Machine Learning Research in the RoboCup Simulation League

Peter Stone
Department of Computer Sciences
The University of Texas at Austin

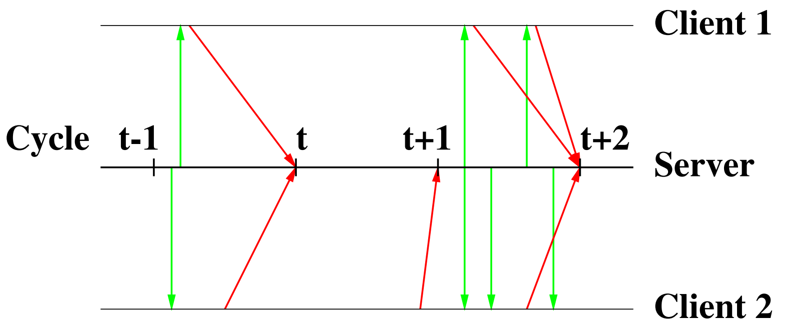
RoboCup — Selected Research challenges

An international AI and Robotics research initiative

- Robotics
  - Mechanics; Locomotion; Vision; Localization
- Multiagent systems
  - Multiple teammates with a common goal
  - Multiple adversaries — not known in advance
- Machine learning
  - Enormous state-space
  - Noisy sensors and actuators
  - Real-time decision making necessary

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)

Learning in the Simulation League

- **Past successes**

  Case studies: Noda, Humboldt, USC/ISI
  Competition: CMUnited, Andou, Karlsruhe
  Whole task: Luke, Darwin United

- **Our past contributions/results**
  - Flexible team member agent architecture
  - Hierarchical, layered learning method
  - TPOT-RL
  - Semi-finalist ’97–’00, champion ’98,’99 (with Veloso, Riley, McAllester)

- **This talk:**
  - A reinforcement learning case study
  - Year-old methods, **new results**

RoboCup and Reinforcement Learning

- Reinforcement Learning — suited to soccer
  - Sequential decision making
  - Achieving delayed goals
  - Handling noise and stochasticity
  - Rapid decision-making
- Challenges
  - Multiple learning agents
  - Large state space
  - Not within realm of theoretical results

Outline

- Motivation
- Keepaway — a subtask of soccer
- Reinforcement learning for keepaway
- Empirical results

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 of bounds
- Performance measure: average possession duration

Department of Computer Sciences
The University of Texas at Austin

Available Skills (from CMUnited-99)

HoldBall(): Remain stationary while keeping possession of the ball.

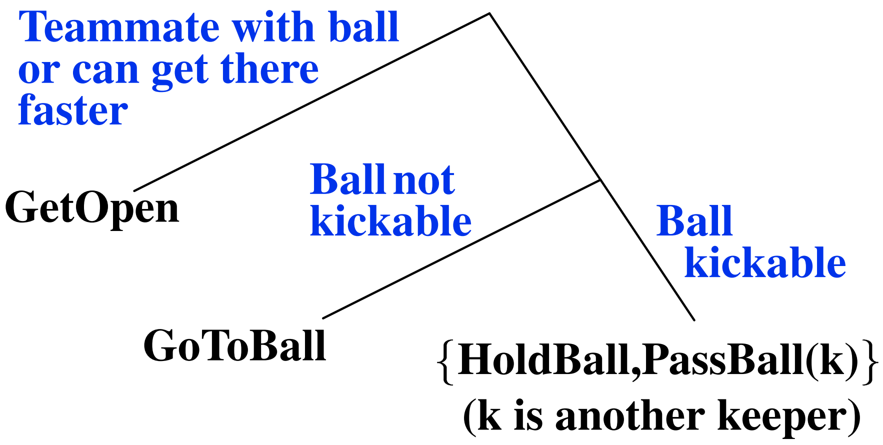
PassBall(k): Kick the ball directly to keeper k.

GoToBall(): Intercept a moving ball or move directly towards a stationary ball.

GetOpen(): Move to a position that is free from opponents and open for a pass from the ball's current position (using SPAR [Veloso et al., 1999])

BlockPass(k): Get in between the ball and keeper k

The Keepers' Policy Space



Example Policies

Random: HoldBall or PassBall(k) randomly
Hold: Always HoldBall
Hand-coded:

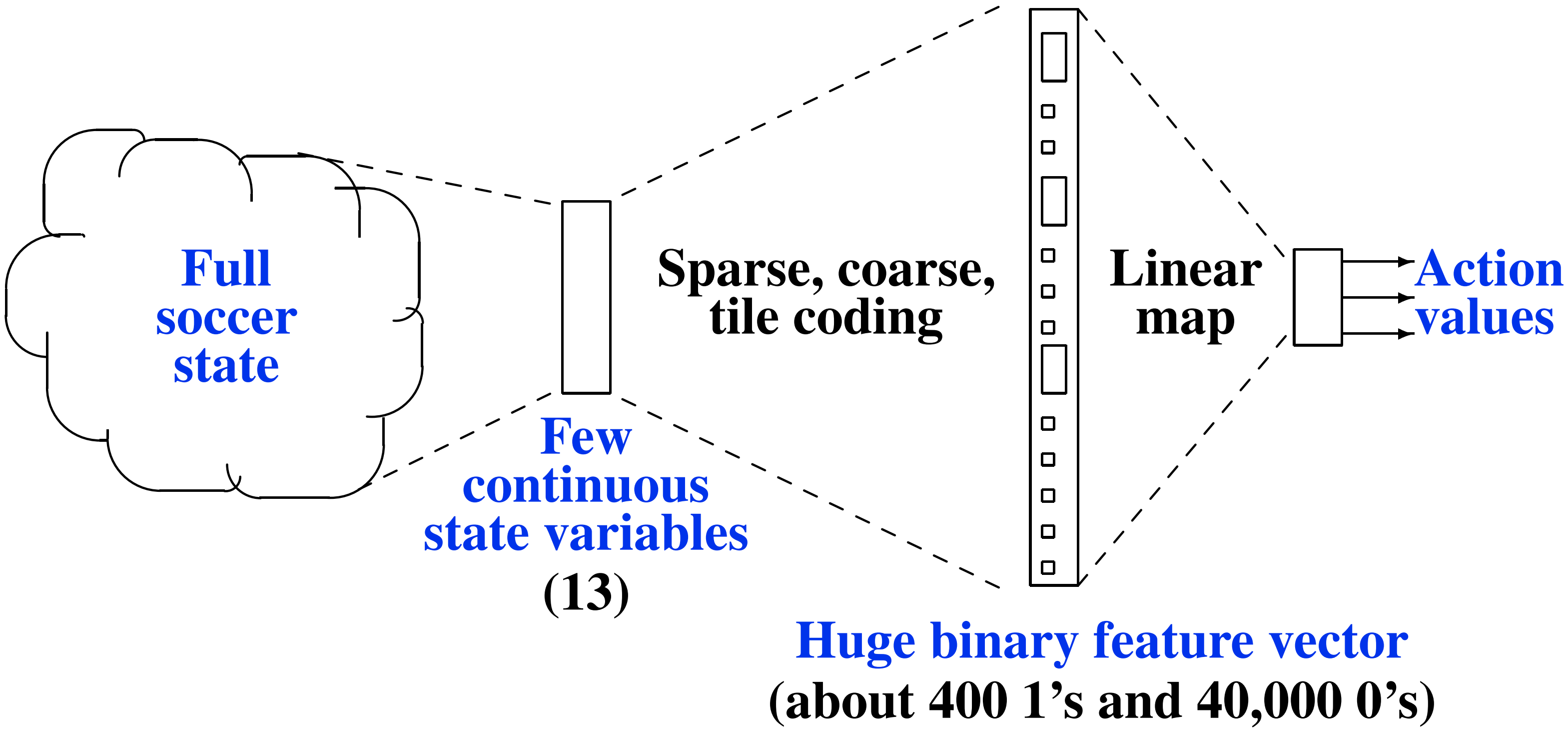
> **If** no taker within 10m: HoldBall
> **Else If** there's a good pass: PassBall(k)
> **Else** HoldBall

Mapping Keepaway to RL

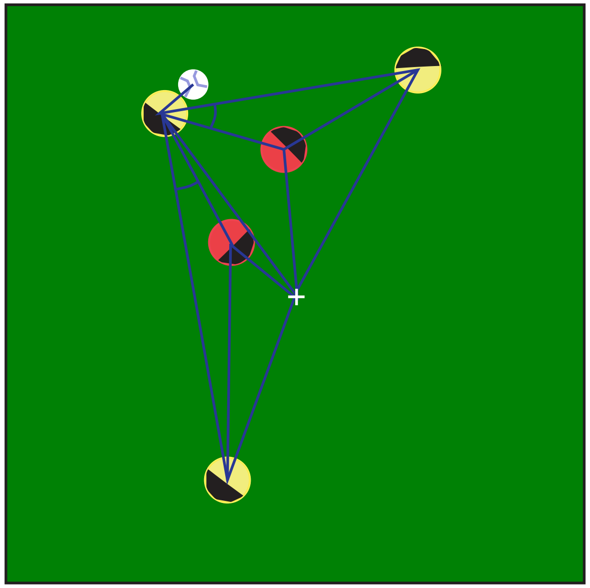
Discrete-time, episodic, distributed RL

- Simulator operates in discrete time steps, $t = 0, 1, 2, \ldots$, each representing 100 msec
- Episode:
  $s_0, a_0, r_1, s_1, \ldots, s_t, a_t, r_{t+1}, s_{t+1}, \ldots, r_T, s_T$
- $a_t \in$ {HoldBall, PassBall(k), GoToBall, GetOpen}
  (Keepers)
- $r_t = 1$
- $V^\pi(s) = E\{T \mid s_0 = s\}$
- Goal: Find π^* that maximizes V for all s

Representation



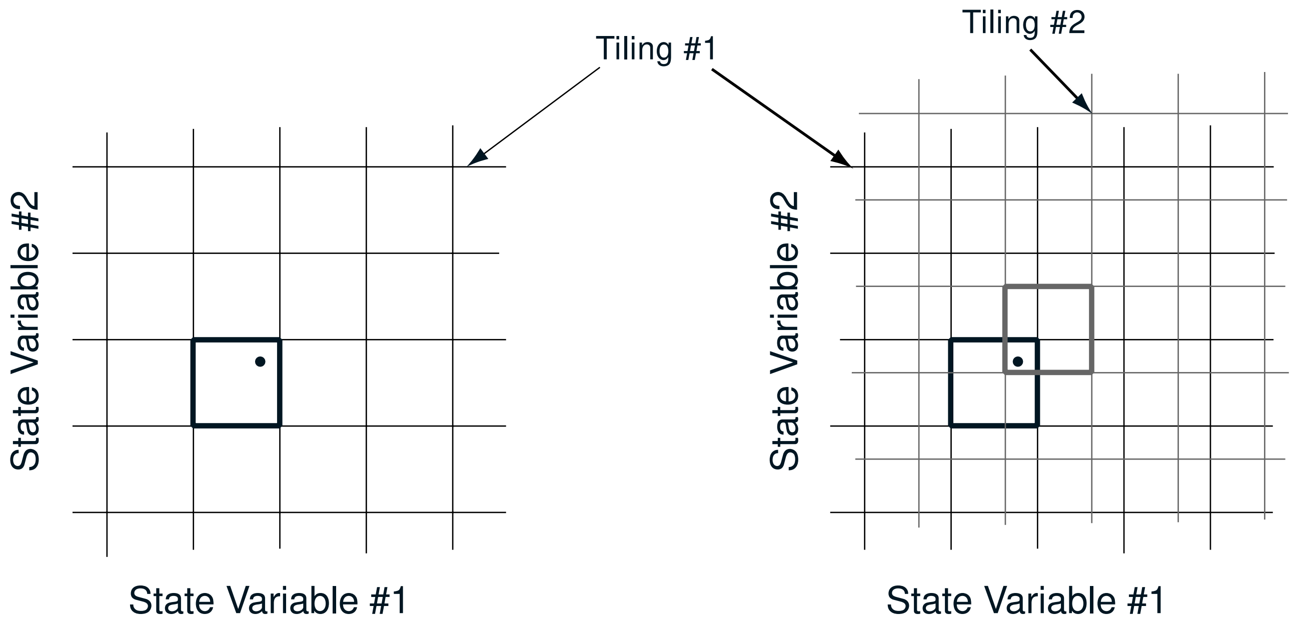
s: 13 Continuous State Variables



- 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]



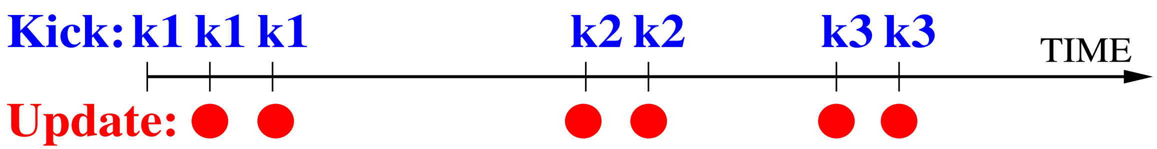
- Tiled state variables individually (13)

Policy Learning

- Learn $Q^\pi(s, a)$: Expected possession time

- Linear Sarsa(λ) — each agent learns independently
  - On-policy method: advantages over e.g. Q-learning
  - Not known to converge, but works (e.g. [Sutton, 1996])

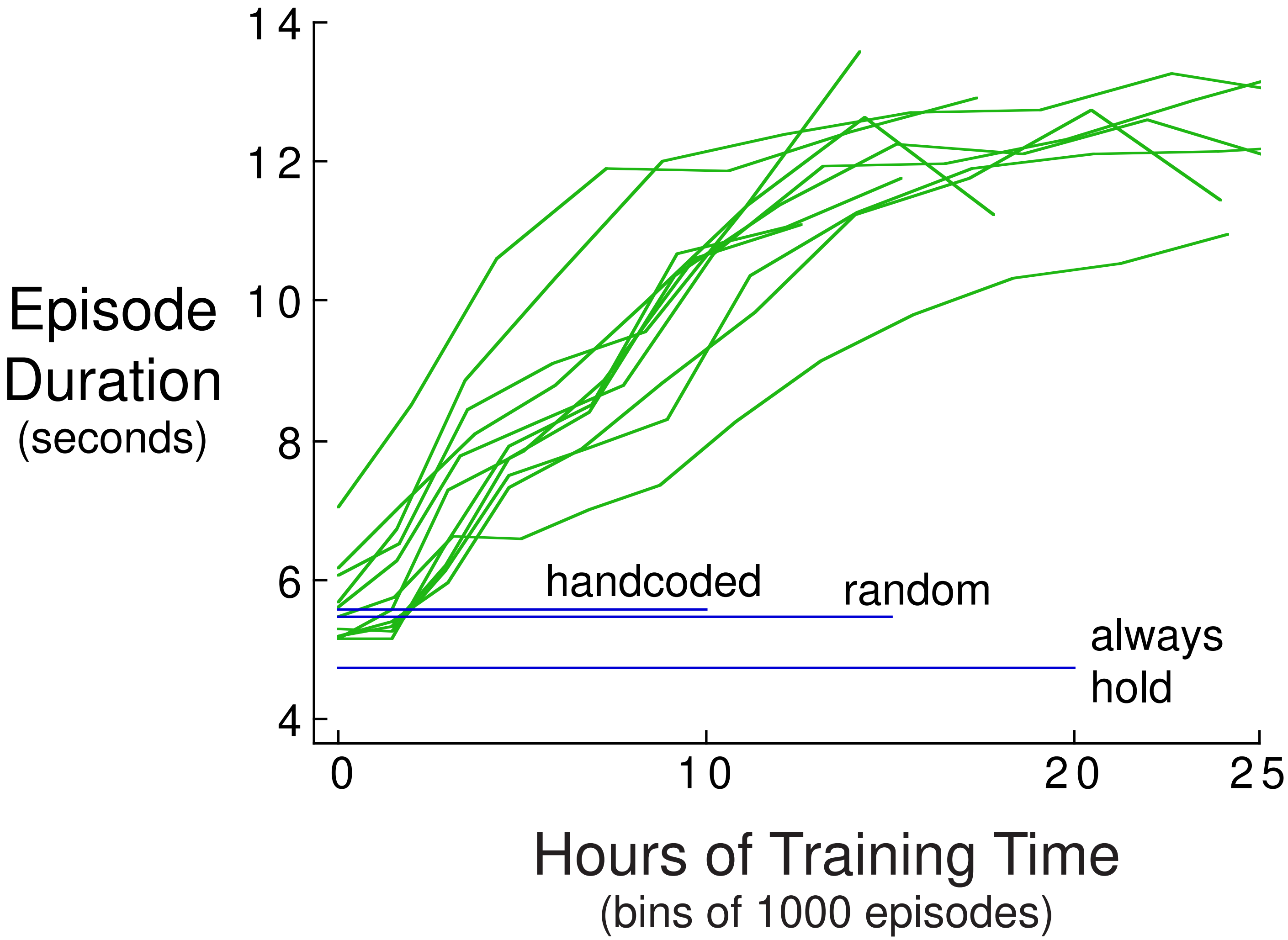
- Only update when ball is kickable for **someone**:
  Semi-Markov Decision Process



Experiments

- Initial:
  - Learning rate: $\alpha = 1/8$
  - Undiscounted vs. discounted reward
  - ϵ-greedy vs. optimistic initialization
  - $\lambda = 0$ (most of the time)
- Keepers learn in 3 vs. 2
  - Vary field size
- Keepers learn in 4 vs. 3
- Takers learn in 4 vs. 3 (preliminary)
- **360^o view angle**
- **No perceptual noise**

Main Result



1 hour = 720 5-second episodes

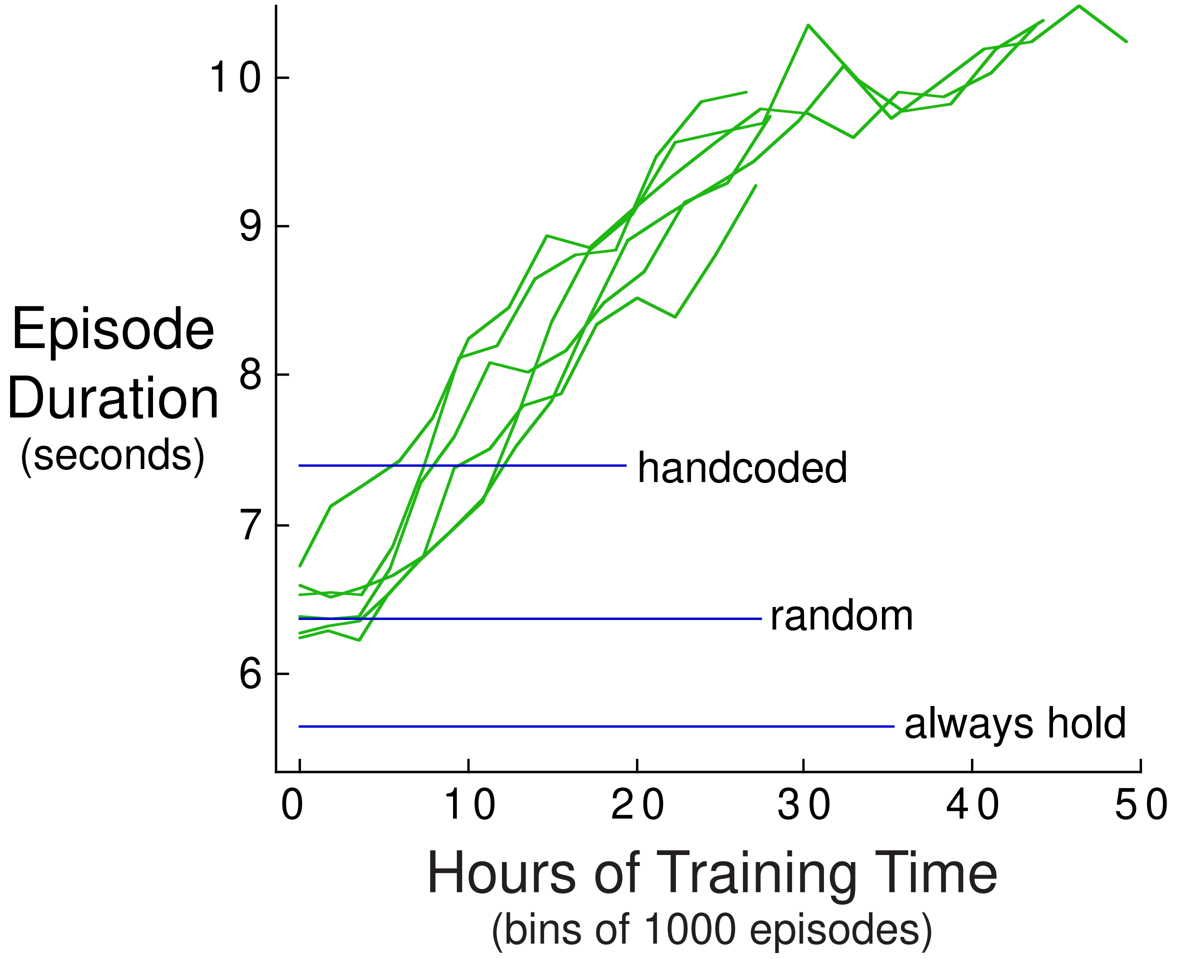
Department of Computer Sciences
The University of Texas at Austin

Varied Field Size

Keepers		Testing Field Size: 15x15	20x20	25x25
Trained on field of size	15x15	**11.0**	9.8	7.2
	20x20	10.7	**15.0**	12.2
	25x25	6.3	10.4	**15.0**
Benchmarks	Hand	4.3	5.6	8.0
	Hold	3.9	4.8	5.2
	Random	4.2	5.5	6.4

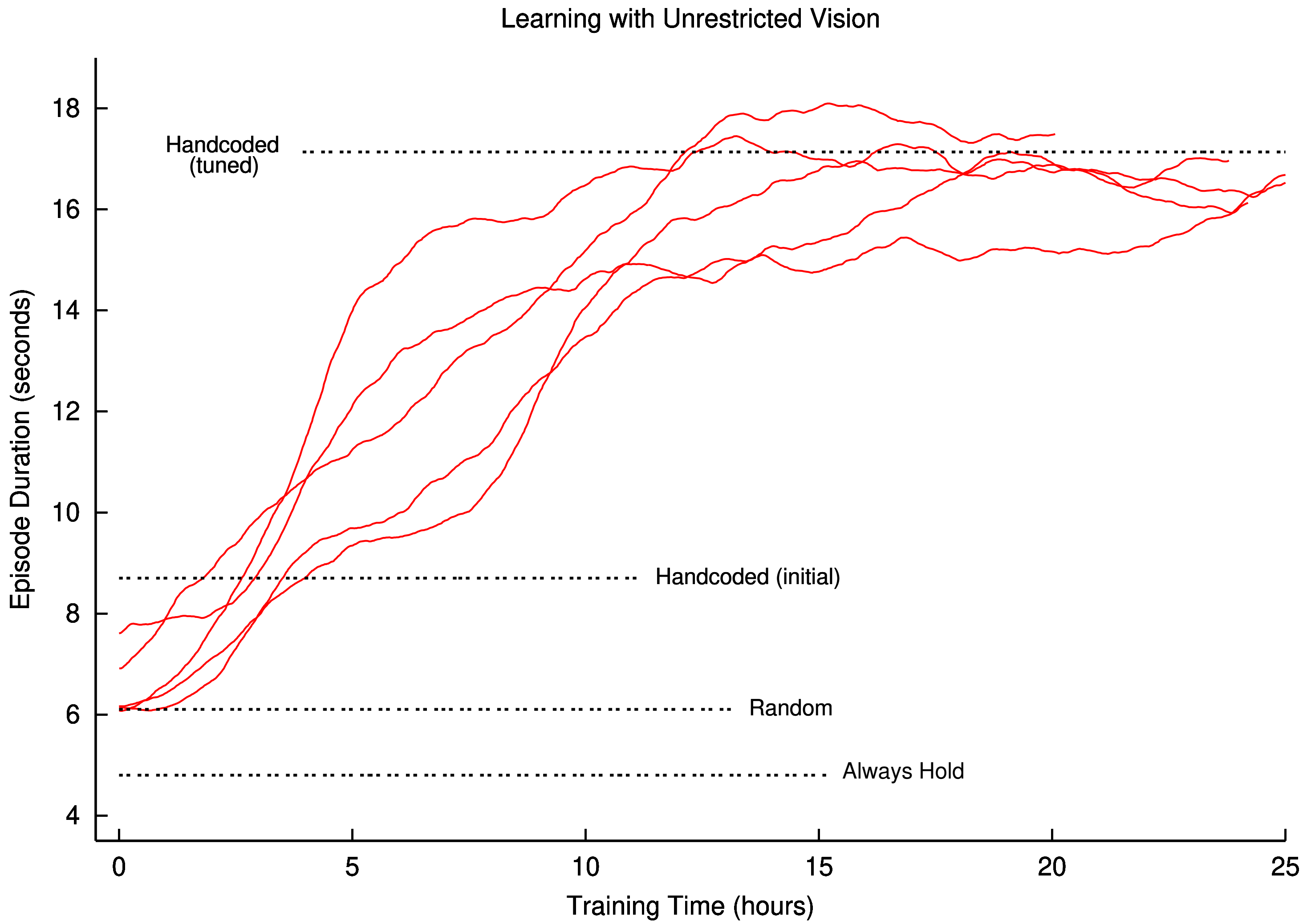
- Single runs
- learning specific to fields
  - mechanism generalizes better than policies

4 vs. 3 Keeper Learning



- Preliminary: taker learning successful as well

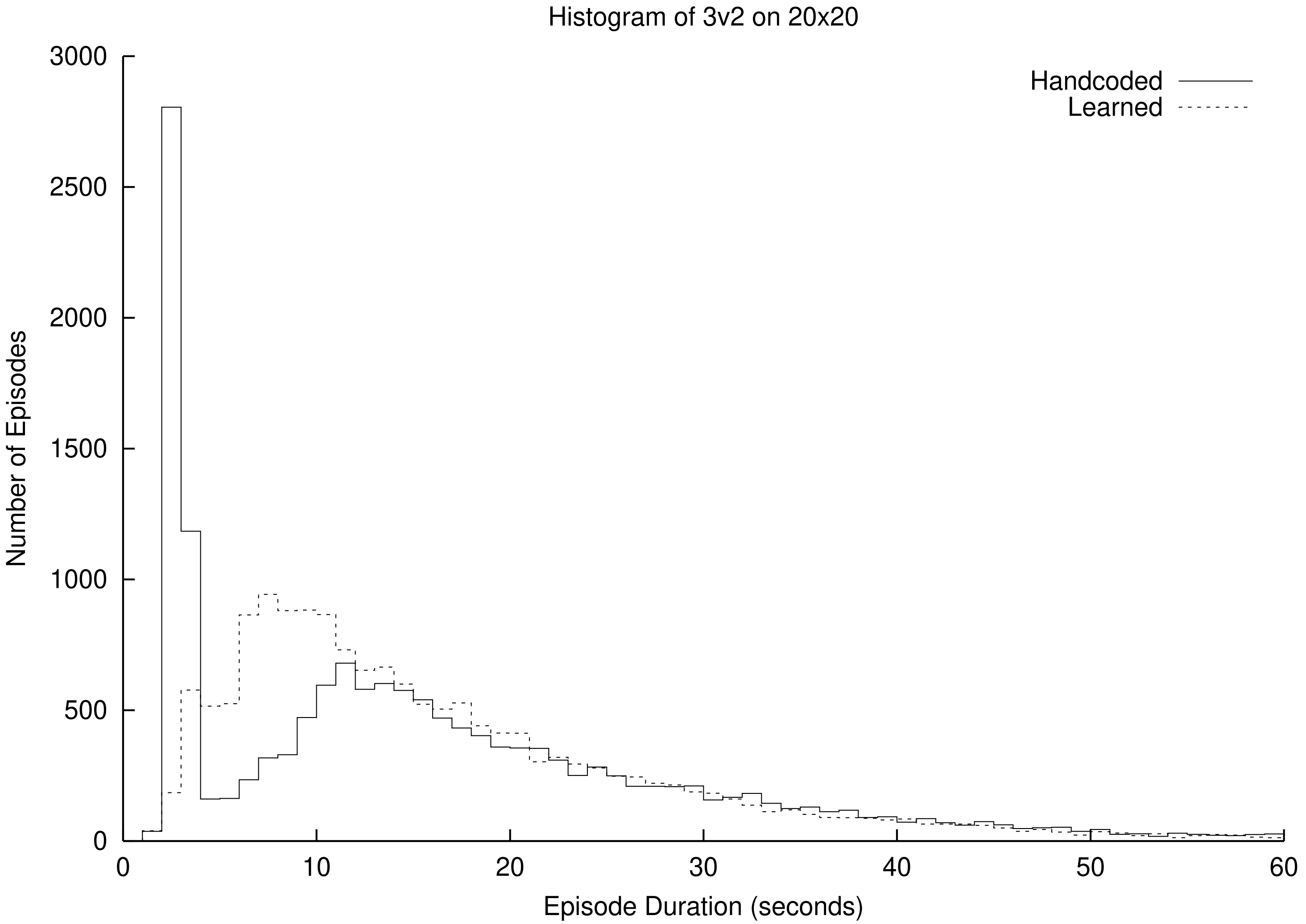
New Results — With Noise



Hand-Coded Algorithm

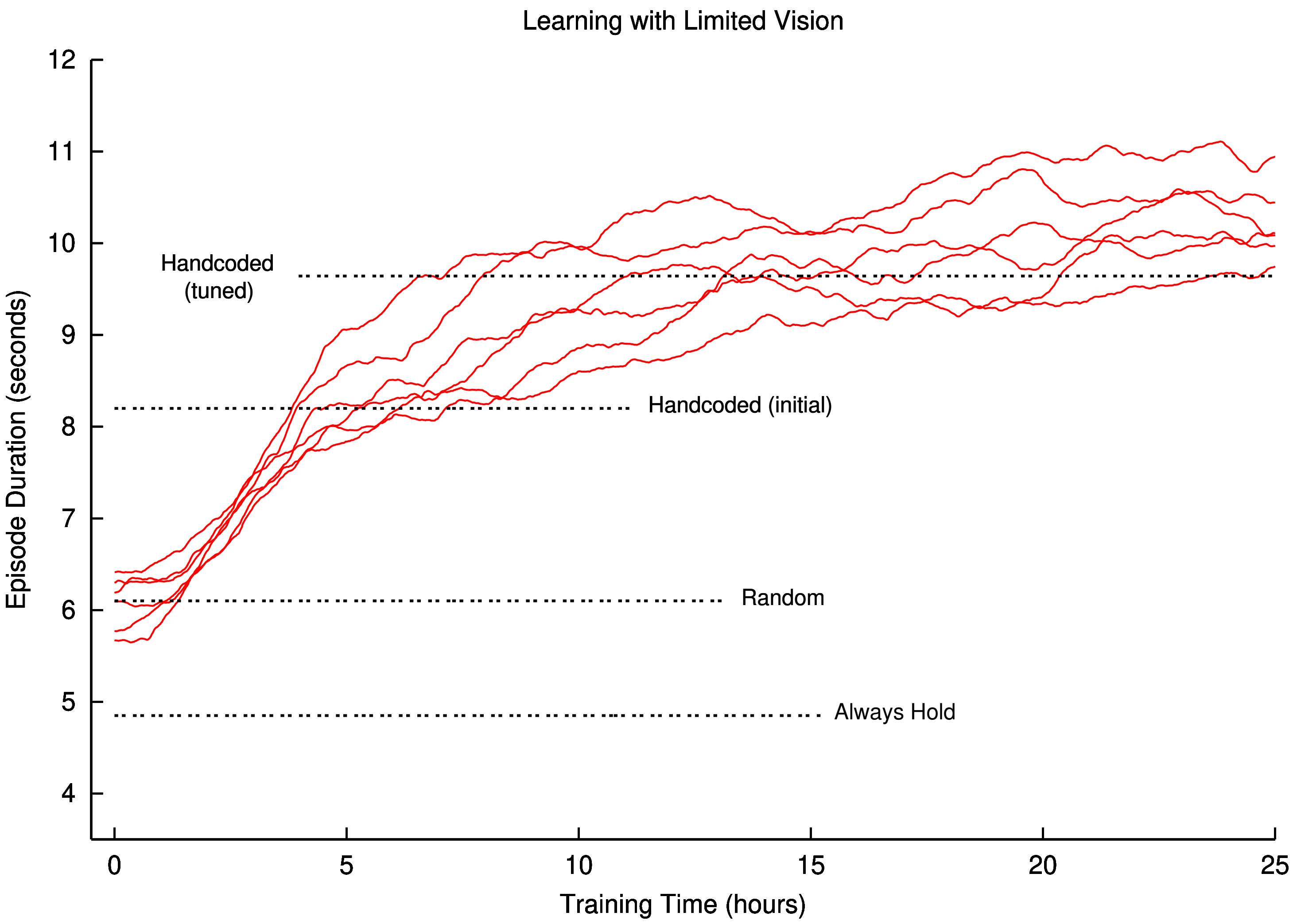
- **If** no opponents within 4 **then** HOLD
- **Else**
  - For each teammate i:
    - score(i) = min_ang(i, Me, T) + 4*min_dist(i, T)
    - (T ranges over takers)
  - best = argmax(score(i))
  - **If** (score(best) $>$ 90) **then** PASS to best
  - **Else** HOLD

Episode Lengths



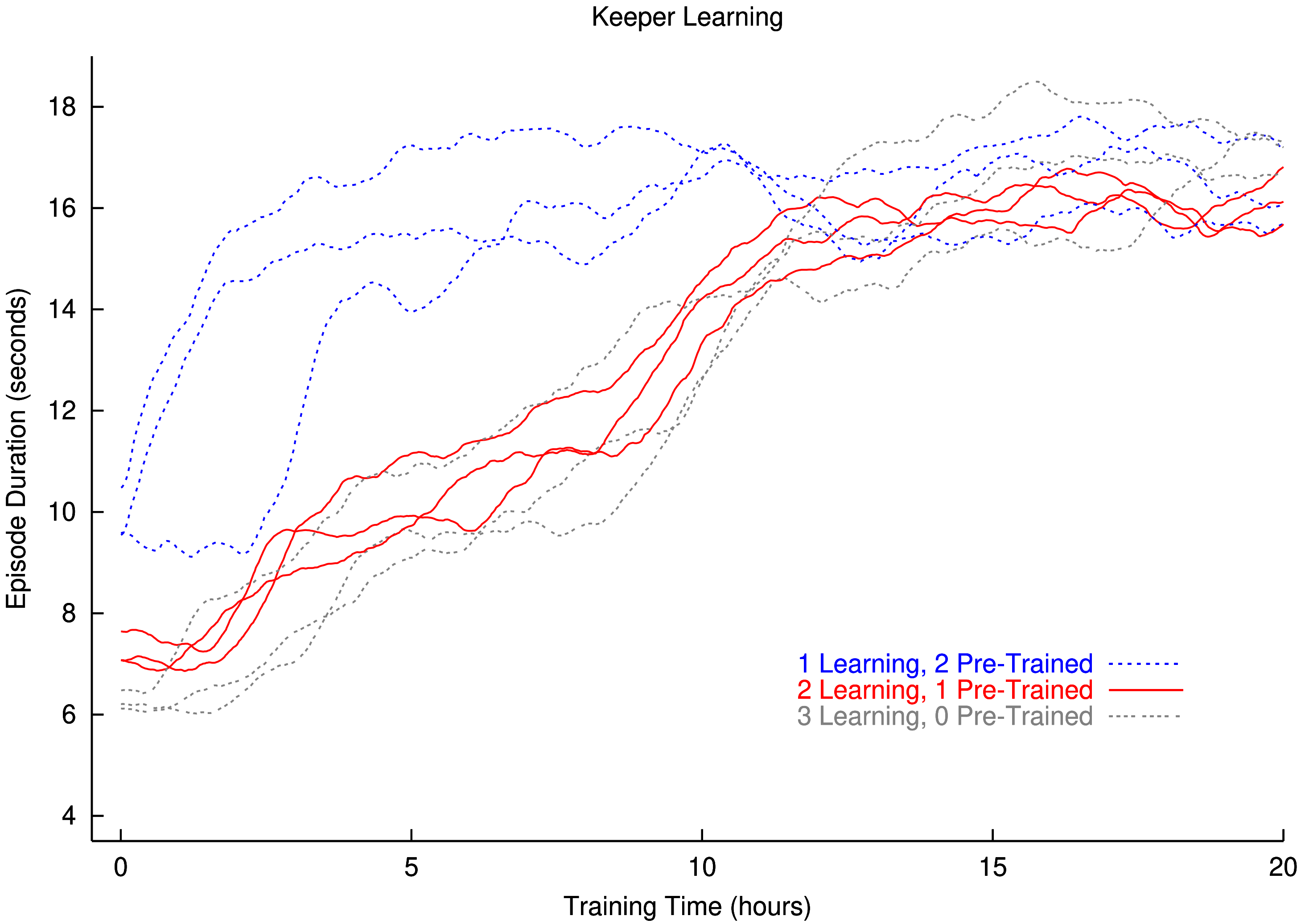
- Same mean: 17.1s
- Different medians: 21.5s (learned) vs. 6.6s (hand)

Limited Vision



- With noise, limited (90^o) vision

Mixing Learned, Hand-Coded Agents



- Multiagent learning is harder!

Summary

$\boxed{\text{Successful keeper policies for keepaway}}$

- Learned via Sarsa(0)
- Beat initial hand-coded policies
- Comparable to tuned hand-coded policies
- Works with 90^o vision and full sensor noise

Future Work

- Do hand-coded policies generalize as well as learning?
- Fully explore taker learning
- Expand task: $>$ players; $>$ actions; $>$ area
- Incorporate into full robotic soccer task

Acknowledgements

Joint work with

Richard S. Sutton and Gregory Kuhlmann