How Machines Learn: From Robot Soccer to Autonomous Traffic

Peter Stone

Department or Computer Sciences
The University of Texas at Austin

Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Question

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

- Autonomous agents
- Multiagent systems
- **Machine learning**
- **Robotics**

Autonomous Intelligent Agents

- They must **sense** their environment.
- They must **decide** what action to take ("think").
- They must **act** in their environment.

Autonomous Intelligent Agents

- They must **sense** their environment.
- They must **decide** what action to take ("think").
- They must **act** in their environment.

***Complete* Intelligent Agents**

Autonomous Intelligent Agents

- They must **sense** their environment.
- They must **decide** what action to take ("think").
- They must **act** in their environment.

***Complete* Intelligent Agents**

- Interact with other agents **(Multiagent systems)**

Autonomous Intelligent Agents

- They must **sense** their environment.
- They must **decide** what action to take ("think").
- They must **act** in their environment.

***Complete* Intelligent Agents**

- Interact with other agents **(Multiagent systems)**
- Improve performance from experience **(Learning agents)**

Autonomous Intelligent Agents

- They must **sense** their environment.
- They must **decide** what action to take ("think").
- They must **act** in their environment.

***Complete* Intelligent Agents**

- Interact with other agents **(Multiagent systems)**
- Improve performance from experience **(Learning agents)**

Autonomous Bidding, Cognitive Systems,
Robot Soccer, **Traffic management**

BE a learning agent

BE a learning agent

- You, as a group, act as a learning agent

BE a learning agent

- You, as a group, act as a learning agent
- **Actions**: Wave, Stand, Clap

BE a learning agent

- You, as a group, act as a learning agent
- **Actions**: Wave, Stand, Clap
- **Observations**: colors, reward

BE a learning agent

- You, as a group, act as a learning agent
- **Actions**: Wave, Stand, Clap
- **Observations**: colors, reward
- **Goal**: Find an optimal *policy*

BE a learning agent

- You, as a group, act as a learning agent
- **Actions**: Wave, Stand, Clap
- **Observations**: colors, reward
- **Goal**: Find an optimal *policy*
  - Way of selecting actions that gets you the most reward

How did you do it?

How did you do it?

- What is your policy?
- What does the world look like?

Formalizing What Just Happened

Knowns:

Formalizing What Just Happened

Knowns:

- $\mathcal{O} = \{$Blue, Red, Green, Black, $\ldots\}$
- Rewards in $\mathbb{R}$
- $\mathcal{A} = \{Wave, Clap, Stand\}$

$$\boxed{o_0, a_0, r_0, o_1, a_1, r_1, o_2, \ldots}$$

Formalizing What Just Happened

Knowns:

- $\mathcal{O} = \{$Blue, Red, Green, Black, $\ldots\}$
- Rewards in $\mathbb{R}$
- $\mathcal{A} = \{Wave, Clap, Stand\}$

$$\boxed{o_0, a_0, r_0, o_1, a_1, r_1, o_2, \ldots}$$

Unknowns:

Formalizing What Just Happened

Knowns:

- $\mathcal{O} = \{$Blue, Red, Green, Black, $\ldots\}$
- Rewards in $\mathbb{R}$
- $\mathcal{A} = \{Wave, Clap, Stand\}$

$$\boxed{o_0, a_0, r_0, o_1, a_1, r_1, o_2, \ldots}$$

Unknowns:

- $\mathcal{S} =$ 4x3 grid
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$
- $\mathcal{P} = \mathcal{S} \mapsto \mathcal{O}$
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$

Formalizing What Just Happened

Knowns:

- $\mathcal{O} = \{$Blue, Red, Green, Black, $\ldots\}$
- Rewards in $\mathbb{R}$
- $\mathcal{A} = \{Wave, Clap, Stand\}$

$$\boxed{o_0, a_0, r_0, o_1, a_1, r_1, o_2, \ldots}$$

Unknowns:

- $\mathcal{S} =$ 4x3 grid
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$
- $\mathcal{P} = \mathcal{S} \mapsto \mathcal{O}$
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$

$$o_i = \mathcal{P}(s_i)$$

Formalizing What Just Happened

Knowns:

- $\mathcal{O} = \{$Blue, Red, Green, Black, $\ldots\}$
- Rewards in $\mathbb{R}$
- $\mathcal{A} = \{Wave, Clap, Stand\}$

$$\boxed{o_0, a_0, r_0, o_1, a_1, r_1, o_2, \ldots}$$

Unknowns:

- $\mathcal{S} =$ 4x3 grid
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$
- $\mathcal{P} = \mathcal{S} \mapsto \mathcal{O}$
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$

$$o_i = \mathcal{P}(s_i) \qquad\qquad s_i = \mathcal{T}(s_{i-1}, a_{i-1})$$

Formalizing What Just Happened

Knowns:

- $\mathcal{O} = \{$Blue, Red, Green, Black, $\ldots\}$
- Rewards in $\mathbb{R}$
- $\mathcal{A} = \{Wave, Clap, Stand\}$

$$\boxed{o_0, a_0, r_0, o_1, a_1, r_1, o_2, \ldots}$$

Unknowns:

- $\mathcal{S} =$ 4x3 grid
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$
- $\mathcal{P} = \mathcal{S} \mapsto \mathcal{O}$
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \mapsto \mathcal{S}$

$$o_i = \mathcal{P}(s_i) \qquad s_i = \mathcal{T}(s_{i-1}, a_{i-1}) \qquad r_i = \mathcal{R}(s_i, a_i)$$

Reinforcement Learning

- **Algorithms to select actions** in such problems

Reinforcement Learning

- **Algorithms to select actions** in such problems
- **Q-learning**: provably converges to the **optimal policy**

Reinforcement Learning

- **Algorithms to select actions** in such problems
- **Q-learning**: provably converges to the **optimal policy**
  - Proof: **contraction mappings** and **fixed point theorem**

A harder problem

- You had 3 actions and saw one of 10 colors

A harder problem

- You had 3 actions and saw one of 10 colors
- What if you had to control 12 joints . . .

A harder problem

- You had 3 actions and saw one of 10 colors
- What if you had to control 12 joints . . .
- . . . and saw something like this 30 times per second?

RoboCup

RoboCup

Goal: By the year 2050, a team of humanoid robots that can beat the human World Cup champion team.

RoboCup

Goal: By the year 2050, a team of humanoid robots that can beat the human World Cup champion team.

- An international **research** initiative

RoboCup

Goal: By the year 2050, a team of humanoid robots that can beat the human World Cup champion team.

- An international **research** initiative
- Drives **research** in many areas:
  - Control algorithms; machine vision, sensing; localization;
  - Distributed computing; real-time systems;
  - Ad hoc networking; mechanical design;

RoboCup

Goal: By the year 2050, a team of humanoid robots that can beat the human World Cup champion team.

- An international **research** initiative
- Drives **research** in many areas:
  - Control algorithms; machine vision, sensing; localization;
  - Distributed computing; real-time systems;
  - Ad hoc networking; mechanical design;
  - **Multiagent systems**; **machine learning**; **robotics**

RoboCup

Goal: By the year 2050, a team of humanoid robots that can beat the human World Cup champion team.

- An international **research** initiative
- Drives **research** in many areas:
  - Control algorithms; machine vision, sensing; localization;
  - Distributed computing; real-time systems;
  - Ad hoc networking; mechanical design;
  - **Multiagent systems**; **machine learning**; **robotics**

Several Different Leagues

RoboCup Soccer

Small-sized League

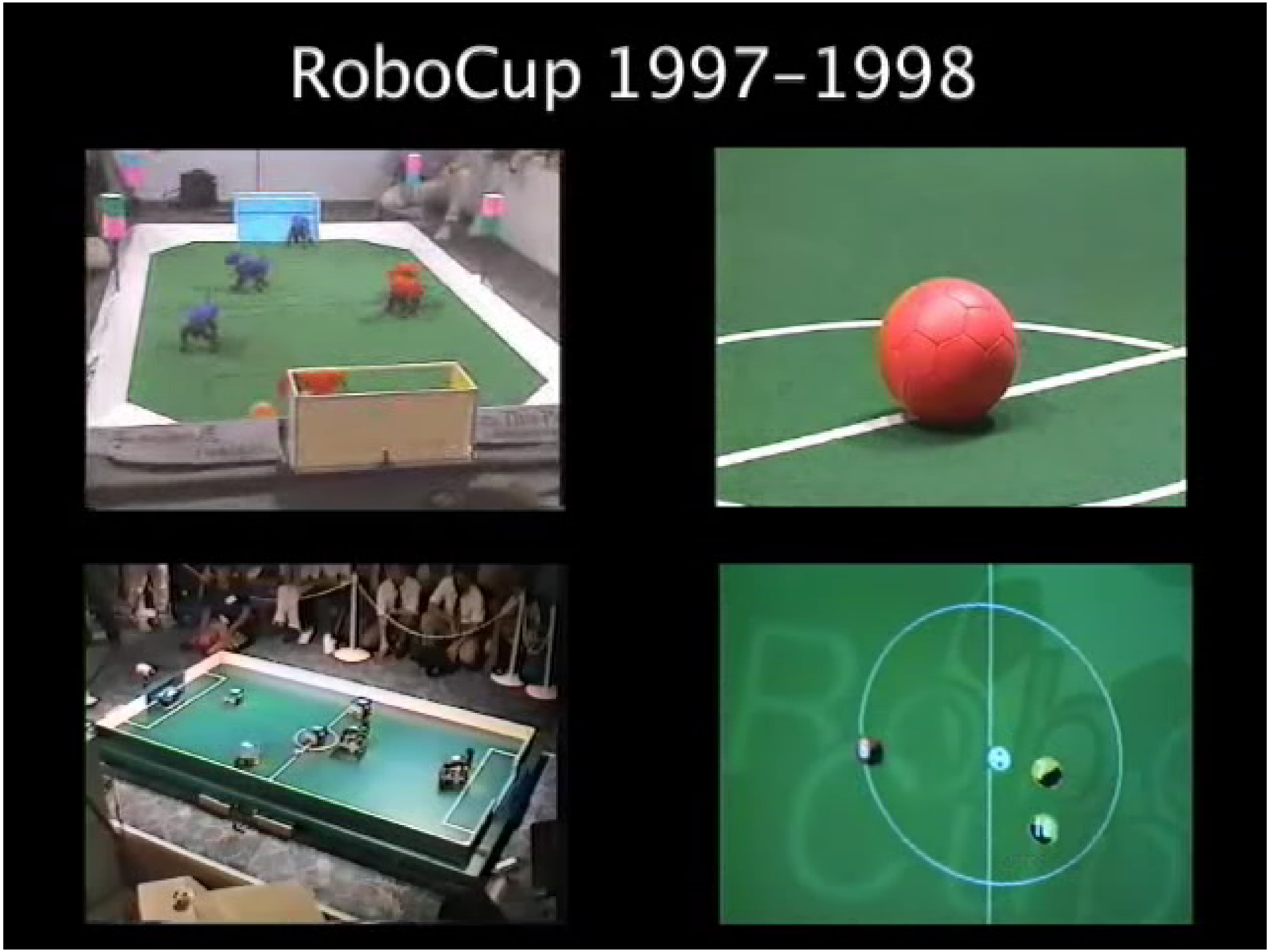
Middle-sized League

Legged Robot League

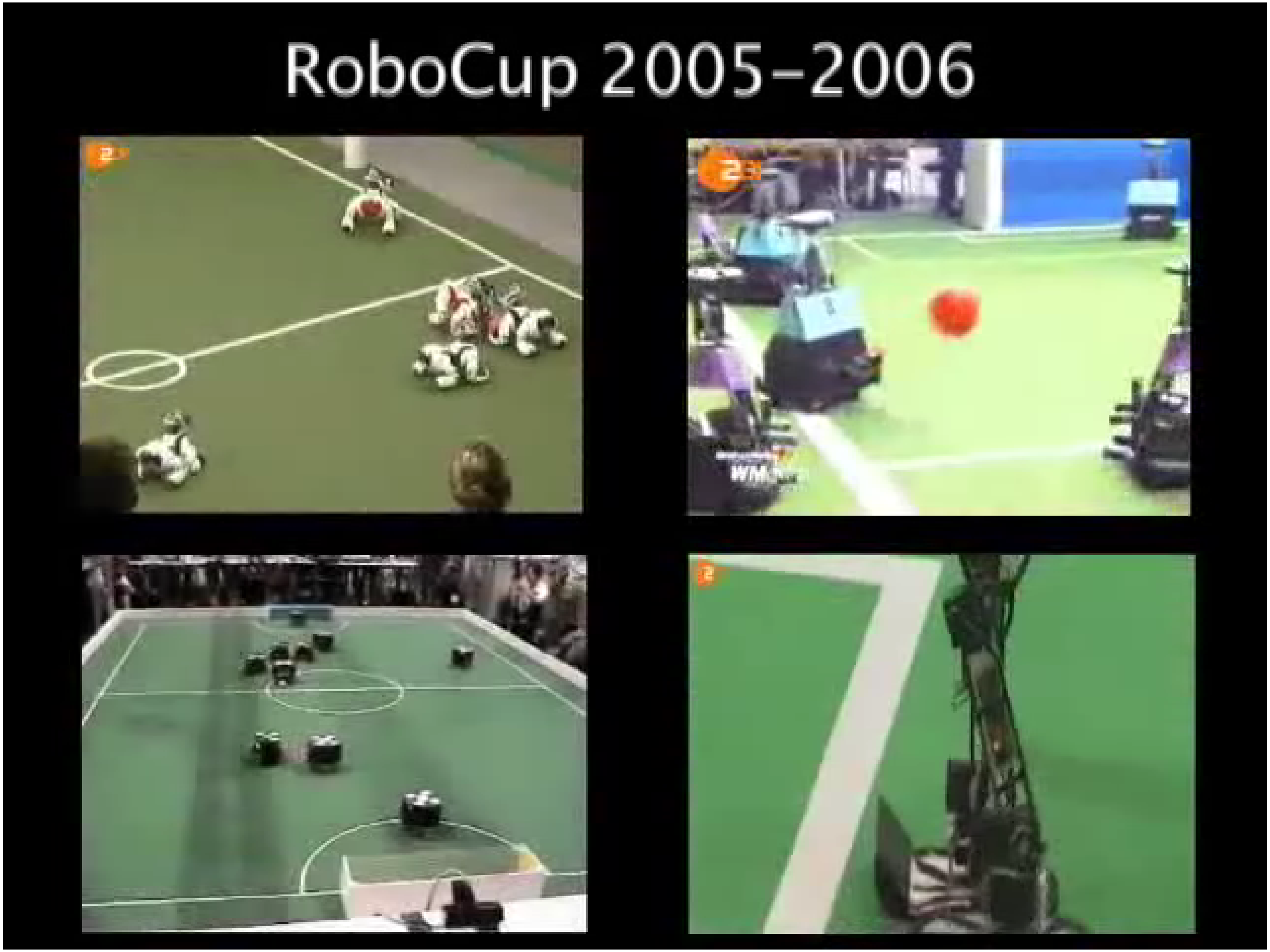
Simulation League

Humanoid League

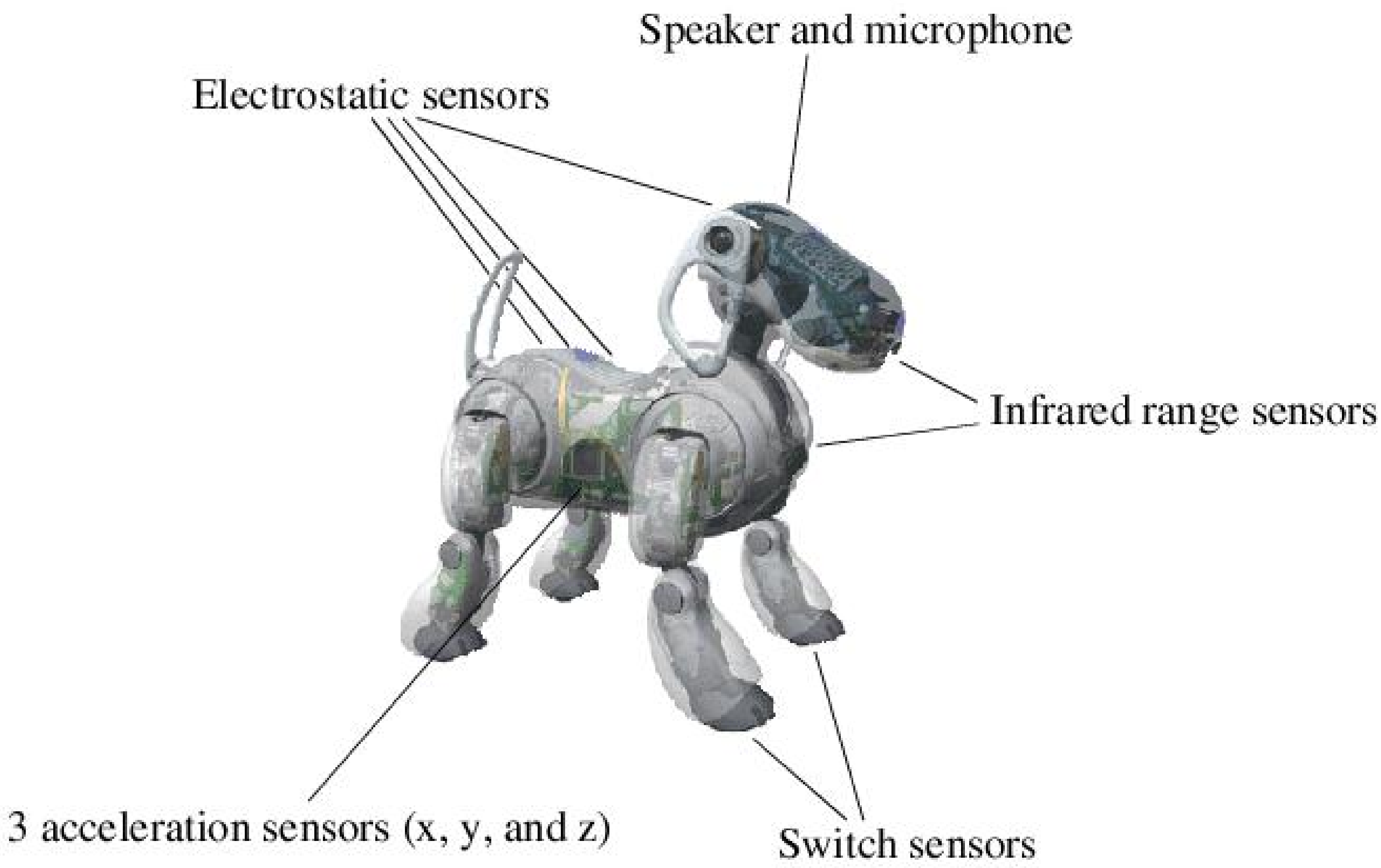
The Early Years



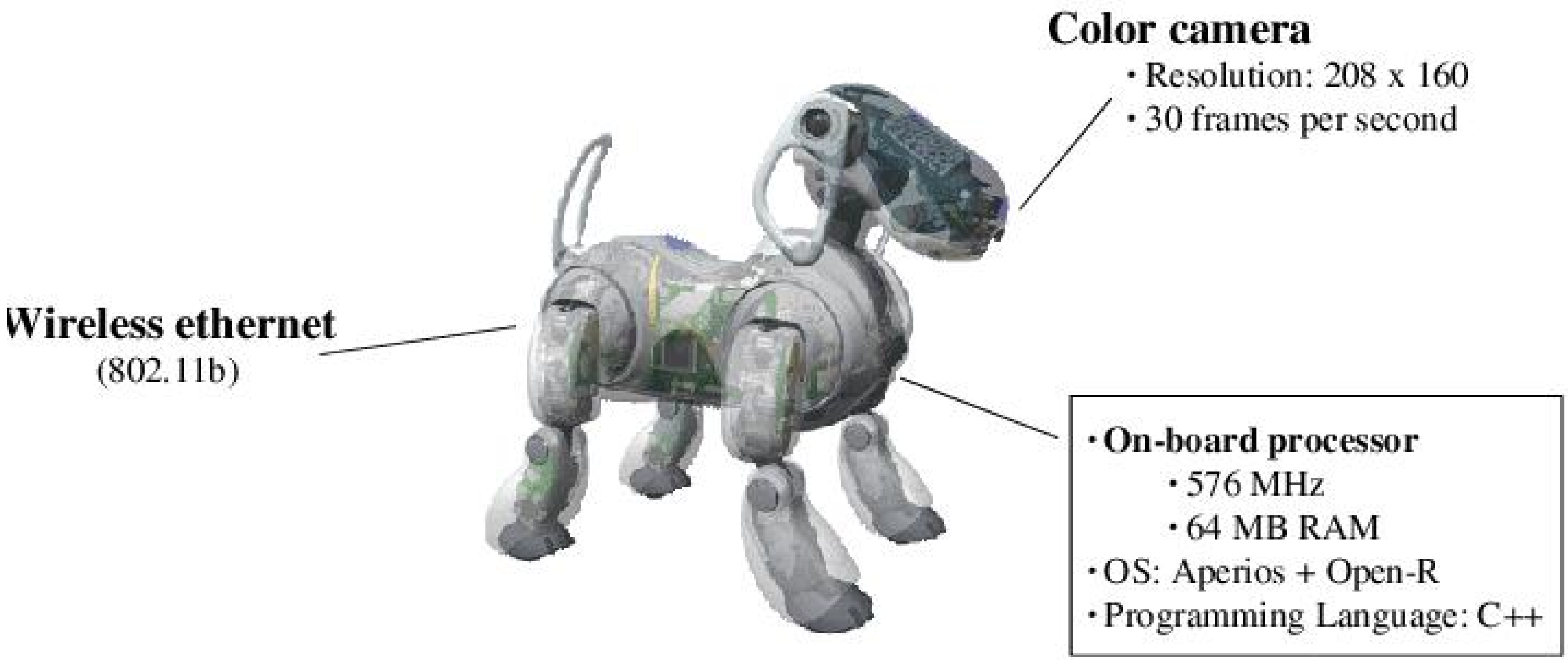
A Decade Later



Sony Aibo (ERS-210A, ERS-7)

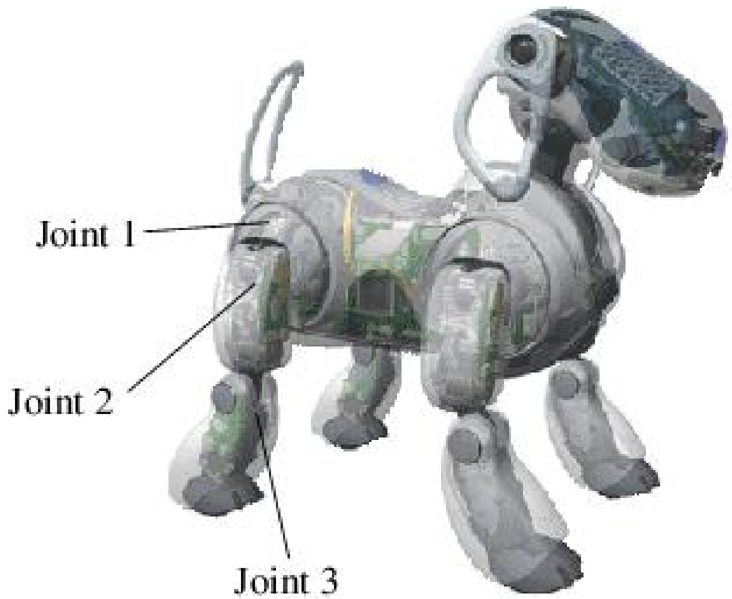


Sony Aibo (ERS-210A, ERS-7)



Sony Aibo (ERS-210A, ERS-7)

20 degrees of freedom



- head: 3 neck, 2 ears, 1 mouth
- **4 legs: 3 joints each**
- tail: 2 DOF

Creating a team — Subtasks

Creating a team — Subtasks

- Vision
- Localization
- Walking
- Ball manipulation (kicking)
- Individual decision making
- Communication/coordination

Creating a team — Subtasks

- Vision
- Localization
- Walking
- Ball manipulation (kicking)
- Individual decision making
- Communication/coordination

Competitions

- Barely "closed the loop" by American Open **(May, '03)**

Competitions

- Barely "closed the loop" by American Open **(May, '03)**
- Improved significantly by Int'l RoboCup **(July, '03)**

Competitions

- Barely "closed the loop" by American Open **(May, '03)**
- Improved significantly by Int'l RoboCup **(July, '03)**
- Won **3rd place** at US Open **(2004, 2005)**
- **Quarterfinalist** at RoboCup **(2004, 2005)**

Competitions

- Barely "closed the loop" by American Open **(May, '03)**
- Improved significantly by Int'l RoboCup **(July, '03)**
- Won **3rd place** at US Open **(2004, 2005)**
- **Quarterfinalist** at RoboCup **(2004, 2005)**
- Highlights:
  - Many saves: 1; 2; 3; 4;
  - Lots of goals: CMU; Penn; Penn; Germany;

  - A nice clear
  - A counterattack goal

Post-competition: the CS research

Post-competition: the CS research

- Model-based joint control (Stronger, S, '04)
- Learning sensor and action models (Stronger, S, '06)
- **Machine learning for fast walking** (Kohl, S, '04)
- **Learning to acquire the ball** (Fidelman, S, '06)
- **Color constancy on mobile robots** (Sridharan, S, '04)
- **Robust particle filter localization** (Sridharan, Kuhlmann, S, '05)
- **Autonomous Color Learning** (Sridharan, S, '05)

Policy Gradient RL to learn fast walk

Goal: Enable an Aibo to walk as fast as possible

Policy Gradient RL to learn fast walk

Goal: Enable an Aibo to walk as fast as possible

- Start with a **parameterized walk**
- **Learn** fastest possible parameters

Policy Gradient RL to learn fast walk

Goal: Enable an Aibo to walk as fast as possible

- Start with a **parameterized walk**
- **Learn** fastest possible parameters
- **No simulator** available:
  - Learn entirely on robots
  - Minimal human intervention

Walking Aibos

- Walks that "come with" Aibo are **slow**
- **RoboCup** soccer: **25+ Aibo teams** internationally
  - Motivates faster walks

Walking Aibos

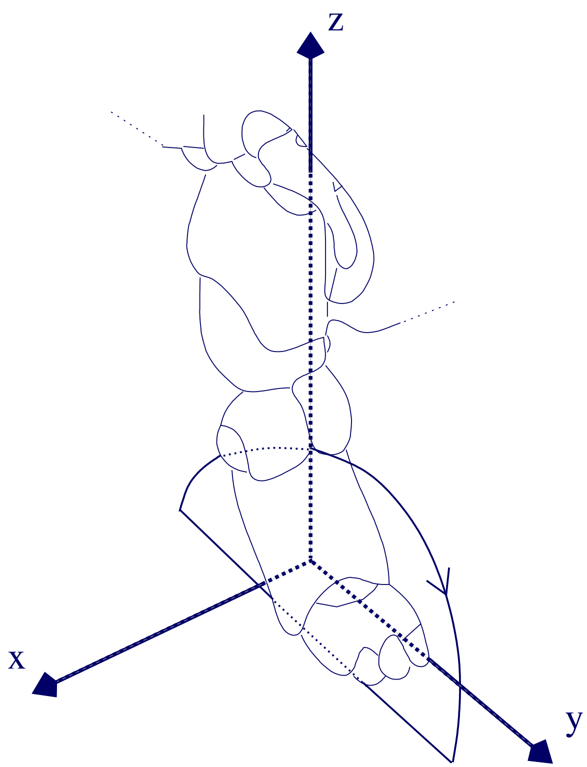
- Walks that "come with" Aibo are **slow**
- **RoboCup** soccer: **25+ Aibo teams** internationally
  - Motivates faster walks

Hand-tuned gaits (2003)			Learned gaits	
German Team	UT Austin Villa	UNSW	Hornby et al. (1999)	Kim & Uther (2003)
230 mm/s	**245**	**254**	**170**	**270** (± 5)

A Parameterized Walk

- Developed from scratch as part of **UT Austin Villa** 2003
- **Trot gait** with elliptical locus on each leg

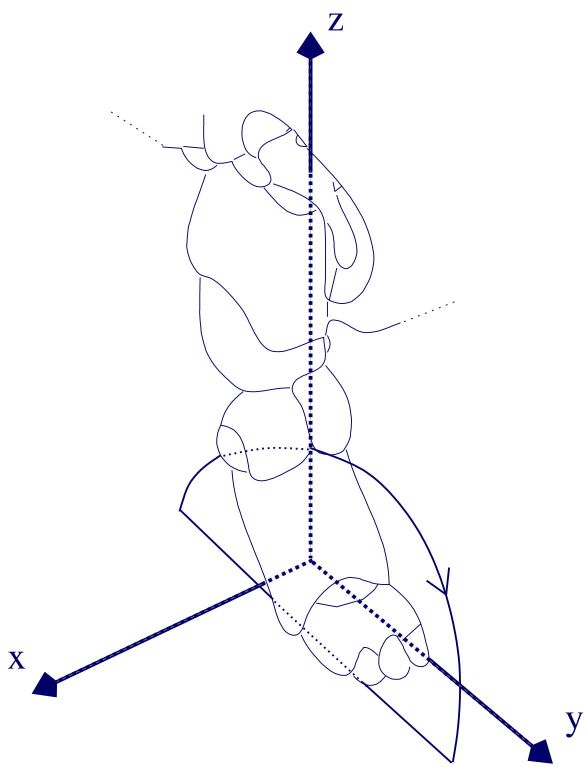
Locus Parameters



- Ellipse length
- Ellipse height
- Position on x axis
- Position on y axis
- Body height
- Timing values

12 continuous parameters

Locus Parameters



- Ellipse length
- Ellipse height
- Position on x axis
- Position on y axis
- Body height
- Timing values

12 continuous parameters

- Hand tuning by April, '03: **140 mm/s**
- Hand tuning by July, '03: **245 mm/s**

Experimental Setup

- Policy $\pi = \{\theta_1, \ldots, \theta_{12}\}$, $V(\pi)$ = walk **speed** when using π

Experimental Setup

- Policy $\pi = \{\theta_1, \ldots, \theta_{12}\}$, $V(\pi)$ = walk **speed** when using π
- Training Scenario
  - Robots **time themselves** traversing fixed distance
  - Multiple traversals (3) per policy to account for **noise**

Experimental Setup

- Policy $\pi = \{\theta_1, \ldots, \theta_{12}\}$, $V(\pi)$ = walk **speed** when using π
- Training Scenario
  - Robots **time themselves** traversing fixed distance
  - Multiple traversals (3) per policy to account for **noise**
  - **Multiple robots** evaluate policies simultaneously
  - Off-board computer **collects results**, **assigns policies**

Experimental Setup

- Policy $\pi = \{\theta_1, \ldots, \theta_{12}\}$, $V(\pi)$ = walk **speed** when using π
- Training Scenario
  - Robots **time themselves** traversing fixed distance
  - Multiple traversals (3) per policy to account for **noise**
  - **Multiple robots** evaluate policies simultaneously
  - Off-board computer **collects results**, **assigns policies**

No human intervention except battery changes

Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$

Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically

Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically
- $\frac{\partial V(\pi)}{\partial \theta_i} \approx V(\{\theta_1 + \epsilon, \ldots, \theta_{12}\}) - V(\{\theta_1 - \epsilon, \ldots, \theta_{12}\})$

Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically
- $\frac{\partial V(\pi)}{\partial \theta_i} \approx V(\{\theta_1 + \epsilon, \ldots, \theta_{12}\}) - V(\{\theta_1 - \epsilon, \ldots, \theta_{12}\})$
  - Requires evaluation of **24** policies

Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically
- $\frac{\partial V(\pi)}{\partial \theta_i} \approx V(\{\theta_1 + \epsilon, \ldots, \theta_{12}\}) - V(\{\theta_1 - \epsilon, \ldots, \theta_{12}\})$
  - Requires evaluation of **24** policies
- Instead, evaluate t **(15)** policies in the neighborhood of π s.t. ith parameter is **randomly** $\theta_i \pm \epsilon$ or 0.

Policy Gradient RL

- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically

- $\frac{\partial V(\pi)}{\partial \theta_i} \approx V(\{\theta_1 + \epsilon, \ldots, \theta_{12}\}) - V(\{\theta_1 - \epsilon, \ldots, \theta_{12}\})$
  - Requires evaluation of **24** policies

- Instead, evaluate t **(15)** policies in the neighborhood of π s.t. ith parameter is **randomly** $\theta_i \pm \epsilon$ or 0.

- $V(\{\theta_1 + \epsilon, \ldots, \theta_{12}\}) \approx Avg_{+\epsilon,1} \equiv$ policies with $\theta_1 + \epsilon$

Policy Gradient RL

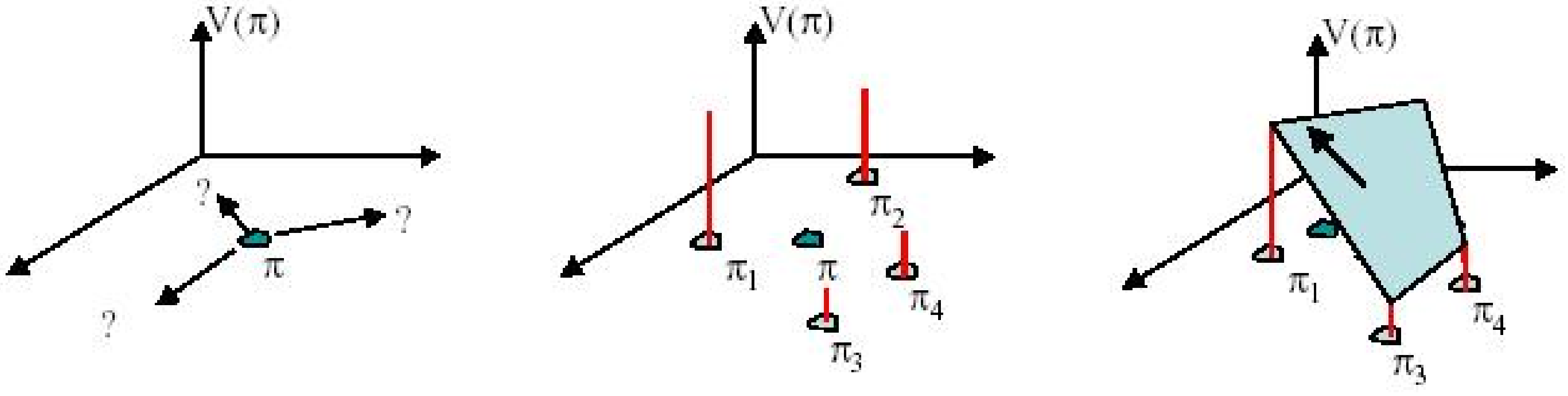
- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically

- $\frac{\partial V(\pi)}{\partial \theta_i} \approx V(\{\theta_1 + \epsilon, \ldots, \theta_{12}\}) - V(\{\theta_1 - \epsilon, \ldots, \theta_{12}\})$
  - Requires evaluation of **24** policies

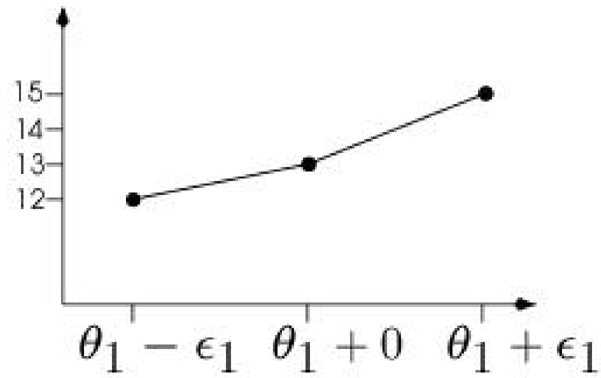
- Instead, evaluate t **(15)** policies in the neighborhood of π s.t. ith parameter is **randomly** $\theta_i \pm \epsilon$ or 0.

- $V(\{\theta_1 + \epsilon, \ldots, \theta_{12}\}) \approx Avg_{+\epsilon,1} \equiv$ policies with $\theta_1 + \epsilon$
  - Expect $t/3$ estimates for each of $\theta_i \pm \epsilon$, 0
  - Each evaluation contributes to **all 12** estimates

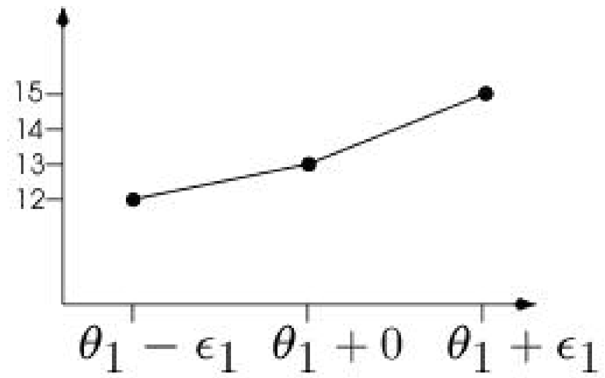
Gradient Estimation



Taking a step

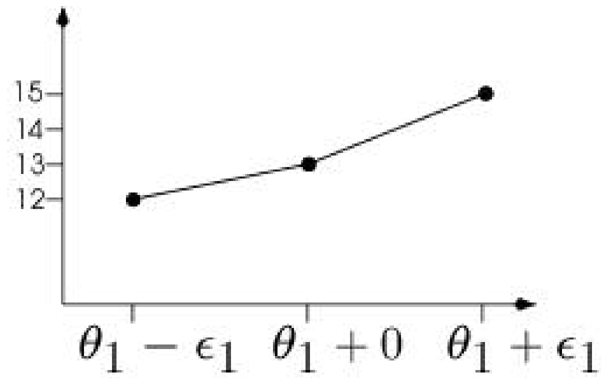


Taking a step



$$A_i = \begin{cases} 0 & \text{if } Avg_{+0,i} > Avg_{+\epsilon,i} \text{ and} \\ & \quad Avg_{+0,i} > Avg_{-\epsilon,i} \\ Avg_{+\epsilon,i} - Avg_{-\epsilon,i} & \text{otherwise} \end{cases} \tag{1}$$

Taking a step



$$A_i = \begin{cases} 0 & \text{if } Avg_{+0,i} > Avg_{+\epsilon,i} \text{ and} \\ & \quad Avg_{+0,i} > Avg_{-\epsilon,i} \\ Avg_{+\epsilon,i} - Avg_{-\epsilon,i} & \text{otherwise} \end{cases} \tag{1}$$

- **Normalize** A, multiply by scalar **step-size** η
- $\pi = \pi + \eta A$

Experiments

- Started from **stable**, but fairly slow gait
- Used **3 robots** simultaneously
- Each iteration takes 45 traversals, $7\frac{1}{2}$ minutes

Experiments

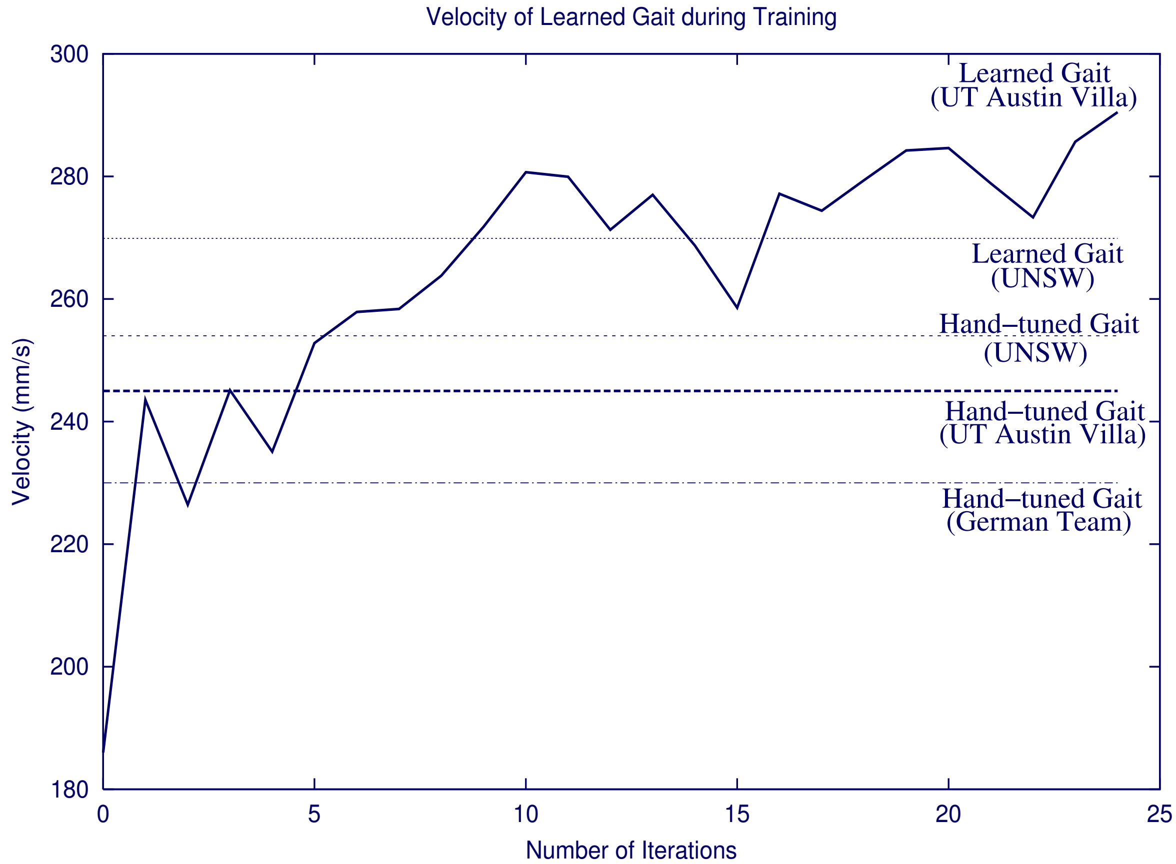
- Started from **stable**, but fairly slow gait
- Used **3 robots** simultaneously
- Each iteration takes 45 traversals, $7\frac{1}{2}$ minutes

Before learning

After learning

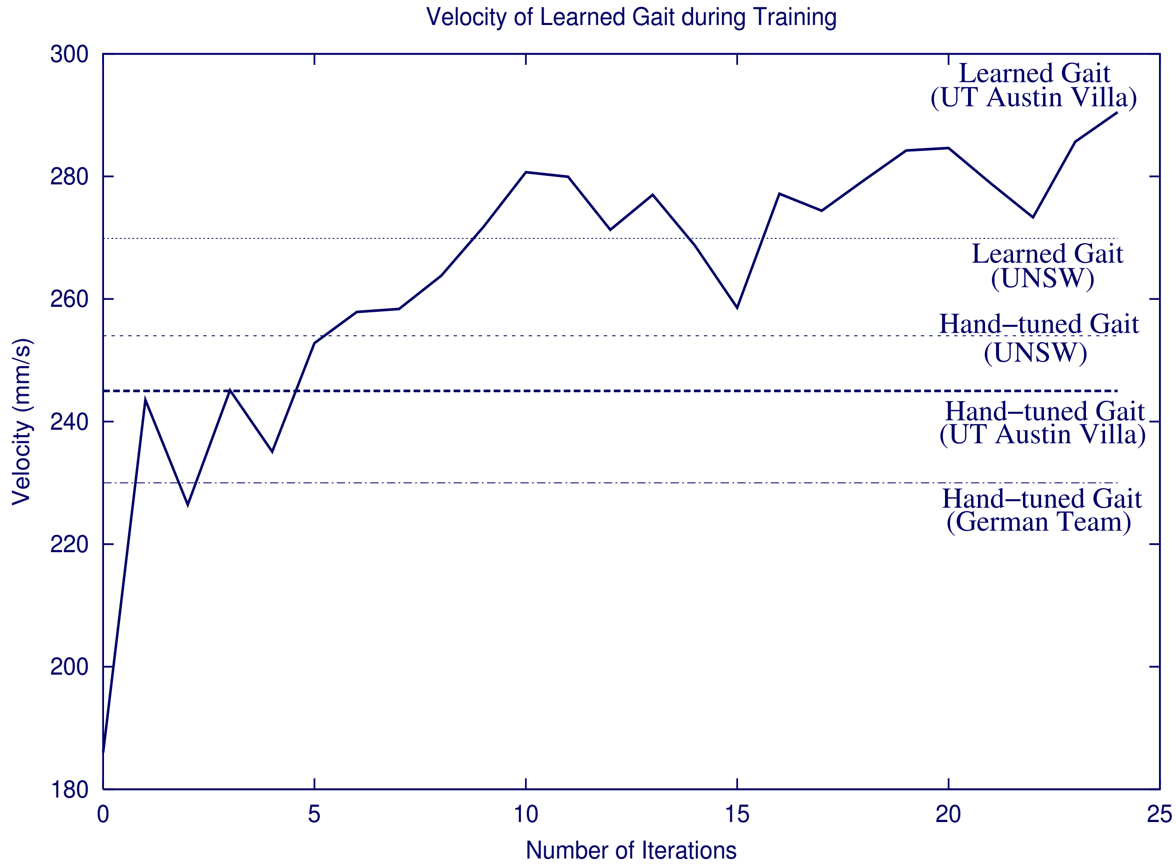
- 24 iterations = **1080 field traversals**, $\approx$ **3 hours**

Results



Velocity of Learned Gait during Training
Velocity (mm/s)
Number of Iterations
300
280
260
240
220
200
180
0
5
10
15
20
25
Learned Gait (UT Austin Villa)
Learned Gait (UNSW)
Hand–tuned Gait (UNSW)
Hand–tuned Gait (UT Austin Villa)
Hand–tuned Gait (German Team)

Results

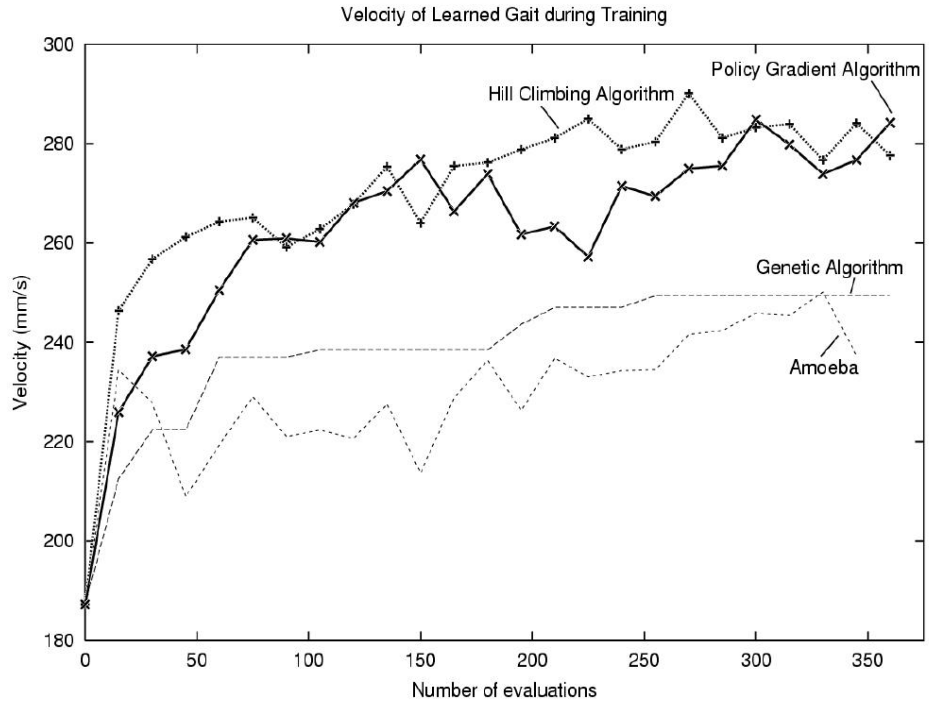


- Additional iterations didn't help
- Spikes: evaluation **noise**? large **step size**?

Learned Parameters

Parameter	Initial Value	ϵ	Best Value
Front ellipse:			
(height)	4.2	0.35	4.081
(x offset)	2.8	0.35	0.574
(y offset)	4.9	0.35	5.152
Rear ellipse:			
(height)	5.6	0.35	6.02
(x offset)	0.0	0.35	0.217
(y offset)	-2.8	0.35	-2.982
Ellipse length	4.893	0.35	5.285
Ellipse skew multiplier	0.035	0.175	0.049
Front height	7.7	0.35	7.483
Rear height	11.2	0.35	10.843
Time to move through locus	0.704	0.016	0.679
Time on ground	0.5	0.05	**0.430**

Algorithmic Comparison, Robot Port



Before learning

After learning

Summary

- Used policy gradient RL to **learn fastest Aibo walk**
- All learning done **on real robots**
- **No human itervention** (except battery changes)

Outline

- Machine learning for fast walking (Kohl, S, '04)
- **Learning to acquire the ball** (Fidelman, S, '06)
- Color constancy on mobile robots (Sridharan, S, '05)
- Autonomous Color Learning (Sridharan, S, '06)

Grasping the Ball

- **Three stages:** walk to ball; slow down; lower chin
- Head proprioception, IR chest sensor $\mapsto$ ball distance
- Movement specified by **4 parameters**

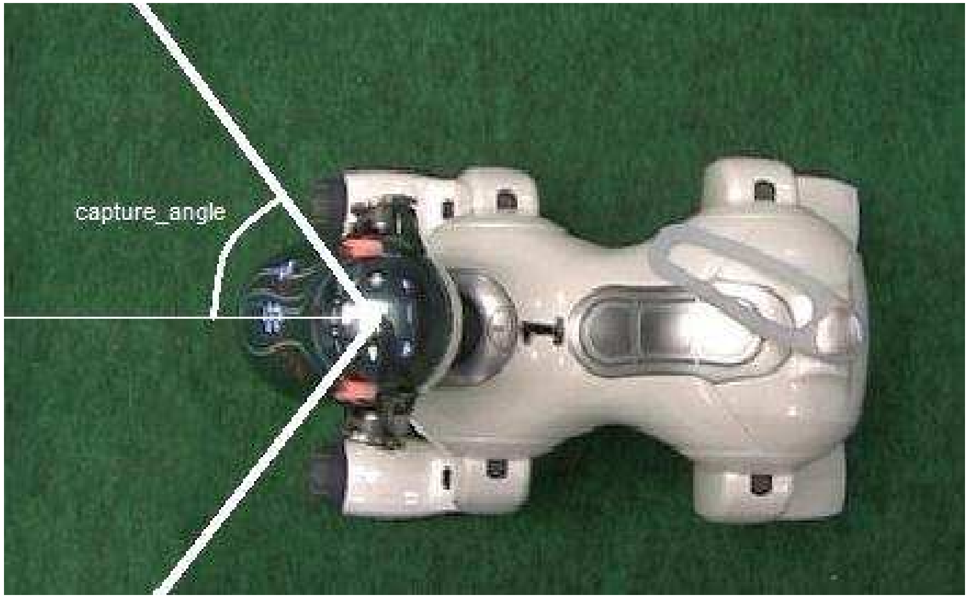
Grasping the Ball

- **Three stages:** walk to ball; slow down; lower chin
- Head proprioception, IR chest sensor $\mapsto$ ball distance
- Movement specified by **4 parameters**

$\boxed{\textbf{Brittle!}}$

Parameterization

- **slowdown_dist:** when to slow down
- **slowdown_factor:** how much to slow down
- **capture_angle:** when to stop turning



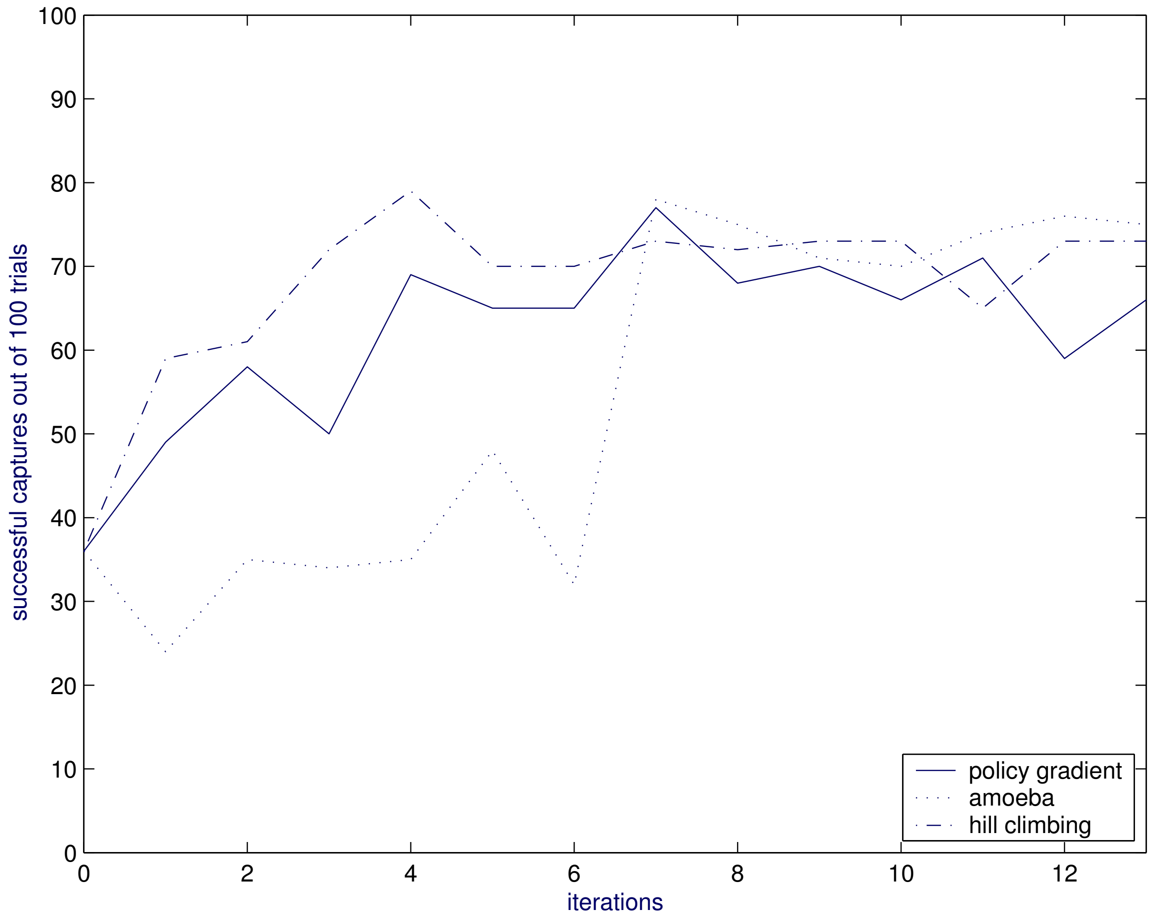
- **capture_dist:** when to put down head

Learning the Chin Pinch

- **Binary, noisy** reinforcement signal: multiple trials
- Robot evaluates self: **no human intervention**

Results

- Evaluation of **policy gradient**, **hill climbing**, **amoeba**



What it learned

Policy	slowdown dist	slowdown factor	capture angle	capture dist	Success rate
Initial	200mm	0.7	15.0^o	110mm	36%
Policy gradient	125mm	1	17.4^o	152mm	64%
Amoeba	208mm	1	33.4^o	162mm	69%
Hill climbing	240mm	1	35.0^o	170mm	66%

Outline

- Machine learning for fast walking (Kohl, S, '04)
- Learning to acquire the ball (Fidelman, S, '06)
- **Color constancy on mobile robots** (Sridharan, S, '05)
- **Autonomous Color Learning** (Sridharan, S, '06)

Color Constancy

- Visual system's ability to recognize **true color** across variations in environment

Color Constancy

- Visual system's ability to recognize **true color** across variations in environment
- Challenge: Nonlinear variations in sensor response with **change in illumination**

Color Constancy

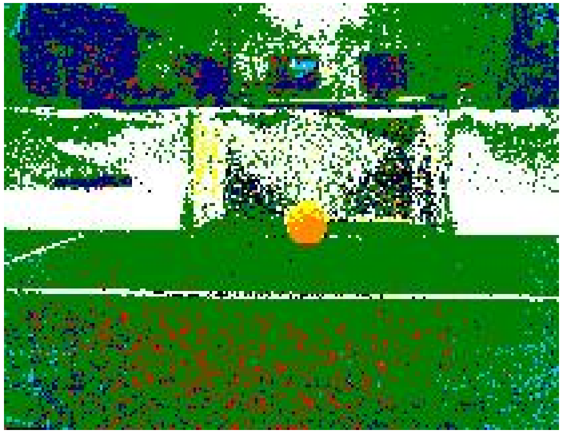
- Visual system's ability to recognize **true color** across variations in environment
- Challenge: Nonlinear variations in sensor response with **change in illumination**
- **Mobile robots**:
  - Computational limitations
  - Changing camera positions

Sample Images

Sample Images

Sample Images

Sample Images



Our Goal

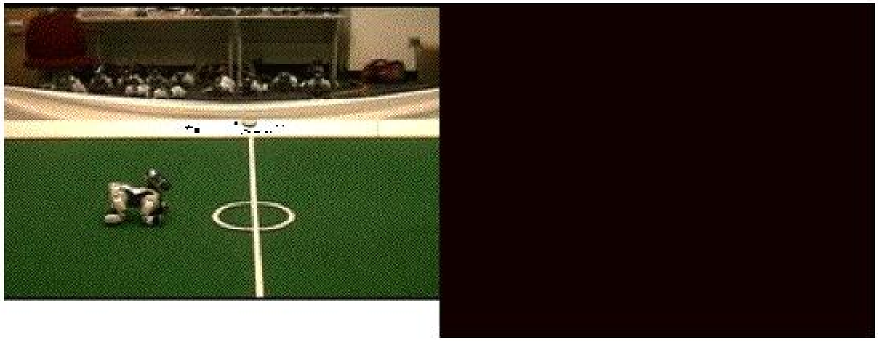
- Match current performance in **changing lighting**
- Experiments on ERS-210A robots

Autonomous Color Learning

- Color Constancy: **more** tediously created maps
  - Hand-labeling many images $\longrightarrow$ hours of **manual effort**

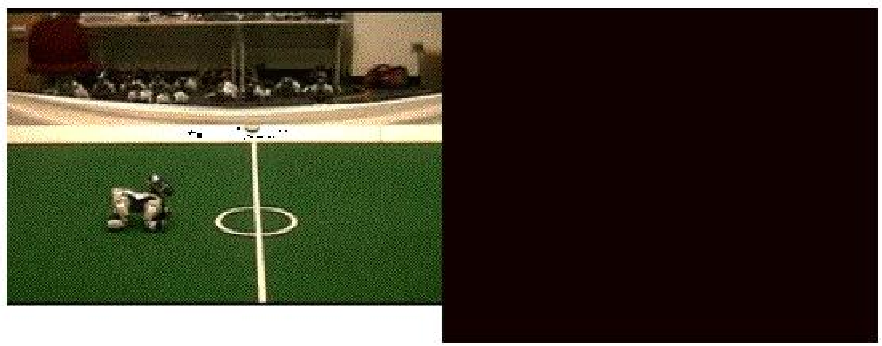
Autonomous Color Learning

- Color Constancy: **more** tediously created maps
  - Hand-labeling many images $\longrightarrow$ hours of **manual effort**
- Use the **structured environment**
  - Robot learns color **distributions**



Autonomous Color Learning

- Color Constancy: **more** tediously created maps
  - Hand-labeling many images $\longrightarrow$ hours of **manual effort**
- Use the **structured environment**
  - Robot learns color **distributions**



- Comparable accuracy, 5 minutes of **robot effort**

Outline

- Learning on **physical robots**
  - No simulation, minimal human intervention

Outline

- Learning on **physical robots**
  - No simulation, minimal human intervention
- **Motion:** learning for fast walking
- **Behavior:** acquiring the ball
- **Vision:** color constancy, autonomous color learning

Outline

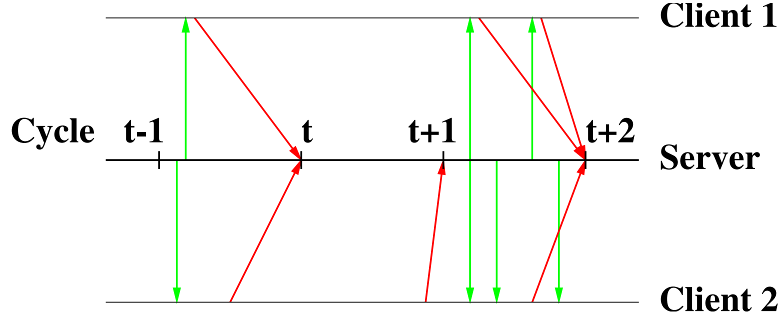
- Learning on **physical robots**
  - No simulation, minimal human intervention
- **Motion:** learning for fast walking
- **Behavior:** acquiring the ball
- **Vision:** color constancy, autonomous color learning
- **Multiagent Strategy:** RL in simulation

RoboCup Simulator

- Distributed: each player a separate client
- Server models dynamics and kinematics

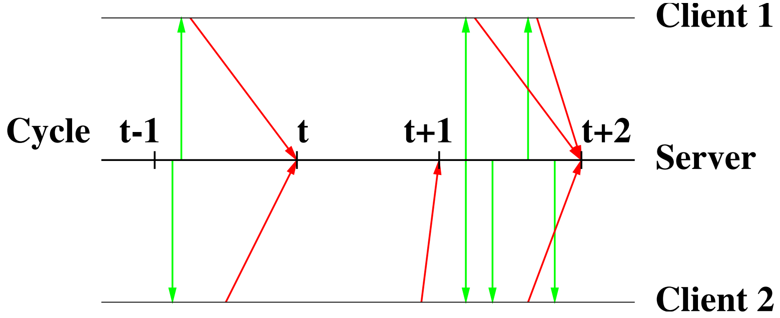
RoboCup Simulator

- Distributed: each player a separate client
- Server models dynamics and kinematics
- Clients receive sensations, send actions



RoboCup Simulator

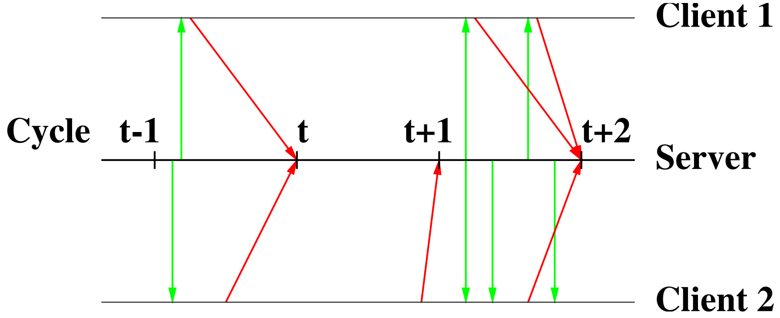
- Distributed: each player a separate client
- Server models dynamics and kinematics
- Clients receive sensations, send actions



- Parametric actions: dash, turn, kick, say

RoboCup Simulator

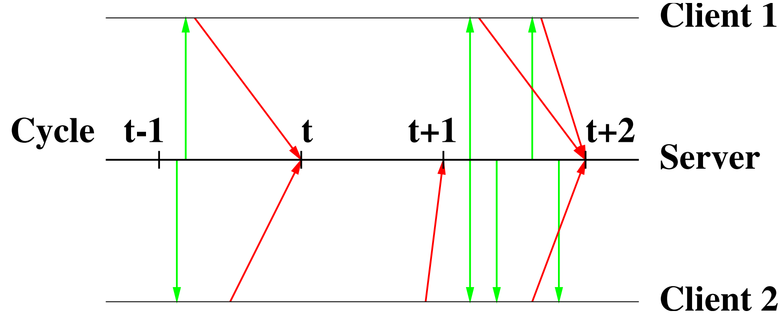
- Distributed: each player a separate client
- Server models dynamics and kinematics
- Clients receive sensations, send actions



- Parametric actions: dash, turn, kick, say
- Abstract, noisy sensors, hidden state
  - Hear sounds from limited distance
  - See relative distance, angle to objects ahead

RoboCup Simulator

- Distributed: each player a separate client
- Server models dynamics and kinematics
- Clients receive sensations, send actions



- Parametric actions: dash, turn, kick, say
- Abstract, noisy sensors, hidden state
  - Hear sounds from limited distance
  - See relative distance, angle to objects ahead
- $> 10^{9^{23}}$ states
- Limited resources : stamina
- Play occurs in real time ($\approx$ human parameters)

3 vs. 2 Keepaway

3 vs. 2 Keepaway

- Play in a **small area** (20m $\times$ 20m)
- **Keepers** try to keep the ball
- **Takers** try to get the ball

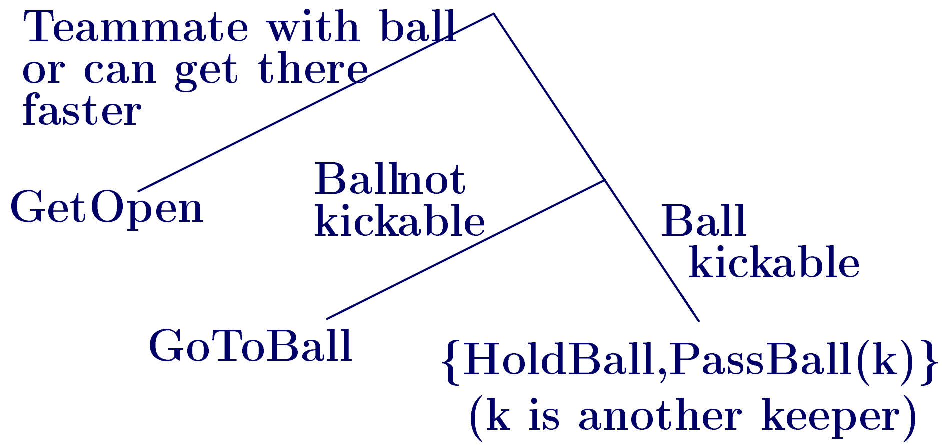
3 vs. 2 Keepaway

- Play in a **small area** (20m $\times$ 20m)
- **Keepers** try to keep the ball
- **Takers** try to get the ball
- **Episode:**
  - Players and ball reset randomly
  - Ball starts near a keeper
  - Ends when taker gets the ball or ball goes out

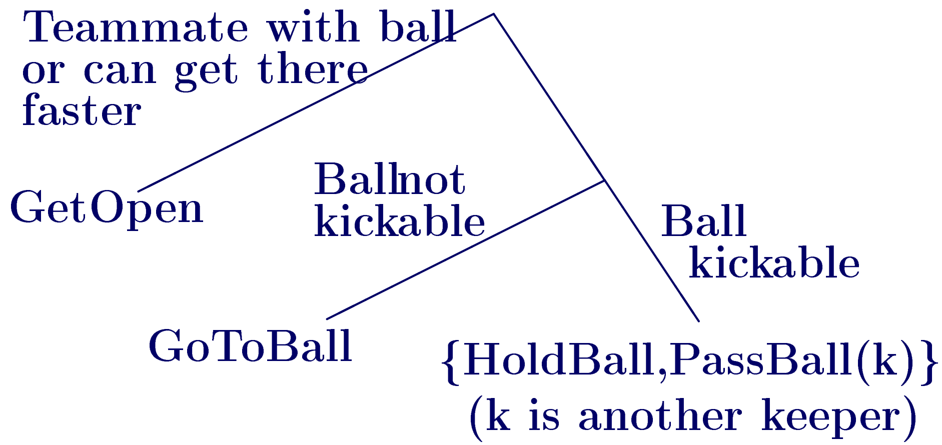
3 vs. 2 Keepaway

- Play in a **small area** (20m $\times$ 20m)
- **Keepers** try to keep the ball
- **Takers** try to get the ball
- **Episode:**
  – Players and ball reset randomly
  – Ball starts near a keeper
  – Ends when taker gets the ball or ball goes out
- Performance measure: **average possession duration**
- Use **CMUnited-99 skills**:
  – HoldBall, PassBall(k), GoToBall, GetOpen

The Keepers' Policy Space



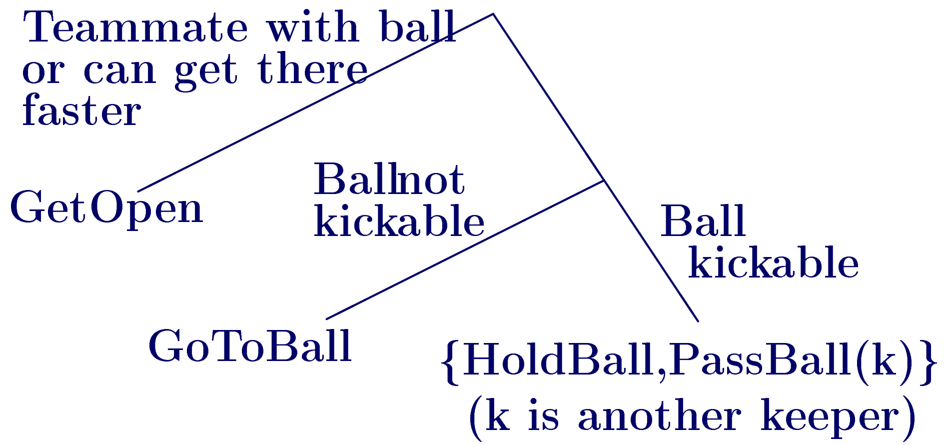
The Keepers' Policy Space



Example Policies

Random: HoldBall or PassBall(k) randomly

The Keepers' Policy Space

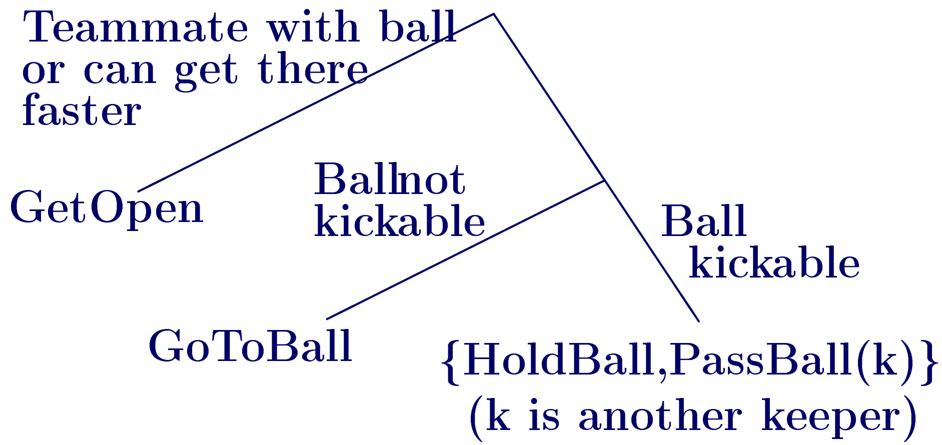


Example Policies

Random: HoldBall or PassBall(k) randomly

Hold: Always HoldBall

The Keepers' Policy Space



Example Policies

Random: HoldBall or PassBall(k) randomly

Hold: Always HoldBall

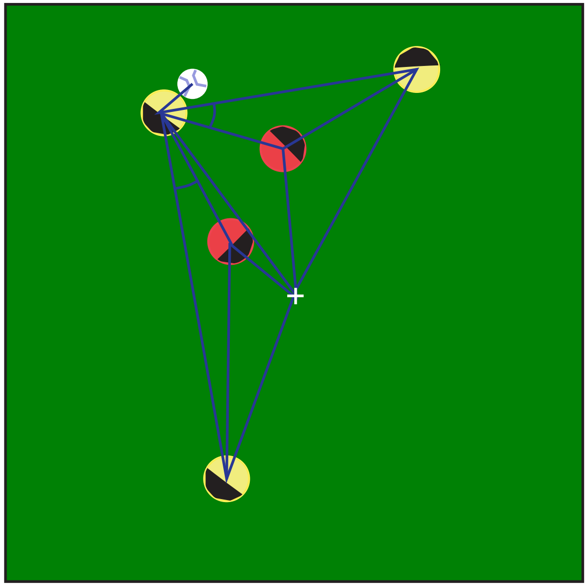
Hand-coded:

If no taker within 10m: HoldBall

Else If there's a good pass: PassBall(k)

Else HoldBall

Keeper's State Variables



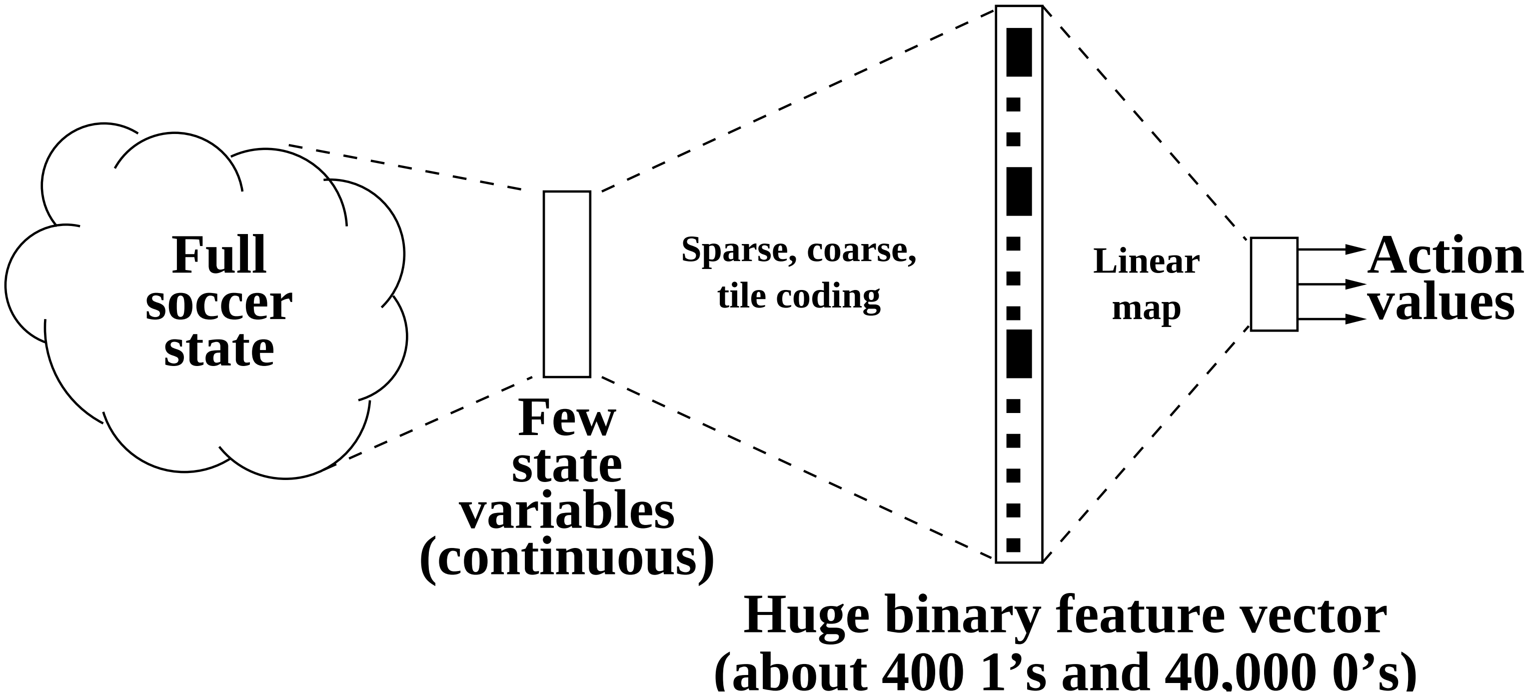
- 11 distances among players, ball, and center
- *2 angles* to takers along passing lanes

Function Approximation: Tile Coding

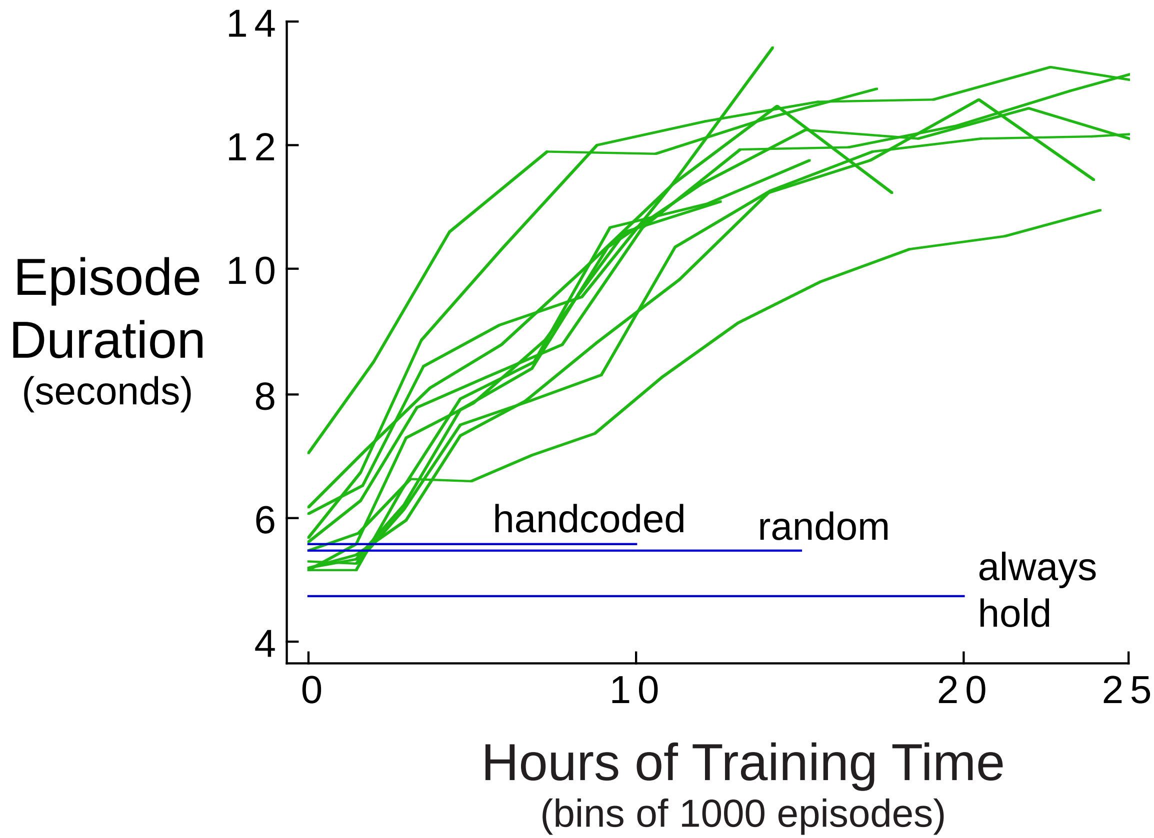
- Form of sparse, coarse coding based on CMACS (Albus, 1981)

Function Approximation: Tile Coding

- Form of sparse, coarse coding based on CMACS (Albus, 1981)

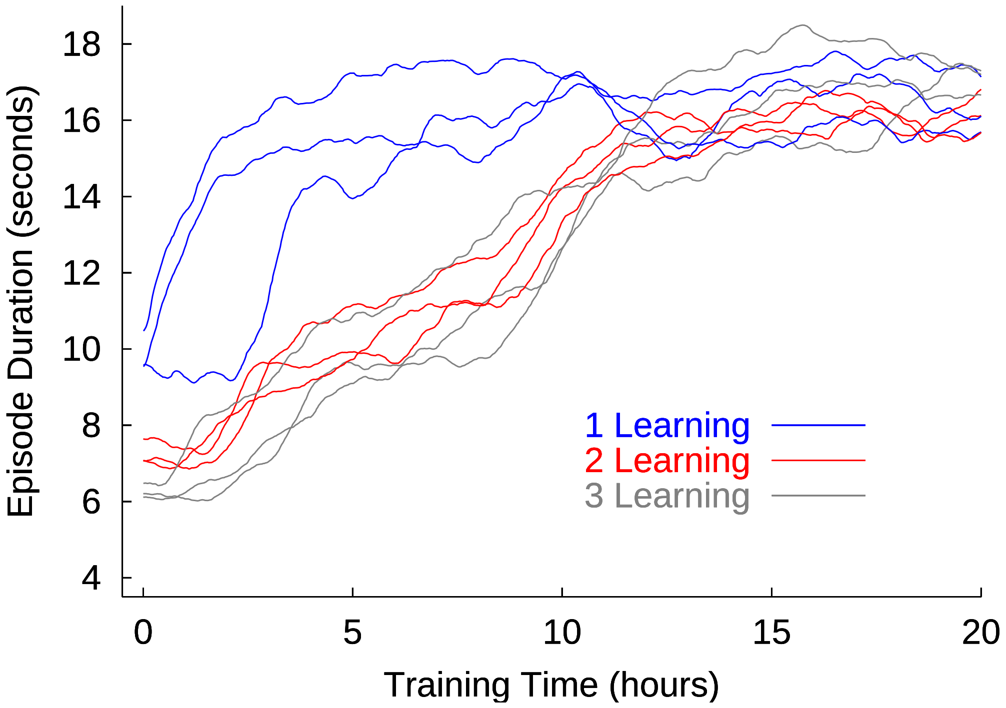


Main Result



1 hour = 720 5-second episodes

Difficulty of Multiagent Learning



- Multiagent learning is harder!

Outline

- Robot soccer on real robots
- Robot soccer in simulation

Outline

- Robot soccer on real robots
- Robot soccer in simulation
- Autonomous driving

Acknowledgements

Thanks to all the Students Involved!

- Kurt Dresner, Nate Kohl, Peggy Fidelman, Mohan Sridharan, Richard Sutton
- Other members of the UT Austin Villa Legged Robot Team
- **http://www.cs.utexas.edu/~AustinVilla**

Acknowledgements

Thanks to all the Students Involved!

- Kurt Dresner, Nate Kohl, Peggy Fidelman, Mohan Sridharan, Richard Sutton
- Other members of the UT Austin Villa Legged Robot Team
- **http://www.cs.utexas.edu/~AustinVilla**
- Fox Sports World for inspiration!