

**USP - ICMC - SSC – Pós-Grad. CCMC
SSC 5888 (RMA) - 1o. Semestre 2009**

**Disciplina de
Robôs Móveis Autônomos
SSC-5888**

Prof. Fernando Santos Osório
Email: fosorio [at] { icmc. usp. br , gmail. com }
Web: <http://www.icmc.usp.br/~fosorio/>

Aula 05: Aprendizado e Robótica

Agenda:

Aprendizado de Máquina em Robótica Móvel: *Machine Learning*

1. *Aprendizado de Comportamentos Simples: ANN, RL*

Desvio de obstáculos, Buscar Alvo, Seguir Paredes

2. *Aprendizado de Seqüências*

Autômatos Finitos (FSM – *Finite State Machine*)

Aprendizado de um Autômato

Controle de Estacionamento de um Veículo Autônomo: SEVA

3. *Robôs com Pernas: Aprendizado do Caminhar*

Controle de um robô com Patas: LEGGEN

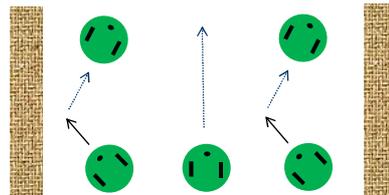
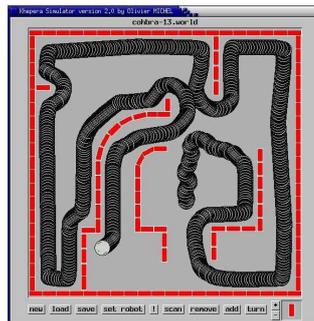
4. *Coordenação em Sistemas Multi-Robóticos*

Estratégias de Ação: Robombeiros

5. *Competições: Agentes Autônomos Inteligentes*

Aprendizado de Comportamentos Simples:

- Desvio de obstáculos
- Buscar Alvo
- Seguir Paredes

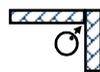
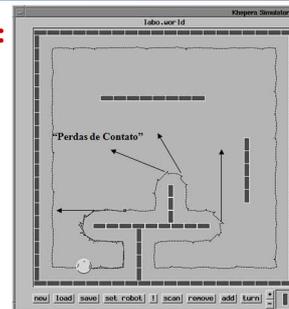


Técnicas de Aprendizado: RL, ANN, GA

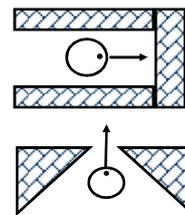
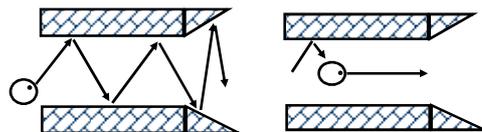
Aprendizado de Comportamentos Simples:

- Desvio de obstáculos
- Buscar Alvo
- Seguir Paredes

Problemas: Mínimos Locais, Falta de Contexto,
 Planejamento Global de Rotas



Alternância: Oscilação
 Esq, Dir, Esq, Dir, Esq, Dir, ...



Técnicas de Aprendizado: RL, ANN, GA

Aprendizado de Seqüências:

➤ **Autômato Finito**

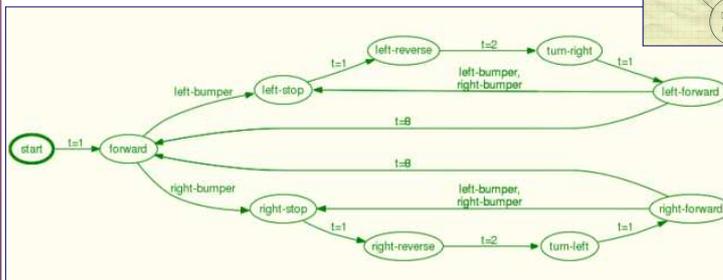
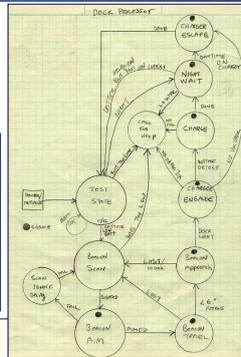
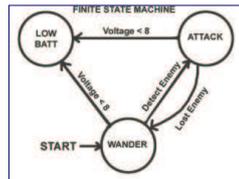
FSM - Finite State Machine

Estado

FSM:
Exemplos da Web...

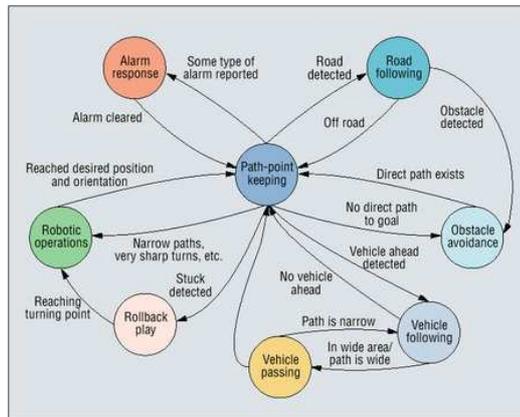
Contexto

Seqüência



Aprendizado de Seqüências:

➤ **Autômato Finito / FSM - Finite State Machine**



DARPA Challenge

Ohio State University at the 2004 DARPA Grand Challenge: Developing a Completely Autonomous Vehicle
 IEEE INTELLIGENT SYSTEMS - Sept./Oct. 2004

Aprendizado de Seqüências:

➤ **Autômato Finito / FSM – *Finite State Machine*: SEVA**

**IEEE IJCNN - International Joint
Conference on Neural Networks 2006
Vancouver - Canadá - July 2006**

Paper - Best Session Presentation Award

SEVA3D:

**Using Artificial Neural Networks to
Autonomous Vehicle Parking Control**



**SEVA3D: Using Artificial Neural Networks
to Autonomous Vehicle Parking Control**

**Applied Computing Research Post-grad Program - PIPCA
Autonomous Vehicles Research Group
[Grupo de Pesquisas em Veículos Autônomos - GPVA]**

**Web: <http://inf.unisinos.br/~osorio/seva3d>
or Google: veiculos autonomos**

**IEEE WCCI - IJCNN 2006
Vancouver, July 2006**

**Milton Roberto Heinen
Prof. Dr. Fernando S. Osório
Prof. M.Sc. Farlei José Heinen
Prof. Dr. Christian Kelber**



Controle da Tarefa de Estacionamento de um Veículo Autônomo através do Aprendizado de um Autômato Finito usando uma Rede Neural J-CC

SBRN'02 - Simpósio Brasileiro de Redes Neurais

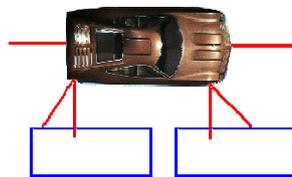


**Fernando Osório
Farlei José Heinen
Luciane Fortes**

Autômato SEVA: Estacionamento Autônomo

Componentes principais:

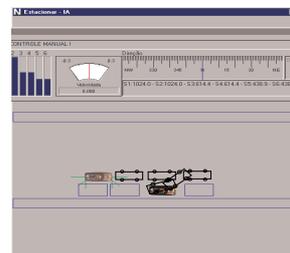
- Modelo de simulação dos *sensores*;
- Modelo de simulação da *cinemática* do veículo (modelo do deslocamento de um carro - Ackerman)
- Comandos do atuador relacionado ao *deslocamento* (avançar / recuar e velocidade);
- Comandos do atuador relacionado ao *giro* do veículo (rotação da direção).



SEVA-A
 Seva Autômato

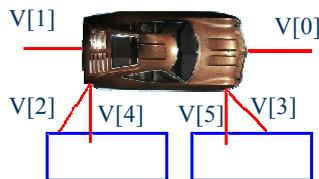
SEVA-H
 Seva Humano

SEVA-N
 Seva Neural



Autômato SEVA: Estacionamento Autônomo

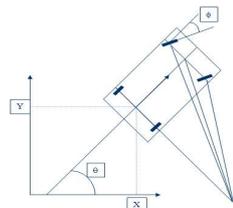
Modelo Sensorial



-Sensores de distância
 Infravermelhos (V[0] a V[5]).

-Sensores posicionados de forma estratégica, especificamente para estacionamento em vagas paralelas.

Modelo Cinemático



Ackerman Steering

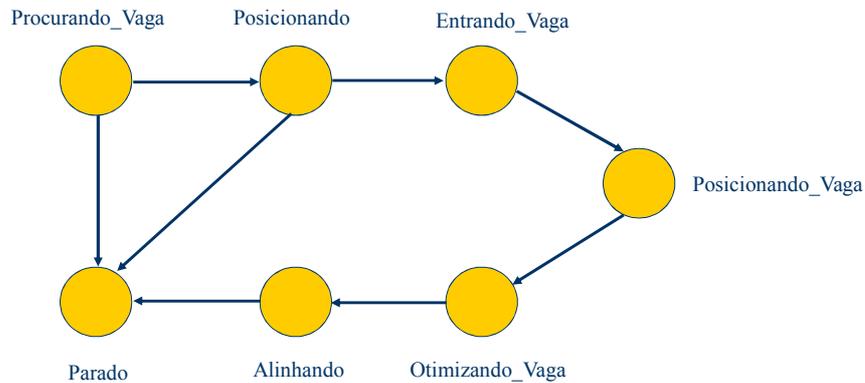
“Comportamento de um automóvel”

$$X = V * \cos(\Phi) * \cos(\theta)$$

$$Y = V * \sin(\Phi) * \cos(\theta)$$

$$\dot{\theta} = V / L * \sin(\Phi)$$

SEVA-A: Autômato de Controle



Autômato de estados finitos (FSA):
Estado E, Sensores S => Estado E', Comando C

11

Maior 2009

```
Se Estado_Atual(Procurando_Vaga) e Próximo_ao_Obstáculo(V[4]) e
Próximo_ao_Obstáculo(V[5])
Então Speed = Avanço_Rápido e RotVel = Direção_Reta;
```

```
Se Estado_Atual(Procurando_Vaga) e Longe_do_Obstáculo(V[2]) e
Longe_do_Obstáculo(V[3]) e Longe_do_Obstáculo(V[4]) e
Longe_do_Obstáculo(V[5])
Então Troca_Estado(Posicionando) e Inicializa(Odômetro);
```

```
Se Estado_Atual(Posicionando)
Então Speed = Avanço_Rápido e RotVel = Direção_Reta;
```

```
Se Estado_Atual(Posicionando) e Longe_do_Obstáculo(V[4]) e
Deslocamento_Suficiente(Odômetro)
Então Estado_atual(Entrando_Vaga) e Inicializa(Odômetro);
```

```
Se Estado_Atual(Entrando_Vaga)
Então Speed = Ré_Rápida e RotVel = Giro_Esquerda_Max;
```

Problemas:

- Usuário deve explicitar as regras, estas regras são pouco robustas, estão sujeitas a erros, são pouco flexíveis e de difícil adaptação a novas situações;

12

Maior 2009

SEVA-N: Rede Neural Recorrente (J-CC)

- * Optamos por um modelo de Rede Neural Artificial baseado na arquitetura MLP (*Multi-Layer Perceptron*), com aprendizado supervisionado do tipo Incremental (*Cascade-Correlation*).
- * Devido a necessidade de nesta implementação se informar a rede sobre o contexto (memória), adotamos uma arquitetura de rede recorrente baseada nas redes de Jordan (*Jordan Nets*) permitindo assim tratar este problema.
- * A rede adotada foi assim denominada de J-CC
>> Jordan-Cascade-Correlation Network <<

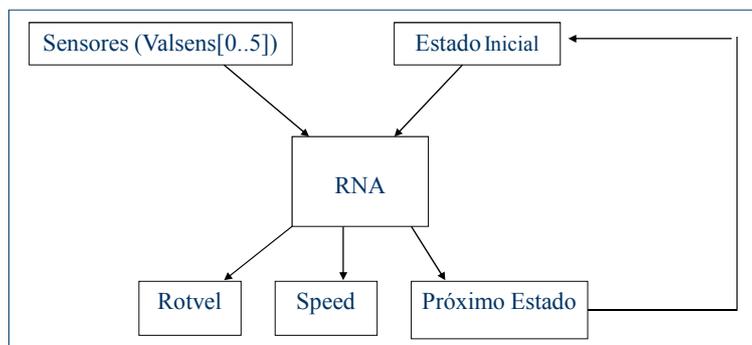
Rede Neural Artificial (RNA J-CC):
Estado E, Sensores S => Estado E', Comando C

13

Mais 2009

As variáveis (atributos) de entrada da rede que empregamos foram:
O estado dos seis sensores e uma indicação do estado atual do processo de estacionamento.

Na saída da rede iremos obter:
O estado dos atuadores (velocidade e rotação), assim como uma indicação do próximo estado do processo de estacionamento



14

Mais 2009

SEVA-N: Resultados da Simulações

Conjunto de 10 Experimentos - Resultados são valores médios

Base de Aprendizado: 392 exemplos

Base de Validação: 392 exemplos

Médias:

- Nro. de Épocas de aprendizado: 572 (Best epoch)
- N° de Neurônios Adicionados: 2
- Taxa de Erro do Aprendizado (% Acertos): 98,469
- N° de Exemplos Classificados Incorretamente: 6

Considerações sobre os experimentos

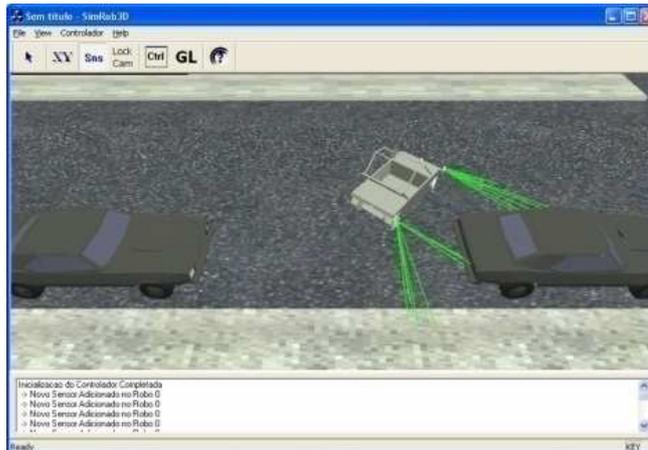
Os resultados obtidos até o presente demonstram que

- Ambos os controladores (SEVA-A e SEVA-N) são capazes de estacionar corretamente um carro, baseados apenas nas informações provenientes de seus sensores externos;
- O automato é bem mais sensível ao “ruído externo”;
- A rede neural J-CC foi capaz da “aprender” o autômato;
- A rede neural realiza perfeitamente a identificação e a troca de estados do autômato.

Autômato SEVA: Estacionamento Autônomo

Aprendizado de Seqüências:

➤ Autômato Finito / FSM - *Finite State Machine*: SEVA 3D



SEVA3D:

Simulação
Virtual em 3D

Sensores :
Realismo 3D
com erro/ruído

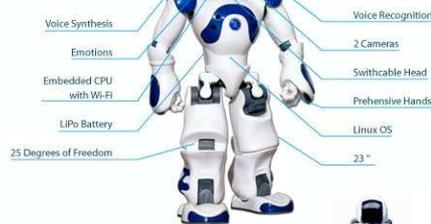
Aprendizado do
FSM-Neural:
A Rede Neural
funciona 100%
com dados
ruidosos e com
sensores em um
Ambiente 3D

Aprendizado de Máquina em Robótica Móvel

Aprendizado de Seqüências: “Robôs com Pernas”

MOTIVAÇÃO - Mobile Articulated Robots:
Walking Machines... Insects, Animals, Humans

Aldebaran Robotics
Nao Robot



http://en.wikipedia.org/wiki/Nao_robot



(a) Honda Asimo



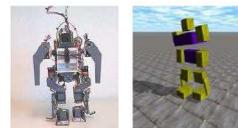
(b) Sony SDR-4X



(c) Kawada H6



(d) Fujitsu HOAP-2

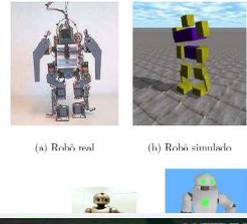
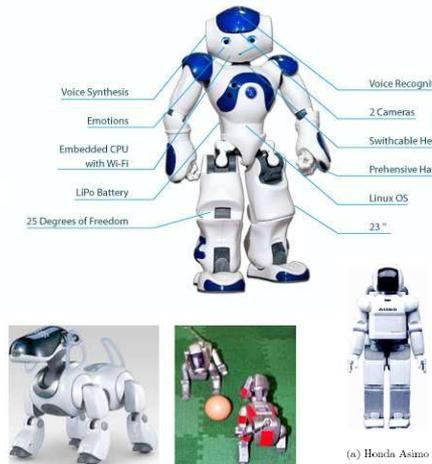


(a) Robô real (b) Robô simulado



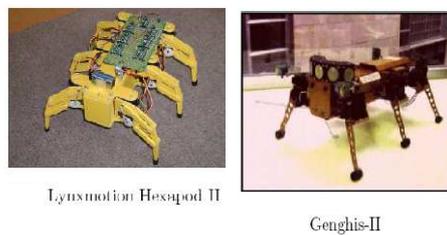
Aprendizado de Seqüências: “Robôs com Pernas”

MOTIVAÇÃO - Mobile Articulated Robots:
 Walking Machines... Insects, Animals, Humans



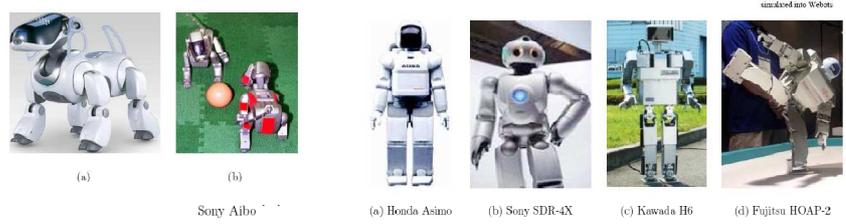
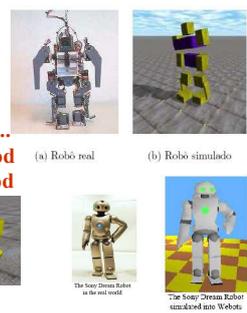
Aprendizado de Seqüências: “Robôs com Pernas”

MOTIVAÇÃO - Mobile Articulated Robots:
 Walking Machines... Insects, Animals, Humans



LEGGEN

Dog
 Horse ...
 Tetrapod
 Hexapod



Aprendizado de Sequências: “Robôs com Pernas”

➤ Autômato Finito / FSM – *Finite State Machine*: LEGGEN



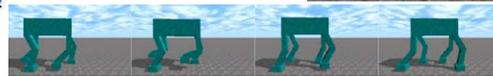
Pós-Graduação em Computação Aplicada - PIPCA
Grupo de Pesquisas em Veículos Autônomos - GPVA
>> Autonomous Vehicles Research Group <<

Web: <http://inf.unisinos.br/~osorio/leggen>
or Google: *veiculos autonomos*

Applying Genetic Algorithms to Control
Gait of Physically based Simulated Robots

IEEE WCCI - CEC 2006
Vancouver, July 2006

Prof. Dr. Fernando S. Osório
Milton Roberto Heinen



Aprendizado de Sequências: “Robôs com Pernas”

➤ Autômato Finito / FSM – *Finite State Machine*: LEGGEN

- * **Virtual Reality Environment and Simulation of Legged Robots**
"Uso de Realidade Virtual para a Simulação do Caminhar em Robôs Móveis"
SVR 2006 - Belém, Brazil
- * **Evaluate different *Robot Models* (hardware configurations)**
"Applying Genetic Algorithms to Control Gait of Physically Based Simulated Robots" - IEEE WCCI / CEC 2006 - Vancouver, Canadá
- * **Evaluate different *Fitness Functions***
"Gait Control Generation for Physically Based Simulated Robots using Genetic Algorithms" - IBERAMIA / SBIA 2006 - Ribeirão Preto, SP
- * **Robot, Vehicles and Human Simulation**
"Increasing Reality in VR Applications: Physical and Behavioral Simulation"
Virtual Concept Summer School - Nov. 2006 - Porto Alegre, RS / VC2006 Cancun
- * **Evaluate Neural Learning of cyclic functions** => SBRN 2006
- * **Evaluate Robot Morphology Evolution** => CLEI 2007

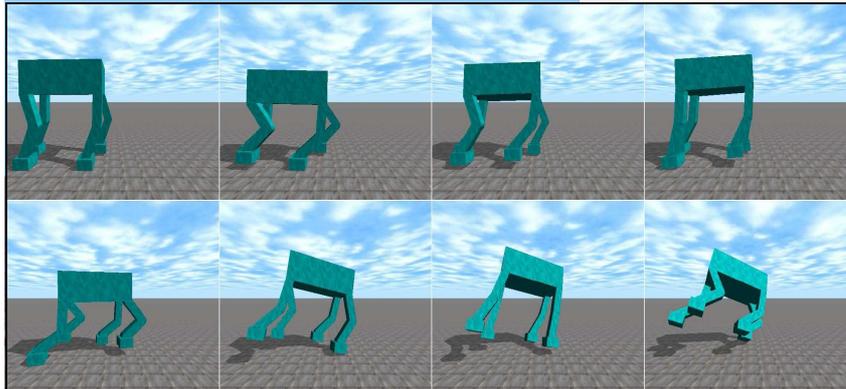
LEGGEN - ROBOT SIMULATION

Simulation of 3D Realistic
 Virtual Legged Robots
LEGGEN Simulator

DIMENSIONS OF THE SIMULATED ROBOTS

tetraL3J			
Part	x	y	z
Body	45.0	15.0	25.0
Thigh	5.0	15.0	5.0
Shin	5.0	15.0	5.0
Paw	8.0	5.0	9.0

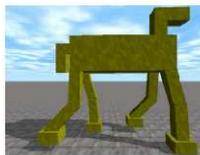
Genetic Evolved Control
 of Articulated Robots (w/legs)



LEGGEN - ROBOT SIMULATION

Simulation of 3D Realistic Virtual Legged Robots
LEGGEN Simulator - Tools:

- 1. OSG - Open Scene Graph (OpenGL + Extensions)**
 | <http://www.openscenegraph.org/> | **Graphics Engine**
- 2. ODE - Open Dynamics Engine**
Rigid Body Physics Simulation
 (gravity, inertia, friction, collision, joints, etc)
 | <http://www.ode.org/> | **Physics Engine**



Dimensões			
Parte	x	y	z
Corpo	45,0cm	15,0cm	25,0cm
Coxa	5,0cm	15,0cm	5,0cm
Canela	5,0cm	15,0cm	5,0cm
Pata	8,0cm	5,0cm	9,0cm

LEGGEN - ROBOT SIMULATION

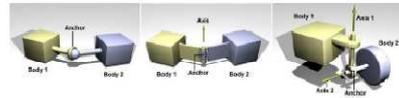
Simulation of 3D Realistic Virtual Legged Robots

LEGGEN Simulator - Tools:

1. OSG - Open Scene Graph (OpenGL + Extensions)

[<http://www.openscenegraph.org/>]

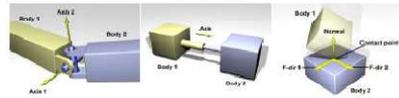
2. ODE - Open Dynamics Engine



(a) Ball Socket

(b) Hinge

(c) Hinge-2



(d) Universal

(e) Slider

(f) Contact

3. GALib

Genetic Algorithms Simulation
<http://www.lancet.mit.edu/ga/>

4. Robot Control: Joint Angles

Finite State Machine
Artificial Neural Nets

27

05 Jun 2007

LEGGEN - ROBOT SIMULATION

Simulation of 3D Realistic Virtual Legged Robots

LEGGEN Simulator - Tools:

1. OSG - Open Scene Graph (OpenGL + Extensions)

[<http://www.openscenegraph.org/>]

2. ODE - Open Dynamics Engine

Control:

- Joints with motors
- Activate motors to set a specific angle for each joint
- Sequence of angles = Walk!

3. GALib

Genetic Algorithms Simulation
<http://www.lancet.mit.edu/ga/>

4. Robot Control: Joint Angles

Finite State Machine
Artificial Neural Nets

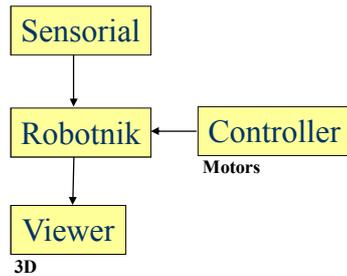
28

05 Jun 2007

LEGGEN - ROBOT SIMULATION

Simulation of 3D Realistic Virtual Legged Robots

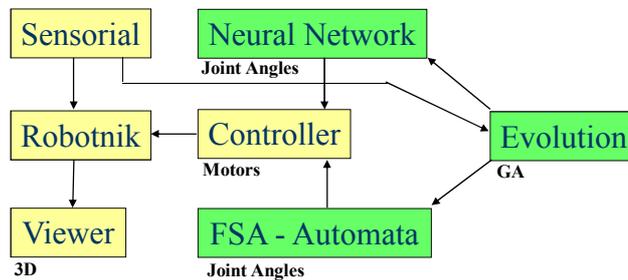
LEGGEN Simulator - Tools:



LEGGEN - ROBOT SIMULATION

Simulation of 3D Realistic Virtual Legged Robots

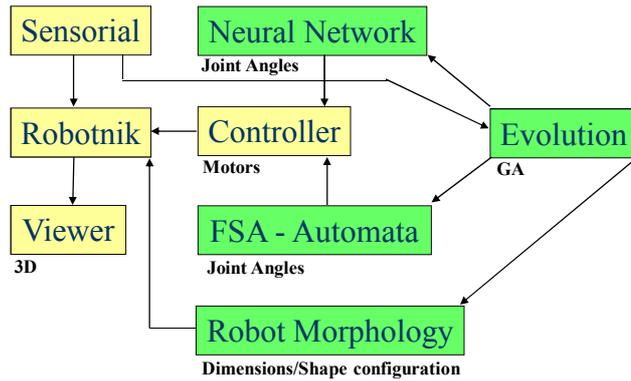
LEGGEN Simulator - Tools:



LEGGEN - ROBOT SIMULATION

Simulation of 3D Realistic Virtual Legged Robots

LEGGEN Simulator - Tools:



Evolve Robot +
 Evolve Control +
 Visualization

LEGGEN SIMULATOR - GA

Genetic Algorithms Simulation

Algorithm 1 Pseudocode of the GA

```

F ← FitnessFunction
G ← NumberOfGenerations
M ← SizeOfPopulation
pc ← FrequencyOfCrossover
pm ← FrequencyOfMutation
Population ← RandomPopulation (M)
CalculateFitness (Population)
while GenerationNum < G do
  Parents ← Population
  Children ← Crossover (Parents, pc)
  NewPopulation ← Mutation (Children, pm)
  Population ← NewPopulation
  CalculateFitness (Population)
end while
return (BestIndividual)
  
```

GA Parameters:

Cross-Over: 0.80
 Mutation: 0.08
 Population: 350
 Generations: 700

Genome:

Evolve FSA
 Evolve ANN

Genetic Algorithms Simulation

Fitness Function:

$$F = \frac{D}{1 + B + a \times G}$$

F = Fitness
 D = Walked Distance
 B = Bumpers (paws touching ground ~ stability)
 G = Gyro (gyroscope/accelerometers ~ stability)

$$B = \sum_{i=1}^P \left(\frac{n_i}{N} - \frac{1}{2} \right)^2$$

where
 P = paws
 ni = bumper contacts of paw i
 N = total bumper sensor readings

$$G = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2 + \sum_{i=1}^N (y_i - \bar{y})^2 + \sum_{i=1}^N (z_i - \bar{z})^2}{N}}$$

G = Sum of x, y, z instability
 where

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N}, \quad \bar{y} = \frac{\sum_{i=1}^N y_i}{N}, \quad \bar{z} = \frac{\sum_{i=1}^N z_i}{N}$$

GA Parameters:

Cross-Over: 0.80
 Mutation: 0.08
 Population: 350
 Generations: 700
 Walking time: 30

GA Finite State Machine

- The robot is placed in the starting position and orientation in the simulation environment;
- The genome is read and the robot control FSM table values are set;
- The physical simulation is executed during a predefined amount of time (sixty seconds in our experiments);
- Gait information and sensor data are captured during each individual physical simulation;
- Fitness is calculated and returned to the GAlib;

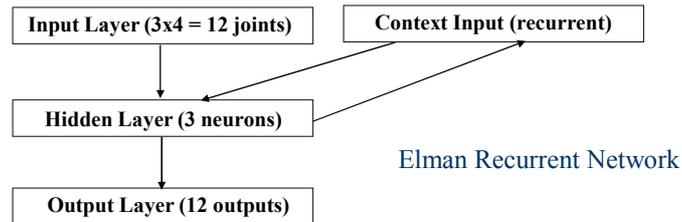
EXAMPLE OF A GENOTYPE

	State 1	State 2	State 3	State 4
Reference velocity	1.1703	2.0705	1.2499	1.5901
Front legs - hip	0.0698	-0.6283	-0.1570	-0.0698
Front legs - knee	1.0473	0.7504	0.2268	0.4363
Front legs - ankle	-1.1172	-0.1221	-0.0698	-0.3665
Rear legs - hip	-0.1570	-0.0698	0.0698	-0.6283
Rear legs - knee	0.2268	0.4363	1.0473	0.7504
Rear legs - ankle	-0.0698	-0.3665	-1.1172	-0.1221

Genome:

- * FSM: Table of Target States (Sequence of Actions)
- * Target angles + Reference Angular velocity
- * Adjust velocity
- * Fitness Function: Sensor based

GA Artificial Neural Network



Input: joint angles at time t $[-1:+1]$
Output: joint angles at time $t+1$ $[-1:+1]$

Temporal sequence of joint angles.

Learning: adjust ANN weights - **Not supervised**
=> Genetic Algorithm weight adjust

Simulation main goals:

Evaluate robot control...

1. Using evolved FSA (automata - sequence of target angles)
2. Using evolved ANN (sequence of joint angles)

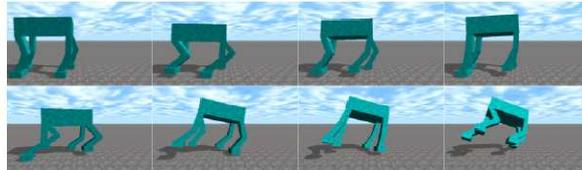
Evolution:



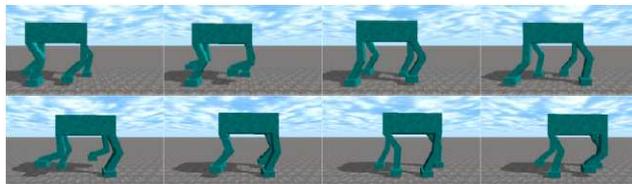
FSA - Target angles evolved using a Genetic Algorithm
ANN - Weights evolved using a Genetic Algorithm
Robot Morphology - Dimension/Shape evolved by a GA
Fitness - Sensors (physical simulation)

EVOLVING ANN AND ROBOTS

Results: Evolving only ANN based robot control...



ANN => "Learning to Walk"



EVOLVING ANN AND ROBOTS

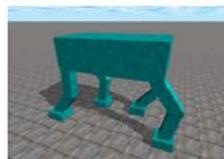
Results:

Evolving morphology...

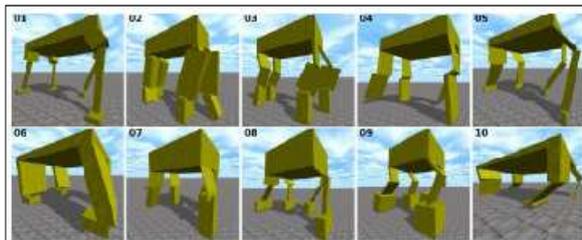
The chromosome includes x, y, z size for each segment

DIMENSIONS OF THE SIMULATED ROBOTS

tetraL3J			
Part	x	y	z
Body	45.0	15.0	25.0
Thigh	5.0	15.0	5.0
Shin	5.0	15.0	5.0
Paw	8.0	5.0	9.0



Typical Robot



Robots with Evolved Morphology

FUTURE WORK ?

Next steps... Autonomous parking control model (SEVA) integrated with LEGGEN control model [IEEE WCCI / IJCNN 2006]

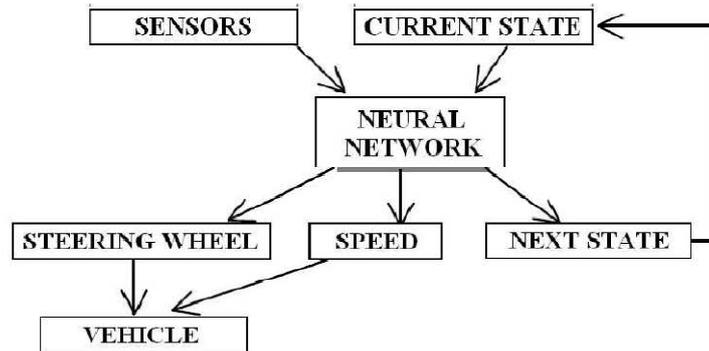


Fig. 7. Artificial neural network model scheme

Aprendizado de Estratégias:

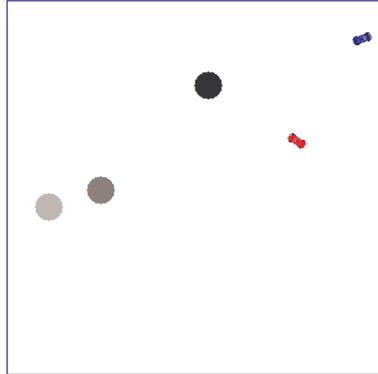
- **Sistemas Multi-Robóticos: Robombeiros**



Comportamentos em Sistemas Multi-Robóticos: Robombeiros
Planejamento de Trajetórias - Otimização usando G.A.

Competições Robóticas:

- Simulação: Robôs, Veículos e Agentes Autônomos
- Aprendizado de Robôs e Agentes Autônomos



IEEE CEC 2007 Car Racing Competition



CIG2009
IEEE Symposium on Computational Intelligence and Games
September 7-10, 2009 - Politecnico di Milano, Milano, Italy

41

Mai 2009

Competições Robóticas:

- Simulação: Robôs, Veículos e Agentes Autônomos
- Aprendizado de Robôs e Agentes Autônomos



Simulated Car Racing

The simulated car racing competition of CIG-2009 is the final event of the 2009 Simulated Car Racing Championship, an event joining the three competitions held at CEC-2009, GECCO-2009, and CIG-2009.

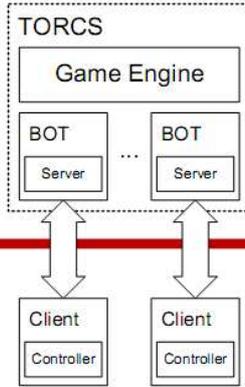
42

Mai 2009

<http://www.ieee-cig.org/> Competitions

Competições Robóticas:

TORCS:
 The Open Racing Car Simulator
<http://torcs.sourceforge.net/>



Simulated Car Racing

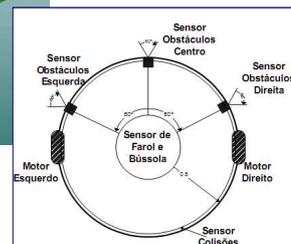
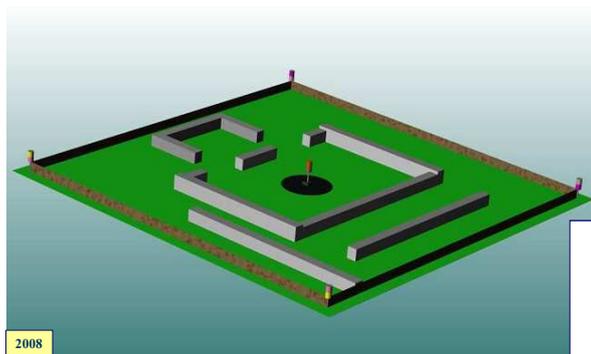
The simulated car racing competition of CIG-2009 is the final event of the 2009 Simulated Car Racing Championship, an event joining the three competitions held at CEC-2009, GECCO-2009, and CIG-2009.

Sensors & Actuators Model



Competições Robóticas:

Ciber-Rato / Micro-Rato:
 Competição Robótica organizada pela Universidade do Aveiro (Pt)



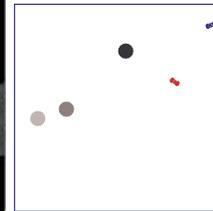
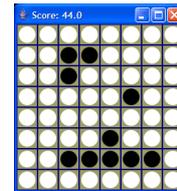
Competições Robóticas:

CIG Competitions 2007:

- * Ms. Pac-Man
- * X-Pilot AI

- * Simulated Car Racing
- * Othello

<http://cswww.essex.ac.uk/cig/2007/>
IEEE CIG 2007 Competitions



45

Maio 2009

Competições Robóticas:

CIG Competitions 2009:

- * The 2K BotPrize
- * The Defcon AI Competition
- * The Simulated Car Racing Competition

<http://www.ieee-cig.org/>
IEEE CIG 2009 Competitions



46

Maio 2009

Referências: Machine Learning

Referências – Aprendizado e Agentes Autônomos:

>> Ferramentas

SNNS / JavaNNS => <http://www.ra.cs.uni-tuebingen.de/SNNS/>
WEKA => <http://www.cs.waikato.ac.nz/ml/weka/>
GA-Lib => <http://lancet.mit.edu/ga/>

>> Referências – Competições Simuladas

Competições: CIG - http://www.ieee-cig.org/?page_id=119 (Games)
CEC - <http://www.cec-2009.org/competitions.shtml>
Car Racing => http://www.ieee-cig.org/?page_id=119 (Simulated)
http://cig.dei.polimi.it/?page_id=80
Pac-Man => <http://www.cec-2009.org/competitions.shtml#pacMan>
Bot Prize => <http://botprize.org/> (UnReal at CEC)
CiberMouse => http://www.ieeta.pt/~lau/web_CiberRTSS07/
Ciber-Rato => <http://microrato.ua.pt/>

Referências: Machine Learning

Referências – Aprendizado e Agentes Autônomos:

>> Referências – Competições com Robôs Reais

Competições:

DARPA Challenge => <http://www.darpa.mil/grandchallenge/>
LARC => <http://robotica.elo.utfsm.cl/competencia/>
RoboCup => <http://www.robocup.org/>
Fire-Fighting Contest => <http://www.trincoll.edu/events/robot/>
Ciber-Rato => <http://microrato.ua.pt/>
OBR => <http://obr.ic.unicamp.br/>
JRI / LARC => http://jri2008.dca.ufrn.br/LARC/index_en.php



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

Web institucional: [Http://www.icmc.usp.br/ssc/](http://www.icmc.usp.br/ssc/)

Página pessoal: [Http://www.icmc.usp.br/~fosorio/](http://www.icmc.usp.br/~fosorio/)

E-mail: [fosorio \[at\] icmc. usp. br](mailto:fosorio@icmc.usp.br) ou [fosorio \[at\] gmail. com](mailto:fosorio@gmail.com)

Disciplina de Robôs Móveis Autônomos

Web Disciplinas: [Http://www.icmc.usp.br/~fosorio/](http://www.icmc.usp.br/~fosorio/)

Web CoTeia: [Http://coteia.icmc.usp.br/mostra.php?ident=575](http://coteia.icmc.usp.br/mostra.php?ident=575)

> Programa, Material de Aulas, Critérios de Avaliação,

> Material de Apoio, Trabalhos Práticos