CS343
Artificial Intelligence

Prof: Peter Stone

Department of Computer Sciences
The University of Texas at Austin

Good Afternoon, Colleagues

Good Afternoon, Colleagues

Are there any questions?

Logistics

- Exercise responses not all checked

Logistics

- Exercise responses not all checked
- Next week's readings: adversarial search

Logistics

- Exercise responses not all checked
- Next week's readings: adversarial search
- Kautz talk on Friday

Pending Questions

- Can you turn continuous domains into discrete?
- Computing gradient locally not globally?

Continuous Local Search to learn fast walk

Goal: Enable an Aibo to walk as fast as possible

Continuous Local Search to learn fast walk

Goal: Enable an Aibo to walk as fast as possible

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

Continuous Local Search 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

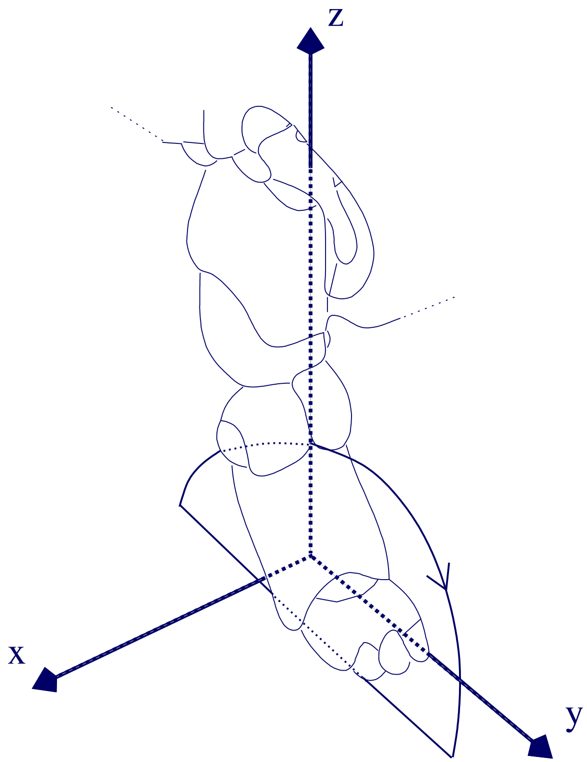
- Walks that "come with" Aibo are **slow**
- **RoboCup** soccer: **25+ Aibo teams** internationally
  - Motivates faster walks

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

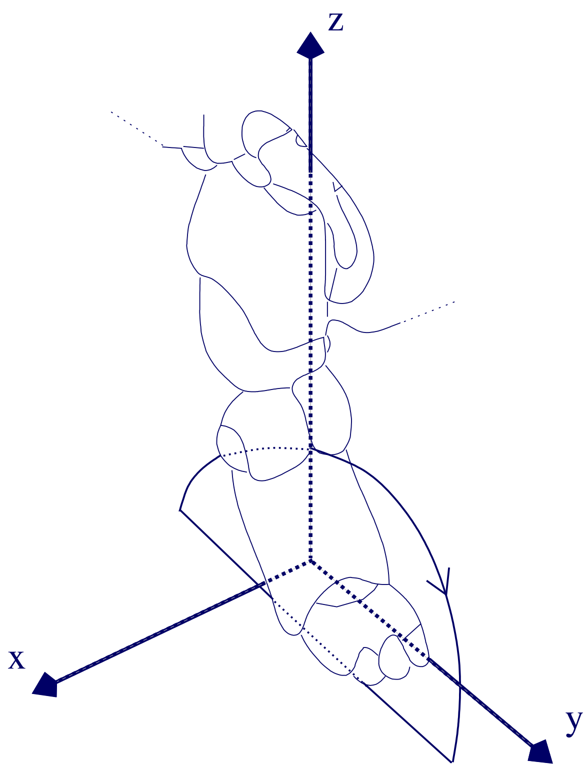
Locus Parameters



- 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**

Parameters To Learn

Parameter	Initial Value
Front ellipse:	
(height)	4.2
(x offset)	2.8
(y offset)	4.9
Rear ellipse:	
(height)	5.6
(x offset)	0.0
(y offset)	-2.8
Ellipse length	4.893
Ellipse skew multiplier	0.035
Front height	7.7
Rear height	11.2
Time to move through locus	0.704
Time on ground	0.5

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

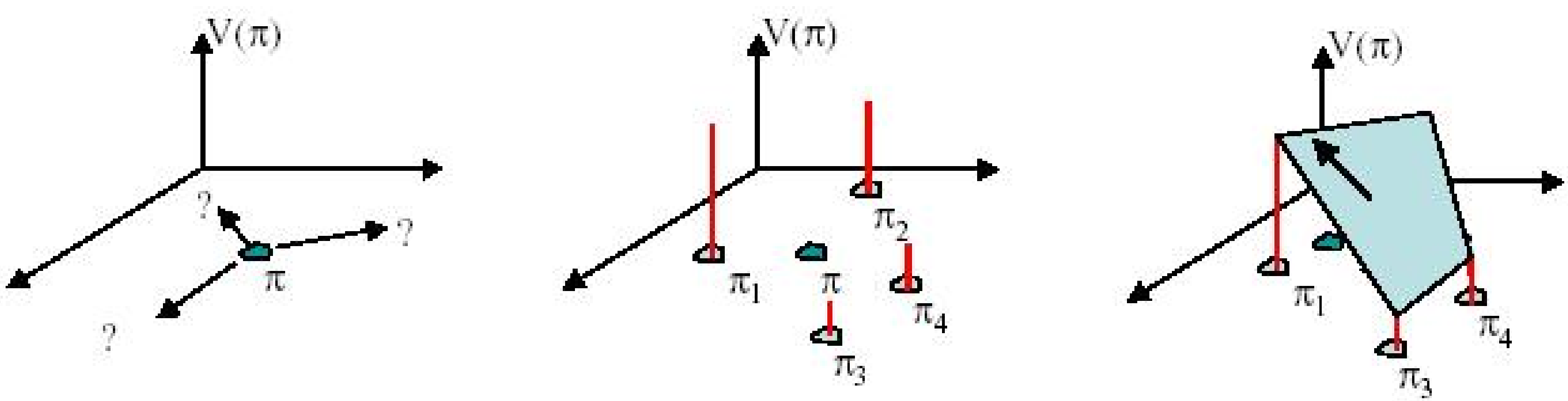
- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically

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
- Evaluate **neighboring policies** to estimate gradient
- Each trial randomly varies **every parameter**

Policy Gradient RL

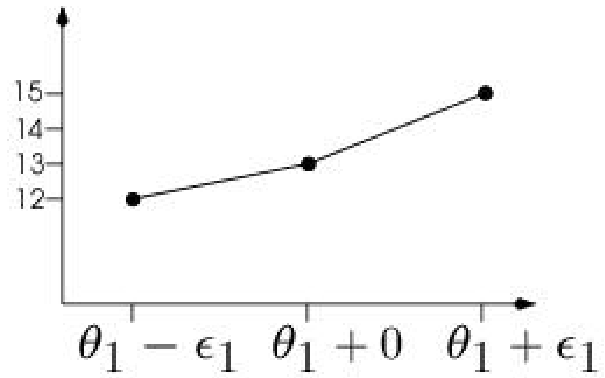
- From π want to move in direction of **gradient** of $V(\pi)$
  - Can't compute $\frac{\partial V(\pi)}{\partial \theta_i}$ directly: **estimate** empirically
- Evaluate **neighboring policies** to estimate gradient
- Each trial randomly varies **every parameter**



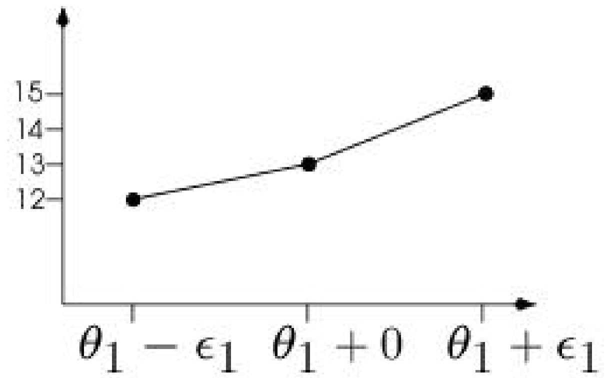
Gradient Estimation

	π_1	$\pi_2 - \pi_N$	Score	
$-\epsilon_1$	$\theta_1 - \epsilon_1$	$\ldots$	11.5	
	$\theta_1 - \epsilon_1$	$\ldots$	12.7	$\Rightarrow$ Average: 12.1
		$\ldots$		
$+0$	$\theta_1 + 0$	$\ldots$	12.3	
	$\theta_1 + 0$	$\ldots$	13.7	$\Rightarrow$ Average: 13.2
		$\ldots$		
$+\epsilon_1$	$\theta_1 + \epsilon_1$	$\ldots$	15.5	
	$\theta_1 + \epsilon_1$	$\ldots$	14.7	$\Rightarrow$ Average: 14.9
		$\ldots$		

Taking a step

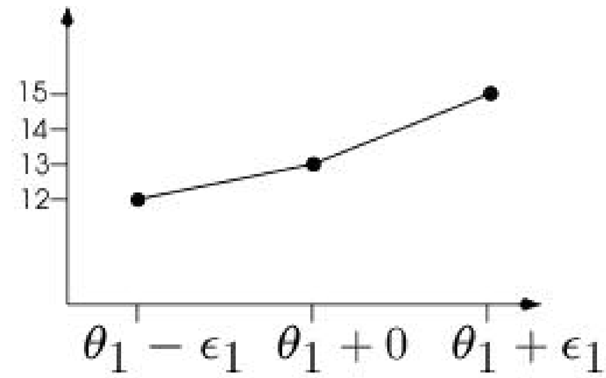


Taking a step



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

Taking a step



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

- **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

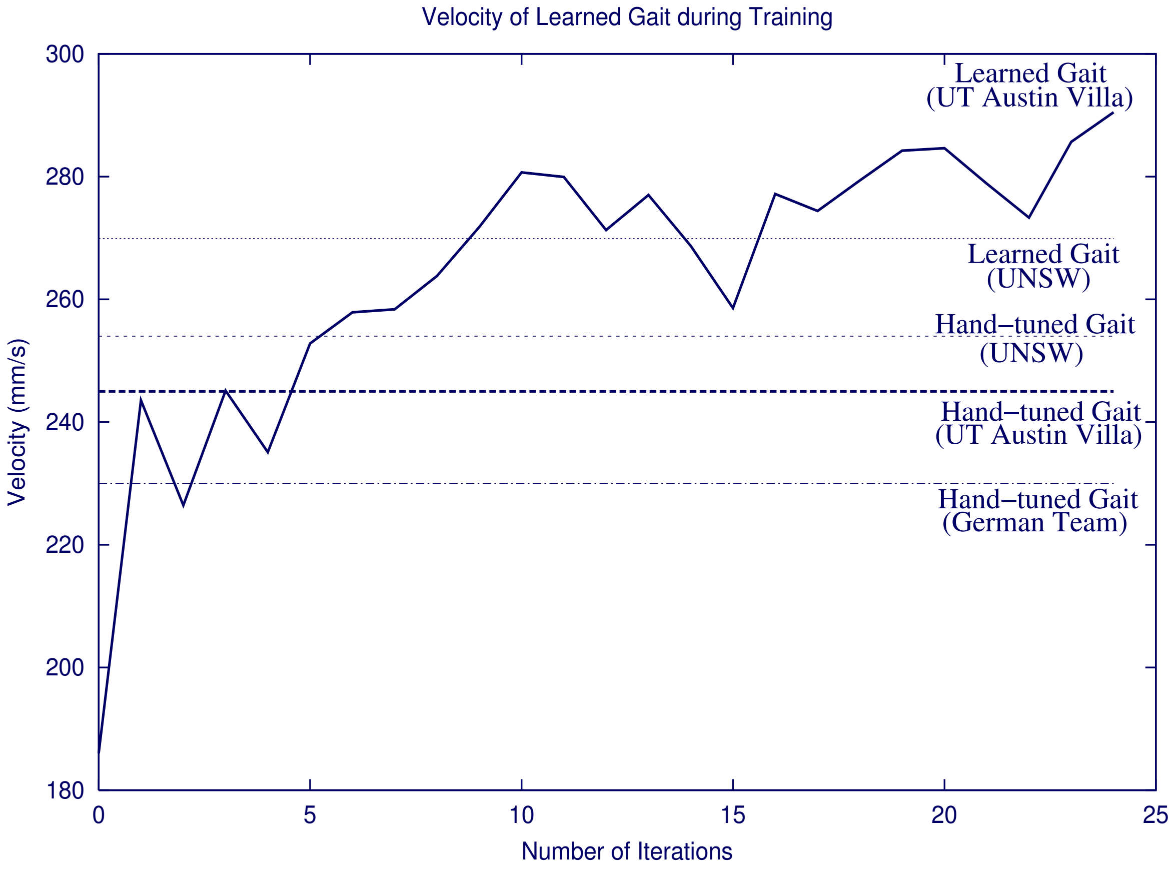
- Started from **stable**, but fairly slow gait
- Used **3 robots** simultaneously
- Each iteration takes 45 traversals, $7\frac{1}{2}$ minutes

Before learning

After learning

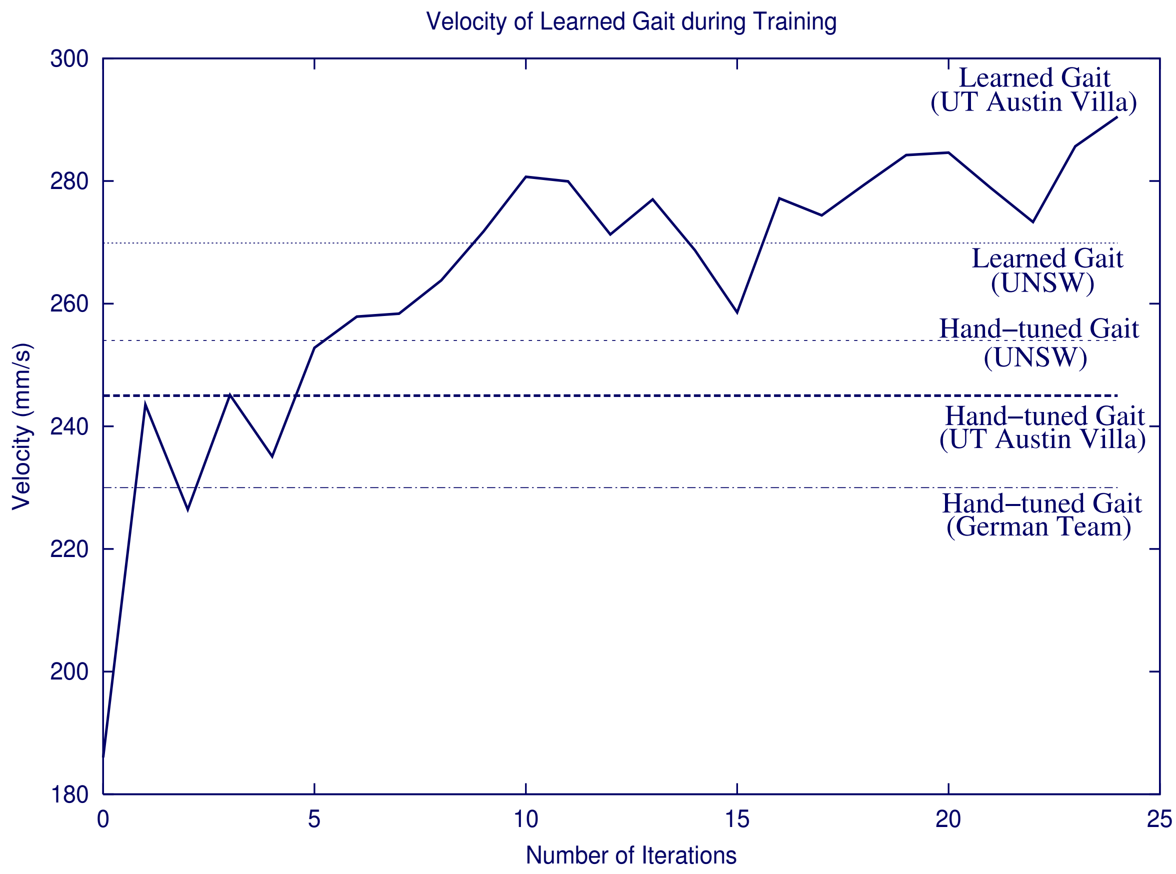
- 24 iterations = **1080 field traversals**, $\approx$ **3 hours**

Results



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

Results

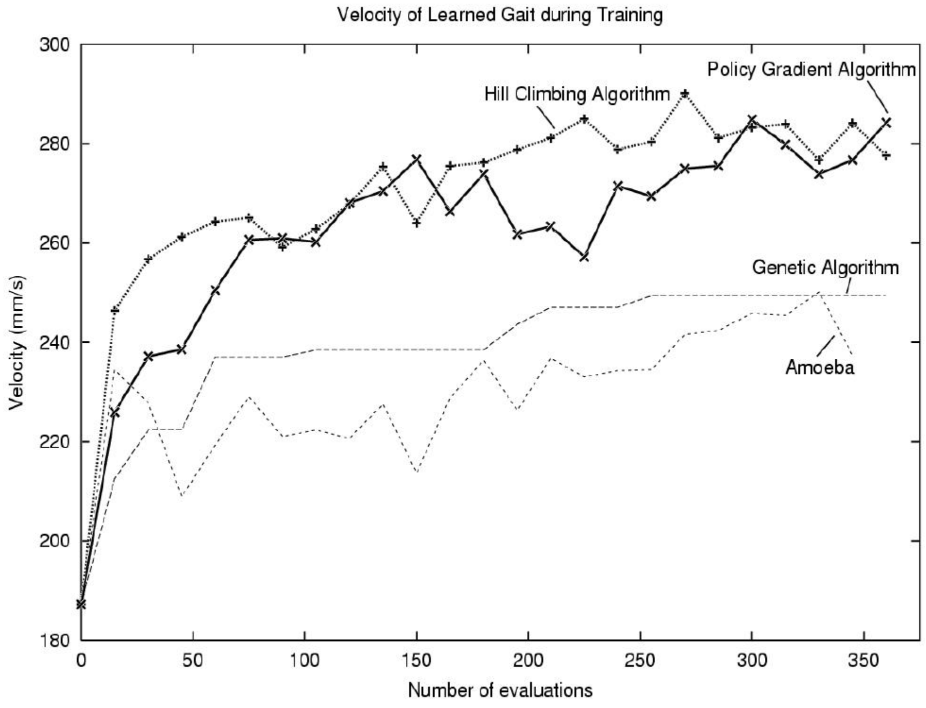


- Additional iterations didn't help
- Spikes: evaluation **noise**? large **step size**?

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)

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**

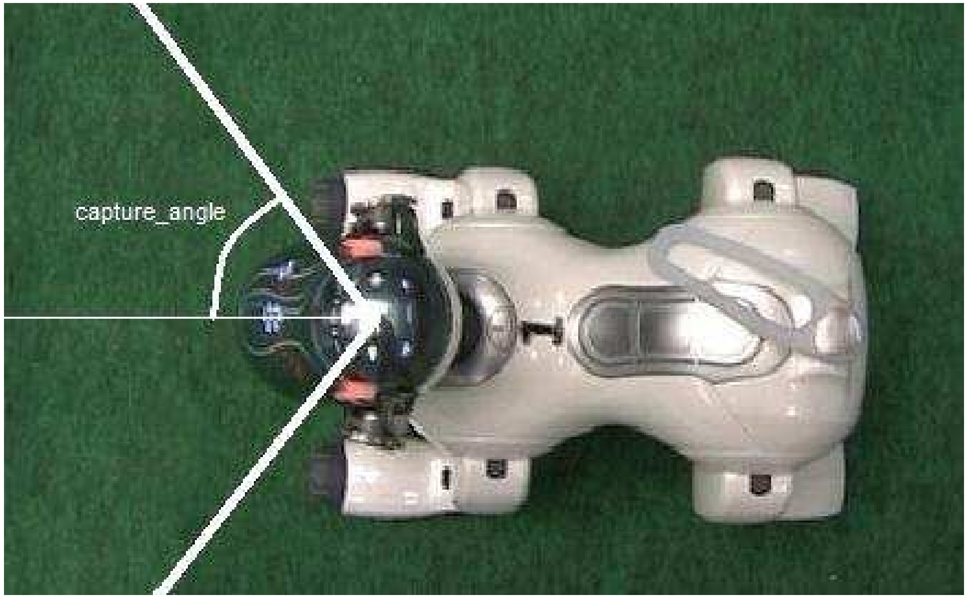
Grasping the Ball

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

Brittle!

Parameterization

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



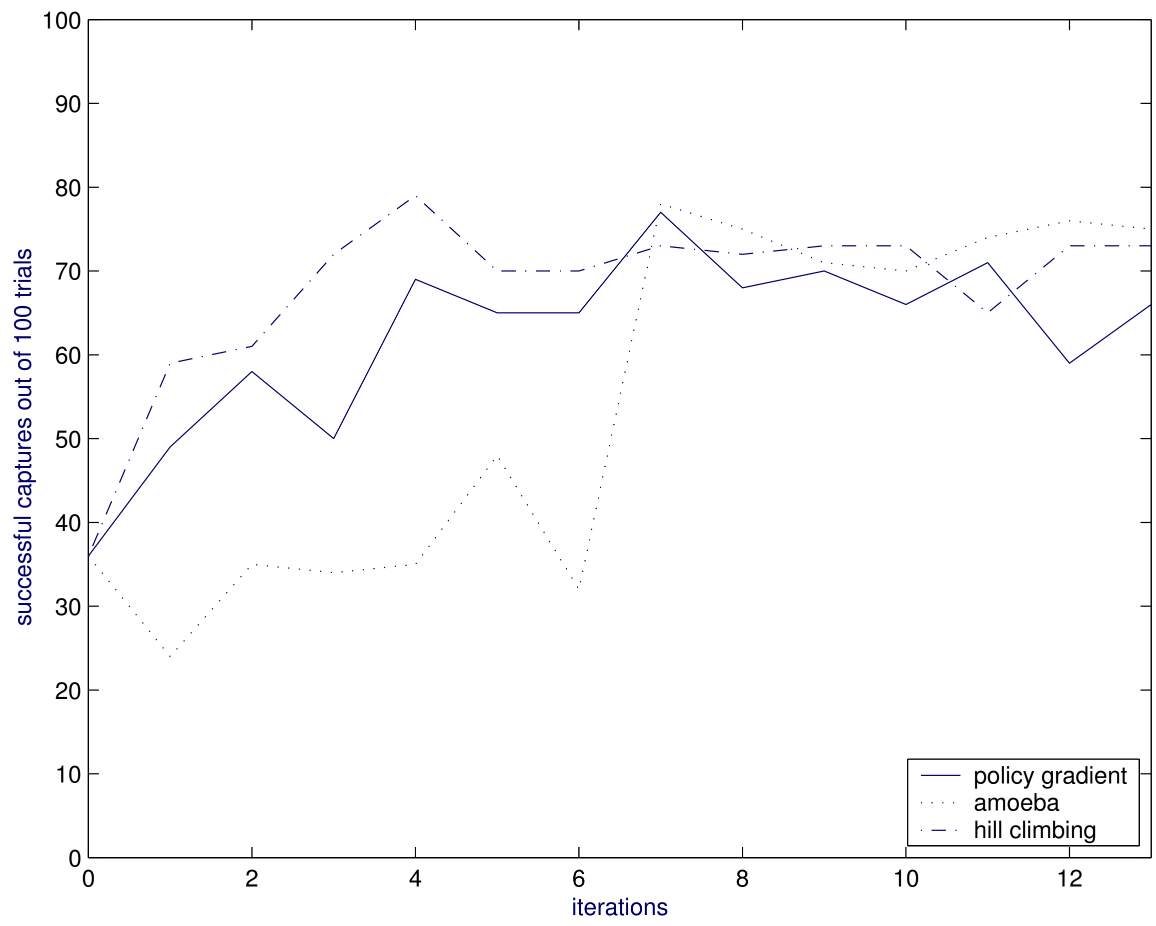
- **capture_dist:** when to put down head

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%

Instance of Layered Learning

- For domains too **complex** for tractably mapping state features $S \longmapsto$ outputs O
- Hierarchical subtask decomposition **given**: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: **exploit data** to train, adapt
- **Learning in one layer feeds into next layer**

Relaxing the Assumptions

- Nondeterministic actions:

Relaxing the Assumptions

- Nondeterministic actions: AND-OR search

Relaxing the Assumptions

- Nondeterministic actions: AND-OR search
- Partial observations:

Relaxing the Assumptions

- Nondeterministic actions: AND-OR search
- Partial observations: Belief states

Relaxing the Assumptions

- Nondeterministic actions: AND-OR search
- Partial observations: Belief states
- Unknown environments:

Relaxing the Assumptions

- Nondeterministic actions: AND-OR search
- Partial observations: Belief states
- Unknown environments: Online search

Relaxing the Assumptions

- Nondeterministic actions: AND-OR search
- Partial observations: Belief states
- Unknown environments: Online search
- Adversaries:

Relaxing the Assumptions

- Nondeterministic actions: AND-OR search
- Partial observations: Belief states
- Unknown environments: Online search
- Adversaries: Next week....