



**USP - ICMC - SSC  
SSC 0714 (RMA) - 1o. Semestre 2010**

**Disciplina de  
Robôs Móveis Autônomos  
SSC-0714**

**Prof. Fernando Santos Osório**

**Email: fosorio [at] { icmc. usp. br , gmail. com }**

**Estagiário PAE: Maurício Acconcia Dias - maccdias [at] gmail.com**

**Web: <http://www.icmc.usp.br/~fosorio/>**

**Wiki ICMC: <http://wiki.icmc.usp.br/index.php/SSC-714>**

**Aula 04 – Arquiteturas de Controle**

**Agenda:**

**Arquiteturas de Controle de Robôs Autônomos**

- 1. Controle e Autonomia**
- 2. Arquitetura de Controle Reativa**  
**Modelo Sensorial-Motor**  
**Controle em Tarefas Reativas**
- 3. Arquitetura de Controle Deliberativa**  
**Mapas e Planejamento de Trajetórias**
- 4. Exemplos de Aplicações**
- 5. Vantagens e Desvantagens**

## Aula 04 – Arquiteturas de Controle

Robôs Móveis Autônomos: Percebe => Decide => Age  
Agentes Autônomos dotados de SENSORES e ATUADORES



3  
Março 2010

Controle e Autonomia

## Aula 04 – Arquiteturas de Controle

Controle e Autonomia

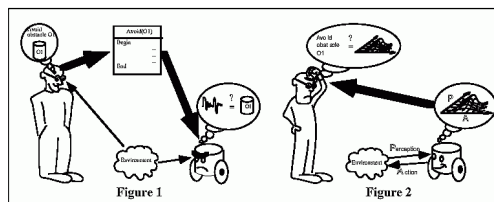
Robôs Móveis:  
Agentes Autônomos dotados de SENSORES e ATUADORES



Como Agir?

Como Interpretar  
as Percepções?

Como Tomar Decisões?



4  
Março 2010

### Controle e Autonomia

#### Arquiteturas de Controle Robótico

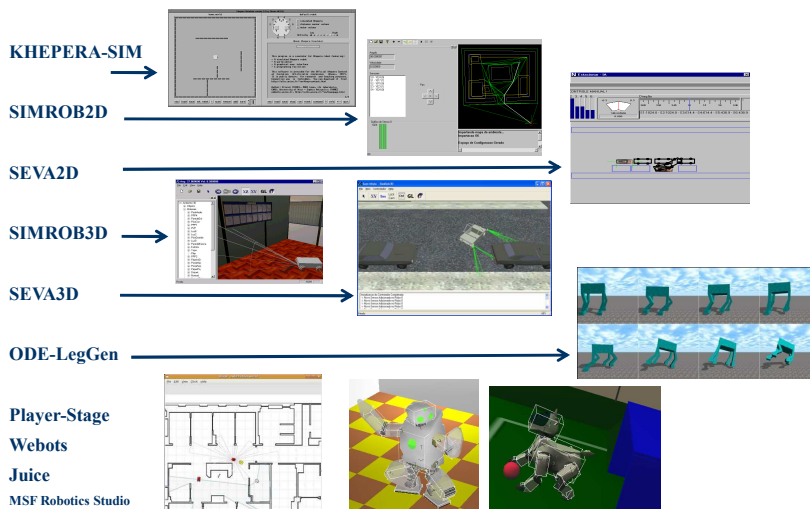
- Arquitetura Reativa
- Arquitetura Deliberativa
- Arquitetura Hierárquica
- Arquitetura Híbrida

*Controle* => Tomada de Decisão para Realizar Ações

*Arquitetura* => Modelo de Tomada de Decisão

*Autonomia* => Independência e Robustez na Tomada de Decisões

### Simuladores: Modelos Sensores / Atuadores e Controle



## Arquitetura de Controle Reativo

### Controle Reativo

**Controle:** Percepção [ => Decisão ] => Ação

**Reativo:** Percepção => Ação

- Reage diretamente aos estímulos externos;
- Esquema sensorio-motor;

**Comportamentos e Tarefas típicas:** *Reactive Behaviour*

- Vagar pelo ambiente, evitando colisões e obstáculos;
- Acompanhar uma parede ou corredor;
- Comportamento direcionado pela luz;
- Ir em direção a uma determinada orientação

**Composição de Comportamentos:** Direção x Obstáculo

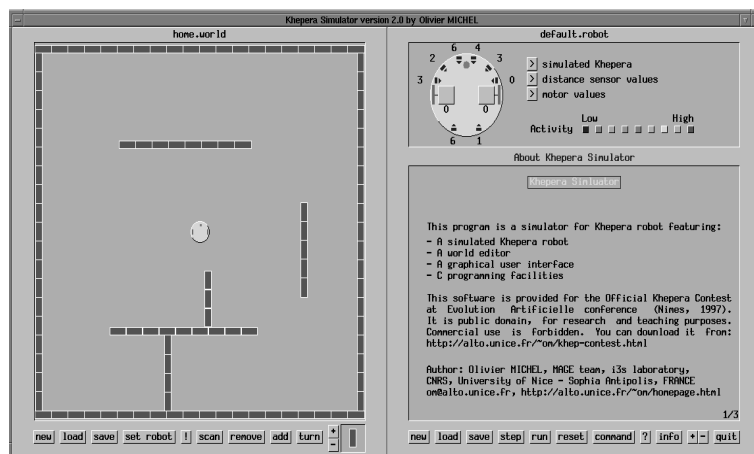
7

Março 2010

## Arquitetura de Controle: Reativo

### Simulador Khepera-SIM: Controle Reativo

Simulador do Khepera / SIM 2.0 Unix / Olivier Mitchell / INRIA Sophia Antipolis



Sensores: 8 IR / Atuadores: 2 motores com cinemática diferencial

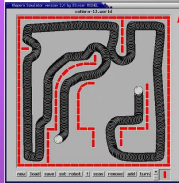
1997/98

8

Março 2010

## Arquitetura de Controle: Reativo

Simulador do Khepera



Robô Khepera



9  
 Março 2010

> 2 DC brushed servo motors with incremental encoders  
 > 8 infrared proximity and ambient light sensors (SFH900)

## Arquitetura de Controle: Reativo

### CONTROLE: Arquiteturas REATIVAS

- Reativo: Integração Sensorial-Motora

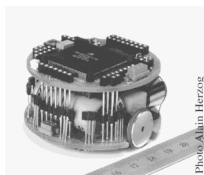
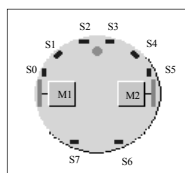
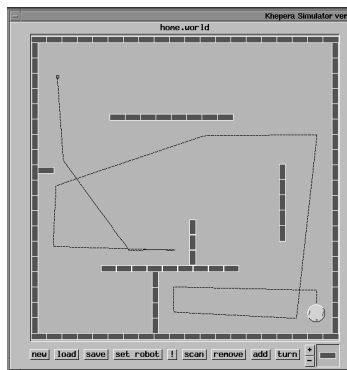


Photo: Alain Herzog



#### Controle Reativo

IF S1 < Limite and  
 S2 < Limite and  
 S3 < Limite and  
 S4 < Limite  
 THEN Action (Go\_Forward)

IF S1 < Limite and  
 S2 < Limite and  
 S3 > Limite and  
 S4 > Limite  
 THEN Action(Turn\_Left)

IF S2 > Limite and  
 S3 > Limite and  
 S2 > S3 and  
 S1 > S4  
 THEN Action(Turn\_Right)

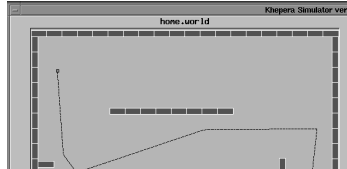
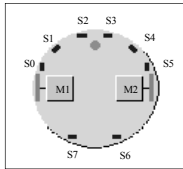
Evitar colisões e obstáculos  
 Sensorial-Motor: Sentir => Agir

10  
 Março 2010

## Arquitetura de Controle: Reativo

### CONTROLE: Arquiteturas REATIVAS

#### • Reativo: Integração Sensorial-Motora



Sensores: 8 - S0 à S7  
 Comandos: 3 ações (L=Turn Left, F=Forward, R=Turn Right)

S0	S1	S2	S3	S4	S5	S6	S7	L	F	R
4	1	3	4	6	3	0	4	0	1	0
1	3	6	3	6	6	6	6	0	1	0
6	3	104	234	772	96	3	6	1	0	0
2	6	104	229	724	107	3	4	1	0	0
563	1023	57	6	6	0	3	1	0	0	1
544	1023	1	3	4	0	5	1	0	0	1

#### Controle Reativo

IF S1 < Limite and  
 S2 < Limite and  
 S3 < Limite and  
 S4 < Limite  
 THEN Action (Go\_Forward)

IF S1 < Limite and  
 S2 < Limite and  
 S3 > Limite and  
 S4 > Limite  
 THEN Action(Turn\_Left)

IF S2 > Limite and  
 S3 > Limite and  
 S2 > S3 and  
 S1 > S4  
 THEN Action(Turn\_Right)

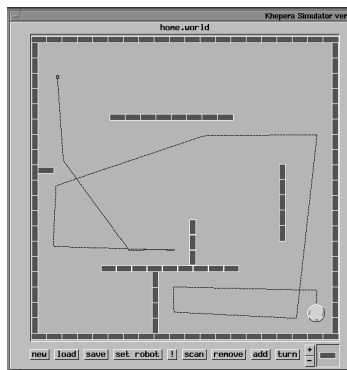
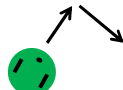
11

Março 2010

## Arquitetura de Controle: Reativo

### CONTROLE: Arquiteturas REATIVAS

#### • Reativo: Integração Sensorial-Motora



Evitar colisões e obstáculos  
 Sensorial-Motor: Sentir => Agir

#### Controle Reativo

IF S1 < Limite and  
 S2 < Limite and  
 S3 < Limite and  
 S4 < Limite  
 THEN Action (Go\_Forward)

IF S1 < Limite and  
 S2 < Limite and  
 S3 > Limite and  
 S4 > Limite  
 THEN Action(Turn\_Left)

IF S2 > Limite and  
 S3 > Limite and  
 S2 > S3 and  
 S1 > S4  
 THEN Action(Turn\_Right)

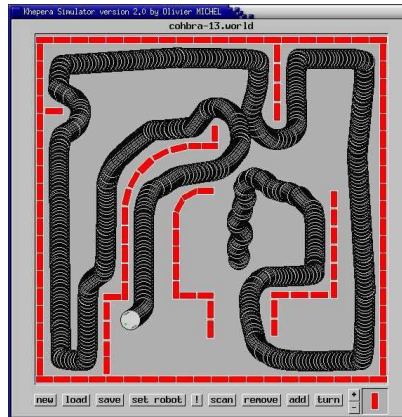
12

Março 2010

## Arquitetura de Controle: Reativo

### CONTROLE: Arquiteturas REATIVAS

#### • Reativo: Integração Sensorial-Motora



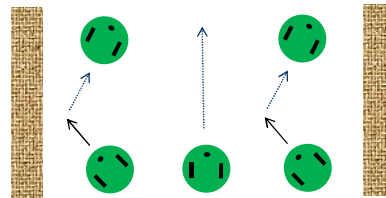
Seguir o contorno de uma parede  
Sensorial-Motor: Sentir => Agir

#### Controle Reativo

Se Longe\_Demais(Dist\_Parede)  
Então Gira\_Direção(Lado\_da\_Parede);

Se Próximo\_Demais(Dist\_Parede)  
Então Gira\_Direção(Lado\_Oposto\_da\_Parede);

Se Distância\_Adequada(Dist\_Parede)  
Então Move\_Para\_Frente();

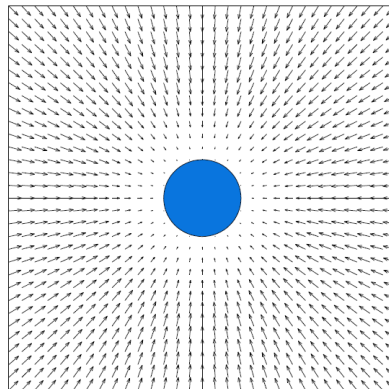


## Arquitetura de Controle: Reativo

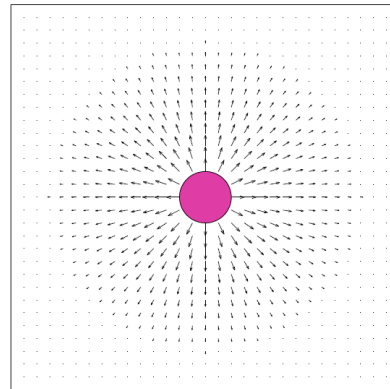
### CONTROLE: Arquiteturas REATIVAS\*

#### • Reativo: Integração Sensorial-Motora • Técnica: Campos Potenciais

Evitar colisões e obstáculos  
Seguir em direção a um alvo  
Sensorial-Motor: Sentir => Agir  
\* Reativo "não puro" baseado em  
informações externas



Força de Atração



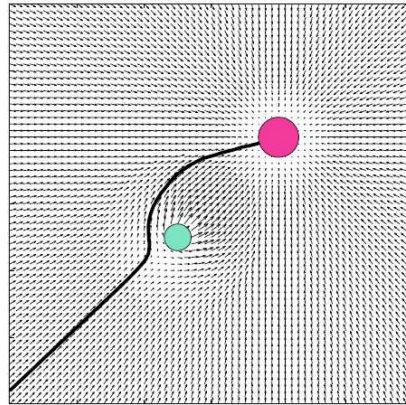
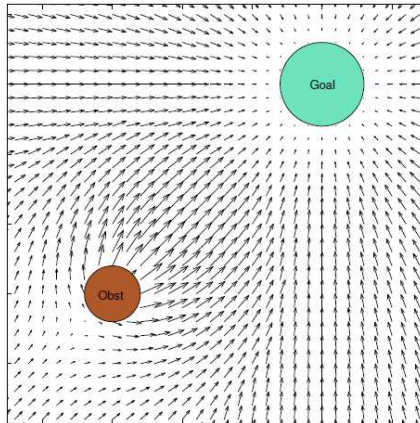
Força de Repulsão

## Arquitetura de Controle: Reativo

**CONTROLE:** Arquiteturas **REATIVAS\***

- Reativo: Integração Sensorial-Motora
- Técnica: Campos Potenciais

Evitar colisões e obstáculos  
 Seguir em direção a um alvo  
 Sensorial-Motor: Sentir => Agir  
 \* Reativo “não puro” baseado em informações externas



15  
 Março 2010

Forças Combinadas: Força de Atração + Força de Repulsão

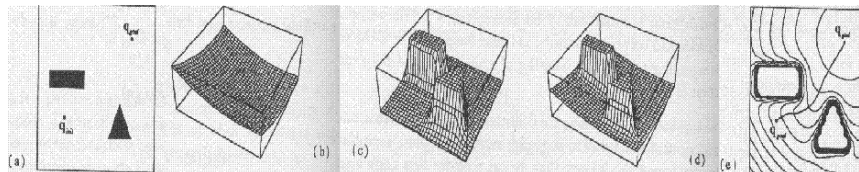
Potential Fields Tutorial  
 Michael A. Goodrich

## Arquitetura de Controle: Reativo

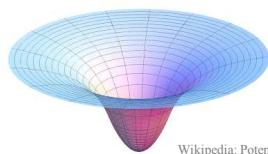
**CONTROLE:** Arquiteturas **REATIVAS\***

- Reativo: Integração Sensorial-Motora
- Técnica: Campos Potenciais

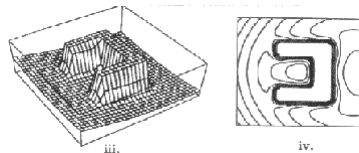
Evitar colisões e obstáculos  
 Seguir em direção a um alvo  
 Sensorial-Motor: Sentir => Agir  
 \* Reativo “não puro” baseado em informações externas



[http://www.scf.usc.edu/~peiyinge/gra\\_planning.html](http://www.scf.usc.edu/~peiyinge/gra_planning.html)



Wikipedia: Potential Filed



Houston, we have a problem!

16  
 Março 2010

Forças Combinadas: Força de Atração + Força de Repulsão



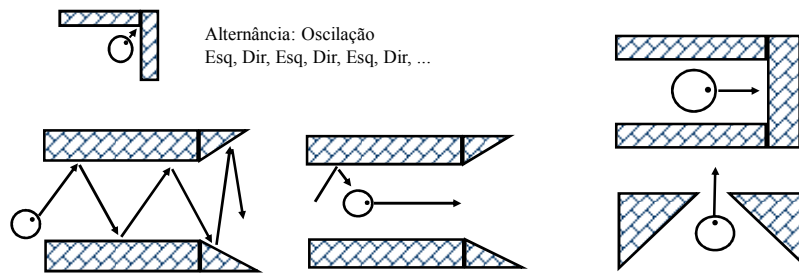
## Arquitetura de Controle: **Reativo**

### CONTROLE: Arquiteturas **REATIVAS\***

- **Reativo: Integração Sensorial-Motora**
- **Técnicas:**
  - Desvio reativo de Obstáculos
  - Campos Potenciais
- **Problemas...**

Evitar colisões e obstáculos  
Seguir em direção a um alvo  
Sensorial-Motor: Sentir => Agir  
\* Reativo "não puro" baseado em informações externas **LOCAIS**

Houston, we have a problem!  
>> Mínimos Locais  
>> Passado, Memória e Contexto

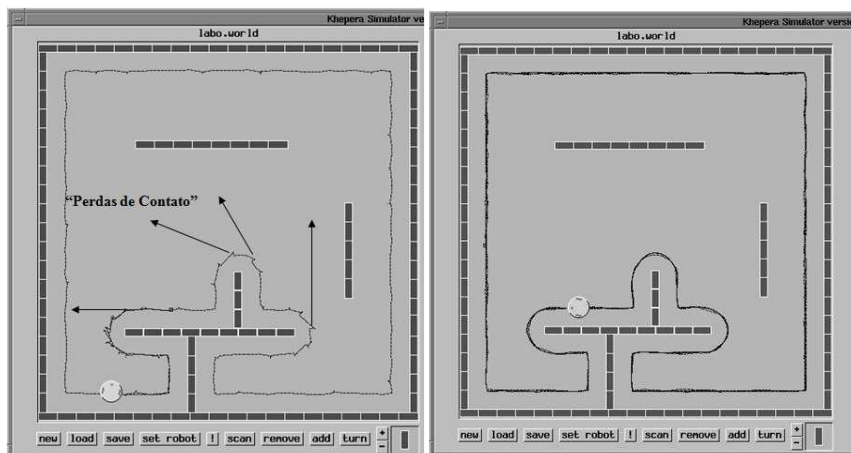


## Arquitetura de Controle: **Reativo**

### CONTROLE: Arquiteturas **REATIVAS\***

- **Reativo: Integração Sensorial-Motora**
- **Problemas...**

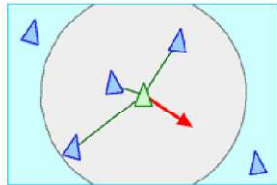
Houston, we have a problem!  
>> Mínimos Locais  
>> **Passado, Memória e Contexto**



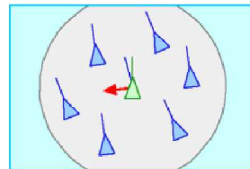
## Arquitetura de Controle: Reativo

### CONTROLE: Arquiteturas REATIVAS

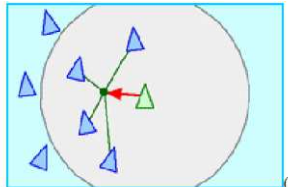
- Reativo: Múltiplos Robôs => Boids / Steering [Craig Reynolds]



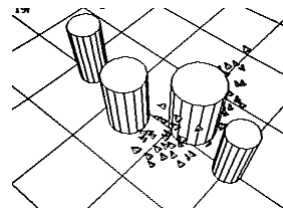
(a) **Separation:** Direcionar o movimento dos agentes para evitar um aglomeramento local dos membros do grupo.



(b) **Alignment:** Direcionar o movimento de modo a seguir o alinhamento médio (*heading*) dos membros do grupo.



(c) **Cohesion:** Direcionar o movimento de modo a ir em direção a posição central média dos membros do grupo.

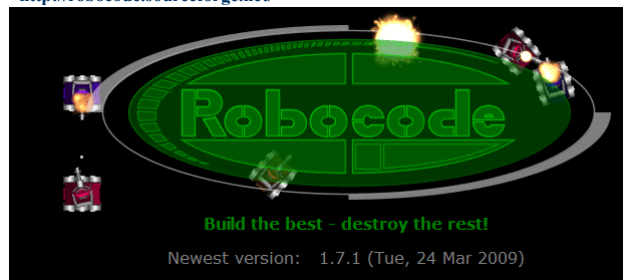


## Arquitetura de Controle: Reativo

### CONTROLE: Arquiteturas REATIVAS\*

- Reativo: *Em Direção ao Deliberativo...* **ROBO CODE**

<http://robocode.sourceforge.net/>



```
/**  
 * Fire when we see a robot  
 */  
public void onScannedRobot(ScannedRobotEvent e)  
{  
    fire(1);  
}
```

```
/**  
 * We were hit! Turn perpendicular to the bullet,  
 * so our seesaw might avoid a future shot.  
 */  
public void onHitByBullet(HitByBulletEvent e)  
{  
    turnLeft(90 - e.getBearing());  
}
```

### Controle Deliberativo

**Controle:** Percepção => [Conhecimento e Decisão] => Ação

**Deliberativo:** Percepção => Planeja, Delibera => Ação

- Possui *conhecimento* sobre a situação do robô e do ambiente;
- Usualmente baseado no uso de mapas e planejamento de trajetórias.

**Tarefas típicas:** *Behaviour : Task Planning, Action Sequence*

- Execução de scripts de ações planejadas previamente;
- Executar uma seqüência de ações previamente determinada;
- Seguir trajetórias especificadas com uso de mapas;
- Execução de Tarefas de Alto Nível;

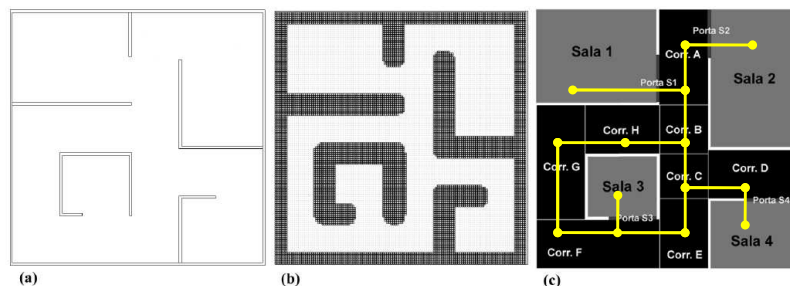
21

Março 2010

### Controle Deliberativo

**MAPAS DO AMBIENTE:** Métrico / Topológico

- Mapa Métrico do tipo GRADE (Grid, Mapa de Ocupação)
- Mapa Métrico do tipo GEOMÉTRICO (Geometria do Ambiente)



22

Março 2010

Farlei Heinen, 2002

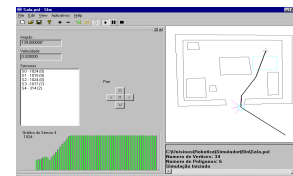
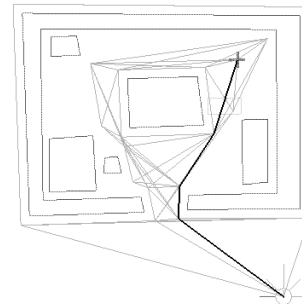
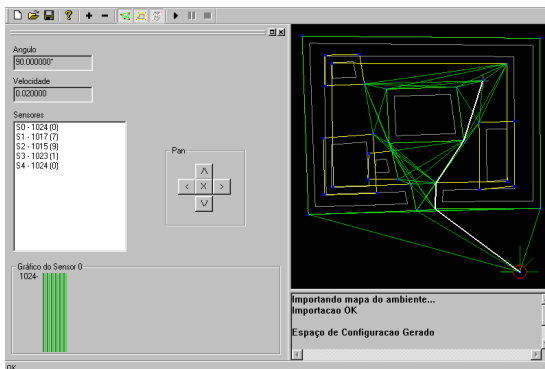
Geométrico (a); Baseado em Grade (b); Topológico/semântico (c);

### SIMROB2D

#### Referência:

Farlei Heinen (Orientador: Fernando Osório )  
Robótica Autônoma: A integração entre planificação e comportamento reativo. 2000.

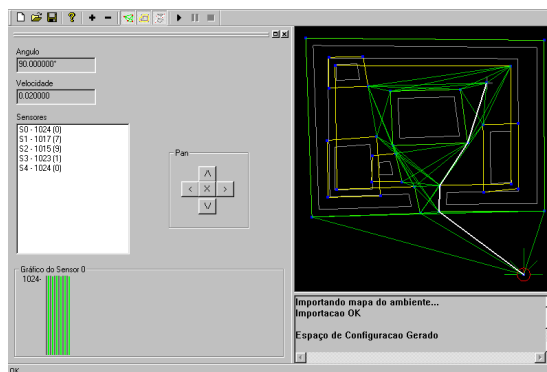
Sensores: 5 IR / Atuadores: 2 motores (diferencial)



23

Março 2010

### SIMROB2D



Farlei Heinen, 2000

#### Planejando a Trajetória do Robo com o uso do Mapa

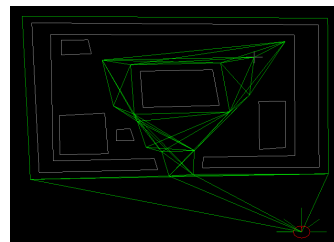
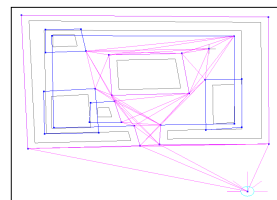
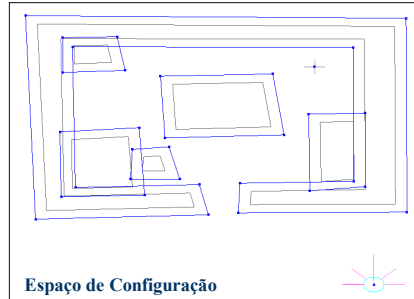
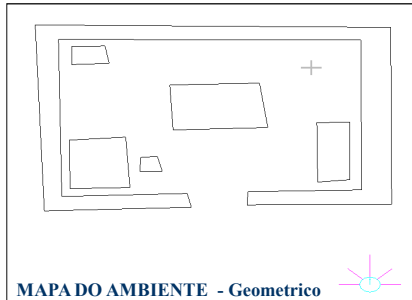
1. Determinar a área onde o robô pode navegar: Espaço de Configuração
2. Gerar o conjunto de rotas possíveis: Grafo de Visibilidade
3. Obter a melhor trajetória dentro das rotas possíveis: Algoritmo de Dijkstra

24

Março 2010

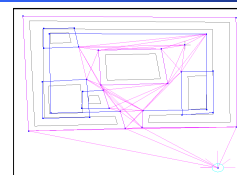
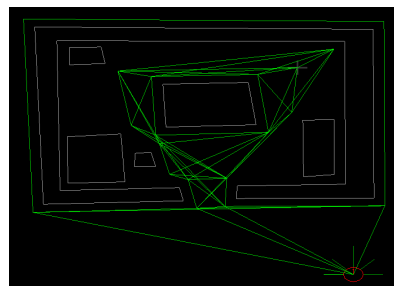
## Arquitetura de Controle: **Deliberativo**

SIMROB2D

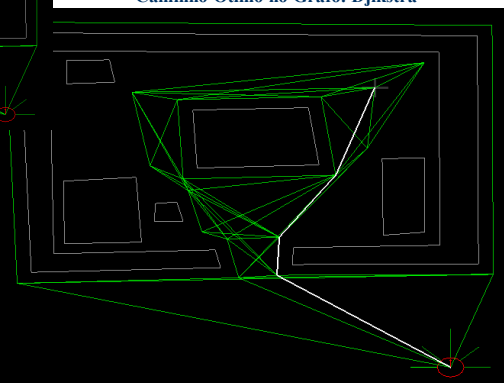
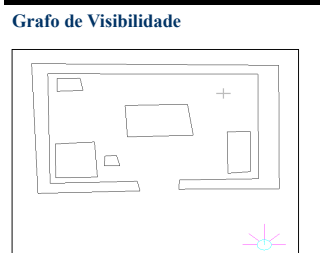


## Arquitetura de Controle: **Deliberativo**

SIMROB2D

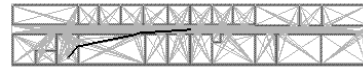
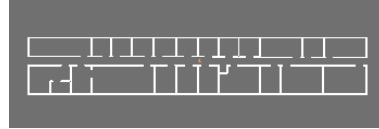
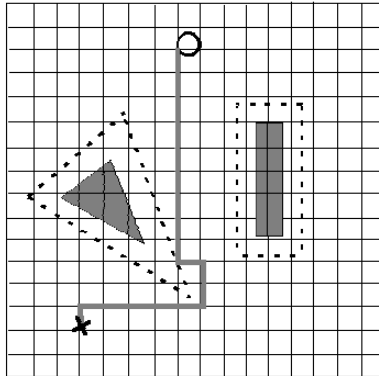


Caminho Ótimo no Grafo: Dijkstra

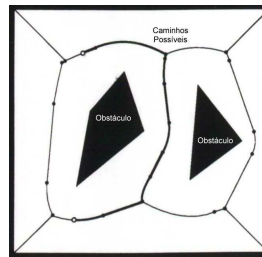
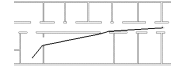


### PLANEJAMENTO DE TRAJETÓRIAS Baseado em Mapas – Múltiplos Algoritmos

Navegação baseada em  
Mapa de Ocupação:  
Planejamento A\*



Navegação baseada em  
Mapa Geométrico:  
Grafo+Dijkstra



Navegação baseada em  
Mapa Geométrico:  
Diagramas de Voronoi

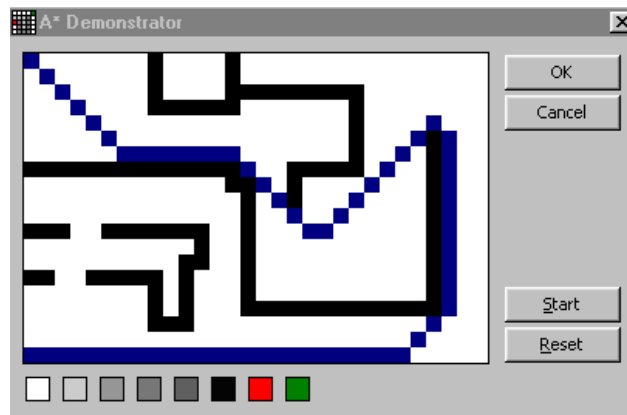
27

Março 2010

Figura 4.3 Navegação baseada em *Grid*

Farlei Heinen, 2002

### A\* Search: [Tutorial *SBGames07*] Busca Heurística de Caminhos

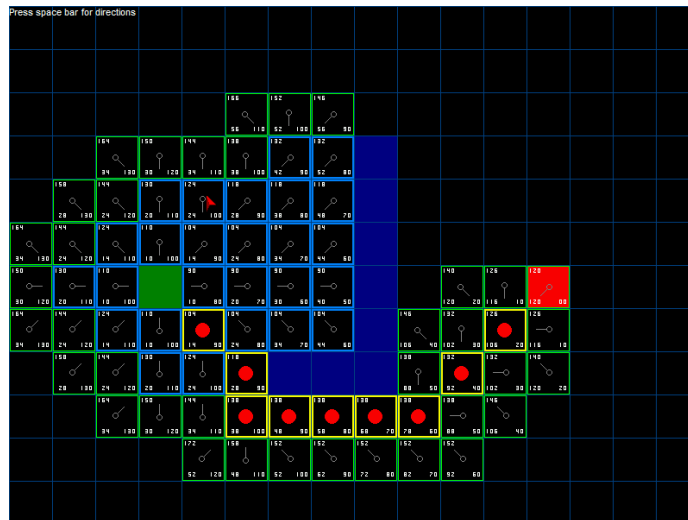


28

Março 2010

[Matthews 2003]

## A\* – Planejamento de Trajetórias [Tutorial SBGames 2007]

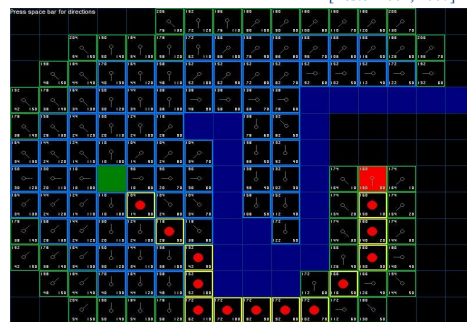
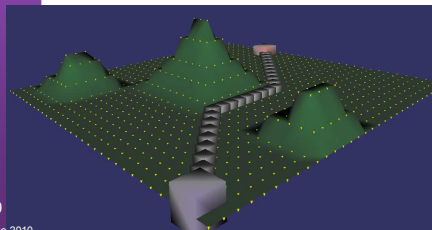
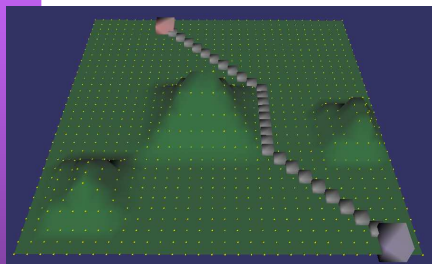


29

Março 2010

## A\* Search: [Tutorial SBGames07] Busca Heurística de Caminhos

[Lester 2004, 2007]



30

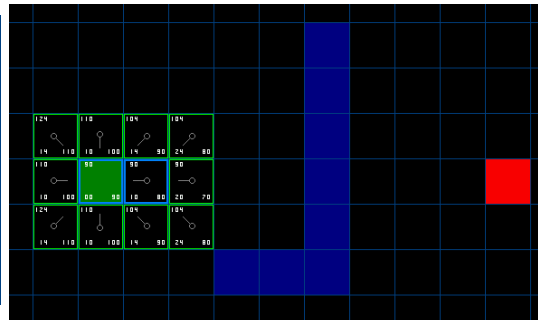
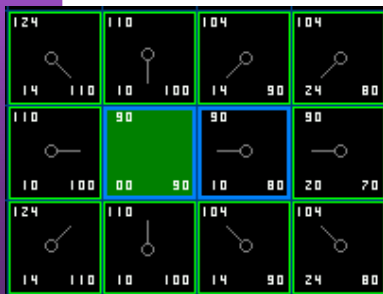
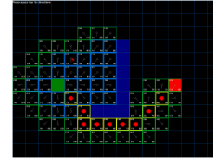
Março 2010

**A\* Search: [Tutorial SBGames07]**

**Busca Heurística de Caminhos**

Custo de um Caminho

$$F(x) = G(x) + H(x) \quad \therefore \quad G = \text{Percorrido}, H = \text{Estimado}$$



Março 2010

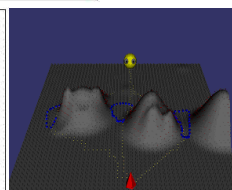
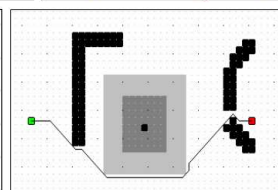
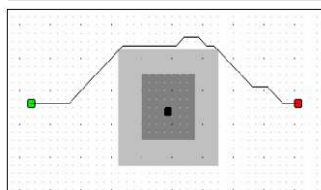
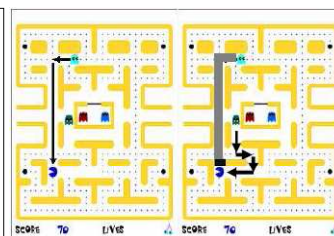
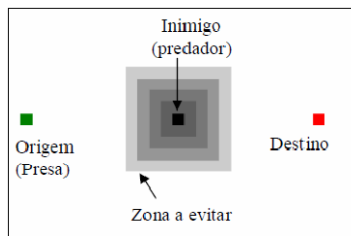
**A\* Search: [Tutorial SBGames07]**

**Busca Heurística de Caminhos**

MAPA: Custo do Terreno a ser Atravessado

Custo de um Caminho

$$F(x) = G(x) + H(x) \quad \therefore \quad G = \text{Percorrido}, H = \text{Estimado}$$



32

Março 2010



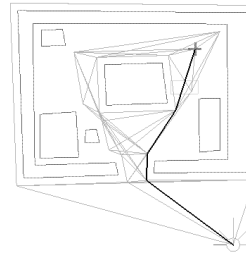
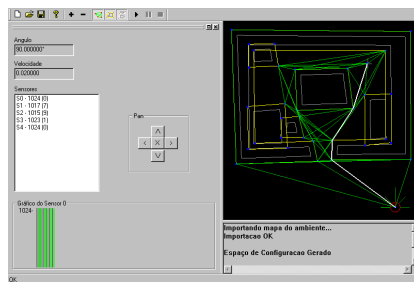
## Arquitetura de Controle: **Deliberativo**

### CONTROLE: Arquiteturas **Deliberativo**

- **Deliberativo: Planejamento**
- **Técnicas:**
  - Grafo de Visibilidade
  - Voronoi
  - A\* ...
- **Problemas...**

Seguir em direção a um alvo  
Planejamento de Trajetória com Mapa  
\* **Deliberativo “puro” baseado em informações externas GLOBAIS**

- Houston, we have a problem!
- >> Mapas com informações imprecisas
  - >> Obstáculos Dinâmicos (não mapeados)
  - >> Erros de localização do robô



*Localização => Tem que ser precisa!  
Mapa do Ambiente => Tem que ser exato!*

## Arquitetura de Controle

CONTROLE: Arquiteturas **Reativas**

CONTROLE: Arquiteturas **Deliberativas**

**Ambas possuem problemas e limitações!**

**Solução?**

**Buscar aproveitar o que de melhor tem cada uma das duas abordagens...**

**SISTEMAS HÍBRIDOS!**

### Referências citadas nesta aula...

**Craig Reynolds – Boids, OpenSteer, Flocks (múltiplos agentes autônomos)**  
<http://www.red3d.com/cwr/boids/>

**Robocode – Batalha de Robôs Simulados**  
<http://robocode.sourceforge.net/>

**Simulador do Khepera**  
<http://diwww.epfl.ch/lami/team/michel/khep-sim/>

**Simulador com Planejamento de Trajetória (SimRob 2D)**  
<http://nccg.unisinos.br/robotica/robotica.html>

### Referências do Algoritmo A\* (A Star)

**A\* Demo1:** [http://www.policvalmanac.org/games/aStarTutorial\\_port.htm](http://www.policvalmanac.org/games/aStarTutorial_port.htm)

**A\* Demo2:** <http://www.gamedev.net/REFERENCE/ARTICLES/ARTICLE2003.ASP>

**A\* Multiplo:** <http://www.inf.unisinos.br/~sbgames/anais/tutorials/Tutorial3.pdf>



### INFORMAÇÕES SOBRE A DISCIPLINA

**USP - Universidade de São Paulo - São Carlos, SP**  
**ICMC - Instituto de Ciências Matemáticas e de Computação**  
**SSC - Departamento de Sistemas de Computação**

**Prof. Fernando Santos OSÓRIO**

**PAE Maurício Acconcia Dias**

**Web institucional:** <Http://www.icmc.usp.br/ssc/>

**Página pessoal:** <Http://www.icmc.usp.br/~fosorio/>

**E-mail:** [fosorio\[at\]icmc.usp.br](mailto:fosorio[at]icmc.usp.br), [gmail.com](mailto:gmail.com) } # [maccddias\[at\]gmail.com](mailto:maccddias[at]gmail.com)

**Disciplina de Robôs Móveis Autônomos**

**Web Disciplinas:** <Http://www.icmc.usp.br/~fosorio/>

**Wiki ICMC:** <http://wiki.icmc.usp.br/index.php/SSC-714>

**> Programa, Material de Aulas, Critérios de Avaliação,**

**> Material de Apoio, Trabalhos Práticos**