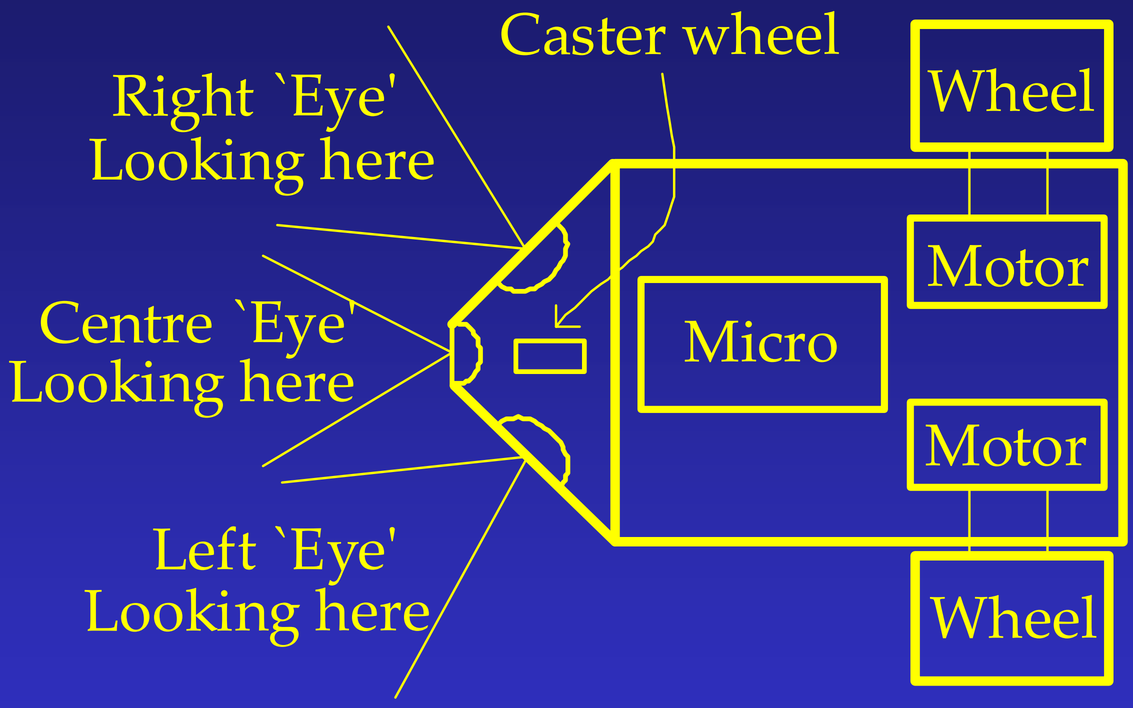
Dynamic Automata For Mobile Robot Learning

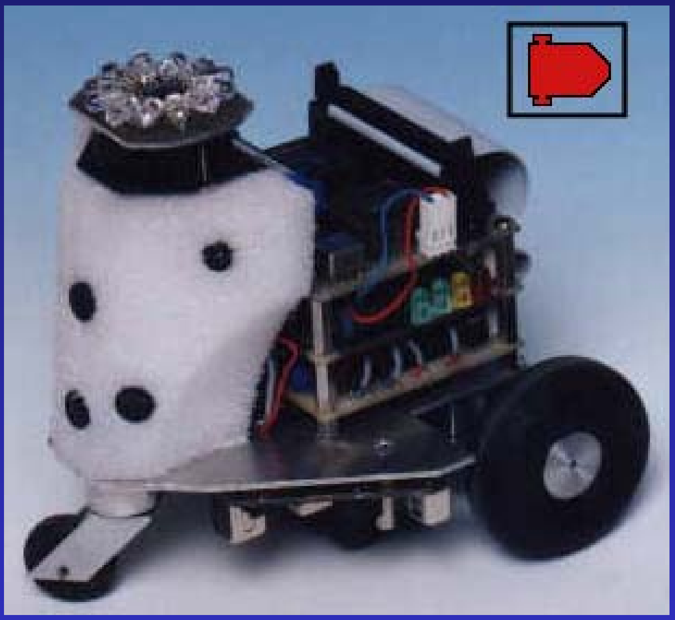
*Isaac Ashdown and **Richard.Mitchell***
Department of Cybernetics, The University of Reading
r.j.mitchell@rdg.ac.uk

Cybernetics' mobile robots learn to explore dynamic environments, perceived via ultrasonic sensors, avoiding obstacles, using a static set of fuzzy automata. To address criticisms of this arbitrary static set , this paper considers the use of a dynamic set of automata together with a new reinforcement learning function which is both scalable to different numbers and types of sensors. The innovations compare successfully with earlier work.

© Dr Richard Mitchell 2005

Cybernetics Robot





Ashdown’s simulator allows robots with more ultrasonic sensors, as well as others such as ‘bump’ sensors.
Single/multiple robots and environments also allowed

© Dr Richard Mitchell 2005

Basic Learning Stategy

9 possible actions (each motor Forward, Off, Backward)

FF FO FB OF OO OB BF BO BB

Associated with each action is a probability

These are grouped as a ***fuzzy automaton***

Robot chooses action (based on weighted roulette wheel)

Robot tries action out

Action evaluated – get goodness factor α

Common sense rules applied to get α:

these do not tell the robot directly how to behave

If $\alpha > 0$, increase probability of action else decrease it.

If probability increased, more likely its action is chosen

© Dr Richard Mitchell 2005

Multiple Automata

One action best when no obstacle; another when one near.

So, have many automata: 5 were chosen, selected by range:

- DD - object distant for both eyes
- ?F - nearest object far from right eye
- F? - nearest object far from left eye
- ?C - nearest object close to right eye
- C? - nearest object close to left eye

(D)istant > (F)ar > (C)lose

? = distance from eye unknown, but object closer to other eye

Overall performance measure: ‘fitness’ based on comparing probabilities of each automata with ‘best’ possible, as determined by Kelly, one of our researchers in 1990s

© Dr Richard Mitchell 2005

Algorithm – For All Time

Determine Current Automaton (on basis of range)

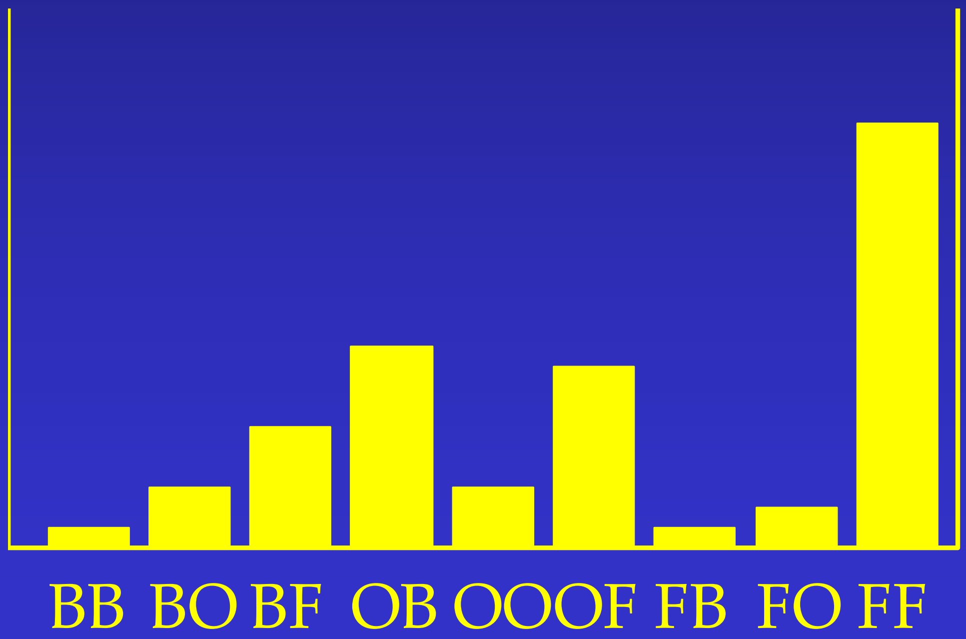
Choose Action (highest prob action most likely)

Evaluate It (find the α based on all sensors)

Adjust Probabilities for the Automata

Graphs used to show automaton

Bar height shows probability of associated action

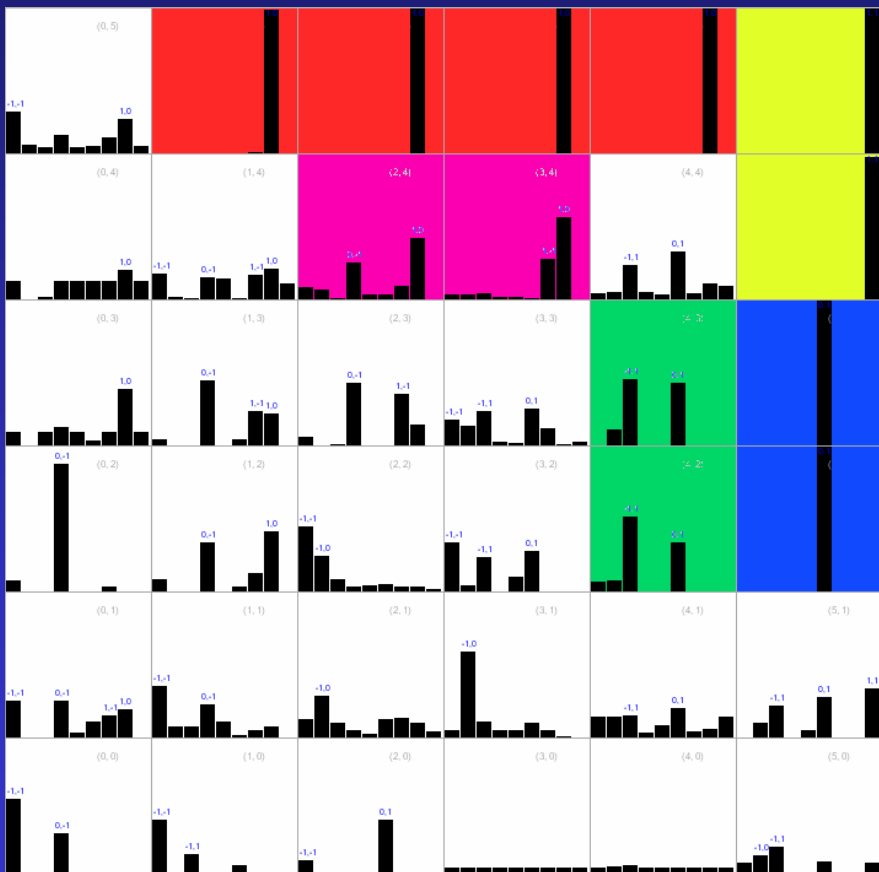


© Dr Richard Mitchell 2005

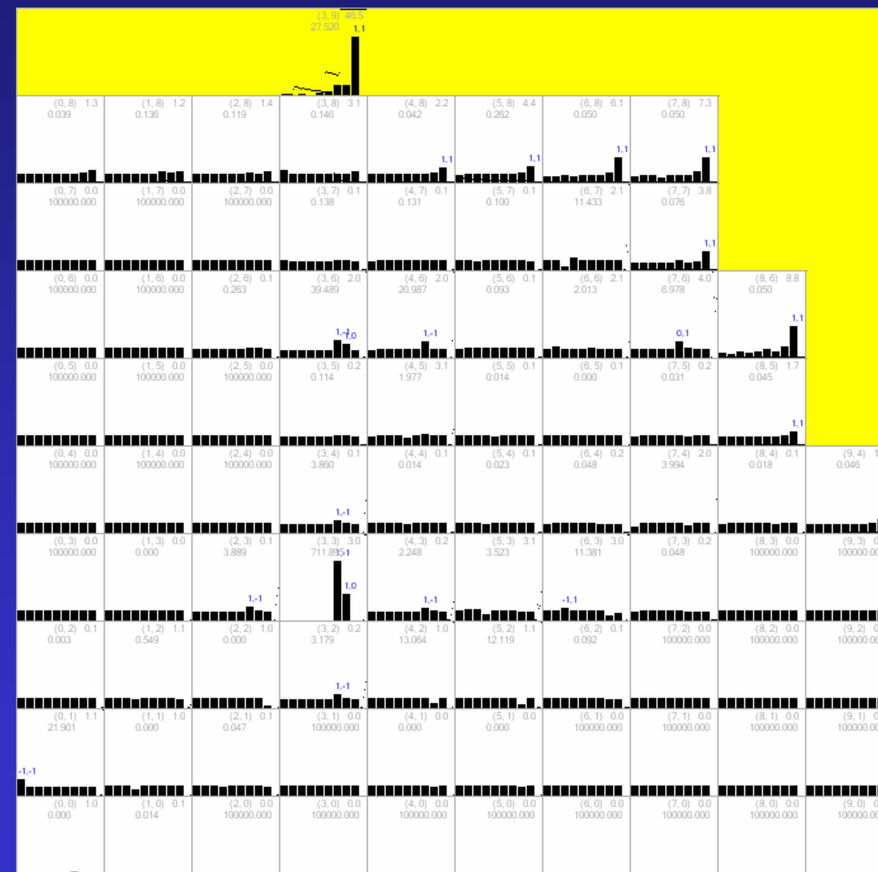
Cybernetics

Problems With Static Automata

Inefficient: should merge



Over generalise: should split



© Dr Richard Mitchell 2005

Aims of Current Project

Investigate automatic method of determining automata

- have many automata : allow to be merged and / or split
- to speed learning, adjust neighbours similarly
  - (influenced by Kohonen network concept)

Have more generic method for generating goodness α

- so scalable to multiple automata/sensors
- aim: robot can learn at least as fast as Kelly's method

Have more generic fitness function

- current one based on optimum when have five automata
- instead estimate how much of the environment explored

© Dr Richard Mitchell 2005

Automata Operation (For appropriate x)

Condition to Merge Two Automata

- Automata run at least x times
- Same top x probabilities correspond to same top x actions
- Sum Square Differences of two automata $< x$
- Instability measure of two automata $< x$

Condition to Split an Automaton

- Instability reached threshold x

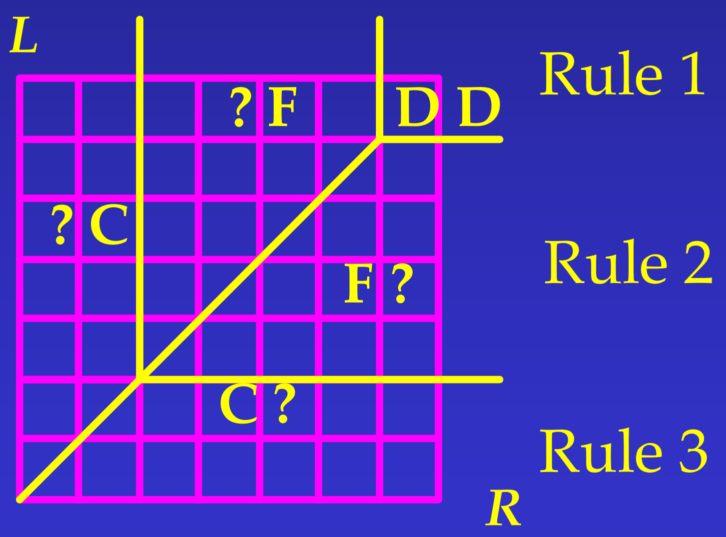
Condition for Automaton to influence a Neighbour

- Influencer x times more stable than neighbour
- Influencer at least x times more defined than neigh
- Influencer run at least x times

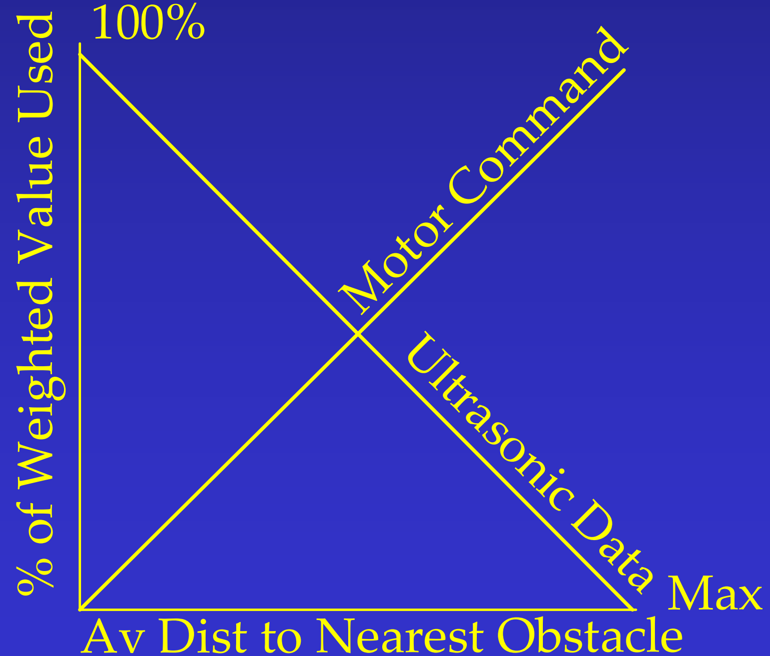
© Dr Richard Mitchell 2005

Setting Goodness Function α

Kelly's 3 separate rules:
1) No object: forward good
2) Mid: use both 1 and 3
3) Close Object: away good



Generic - less arbitrary:
Scalable to different sensors
Pre-set weight * sensor change
Balances 'move forward' and 'avoid obstacle' behaviours:

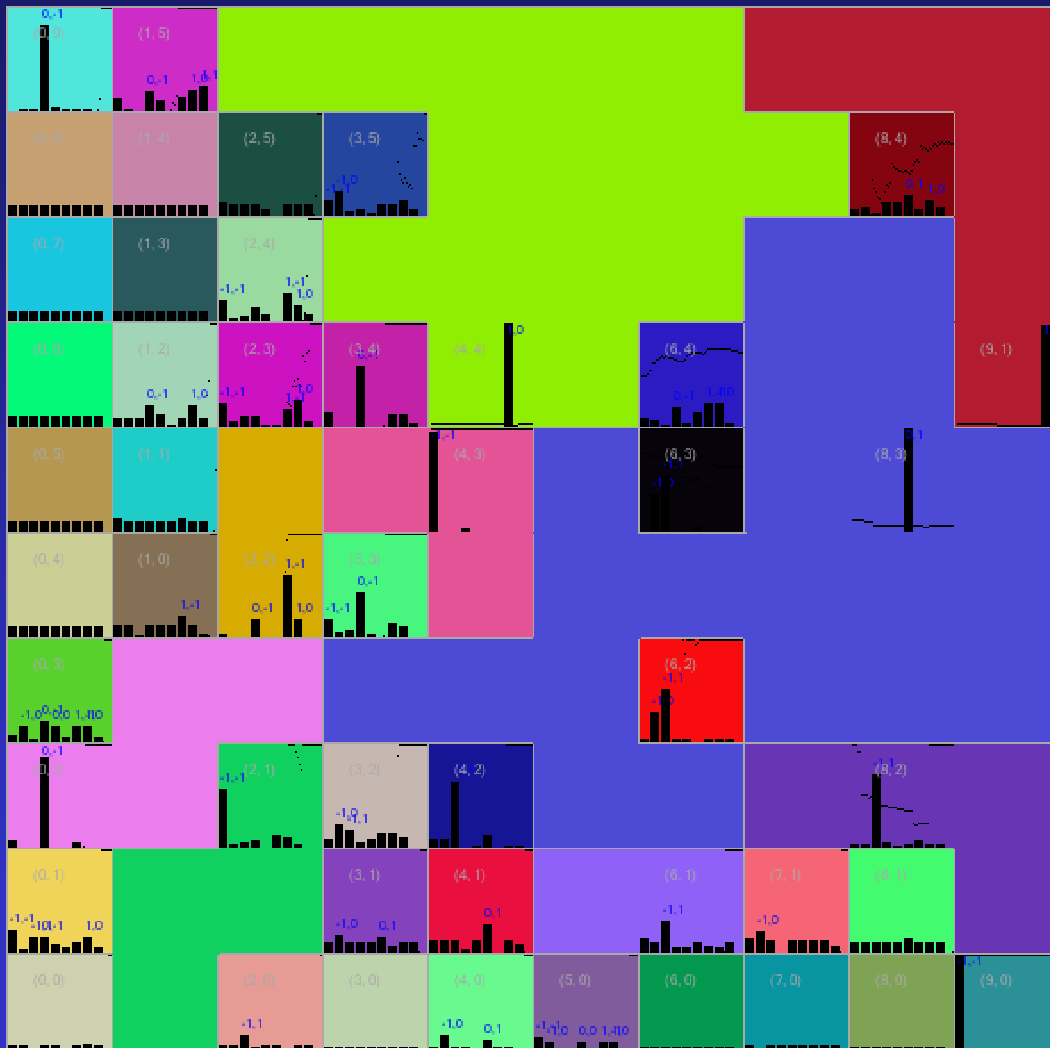


© Dr Richard Mitchell 2005

Automata After Example Run

L

DD

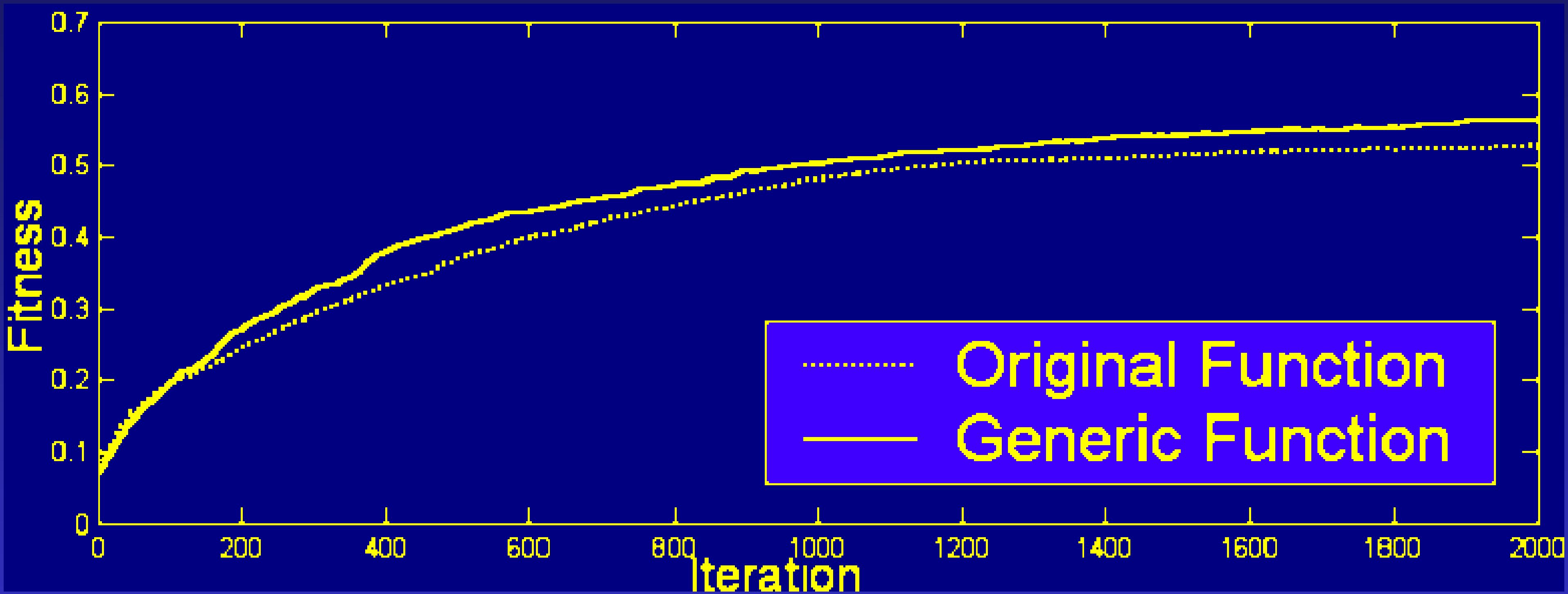


R

Comment – if avoided well, robot has not encountered the 'close' situations as often, hence less merging there

© Dr Richard Mitchell 2005

Learning Function Comparison



New method slightly better, fortunately not worse!

© Dr Richard Mitchell 2005

Conclusions

New learning function scalable and improvement over old

Dynamic automata successfully self-organised and drove behaviour, but

No evidence dynamic automata more efficient than static

Other Work

Confirmed Punishment and Reward better than either

Shared Experience Learning improves speed

Future Work

Apply to real robots (or mix of real and simulated)

Thanks to *Isaac Ashdown* for his hard work on project

© Dr Richard Mitchell 2005