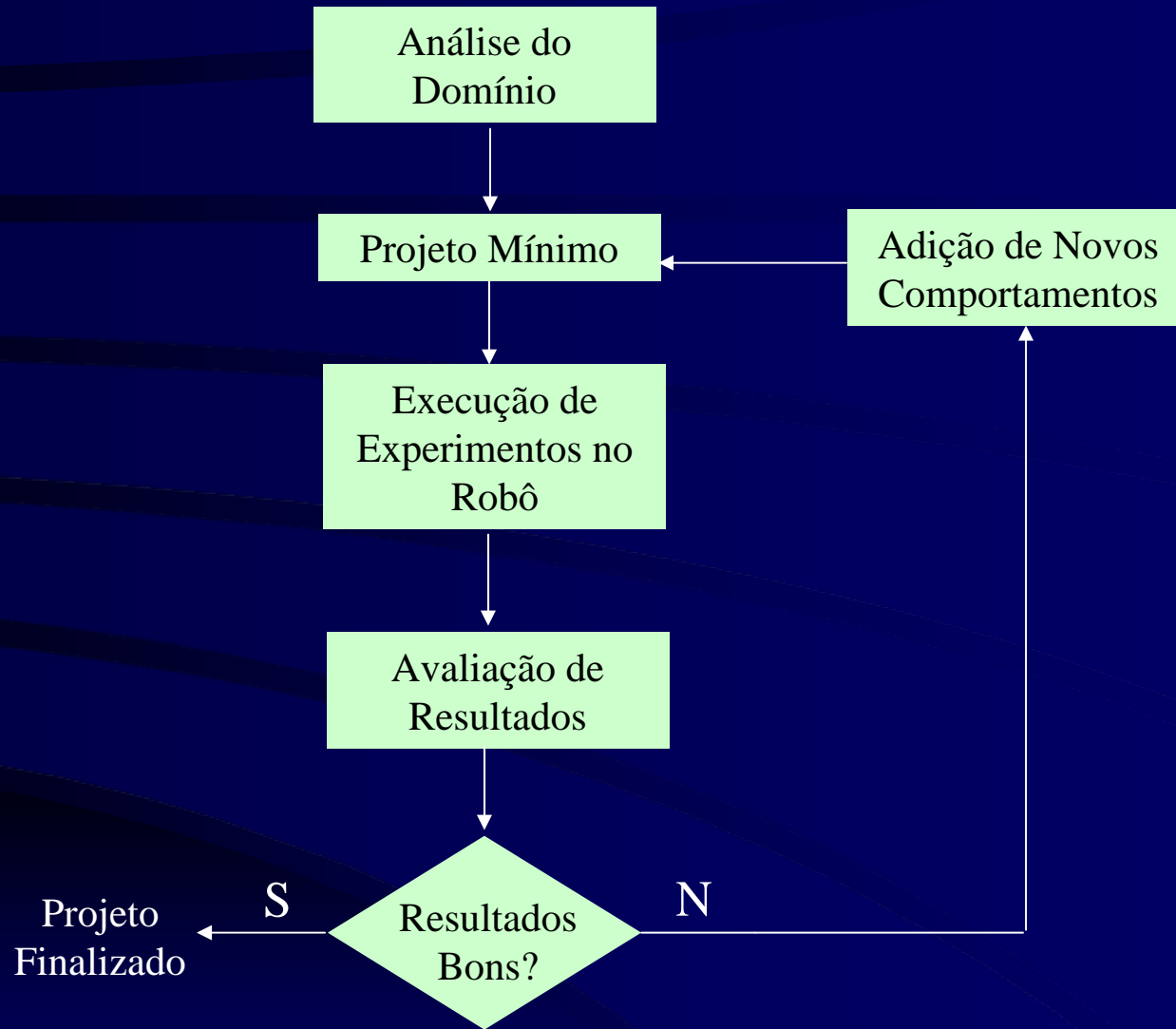
Análise experimental pode indicar necessidade de aperfeiçoamento dos comportamentos projetados



2.5.3 Métodos Guiados por Experimentação

Inicialmente: conjunto mínimo de comportamentos.

A seguir: realizam-se experimentos e novos comportamentos são adicionados para compensar defeitos daqueles previamente projetados, até que se obtenha um desempenho satisfatório.



2.6 Codificação de Comportamentos

- Elementos de um comportamento que representam sua interação com o ambiente (sensores e atuadores) devem ter seus valores produzidos (ou resultantes) definidos e quantificados de forma precisa e consistente para a realização e avaliação do projeto.
- A definição deve ser feita a partir de um conjunto de parâmetros bem definidos.
- A interação entre os elementos também deve ser representada de forma precisa e matematicamente consistente.

2.6.1 Parametrização de Ações

- Ações de locomoção objetivam transladar ou rotacionar o robô.
- Cada ação pode corresponder a integração de várias realizações de controladores de baixo nível.
- Possíveis ações: girar de 90° para a esquerda, avançar 20cm, manter velocidade constante, etc.
- Em geral, uma ação é parametrizada por uma **magnitude** e uma **direção**.

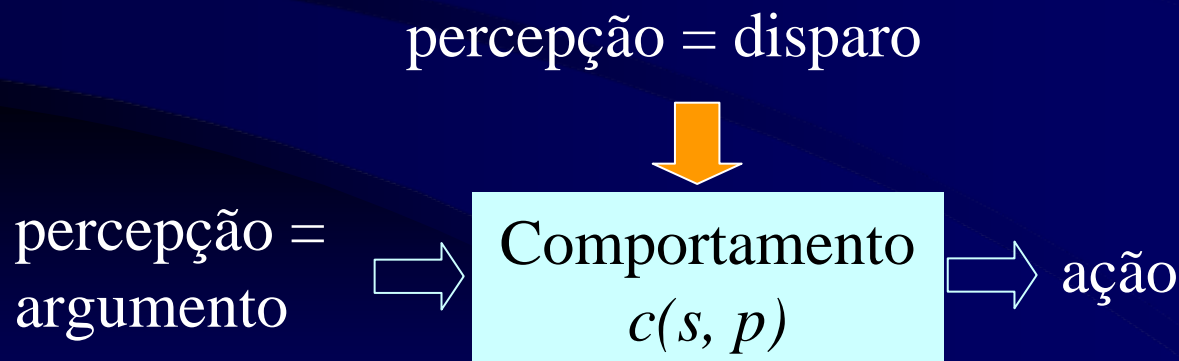
2.6.2 Parametrização de Estímulos

- Depende muito do sensor.
- Normalmente, existe um limiar L_{es} a partir do qual o estímulo é detectável pelo sensor: este limiar é uma **característica do sensor**.
- Existe também um limiar L_{sr} a partir do qual o estímulo obtido pelo sensor pode disparar (evocar) um comportamento. Este limiar é um **parâmetro do projeto**.

Sensoreamento e Disparo de Comportamentos

Duas funções possíveis para percepção sensorial:

- Argumento para a função $c(s, p)$.
- Disparo (*releasing*) de comportamentos que não utilizam necessariamente o comportamento em questão.



2.6.3 Formalização para Comportamentos e Coordenação de Comportamentos

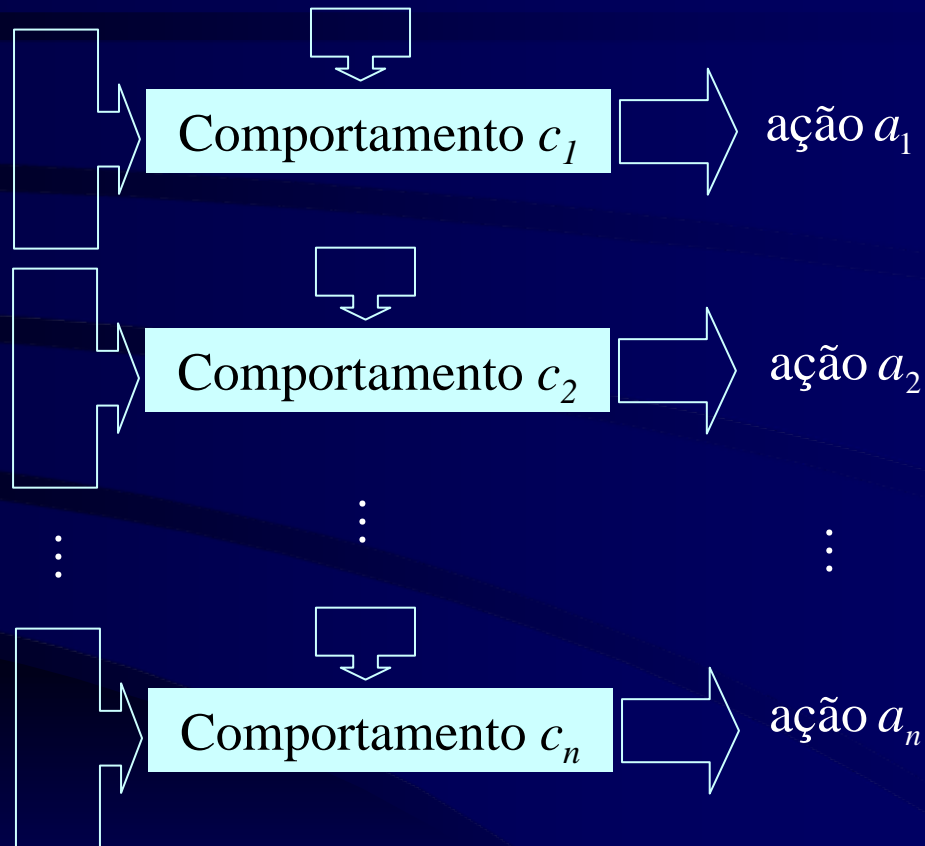
- O projeto de um robô móvel para execução de uma dada tarefa é um projeto de Engenharia.
- Projetos em Engenharia exigem uma formalização concisa e não-ambígua.
- Uma formalização corresponde a uma descrição matemática precisa, simples e suficientemente geral para permitir aplicabilidade a vários problemas.

Cada comportamento c_i mapeia um conjunto de estímulos e um plano em uma ação:



$$a_i = c_i \left(\left[s_i^1, s_i^2, \dots, s_i^{k_i} \right], p_i, w_i \right)$$

Um robô móvel atua a partir de um conjunto de n comportamentos:



Notação:

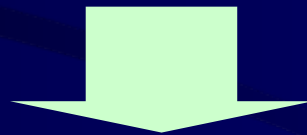
$$\begin{aligned} a_1 &= c_1 \left([s_1^1, s_1^2, \dots, s_1^{k_1}], p_1, w_1 \right) \\ a_2 &= c_2 \left([s_2^1, s_2^2, \dots, s_2^{k_2}], p_2, w_2 \right) \\ &\vdots \\ a_n &= c_n \left([s_n^1, s_n^2, \dots, s_n^{k_n}], p_n, w_n \right) \end{aligned}$$

Uma notação vetorial para os comportamentos:

$$\mathbf{S} = [s_1 \ s_2 \ \dots \ s_n]^T \quad \mathbf{P} = [p_1 \ p_2 \ \dots \ p_n]^T \quad \mathbf{W} = [w_1 \ w_2 \ \dots \ w_n]^T$$

$$\mathbf{C}(\mathbf{S}, \mathbf{P}, \mathbf{W}) = [c_1(s_1, p_1, w_1) \ c_2(s_2, p_2, w_2) \ \dots \ c_n(s_n, p_n, w_n)]^T$$

$$\mathbf{A} = [a_1 \ a_2 \ \dots \ a_n]^T$$

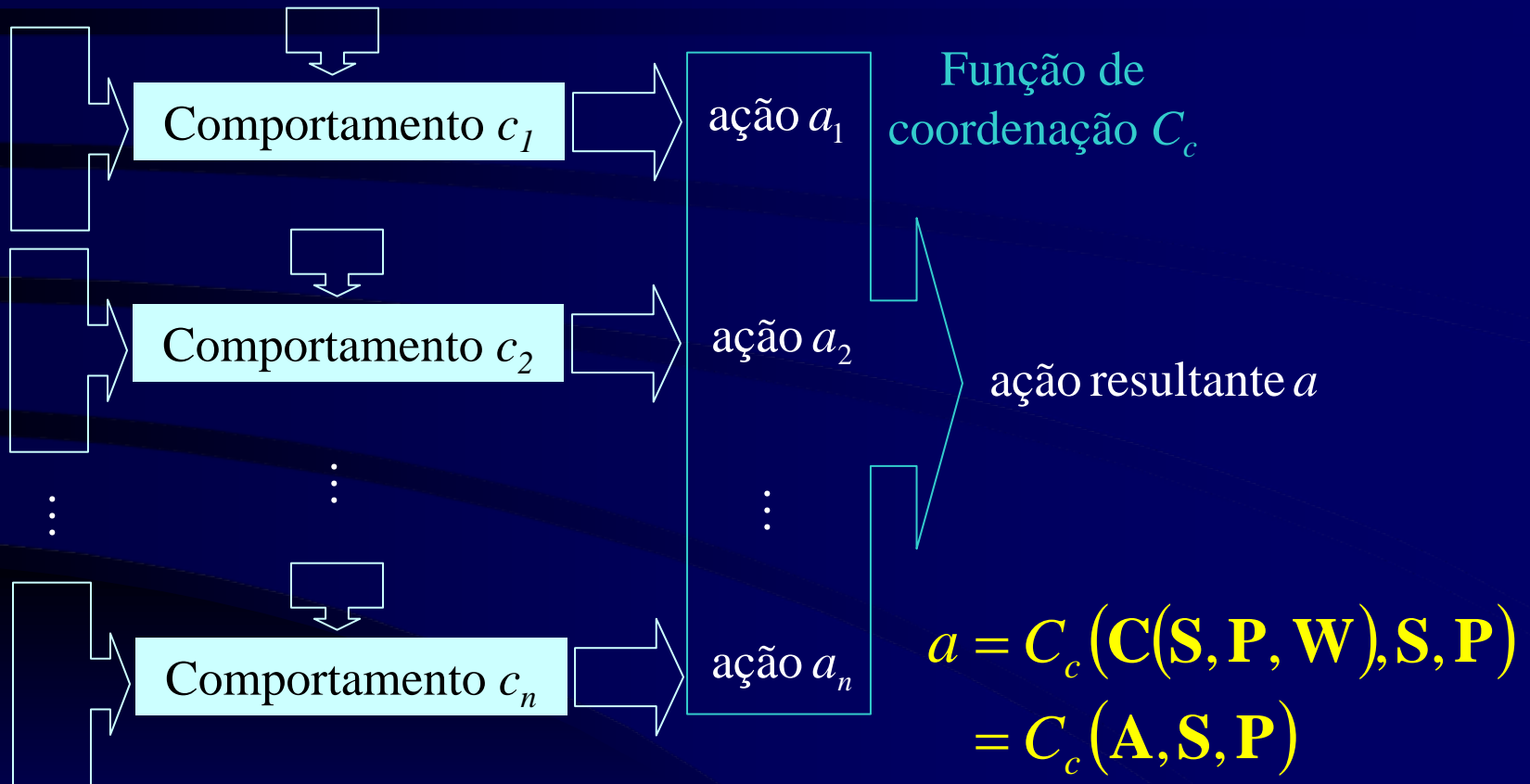


$$\mathbf{A} = \mathbf{C}(\mathbf{S}, \mathbf{P}, \mathbf{W})$$

Mapeamentos Discretos e Contínuos

- Formalização admite codificações discretas ou contínuas.
- Codificações discretas: conjunto finito de pares (estímulos sensoriais, planos) \rightarrow ação. Exemplo: coleção de regras do tipo SE-ENTÃO.
- Codificações contínuas: representação apenas através da relação funcional. Exemplo: campos potenciais.

Os n comportamentos devem ser coordenados por uma função de coordenação:



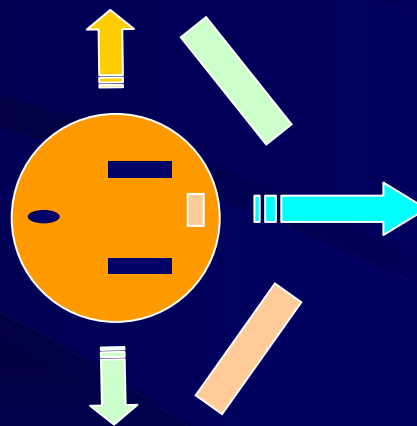
Um Exemplo Simples

- Comportamento c_1 : Evitar Obstáculos
- Comportamento c_2 : Explorar Ambiente

$$\mathbf{C}(\mathbf{S}, \mathbf{P}, \mathbf{W}) = [c_1(\mathbf{s}_1) \quad c_2(\mathbf{w}_2)]^T$$

Comportamento c_1 : Evitar Obstáculos

$$c_1(\mathbf{s}_1) = \begin{cases} \text{girar à direita, se } s_1^e > L \\ \text{girar à esquerda, se } s_1^d > L \\ \text{avançar, caso contrário} \end{cases}$$



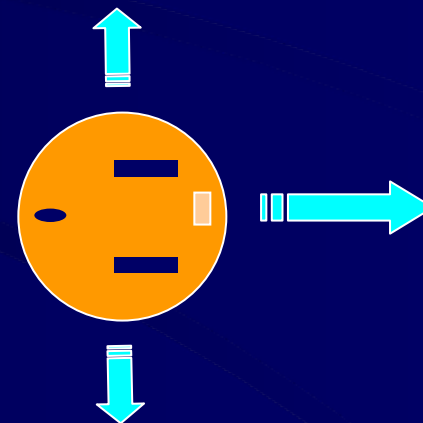
Comportamento c_2 : Explorar Ambiente

$$c_2(w_2) = \begin{cases} \text{avançar, se } w_2 \geq 1 \\ \text{girar à direita, se } 0.5 \leq w_2 < 1 \\ \text{girar à esquerda, se } w_2 < 0.5 \end{cases}$$

$$P(w_2 \geq 1) = 0.5$$

$$P(0.5 \leq w_2 < 1) = 0.25$$

$$P(w_2 < 0.5) = 0.25$$



Coordenação (exemplo 1)

$$\begin{aligned} a &= C_c(\mathbf{C}(\mathbf{S}, \mathbf{P}, \mathbf{W}), \mathbf{S}, \mathbf{P}) \\ &= C_c\left(\left[\begin{array}{c} c_1(\mathbf{s}_1) \\ c_2(w_2) \end{array}\right]^T, \mathbf{s}_1\right) = \begin{cases} c_1(\mathbf{s}_1), & \text{se } s_1^i \geq L_{sr} \\ c_2(w_2), & \text{caso contrário} \end{cases} \end{aligned}$$

Função de coordenação reutiliza a informação sensorial do comportamento c_1

Coordenação (exemplo 2)

$$\begin{aligned} a &= C_c(\mathbf{C}(\mathbf{S}, \mathbf{P}, \mathbf{W}), \mathbf{S}, \mathbf{P}) \\ &= C_c\left(\left[\begin{array}{c} c_1(\mathbf{s}_1) \\ c_2(w_2) \end{array}\right]^T\right) = \begin{cases} c_1(\mathbf{s}_1), & \text{se } c_1(\mathbf{s}_1) \neq \lambda \\ c_2(w_2), & \text{caso contrário} \end{cases} \end{aligned}$$

$$c_1(\mathbf{s}_1) \neq \lambda \Leftrightarrow s_1^i \geq L_{sr}$$

Comportamento c_1 desativado quando desnecessário \rightarrow função de coordenação não utiliza informação sensorial.

2.6.4 Funções de Coordenação de Comportamentos mais Importantes

- **Coordenação Competitiva:** a ação resultante num dado instante é selecionada a partir de uma competição entre os comportamentos ativos.
- **Coordenação Cooperativa:** a função de coordenação produz uma ação resultante para a qual contribuem todos os comportamentos ativos.

Coordenação Competitiva: Tipos de Competição

- Hierarquia Pré-Definida
- Definição Dinâmica da Hierarquia
- Baseada em Votação

Competição com Hierarquia Pré-Definida

- Hierarquia definida por plano pré-projetado.
- Forma da função de coordenação:

$$a = C_c \left(C(S, P, W), P_{prior} \right)$$

Competição com Definição Dinâmica da Hierarquia

- Hierarquia definida dinamicamente, de acordo com a informação sensorial e/ou estágio de execução da tarefa.
- Forma da função de coordenação:

$$a = C_c(\mathbf{C}(\mathbf{S}, \mathbf{P}, \mathbf{W}), \mathbf{S}, \mathbf{P})$$

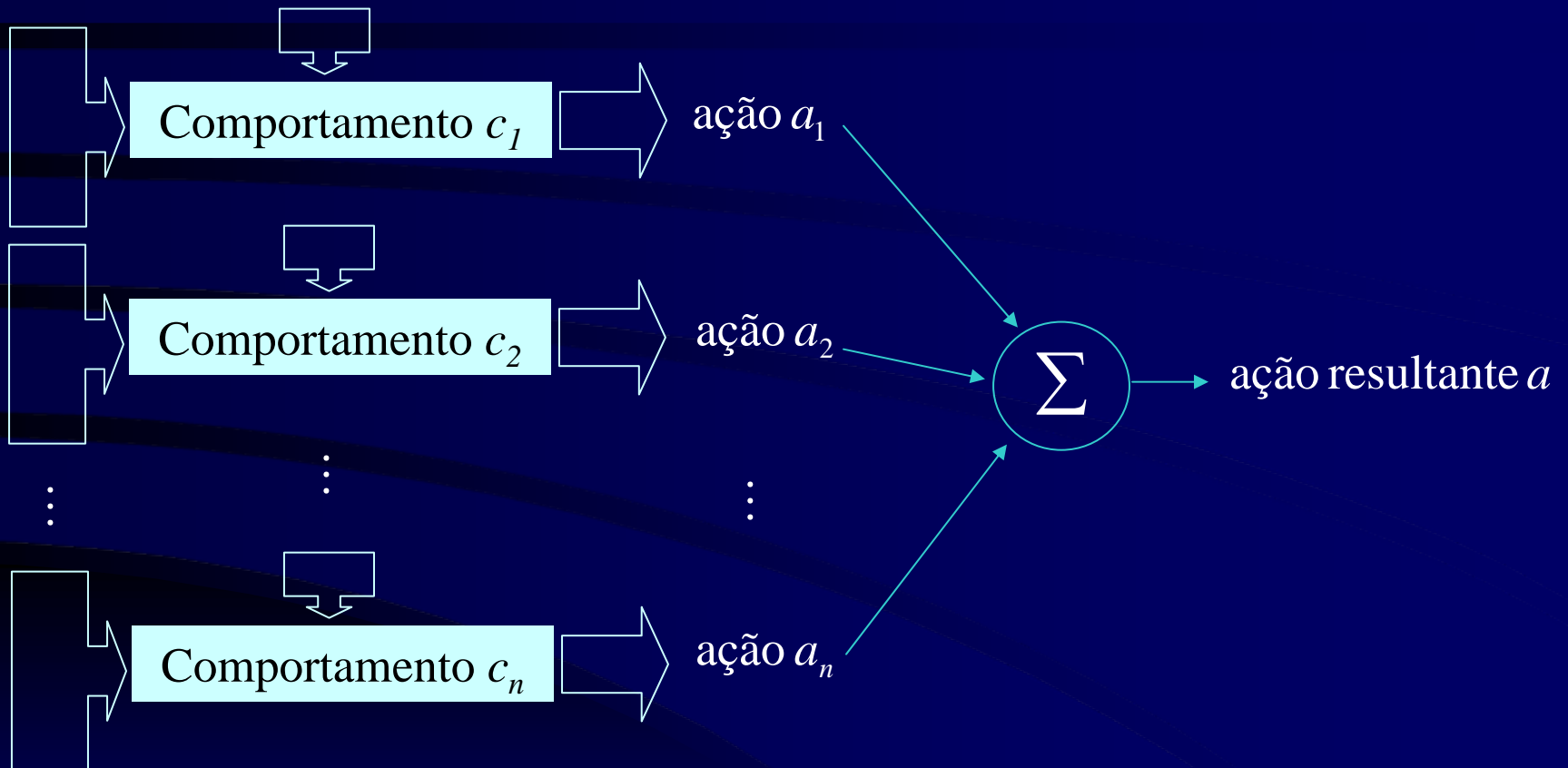
Competição Baseada em Votação

- Comportamentos ativos “votam” em ações candidatas.
- A ação mais votada é “eleita” (função de coordenação conta os votos e executa a vencedora).
- Empates: hierarquia pré-definida ou definida dinamicamente é chamada a atuar.

Coordenação Cooperativa

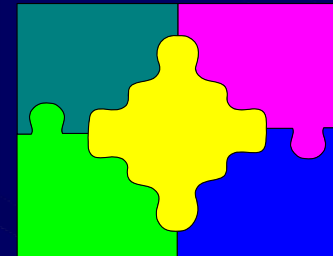
- Todos os comportamentos contribuem em algum grau para a ação resultante.
- Cooperação representada por uma operação envolvendo os comportamentos ativos.
- Exemplo: representação de comportamentos por campos potenciais.

Coordenação Cooperativa: Exemplo



O que é uma arquitetura?

- Arquiteturas definem como o robô é projetado, a partir de uma coleção de blocos fundamentais:
 - módulos correspondentes a comportamentos,
 - módulos de inibição ou ativação dos comportamentos,
 - módulos de planejamento,
 - etc.



Tipos de arquiteturas

- Reativas
- Deliberativas
- Híbridas

Definidas segundo suas *primitivas* e a dinâmica envolvida.

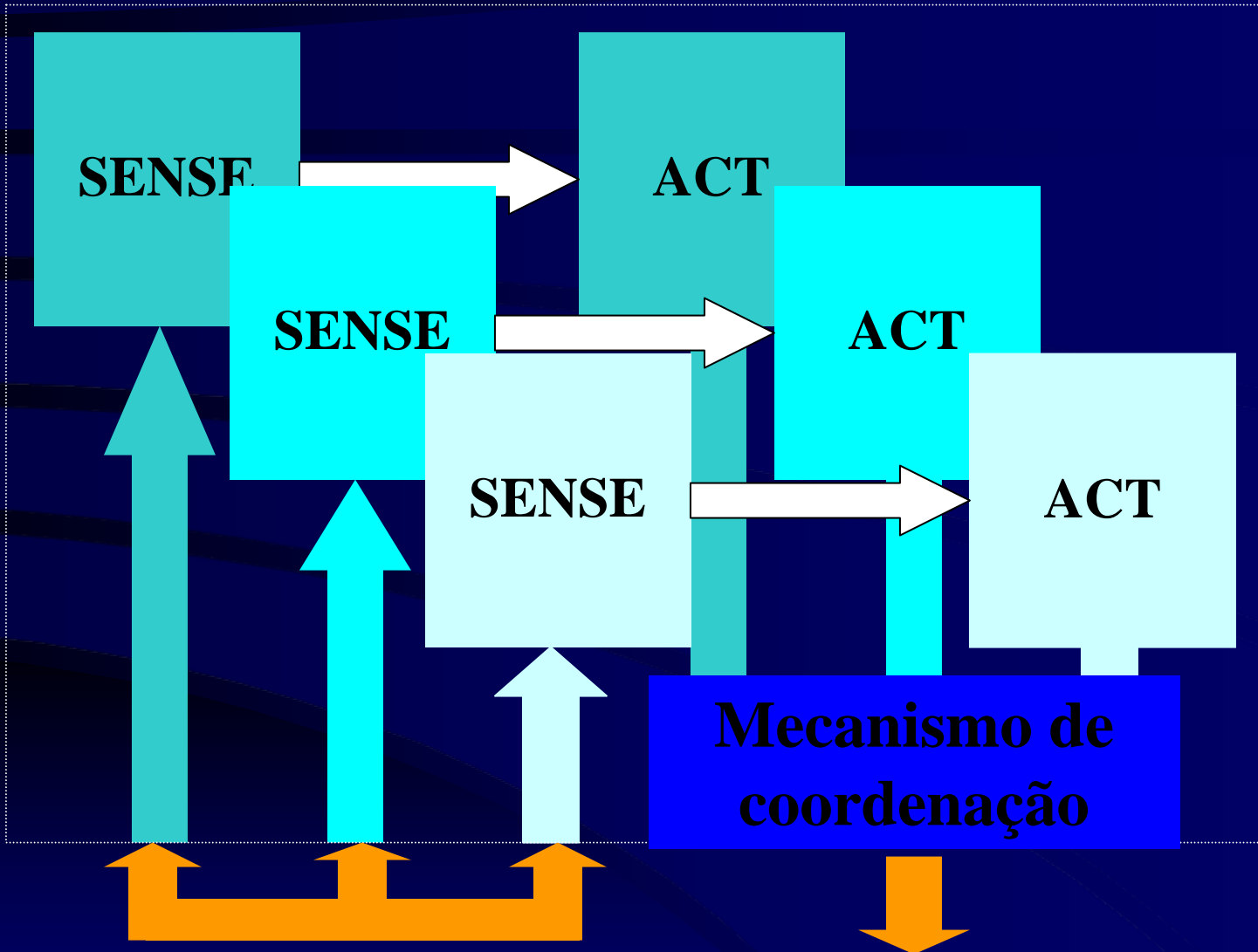
Arquiteturas: Primitivas

- PERCEPÇÃO (**SENSE**)
 - Informação obtida via sensores e adaptada para uso por estruturas internas do robô.
- PLANEJAMENTO (**PLAN**)
 - Determinação de ações a serem executadas, a partir de um *modelo interno* do robô, do mundo exterior e da tarefa (domínio).
- ATUAÇÃO (**ACT**)
 - Execução das ações através de atuadores.

1. Arquiteturas Reativas

- Surgidas no final dos anos 80.
- Fundamentadas em estudos do comportamento animal (Etologia).
- Baseadas em processamento **paralelo** (vários comportamentos simultaneamente ativos).
- Processo de percepção dependente da ação: robô constrói ou usa modelos mínimos para execução de ações.

Arquiteturas Reativas



Arquiteturas Reativas: Vantagens

- Comportamentos reativos são normalmente simples de projetar.
- Paralelismo, simplicidade de comportamentos individuais e ligação direta *SENSE-ACT* permitem operação em **tempo real**.
- Processamento **local** da informação sensorial.
- Prototipação rápida para poucos comportamentos.

Arquiteturas Reativas: Problemas

- Implementação de um grande conjunto de comportamentos é uma tarefa difícil.
- Combinação de comportamentos reativos não garantem sucesso na execução da tarefa (defensores da abordagem falam em ***Inteligência emergente***)
- Difícil definição de um conjunto mínimo de comportamentos reativos no caso geral.

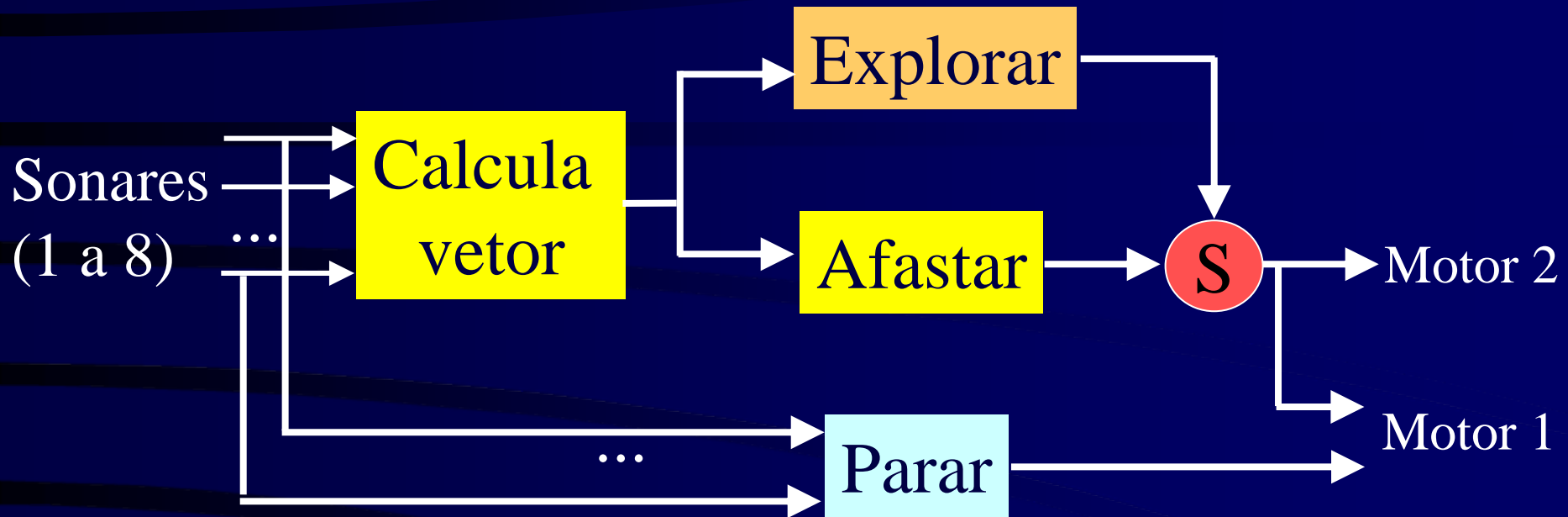
1.1. Arquitetura *Subsumption*

- Divisão hierárquica de comportamentos (**camadas de competência**).
 - Níveis mais baixos → funções de “sobrevivência”.
 - Níveis mais altos → funções dirigidas aos objetivos da tarefa.

1.1. Arquitetura *Subsumption*

- Mecanismo de prioridade de comportamentos (coordenação competitiva de comportamentos com hierarquia pré-definida):
 - Supressão: saída do nível mais alto fica atuante (nível mais baixo continua ativo, porém sua ação não produz efeito devido à supressão);
 - Inibição: comportamento do nível mais baixo é desativado pelo de nível superior.

Implementação simples na *Subsumption*



Calcula vetor: compõe as medidas num vetor v_{cv} que indica posição e distância de um obstáculo (imaginário).

Explorar: compõe um vetor aleatório com o oposto de v_{cv} .

Afastar: 1. vira para direção oposta a v_{cv} e 2. avança em distância proporcional à magnitude de v_{cv} .

Parar: pára o robô (ou **Parar** ou **Calcula_vetor** fica ativo)

1.2. Arquitetura Baseada em Campos Potenciais

- Comportamentos num mesmo nível de prioridades, sempre ativos.
- Cada comportamento produz uma ação, representada vetorialmente (**campos potenciais**).
- Coodenação **cooperativa** de comportamentos (dada pela soma vetorial).
- Ação executada: resultante da soma vetorial de todas as ações.

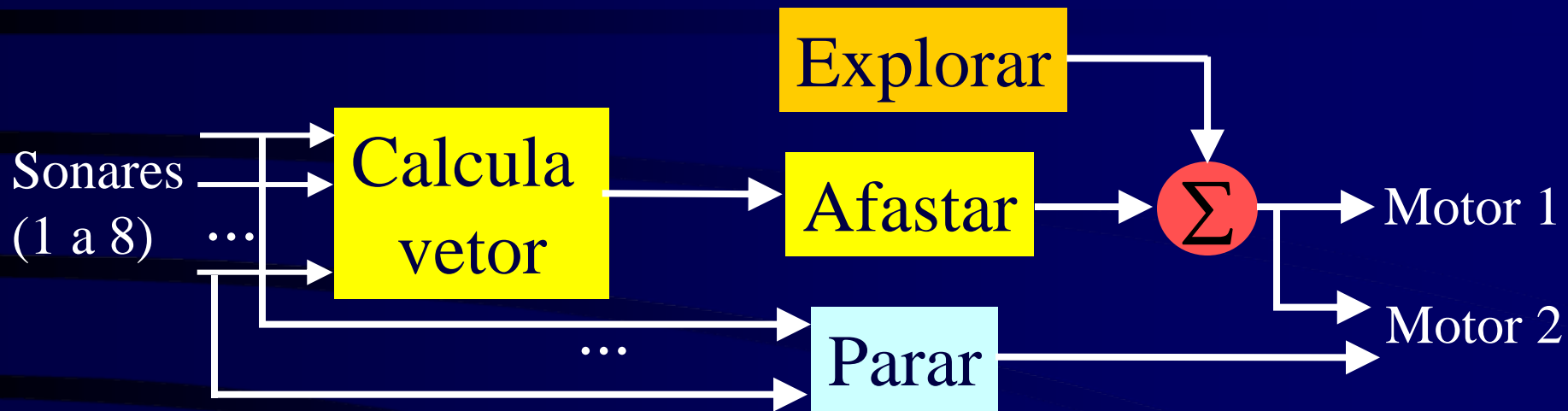
Tipos de Campos Potenciais

- **Uniforme**: mesma magnitude e direção.
 - Ex.: avançar.
- **Perpendicular** a alguma fronteira.
 - Ex.: evitar paredes.
- **Atrativo**: vetores com magnitude inversamente proporcional à distância do ponto atrator e em direção ao atrator.
 - Ex.: tropismo.

Tipos de Campos Potenciais

- **Repulsivo**: oposto do campo atrativo.
 - Ex.: evitar obstáculos.
- **Tangencial**: vetores perpendiculares a linhas radiais a partir do centro de um objeto.
 - Ex.: margear obstáculos.

Implementação na Arquitetura Baseada em Campos Potenciais



Explorar: direção aleatória de movimento.

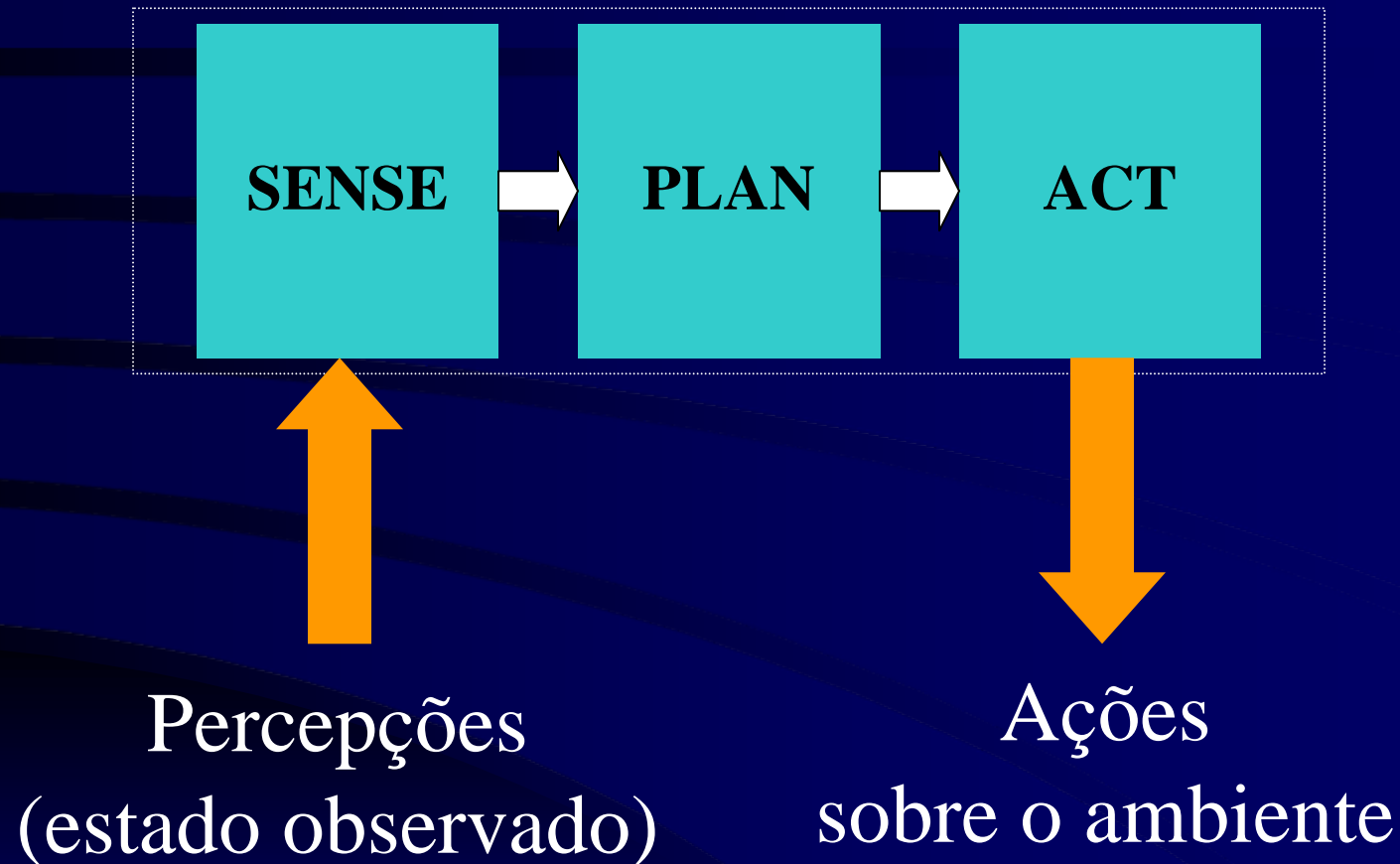
Afastar: 1. vira para direção oposta a v_{cv} e 2. avança em distância proporcional à magnitude de v_{cv} (sendo v_{cv} a saída de Calcula_vetor)

Parar: pára o robô (comportamento emergencial).

2. Arquiteturas Deliberativas

- Usada no primeiro robô inteligente (Shakey, 1967).
- Principal paradigma até o final dos anos 80.
- Baseada em processamento **sequencial**.
- Processo de percepção monolítico, independente da tarefa: robô constrói ou usa um modelo do “mundo” (mundo externo + robô)

Arquiteturas Deliberativas



Arquiteturas Deliberativas: Vantagem

- Ordenação sequencial facilita sub-projetos independentes.
- Geralmente possui **controle hierárquico**.
 - Planejamento subdividido em módulos.
 - Do nível superior ao inferior, cresce a **restrição de tempo de resposta** e diminui o **espaço físico de interesse**.

Planejamento Hierárquico



Arquiteturas Deliberativas: Problemas

- Criação do modelo do “mundo” tem custo computacional alto, e pode não ser necessário.

Exemplo: para abrir uma porta específica, não preciso ter o modelo de todas as outras portas, janelas, móveis, etc do ambiente.

- Separação SENSE/ACT (pelo PLAN) pode ser artificial: elimina possibilidade de ações do tipo estímulo-resposta.
- Modelagem da incerteza (ruído em sensores e atuadores) é complicada.

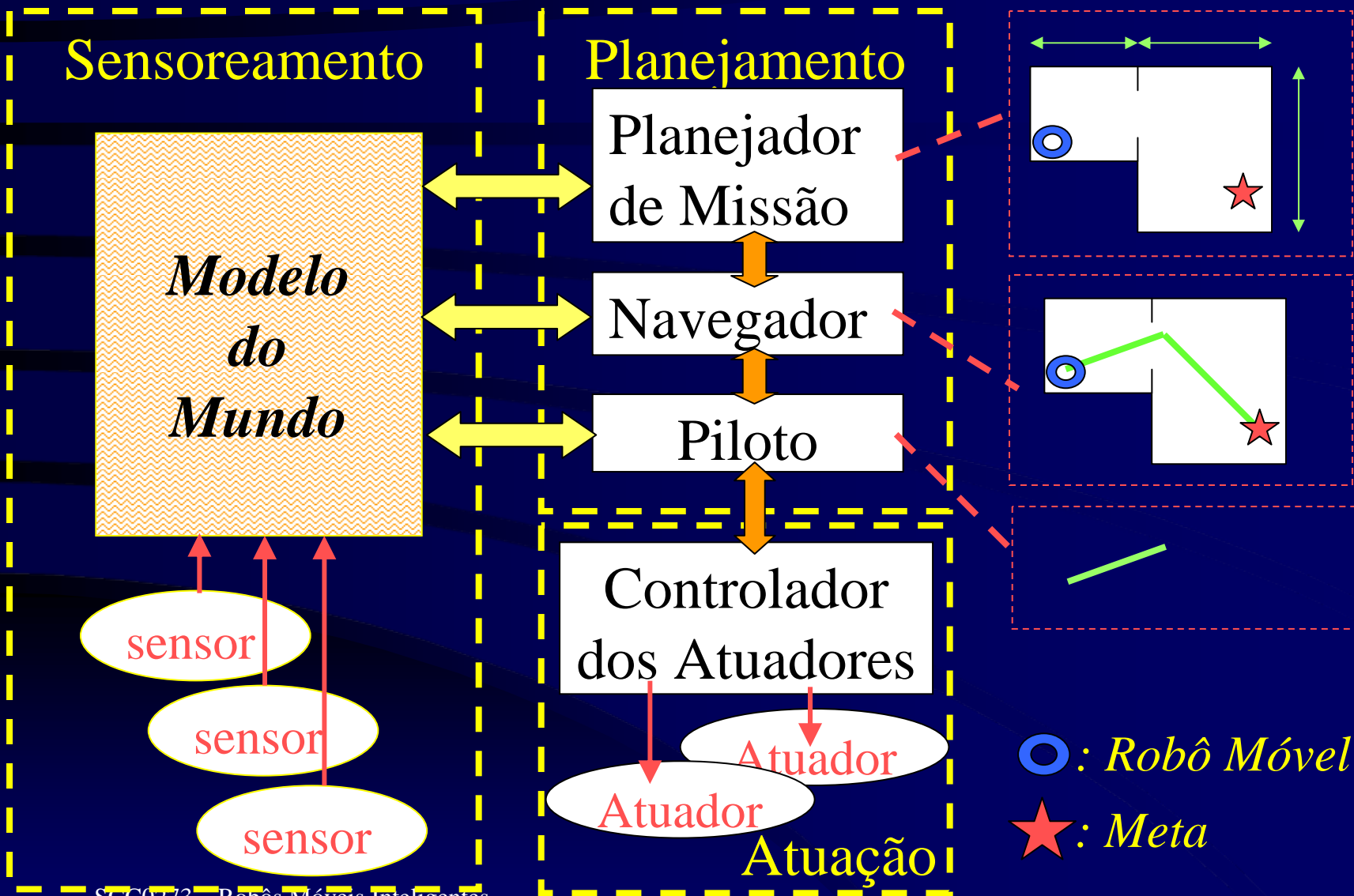
Arquiteturas Deliberativas

- Exemplos:
 - Arquitetura NHC - **N**ested **H**ierarquical **C**ontroller
 - Arquitetura RCS - **R**ead-time **C**ontrol **S**ystem

2.1. Arquitetura NHC

- **N**ested **H**ierarquical **C**ontroller [Meystel 1990]
- Uso em robôs móveis em tarefas de navegação.
- Funcionamento:
 1. Robô coleta informação do ambiente (constrói mapa);
 2. Após construção do modelo, planeja:
 - **missão**: desmembra em partes (localização atual + submetas)
 - **navegador**: encontra caminho até a meta, definido por segmentos (envia 1 segmento por vez ao módulo inferior)
 - **piloto**: gera ações específicas (direção, velocidade, etc) para seguir a trajetória.
 - a cada ação, percepção monitora execução.

Arquitetura NHC



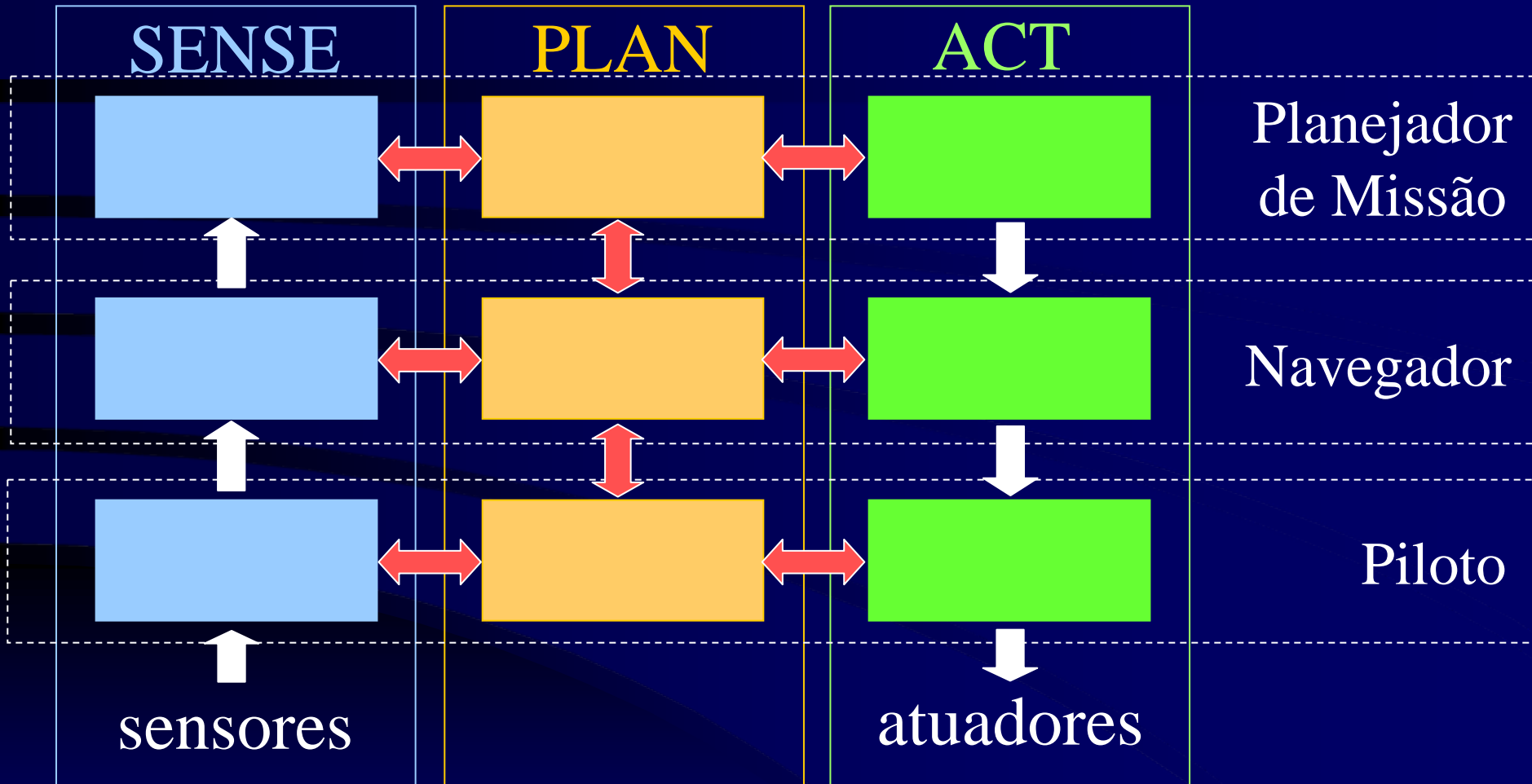
Arquiteturas Deliberativas: Exemplo RCS

- Real-time Control System [Albus 1997]
- Desenvolvida no NIST – *National Institute of Standards and Technology*:
 - para auxiliar empresas de manufatura a construir robôs móveis inteligentes.
- Utilização:
 - Robôs comerciais de limpeza de chão.
 - Robô para mineração.
 - Robô para exploração subaquática.

Arquiteturas Deliberativas: Exemplo RCS

- Baseado na NHC, porém:
 - Possui módulos que auxiliam a inserção de inteligência (interação humana).
 - O modelo de mundo é organizado em uma hierarquia, com níveis de abstração.
 - O módulo de planejamento testa o plano gerado (simula) antes de transformá-lo em ações.

Arquitetura RCS



Arquiteturas Híbridas

- Incorporam um elemento de planejamento sobre a definição e seleção de comportamentos reativos.
- Corresponde a uma arquitetura reativa controlada por um plano de execução e sequenciamento.
- Permitem a reconfiguração dinâmica de sistemas de controle reativo.

Arquiteturas Híbridas: Vantagem

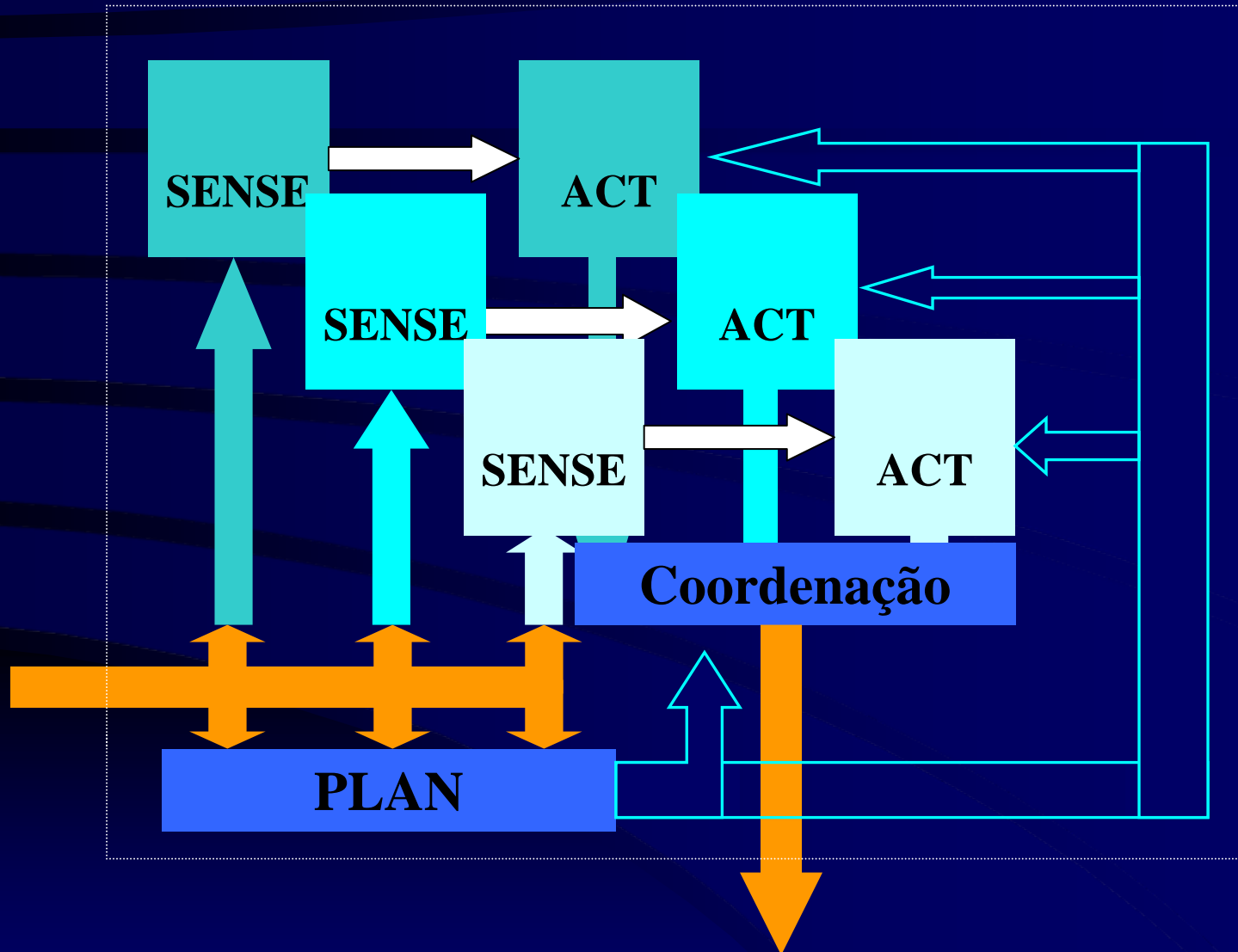
A integração de deliberação e controle reativo devem ser levadas em conta para produzir um sistema flexível robusto e inteligente.

Arquiteturas Híbridas: Problema

A integração de deliberação e controle reativo é um problema complexo.

- Se tentar modelar e pré-planejar todas as eventualidades corre-se o risco de que o processo de planejamento nunca termine.
- Por outro lado, não é seguro para o robô fazer suposições grosseiras sobre o mundo, que não reflitam sua natureza dinâmica.

Arquiteturas Híbridas



Arquiteturas Híbridas: Componentes

- **Sequenciador**: define o conjunto de comportamentos reativos para uma subtarefa.
- **Gerenciador de Recursos**: aloca recursos (sensores, tempo, ...)
- **Cartógrafo**: mantém um mapa ou informação espacial do ambiente.
- **Planejador de Missões**: define um plano para execução da tarefa (subplanos).
- **Monitoramento de Desempenho**: Avalia o desempenho do robô na execução da tarefa.

Arquiteturas Híbridas: Ex. AuRA

- **A**utonomous **R**obot **A**rchitecture [Arkin 1986]
- É um sistema híbrido deliberativo/reativo que configura o sistema de controle reativo antes da execução e o reconfigura para eventuais falhas do sistema.

Arquiteturas Híbridas: Ex. AuRA

Componente Hierárquico



- **Planejador da missão**: metas de alto nível e restrições.
- **Raciocinador Espacial**: através de um cartógrafo constrói planos.
- **Seqüenciador do plano**: traduz cada passo como um comportamento reativo.
- **Controlador Schema**: é responsável por controlar e monitorar os processos comportamentais em tempo de execução.

Componente Reativo

Arquiteturas Híbridas: Ex. Atlantis



Ex.: Sojourner



- **Gat, 1991**
- **Deliberador**: planejamento e modelagem do mundo.
- **Sequenciador**: inicia e termina atividades de baixo nível e contorna falhas do sistema reativo para completar a tarefa.
- **Controlador reativo**: um gerenciador de uma coleção de atividades primitivas