Action Planning in Anticipatory Classifier Systems

C.Y. Chuang

Dec 20, 2006
NCLab Group Meeting Presentation

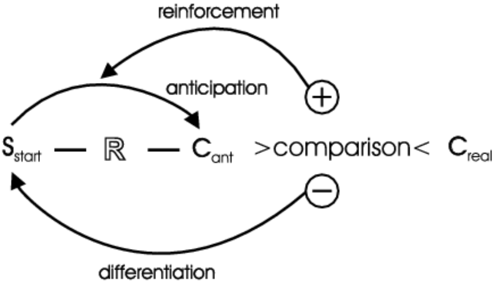
- Stolzmann, W., "Latent Learning in Khepera Robots with Anticipatory Classifier Systems",
  Genetic and Evolutionary Computation Conference 1999, Workshop Program

- Stolzmann, W., "An Introduction to Anticipatory Classifier System",
  Learning Classifier System: From Foundation to Applications, 2000, book chapter

Review of ACS

Anticipatory Behavioral Control

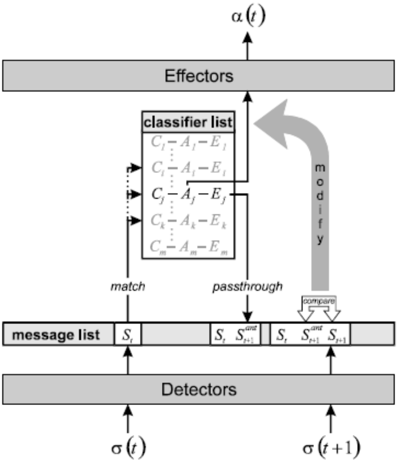
1993 Hoffmann: theory of anticipatory behavioral contorl



Basic Components of ACS

Four basic components of ACS:

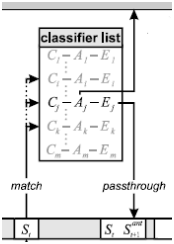
1. input interface
2. output interface
3. classifier list
4. message list



ACS Classifier

Three parts of ACS classifier:

1. Condition part C
   - $C \in \{0, 1, \#\}^n$
   - #: DON'T CARE
2. Action part A
3. Expectation part E
   - $E \in \{0, 1, \#\}^n$
   - #: PASS-THROUGH



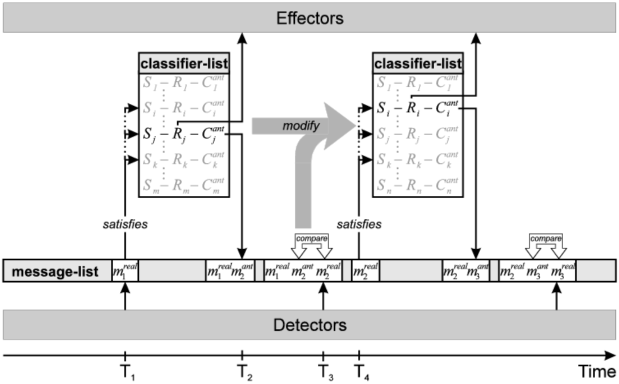
Ex. C-A-E: 0#0#-10-###1

- By 0101 $\Rightarrow$ 0101-10-0101
- By 0000 $\Rightarrow$ 0000-10-0001

Two strength values:

1. q: accuracy of anticipation
2. r: reward from environment

Behavioral Acts



Anticipation-Learning

Basic Idea:

If S_{t+1} was anticipated correctly ($S_{t+1} = S_{t+1}^{ant}$),
then the quality q should be increased.

If the anticipation was wrong ($S_{t+1} \neq S_{t+1}^{ant}$),
then generate a new classifier that anticipates S_{t+1} correctly.

If it is not possible to generate such a classifier,
then the quality q should be decreased.

Woods Environment & Performance Measure

1. Average steps to food:

   Optimal: 2.5

   $(1+2+2+3+3+4) \div 6 = 2.5$

2. Achieved knowledge:

   Reliable classifiers: $q_c \geq \theta_r$

   Reliable anticipations.

T	T	T	T	T
T	3	2	F_1	T
T	4	T	T	T
T	5	6	T	T
T	7	T	T	T
T	T	T	T	T

MazeF2

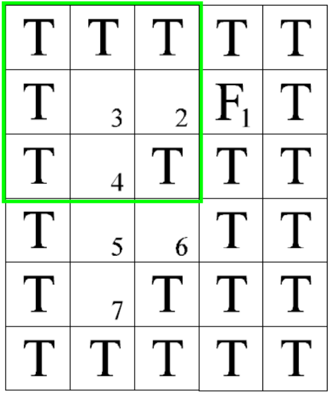
Classifier Encoding

Classifier: C - A - E

- C: $\{$t(tree), b(blank), f(food), $\#\}^8$
- A: $\{\leftarrow, \rightarrow, \uparrow, \downarrow, \nearrow, \searrow, \nwarrow, \swarrow\}$
- E: $\{$t(tree), b(blank), f(food), $\#\}^8$

Initial Classifier List

########	↑	########
########	→	########
########	↓	########
########	←	########
########	↗	########
########	↘	########
########	↙	########
########	↖	########



MazeF2

$$S_t: \begin{matrix} t & t & t \\ t & & b \\ t & b & t \end{matrix}$$

$$c: \begin{matrix} \# & \# & \# \\ \# & & \# \\ \# & \# & \# \end{matrix} \rightarrow \begin{matrix} \# & \# & \# \\ \# & & \# \\ \# & \# & \# \end{matrix}$$

$$S_{t+1}^{ant}: \begin{matrix} t & t & t \\ t & & b \\ t & b & t \end{matrix} \quad S_{t+1}: \begin{matrix} t & t & t \\ b & & f \\ b & t & t \end{matrix}$$

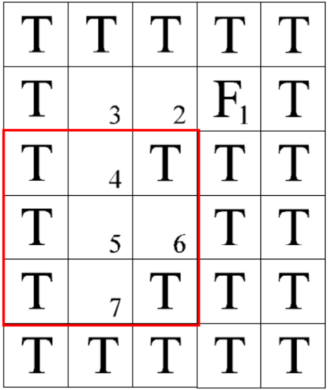
$$c_n: \begin{matrix} \# & \# & \# \\ t & & b \\ t & b & \# \end{matrix} \rightarrow \begin{matrix} \# & \# & \# \\ b & & f \\ b & t & \# \end{matrix}$$

An Extension of ACS

Problem of Original ACS

It's not possible to learn all deterministic multi-step environment.

Example: Not Correctable Situation



MazeF2

$$S_t: \begin{array}{ccc} t & b & t \\ t & & b \\ t & b & t \end{array}$$

$$c_n: \begin{array}{ccc} \# & \# & \# \\ t & & b \\ t & b & \# \end{array} \longrightarrow \begin{array}{ccc} \# & \# & \# \\ b & & f \\ b & t & \# \end{array}$$

$$S^{ant}_{t+1}: \begin{array}{ccc} t & b & t \\ b & & f \\ b & t & t \end{array} \qquad S_{t+1}: \begin{array}{ccc} b & t & t \\ b & & t \\ b & t & t \end{array}$$

Detail Description

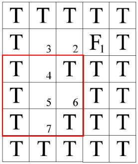
It's not possible for an ACS to solve a task where at least one behavioral act, whose behavioral consequences depend on an environmental attribute that is not changed by the action, plays an important role.

Specification of Unchanging Components: 1. Mark

Let S_p be the current state and $c = C\text{-}A\text{-}E$ the active classifier.

If the application of c does NOT lead to the *expected case*, then c remembers S_p.

The remembered S_p is called a mark (M).



MazeF2

$$S_p\!: \begin{array}{ccc} t & b & t \\ t & & b \\ t & b & t \end{array}\;,\ \text{using } c\!: \begin{array}{ccc} \# & \# & \# \\ t & & b \\ t & b & \# \end{array} \longrightarrow \begin{array}{ccc} \# & \# & \# \\ b & & f \\ b & t & \# \end{array}$$

$$S^{ant}_{p+1}\!: \begin{array}{ccc} \color{red}{t} & \color{red}{b} & t \\ b & & \color{red}{f} \\ b & t & t \end{array} \qquad S_{p+1}\!: \begin{array}{ccc} \color{red}{b} & \color{red}{t} & t \\ b & & \color{red}{t} \\ b & t & t \end{array}$$

$$c\text{-}M: \begin{array}{ccc} \# & \# & \# \\ t & & b \\ t & b & \# \end{array} \longrightarrow \begin{array}{ccc} \# & \# & \# \\ b & & f \\ b & t & \# \end{array} - \begin{array}{ccc} t & b & t \\ t & & b \\ t & b & t \end{array}$$

Specification of Unchanging Components: 2. Expected Case

Let S_q be a later state that leads to a behavioral act where c is applied and leads to the *expected case*.

T	T	T	T	T
T	3	2	F1	T
T	4	T	T	T
T	5	6	T	T
T	7	T	T	T
T	T	T	T	T

MazeF2

$$S_q:\ \begin{matrix} t & t & t \\ t & & b \\ t & b & t \end{matrix}$$

$$\text{using } c\text{-}M:\ \begin{matrix} \# & \# & \# \\ t & & b \\ t & b & \# \end{matrix} \rightarrow \begin{matrix} \# & \# & \# \\ b & & f \\ b & t & \# \end{matrix} \text{ - } \begin{matrix} t & b & t \\ t & & b \\ t & b & t \end{matrix}$$

$$S^{ant}_{q+1}:\ \begin{matrix} t & t & t \\ b & & f \\ b & t & t \end{matrix} \qquad S_{q+1}:\ \begin{matrix} t & t & t \\ b & & f \\ b & t & t \end{matrix}$$

Specification of Unchanging Components: 3. Specification

A component is randomly selected out of all components with the following property:

- S_p (i.e M) is different from S_q in this component.
- C and E consist of a $\#$ in this component.

If such a component i exists then the i-th component of C and E is respectively replaced by the i-th component of S_q.

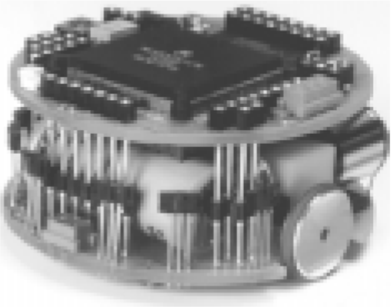
$$S_p \text{ (i.e. } M\text{): } \begin{array}{ccc} t & b & t \\ t & & b \\ t & b & t \end{array} \qquad S_q: \begin{array}{ccc} t & t & t \\ t & & b \\ t & b & t \end{array}$$

$$c\text{-}M: \begin{array}{ccc} \# & \# & \# \\ t & & b \\ t & b & \# \end{array} \longrightarrow \begin{array}{ccc} \# & \# & \# \\ b & & f \\ b & t & \# \end{array} \text{ - } \begin{array}{ccc} t & b & t \\ t & & b \\ t & b & t \end{array}$$

$$c_{new}: \begin{array}{ccc} \# & t & \# \\ t & & b \\ t & b & \# \end{array} \longrightarrow \begin{array}{ccc} \# & t & \# \\ b & & f \\ b & t & \# \end{array}$$

ACS for Khepera Robot in T-maze

Khepera Robot



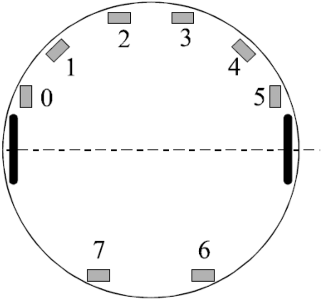


Figure 3 A Khepera robot

Environment for Khepera Robot

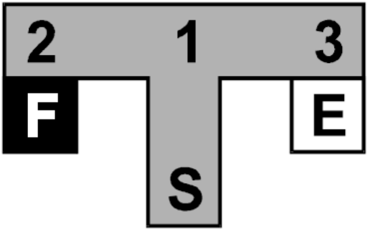


Figure 2 A simple T-maze

Detector

Input: $(d_1, d_2, d_3, d_4, d_5)$

$$d_1\colon \begin{cases} 1, & \text{if there is nothing in front of the robot;} \\ 0, & \text{if there is a wall in front of the robot.} \end{cases}$$

d_2: as d_1, but on the left of the robot.

d_3: as d_1, but behind the robot.

d_4: as d_1, but on the right of the robot.

$$d_5\colon \begin{cases} 1, & \text{if there is a infra-red light near the robot;} \\ 0, & \text{if there is no infra-red light near the robot.} \end{cases}$$

ACS's Perceptions of T-maze

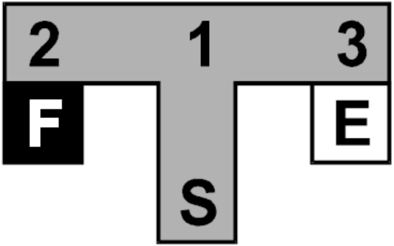


Figure 2 A simple T-maz

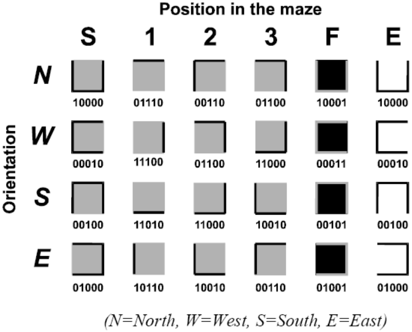


Figure 4 The ACS's perception of the T-maze

Effector

Output: (e_1)

$$e_1\colon \begin{cases} l, & \text{to make a 90 degree left turn;} \\ r, & \text{to make a 90 degree right turn;} \\ f, & \text{to go forward.} \end{cases}$$

Perceptual Aliasing

- The environment is only partially observable.
- Different environmental states might be equal for the robot.

Perceptual Aliasing

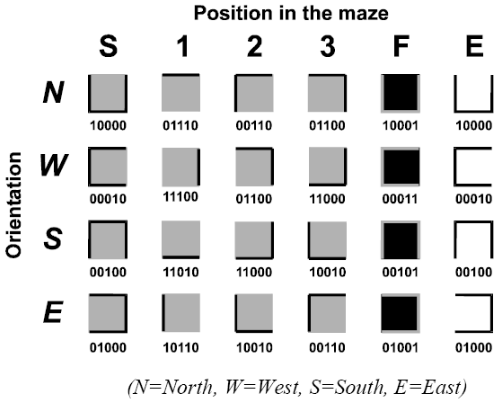


Figure 4 The ACS's perception of the T-maze

Solutions for Perceptual Aliasing

- Add memory.
- Behavioral sequences.

Behavioral Sequences

Activation of $c_1 = C_1$-A_1-E_1 is followed by $c_2 = C_2$-A_2-E_2

Generate $c_{new} = C_{new}$-A_{new}-E_{new}

- $C_{new} = passthrough(C_2, C_1)$
- $A_{new} = A_1 A_2$
- $E_{new} = passthrough(E_1, E_2)$

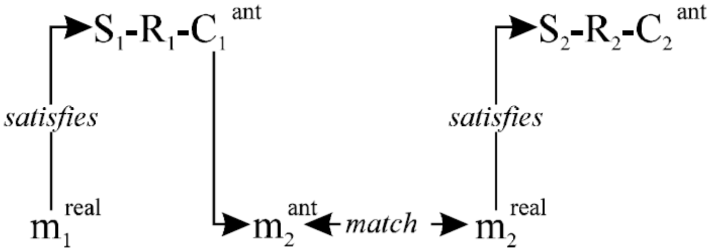
Action Planning in ACS

Assumption

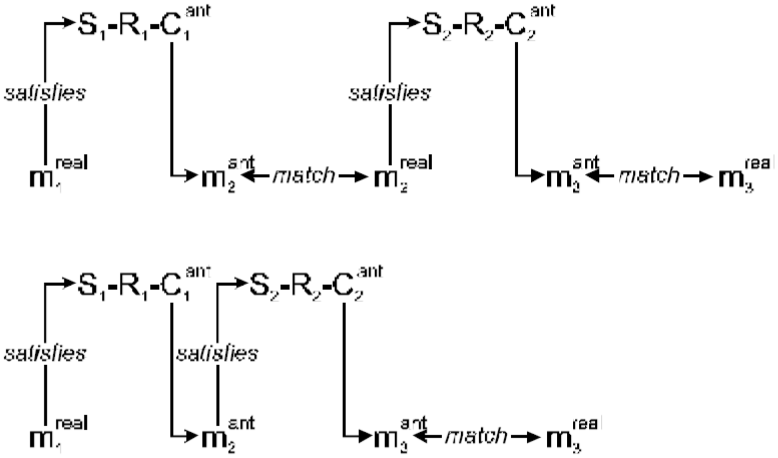
During the exploration phase,

the ACS has learned an internal model of the environment.

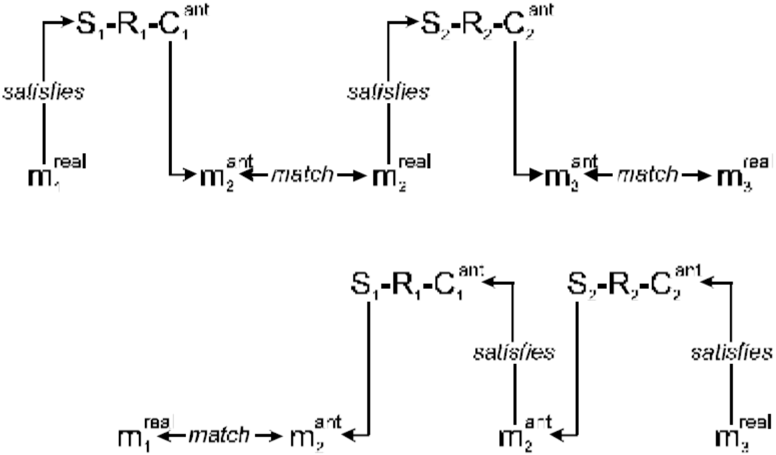
Internal Model



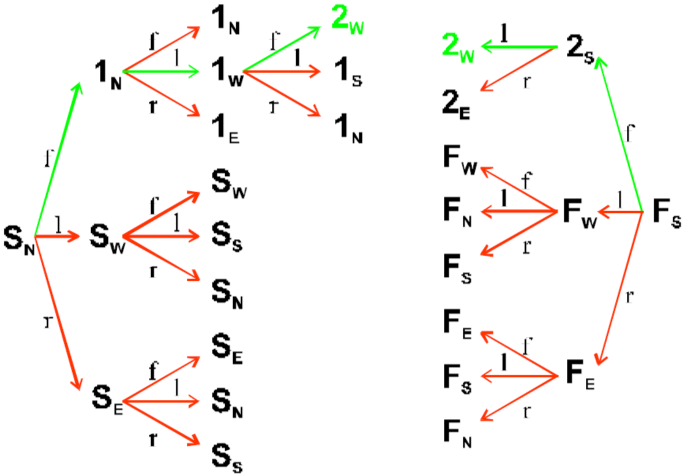
Forward Planning



Backward Planning



Bidirectional Planning



Result

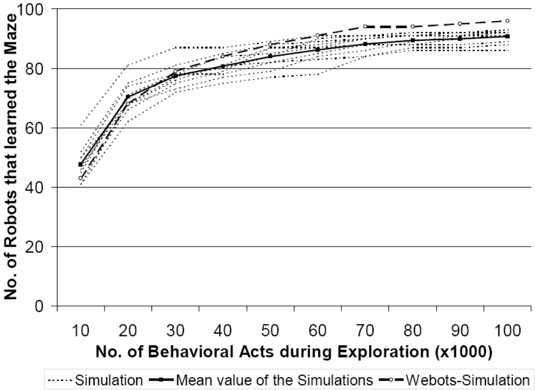


Figure 9: Results of the simulations with food in F

Result

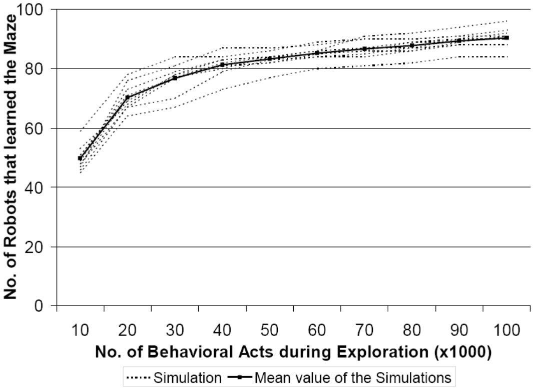
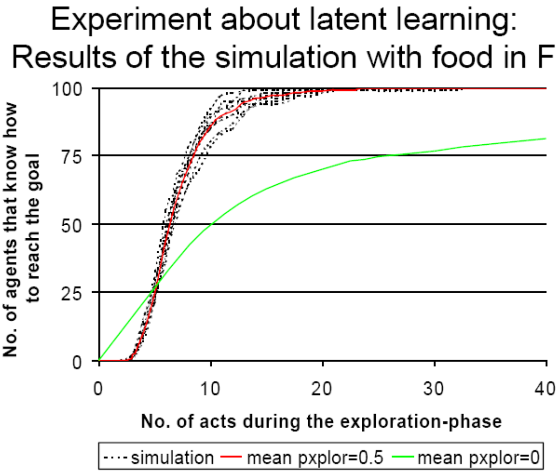


Figure 10: Results of the simulations with food in E

Result


Experiment about latent learning:
Results of the simulation with food in F
No. of agents that know how to reach the goal
100
75
50
25
0
0
10
20
30
40
No. of acts during the exploration-phase
simulation
mean pxplor=0.5
mean pxplor=0