

# Distributed Learning Automaton

*Aleksander M. Stensby  
and  
Ole-Alexander Moy*

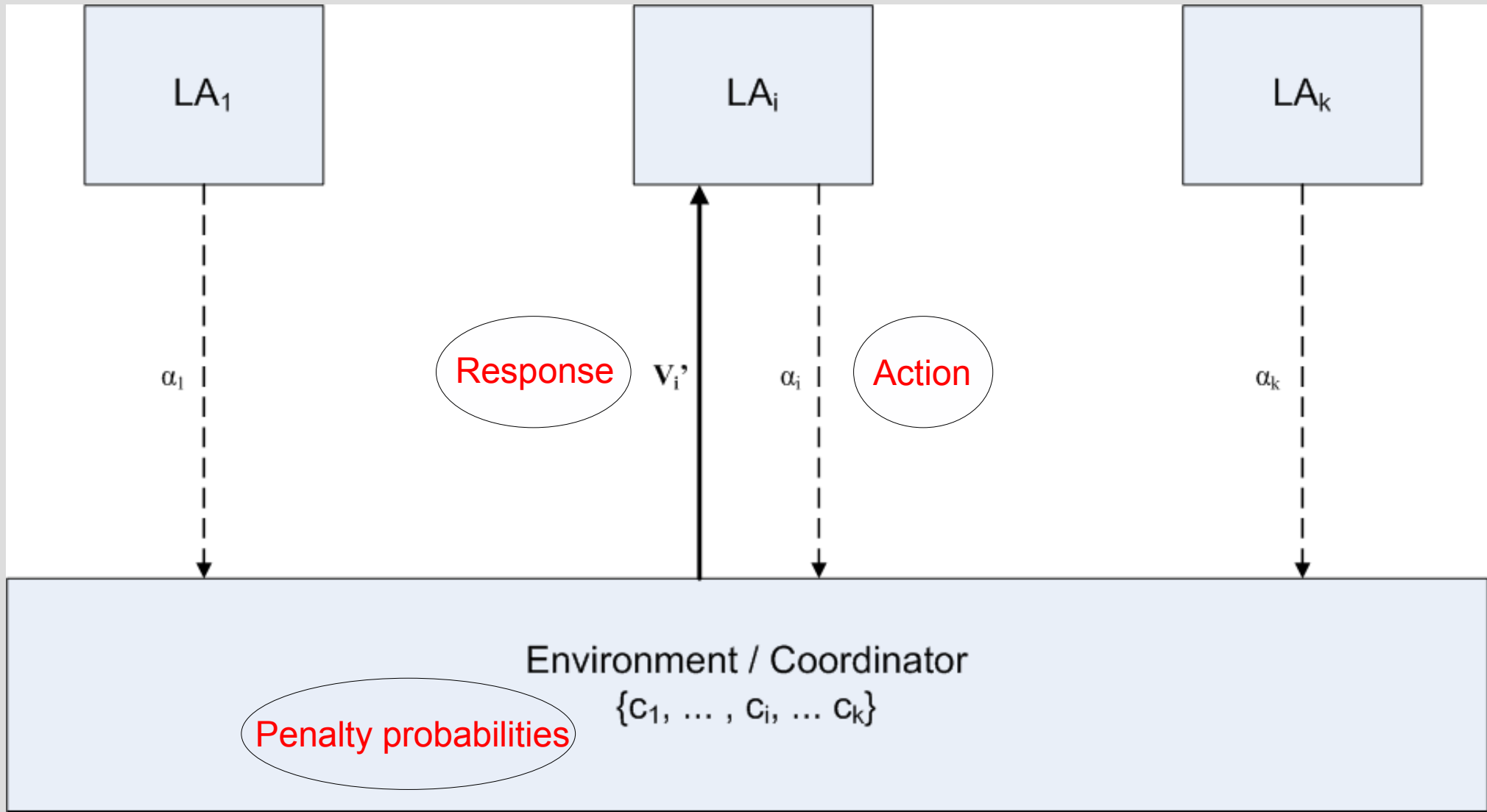
# Distributed Learning Automaton

- Novel and previously unpublished scheme
- Thesis:
  - **«The DLA presented forms an effective scheme for distributed learning»**
- Claims:
  - *«The DLA allows the best action to be found in a decentralized manner.»*
  - *«The DLA scales better than the  $L_{R-I}$  scheme with respect to the number of actions available to the automaton.»*

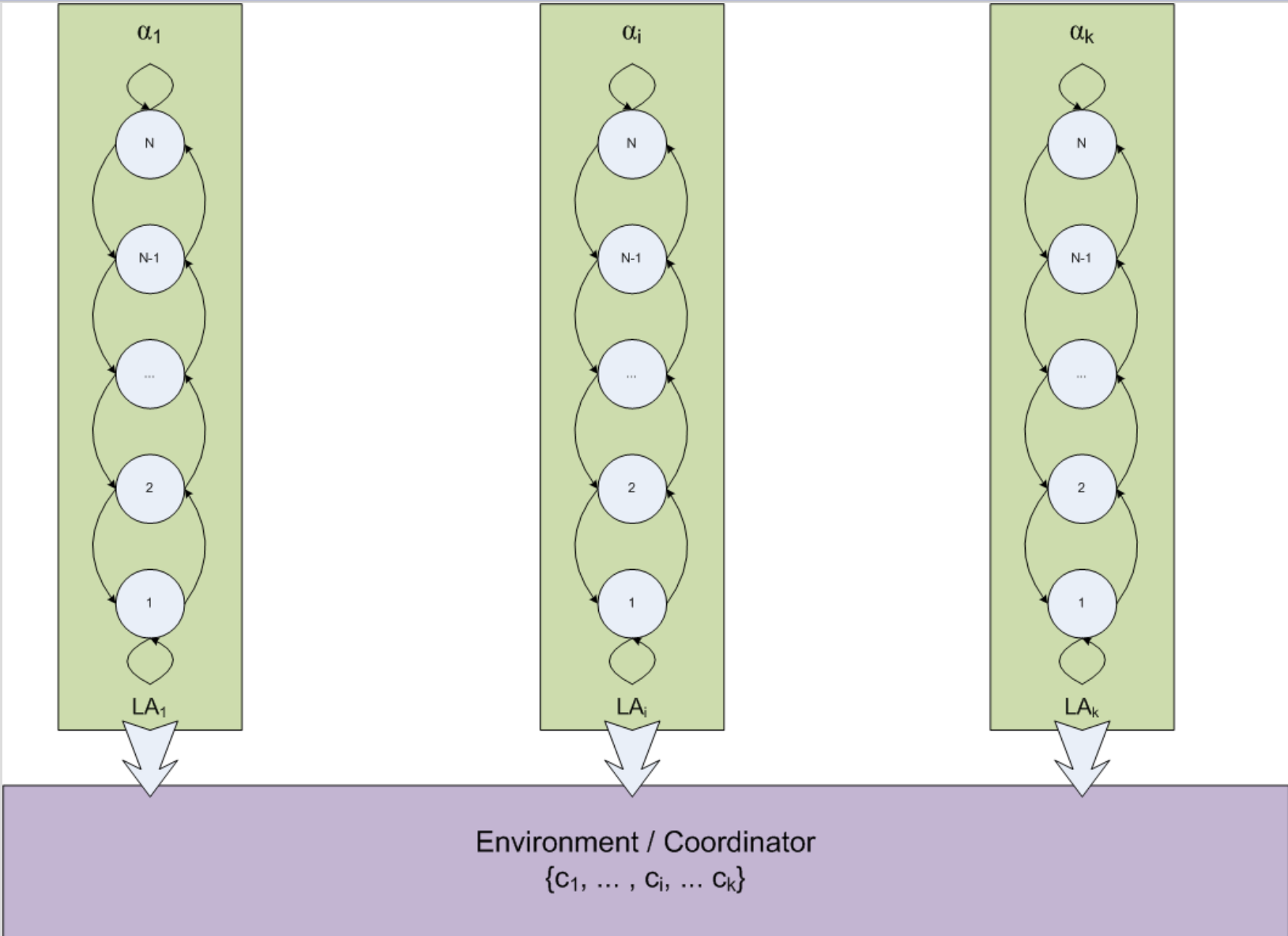
# Why distribution?

- Operation in complex environments
- Scalability
- Single LA: Action space grow exponentially
  
- Geographical location
- Decentralized decision making  
(*Wireless sensor networks*)

# The DLA



# The DLA



# The DLA

- Automata does not know about each other
- Communicate with Environment
  - Message passing (Sockets)
  - First Come, First Served
- Environment:
  - **Never same response twice in a row!**

# Updating schemes

- DLA

$$\begin{aligned} s_i(t+1) &:= s_i(t) + 1, && \text{if } v'_i(t) = 0 \text{ and } 1 \leq s_i(t) < N \\ &:= s_i(t) - 1, && \text{if } v'_i(t) = 1 \text{ and } 1 < s_i(t) \leq N \\ &:= s_i(t), && \text{otherwise} \end{aligned}$$

- $\mathcal{L}_{R-I}$

Probability of action  $\frac{s_i(t)}{N}$

$$p_i(n+1) = \begin{cases} p_i(n) + a(1 - p_i(n)) & \alpha(n) = \alpha_i, \beta(n) = 0 \\ p_i(n) & \alpha(n) = \alpha_i, \beta(n) = 1 \end{cases} \text{ for all } j \neq i.$$

$$p_j(n+1) = \begin{cases} (1 - a)p_j(n) & \alpha(n) = \alpha_i, \beta(n) = 0 \\ p_j(n) & \alpha(n) = \alpha_i, \beta(n) = 1 \end{cases} \text{ for all } j \neq i.$$

# Testing

Estimated Reward Probability

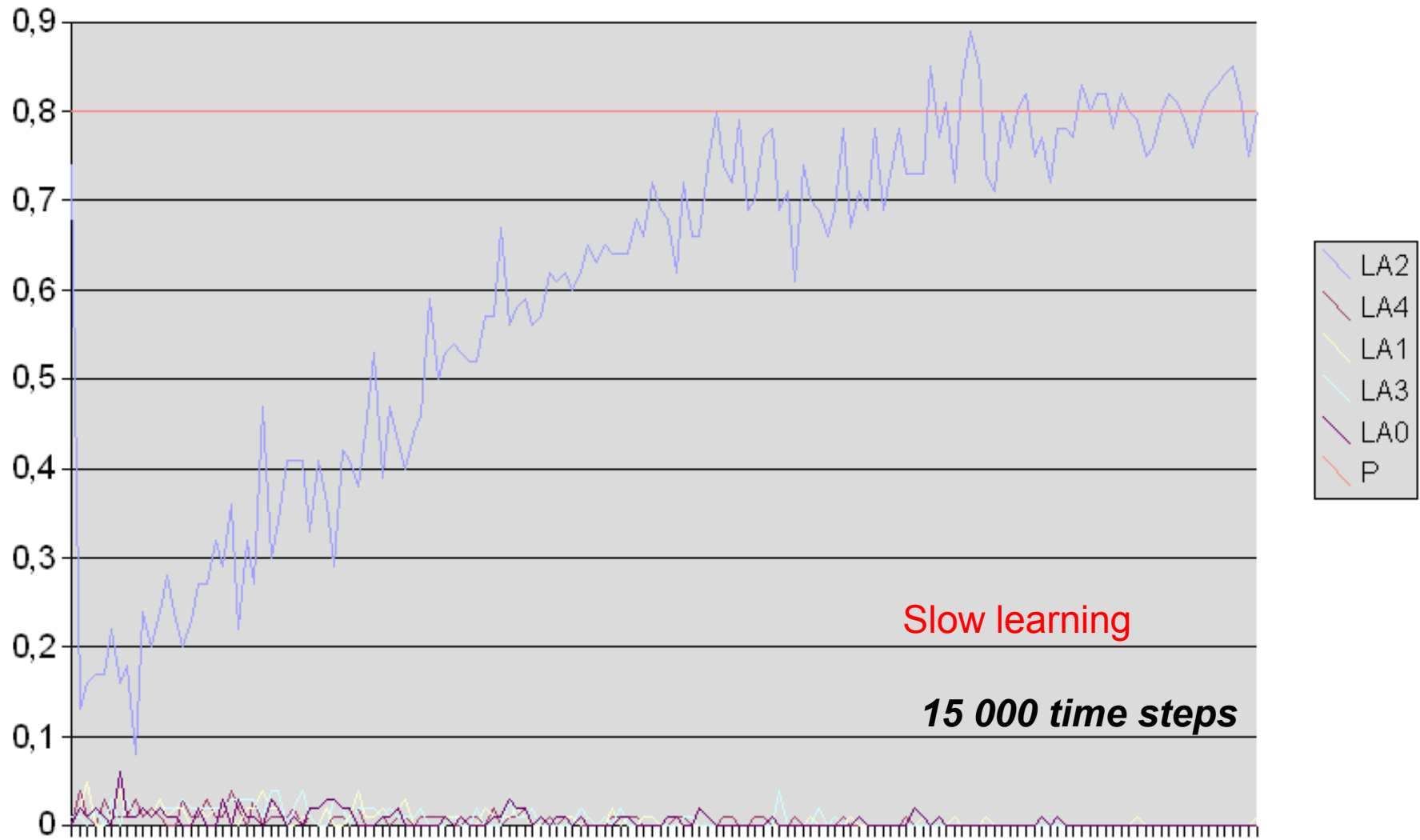
$$E[Q(n)] = \left( \frac{\text{Rewards}[n]}{Z} \right)$$

- DLA Parameters:
  - Number of states (N)
  - Number of actions ( $k$ )
  - Number of time steps (T)
  - Number of tests (Z)
  - Penalty probabilities
- $L_{R-I}$  Parameters:
  - Learning parameter ( $a$ )
  - Number of actions ( $k$ )
  - Number of time steps (T)
  - Number of tests (Z)
  - Penalty probabilities

# DLA 5 Actions

High accuracy

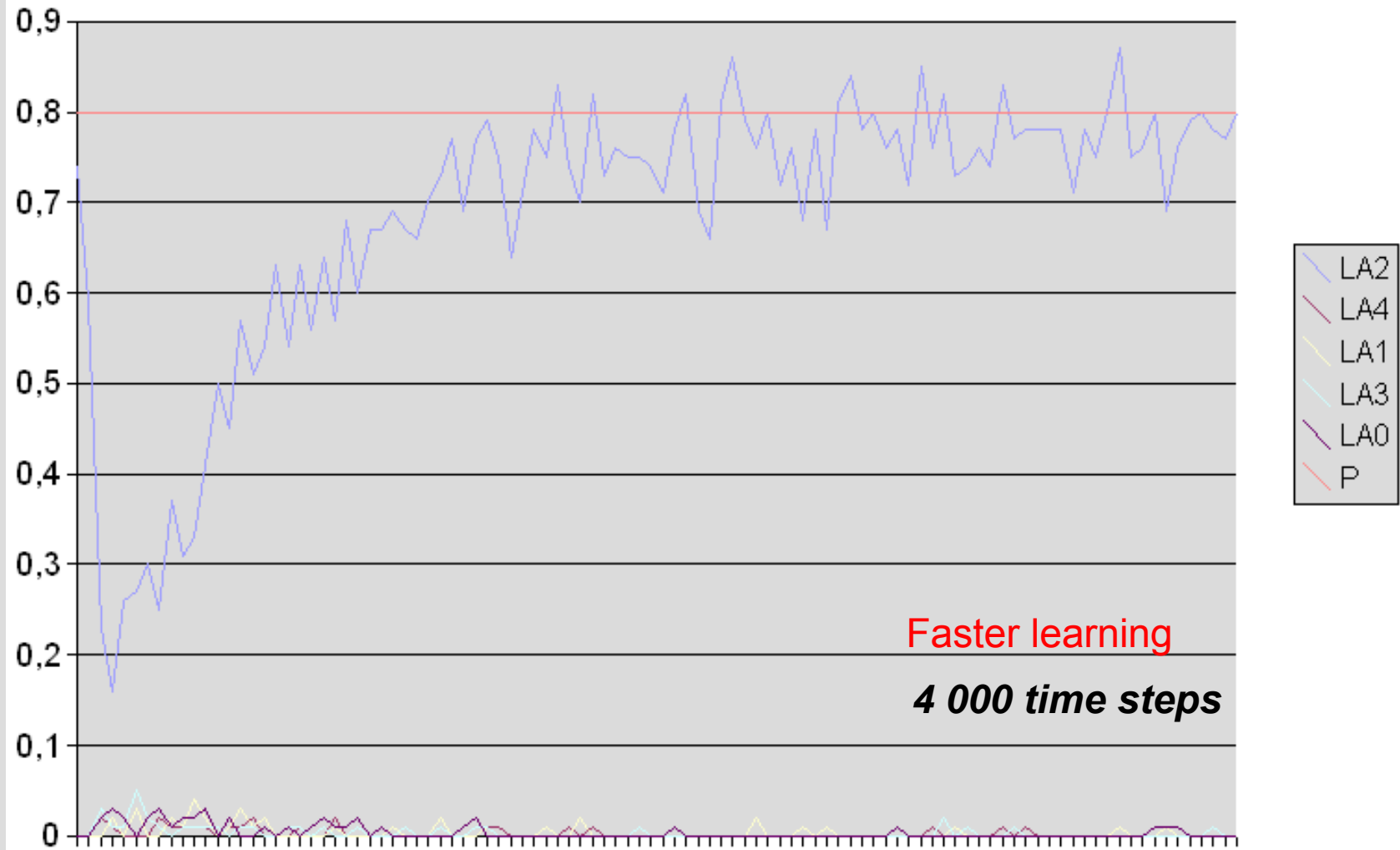
## Estimated Reward Probabilities (N=1000)



# DLA 5 Actions

Lower accuracy

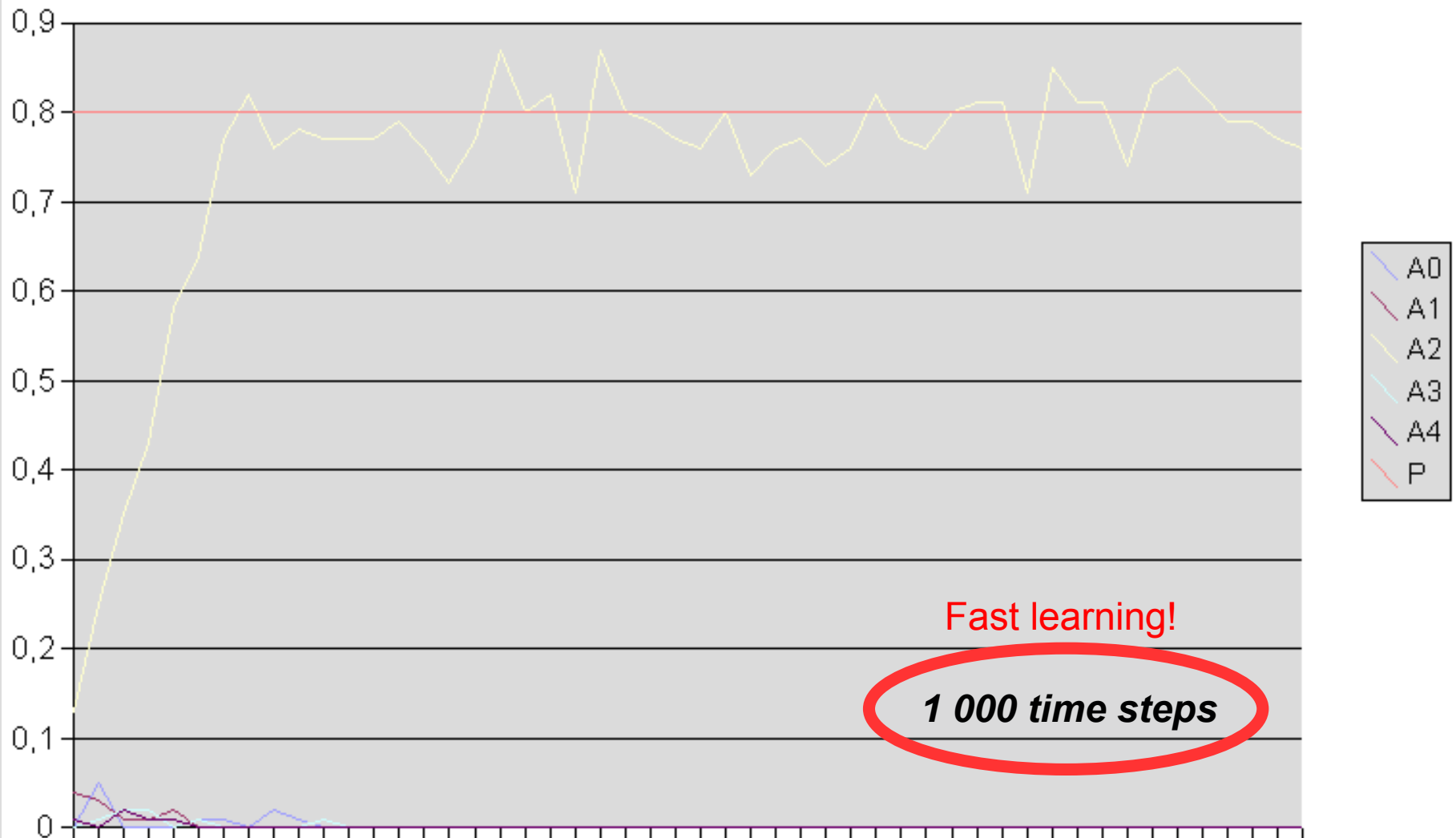
## Reward Probabilities (100 states)



# $L_{R-I}$ 5 actions

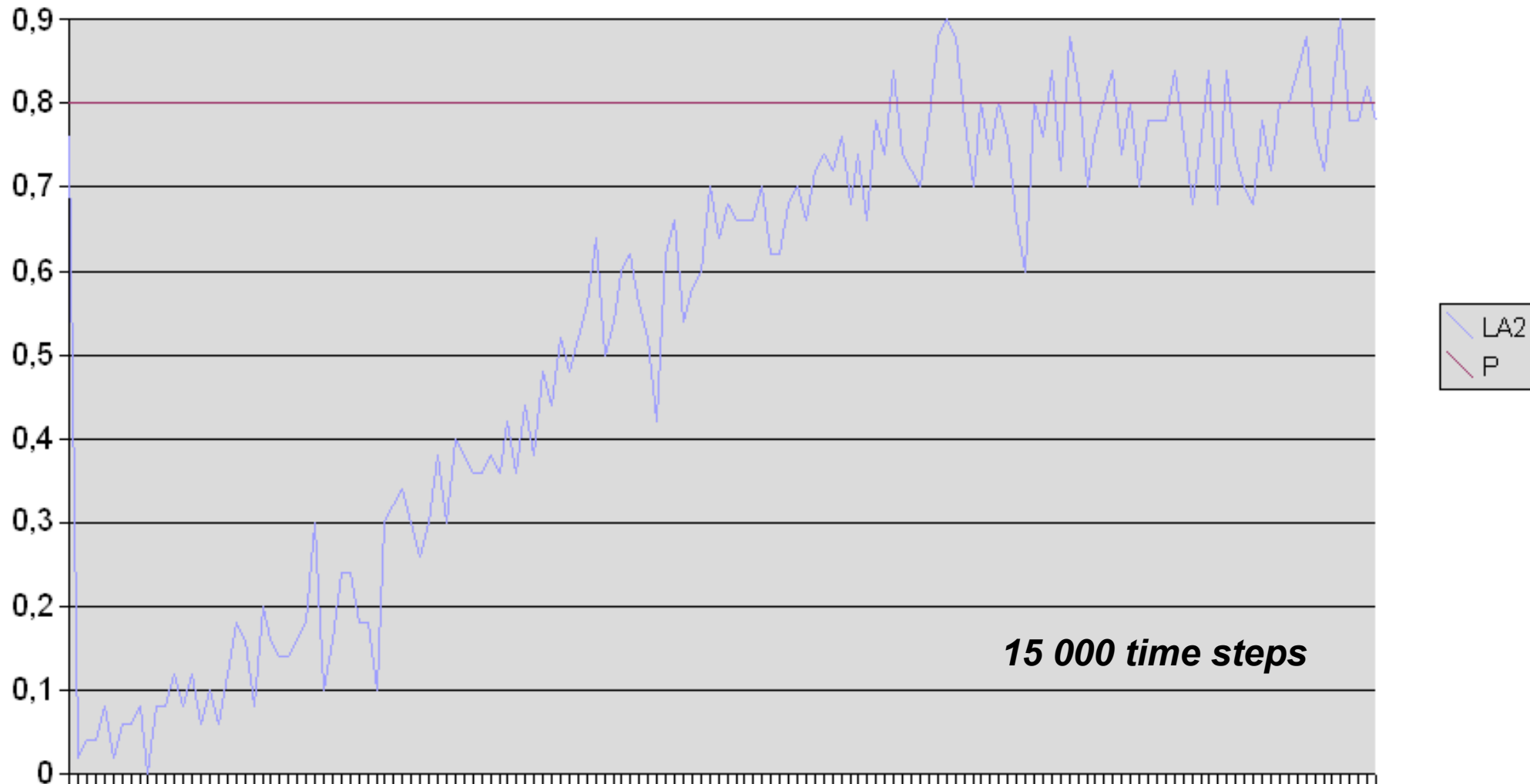
Noise

## Estimated Reward Probability ( $a = 0.1$ )



# DLA 50 Actions

Estimated Reward Probability (N = 1000)



# DLA 5 Actions

Estimated Reward Probabilities (N=1000)



# $L_{R-I}$ 50 actions

Estimated Reward Probability ( $\alpha = 0.1$ )



# Conclusion

- Able to find the best action in a *decentralized* manner.
- Scaling: *Superior* to the  $L_{R-I}$  scheme!
- Weakness: High number of internal states demand a high number of time steps for the estimated value to converge to the optimal value → *slow, but accurate* learning
  - « $L_{R-I}$  is more efficient, but does not scale»

**Questions?**