An Idiotypic Immune Network as a Short-Term Learning Architecture for Mobile Robots

Amanda Whitbrook, Uwe Aickelin, Jonathan Garibaldi

Introduction

Problem
- Robots with no *a priori* knowledge
  - Learning period - increased task time
- Risk of damage
- Robot with *pre-engineered* knowledge
  - No adaptation, flexibility, learning capacity

Inspiration
- A good compromise - take inspiration from immune system.
- Immune system - both Long-Term Learning (LTL) and Short-Term Learning (STL)
  - LTL - antibodies built from gene libraries that have evolved over the lifetime of the *species*
  - STL - antibody repertoire changes in response to antigens. Antibody concentrations change. Happens over lifetime of *individual*
Introduction
Proposed Solution

- Couple LTL and STL as the immune system does
- LTL = Genetic Algorithm with Reinforcement Learning (RL)
  - Robot has a priori knowledge but it isn’t engineered
  - Use the evolved behaviours to seed the STL phase
- STL = Idiotypic Immune System (Adaptive and flexible) + RL
  - Robot uses the evolved behaviours in an adaptive way

What’s new?

- Combining an accelerated GA in simulation with an idiotypic AIS in real world is a novel combination
- Behaviours are encoded in a novel way to give greater diversity - more later
- Would like to gain insight into the following research questions:
  - Is an LTL-STL strategy better than an STL-only strategy?
  - Can the evolved behaviours be transferred directly to the AIS and the real world?
  - Is antibody replacement necessary in the STL phase?
  - Does idiotypic selection improve performance?

Overcoming past drawbacks

- Most robotic idiotypic networks have used a small number of pre-engineered antibodies
  - Limits the self-discovery and learning properties
  - Most have just evolved the network connections
  - Here, the behaviours themselves are evolved and the initial connections are derived from the RL scores in the GA
  - Provides greater flexibility for the system
- Use of very rapid simulations using Webots simulator (200x real time) means that AIS can be seeded within a realistic time frame (within half an hour)

E-puck Robot

- Webots simulator natively supports the e-puck
- Can use the same programs in the simulator and transfer them to the real e-puck
- 8 IR sensors
- Frontal camera
- Two independent drive wheels
Test Environments and Problem
LTL Phase (GA)

Test Environments and Problem
STL Phase (AIS) - Simulated

Test Environments and Problem
STL Phase (AIS) - Real

System Architecture
Jerne’s Idiotypic Network

- Antibodies have an idiotope as well as a paratope.
- They suppress and stimulate each other as well as being stimulated by antigens
System Architecture

Farmer’s Model

- Farmer’s computational model of Jerne’s Idiotypic Network theory very popular for mobile robotics
- Models continuous antibody concentration change
- When applied to robotics:
  - Antibodies --- Robot behaviours
  - Antigens --- Environmental information
  - In theory flexible and dynamic because antibody that best matches current antigen is not necessarily selected

System Architecture

Antigens - Environmental Information

<table>
<thead>
<tr>
<th>CODE</th>
<th>TYPE</th>
<th>NAME</th>
<th>NOTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>MARKER</td>
<td>MARKER UNSEEN</td>
<td>NO OBSTACLE</td>
</tr>
<tr>
<td>1</td>
<td>MARKER</td>
<td>MARKER SEEN</td>
<td>NO OBSTACLE</td>
</tr>
<tr>
<td>2</td>
<td>OBSTACLE</td>
<td>OBSTACLE RIGHT</td>
<td>MAX IR = 250 – 2400, (IR = 0, 1, 2)</td>
</tr>
<tr>
<td>3</td>
<td>OBSTACLE</td>
<td>OBSTACLE REAR</td>
<td>MAX IR = 250 – 2400, (IR = 3, 4)</td>
</tr>
<tr>
<td>4</td>
<td>OBSTACLE</td>
<td>OBSTACLE LEFT</td>
<td>MAX IR = 250 – 2400, (IR = 5, 6, 7)</td>
</tr>
<tr>
<td>5</td>
<td>OBSTACLE</td>
<td>COLLISION RIGHT</td>
<td>MAX IR = 2400 +, (IR = 0, 1, 2)</td>
</tr>
<tr>
<td>6</td>
<td>OBSTACLE</td>
<td>COLLISION REAR</td>
<td>MAX IR = 2400 +, (IR = 3, 4)</td>
</tr>
<tr>
<td>7</td>
<td>OBSTACLE</td>
<td>COLLISION LEFT</td>
<td>MAX IR = 2400 +, (IR = 5, 6, 7)</td>
</tr>
</tbody>
</table>

System Architecture

Antibodies - Behaviours

<table>
<thead>
<tr>
<th>TYPE</th>
<th>SPEED</th>
<th>FREQ. TURN</th>
<th>ANGLE TURN</th>
<th>DIR. TURN</th>
<th>FREQ. R. TURN</th>
<th>ANGLE R TURN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
<td>MAX</td>
<td>MIN</td>
<td>MAX</td>
<td>MIN</td>
<td>MAX</td>
</tr>
<tr>
<td>0</td>
<td>WANDER SINGLE</td>
<td>50</td>
<td>400</td>
<td>10</td>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>WANDER BOTH</td>
<td>50</td>
<td>400</td>
<td>10</td>
<td>90</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>FORWARD TURN</td>
<td>50</td>
<td>400</td>
<td>-</td>
<td>-</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>STATIC TURN</td>
<td>50</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>100</td>
</tr>
<tr>
<td>4</td>
<td>REVERSE TURN</td>
<td>300</td>
<td>400</td>
<td>-</td>
<td>-</td>
<td>20</td>
</tr>
<tr>
<td>5</td>
<td>TRACK MARKERS</td>
<td>50</td>
<td>400</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

- Fusion of basic behaviour types with a number of different attributes that can take many values
- Many different behaviours possible

Summary of LTL Phase (GA)

- Antibody attribute values are crossed, mutated etc.
- Fitness is the number of collisions + task time
- Uses RL to speed up convergence
- Need good balance between behaviour diversity and speed of convergence
- Concluded that it’s best to use:
  - Five separate populations of 10 robots
  - Mutation rate of 5%
- Able to deliver starting behaviours within 10 minutes in static world and within 25 minutes within dynamic world
- The five fittest robots from the final generation pass their behaviours to the AIS
GA selects 5 fittest robots
- Each robot has 8 antibodies, one for each antigen
- $x = 5$ (no. of robots), $y = 8$ (no. of antigens)
- Paratope is an $xy$ or $5 \times 8$ matching matrix
- Element values are the normalized result of multiplying the final RL scores, $L$ (after GA phase) by the fitness of the set $\mu$
- Paratope constantly changes

System Architecture
Paratope

Antigen - How good is each antibody?

<table>
<thead>
<tr>
<th>Antigen</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>$L_{00} \mu_0$</td>
<td>$L_{10} \mu_0$</td>
<td>$L_{20} \mu_0$</td>
<td>$L_{30} \mu_0$</td>
<td>$L_{40} \mu_0$</td>
<td>$L_{50} \mu_0$</td>
<td>$L_{60} \mu_0$</td>
<td>$L_{70} \mu_0$</td>
</tr>
<tr>
<td>1</td>
<td>$L_{01} \mu_1$</td>
<td>$L_{11} \mu_1$</td>
<td>$L_{21} \mu_1$</td>
<td>$L_{31} \mu_1$</td>
<td>$L_{41} \mu_1$</td>
<td>$L_{51} \mu_1$</td>
<td>$L_{61} \mu_1$</td>
<td>$L_{71} \mu_1$</td>
</tr>
<tr>
<td>2</td>
<td>$L_{02} \mu_2$</td>
<td>$L_{12} \mu_2$</td>
<td>$L_{22} \mu_2$</td>
<td>$L_{32} \mu_2$</td>
<td>$L_{42} \mu_2$</td>
<td>$L_{52} \mu_2$</td>
<td>$L_{62} \mu_2$</td>
<td>$L_{72} \mu_2$</td>
</tr>
<tr>
<td>3</td>
<td>$L_{03} \mu_3$</td>
<td>$L_{13} \mu_3$</td>
<td>$L_{23} \mu_3$</td>
<td>$L_{33} \mu_3$</td>
<td>$L_{43} \mu_3$</td>
<td>$L_{53} \mu_3$</td>
<td>$L_{63} \mu_3$</td>
<td>$L_{73} \mu_3$</td>
</tr>
<tr>
<td>4</td>
<td>$L_{04} \mu_4$</td>
<td>$L_{14} \mu_4$</td>
<td>$L_{24} \mu_4$</td>
<td>$L_{34} \mu_4$</td>
<td>$L_{44} \mu_4$</td>
<td>$L_{54} \mu_4$</td>
<td>$L_{64} \mu_4$</td>
<td>$L_{74} \mu_4$</td>
</tr>
</tbody>
</table>

System Architecture
Idiotope

Example Idiotope - Is it a poor antibody?

<table>
<thead>
<tr>
<th>Antigen</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

NB Idiotope constantly changes
System Architecture
Antibody Selection Process

- A 1-stage process for non-idiotypic selection
  1. Determine index of presenting antigen $m$, then examine the five paratope values $P_{im}$ ($i = 0, ..., x-1$) of the matching antibodies and select the one with the highest value, $\alpha$.

- A 3-stage process for idiotypic selection
  1. Select $\alpha$ as above
  2. Calculate stimulatory and suppressive effects of $\alpha$ on all antibodies in the system and re-calculate $P_{im}$ values

System Architecture
Stage 2 - Stimulation and Suppression

- Stimulation calc - compare the idiotope of $\alpha$ with the paratopes of the other antibodies - reward antibodies similar to $\alpha$
  $$\varepsilon_{im} = k_1 \sum_{j=0}^{x-1} (1 - P_{ij}) \sum_{c, n} C_{ij} C_{nj}$$

- Suppression calc - compare the paratope of $\alpha$ with the idiotopes of the other antibodies - penalize antibodies different to $\alpha$
  $$\delta_{im} = k_2 \sum_{j=0}^{x-1} P_{nj} \sum_{c, n} C_{ij} C_{nj}$$

- New strength-of-match values given by:
  $$P_{im}' = (P_{im})_2 = (P_{im})_1 + \varepsilon_{im} - \delta_{im}$$

System Architecture
Stage 3 - Activation

- Adjust numbers of clones and re-calculate concentrations
- Clone numbers fluctuate according to a variation of Farmer’s equation:
  $$N_{im(2)} = b P_{im(2)} + N_{im(1)} (1 - k_3)$$
- $m$ is the index of the presenting antigen, $b$ and $k_3$ are constants
- Concentration of every antibody in the system changes according to:
  $$C_{ij} = \frac{\Phi N_{ij}}{\sum_{k=0}^{x} \sum_{l=0}^{x} N_{kl}}$$

Then work out activation $\lambda_{im}$ of each antibody matching $m$ and choose the antibody with the highest activation, $\beta$
- Sometimes $\alpha = \beta$, sometimes $\alpha \neq \beta$
Experimental Procedures

- Testing the following systems:
  - Seeded with idiotypic effects ($b = 100$, $k_3 = 0$, $k_1 = 0.85$, $k_2 = 1.10$)
  - Seeded with RL only
  - Unseeded with idiotypic effects ($b = 100$, $k_3 = 0$, $k_1 = 0.85$, $k_2 = 1.10$)
  - Unseeded with RL only
- 30 trials performed in each simulated world for each system
- 20 trials performed in real world for each system
- Measure time taken, collisions and solution quality as $q = (t + 8c)/2$
- Limit time to 4000s and collisions to 100, robots exceeding these limits fail
- Conduct two-tailed standard $t$-tests on results sets
- Accept differences as significant only at 99% level

Real World Demos

- Robot finding block

Results #1 - Scores and Fail Rates

<table>
<thead>
<tr>
<th>System</th>
<th>Simulated World 1</th>
<th>Simulated World 2</th>
<th>Real World</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$c$</td>
<td>$r$</td>
<td>$q$</td>
</tr>
<tr>
<td>SIE</td>
<td>1</td>
<td>562</td>
<td>284</td>
</tr>
<tr>
<td>SRL</td>
<td>8</td>
<td>1298</td>
<td>679</td>
</tr>
<tr>
<td>UIE</td>
<td>26</td>
<td>1513</td>
<td>862</td>
</tr>
<tr>
<td>URL</td>
<td>45</td>
<td>2150</td>
<td>1253</td>
</tr>
<tr>
<td>System</td>
<td>Mean % failures in all worlds</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$c$</td>
<td>$r$</td>
<td>$q$</td>
</tr>
<tr>
<td>SIE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SRL</td>
<td>3</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>UIE</td>
<td>46</td>
<td>13</td>
<td>49</td>
</tr>
<tr>
<td>URL</td>
<td>59</td>
<td>24</td>
<td>68</td>
</tr>
</tbody>
</table>

Results #2 - Significant Differences

<table>
<thead>
<tr>
<th>Systems</th>
<th>Simulated World 1</th>
<th>Simulated World 2</th>
<th>Real World</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$c$</td>
<td>$r$</td>
<td>$q$</td>
</tr>
<tr>
<td>SIE</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>SRL</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>UIE</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>URL</td>
<td>98</td>
<td>99</td>
<td>72</td>
</tr>
<tr>
<td>SRL</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>UIE</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>URL</td>
<td>87</td>
<td>90</td>
<td>93</td>
</tr>
</tbody>
</table>
Results #3 - Summary

- Seeded idiotypic system:
  - Significantly better than ALL unseeded systems
  - 0% fail rate, others do not
  - Mostly significantly better than seeded non-idiotypic system

- For unseeded systems:
  - It makes no difference whether idiotypic selection is used or not

Conclusions

- Have coupled an accelerated GA in simulation with an idiotypic AIS in real world
- Have encoded behaviours in a way that allows diverse ones to evolve
- Have provided evidence to show that:
  - An LTL-STL strategy is better than an STL-only strategy
  - The evolved behaviours can be transferred directly to the AIS and the real world
  - Antibody replacement is not necessary in the STL phase when good seeding and idiotypic selection is used.
  - Idiotypic selection improves performance significantly

- Provided a more flexible idiotypic AIS system with a realistic training period.
- Conducted testing in more complex environments than previous approaches