Using Machine Learning to Control Coupled, Dynamical Life Support Systems

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Houston TX 77058
kortenkamp@jsc.nasa.gov
http://www.traclabs.com/~korten
Team

- **NASA JSC**
  - David Kortenkamp, Pete Bonasso
- **NASA Ames**
  - Justin Boyan
- **Rice University**
  - Devika Subramanian
- **Carnegie Mellon University**
  - Jeff Schneider
- **Naval Research Laboratory**
  - Alan Schultz
Advanced Life Support Systems

- **Regenerative**
  - produce own food
  - recycle water and air
- **Low margins, volume, mass, energy and labor**
- **Limited resupply**
- **Highly interconnected**
- **Require optimization and tight control**
- **Desire for autonomy**

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Control Issues

- Advanced Life Support (ALS) systems are:
  - Dynamic – it is not sufficient only to find a single a priori setting
  - Non-stationary – presence of adaptive organisms such as humans, plants and bacteria as well as degradation requires adaptation
  - Safety-sensitive – crew depends on system for life support, verification and validation are important
  - Coupled – multiple heterogeneous systems (water, air, plants, people) all effect one another in obvious and subtle ways
Previous and Current Control Systems

- Several experiments at JSC based on the 3T control architecture
- 3T
  - planning
  - sequencing
  - control
Phase III Crewed Test

- Four crew members for 91 days in a closed chamber.
- Wheat crop in another chamber.
- 3T managed transfer of gases between the two chambers.
- Operated reliably round-the-clock for 73 days (10/6/97-12/19/97).
- Typically ran without human supervision or intervention.

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Air Revitalization System

- Simulation of an ARS using a discrete event simulator
- 3T control (skills and RAPs) integrated with Livingstone MIR (from Ames)
- Planning currently being added
Water Recovery System

- Four integrated subsystems:
  - Biological water processor
  - Reverse osmosis system
  - Air evaporation system
  - Post-processor
- 3T skills (over 75 separate skills)
- 3T RAPs
- ~200 sensors and actuators

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Lessons Learned

- Change in crew role from vigilance to supervision
  - System should let them know if there is something to be looked at
  - System should provide summaries of control actions

- Coupled systems
  - Full ALSS = WRS + ARS + plants + food production + climate control becomes planning/scheduling problem, not a control problem
  - AWRS had four subsystems itself – no planning.
Lessons Learned 2

- Small changes to sensor calibration or the underlying biological/chemical processes requires expensive recoding of control procedures.
- Changes to the desired operating regime (e.g., optimizing for a different resource) requires expensive recoding of control procedures.
- Complex interactions are difficult to predict.
- Adaptation of control code is required for long-duration, autonomous missions.
Learning in ALS Systems

- Some of the control will be hand-coded and fixed
- Some portion will need to adapt as the system runs
- Many open research questions
  - On-line vs. off-line learning
  - Limits of experimentation with the real system
  - Fidelity of models and relationship to learning quality
  - Abstraction of state and action space (making system aware of hidden states)
  - Crew interaction with learning system (inspectability and instructability)
The Role of Learning

- Detecting signatures
  - Parsing real-world data stream to recognize events
- Refining models
  - Using feedback from actual system to adjust models
- Robust design
  - Searching through design criteria for optimal solution
- Learning/optimizing sequences
- Integrating with autonomous control
- Adaptive crew interfaces
- Control system design methodology
  - Using learning algorithms to find important variables and interactions
- Optimizing resource allocation

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
ALS State Space

• Potential state space is enormous and hybrid (i.e., mix of discrete and continuous) so we need to abstract

• Possible abstractions are
  – Current levels of consumables (air, water, food)
  – Quality of air and water and health of plants
  – Flow paths for water and air through the system
  – Current energy allocations to subsystems
  – The current phase of operation
  – Crew health/happiness
  – Temperatures and other environmental measures

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
ALS Action Space

- Potential action space is large and hybrid
- Combination of physical actions to produce abstract actions
  - Allocation of energy amongst subsystems
  - Use of consumable stores
  - Crew activity
  - Routing of air/water flows
  - Planting/harvesting of crops (when and which)
  - Adjusting crop light levels
  - Adjusting climate controls
  - Venting of gases to the outside atmosphere
ALS Rewards and Feedback

- **Final or end state rewards**
  - Duration of mission with different controllers
  - Total crew productivity over mission duration
  - Total amount of air, water, food or energy available in system or stores

- **Progress measures**
  - Quality of air and water and health of plants
  - Plant growth rates and plant food output
  - Climate feedback (keeping climate parameters within boundaries)
  - Health/satisfaction of crew
Experiments with a Simplified Simulation

• Goal was to determine if simple ML techniques worked on simple ALS simulation. Then scale up each accordingly.

• Simple, deterministic simulation:
  – Air, water, plants and crew modules
  – Simulation time in ‘ticks’ (one hour)
  – Single tick runs each process once
  – Each process takes energy and dirty resources and produces a resource (clean air, clean water or food)
  – Crew consumes clean resources and produces dirty
  – Stores for food, water and air.
The Simulation
Simulation Resource Rates

• **Low activity level**
  - dirty water \( (W_d) = 0.95 \) clean water \( (W_c) \)
  - dirty air \( (A_d) = 0.95 \) clean air \( (A_c) \)
  - Science = \((food + W_c + A_c) * 0.30\)

• **Medium activity level**
  - dirty water \( (W_d) = 0.85 \) clean water \( (W_c) \)
  - dirty air \( (A_d) = 0.85 \) clean air \( (A_c) \)
  - Science = \((food + W_c + A_c) * 0.60\)

• **High activity level**
  - dirty water \( (W_d) = 0.75 \) clean water \( (W_c) \)
  - dirty air \( (A_d) = 0.95 \) clean air \( (A_c) \)
  - Science = \((food + W_c + A_c) * 0.90\)
ML Experiments

• **Goal**
  - Find control policy that maximizes either mission length (i.e., ticks), total science or combination of both

• **Assumptions**
  - Markov process

• **Experimented with**
  - Reinforcement learning (search in value space)
  - Genetic algorithms (search in policy space)

• **Developed a novel genetic algorithm approach**
Reinforcement Learning

- **Q learning**
  \[
  Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[r + \max_{a' \in A} Q(s', a')] 
  \]

- **Action space**
  - Energy levels (increase or decrease), activity level (increase or decrease), use of stores

- **State space**
  - Energy levels, activity level

- **Reward**
  - Change in science at each tick (when maximizing science)
  - Increment of one for each tick (when maximizing duration)
  - End of mission (negative reward)
## Results

### Samples science ticks

<table>
<thead>
<tr>
<th>Samples</th>
<th>science</th>
<th>ticks</th>
</tr>
</thead>
<tbody>
<tr>
<td>502</td>
<td>150.448308</td>
<td>13</td>
</tr>
<tr>
<td>1018</td>
<td>318.30795</td>
<td>20</td>
</tr>
<tr>
<td>1525</td>
<td>179.515967</td>
<td>30</td>
</tr>
<tr>
<td>2037</td>
<td>318.30795</td>
<td>20</td>
</tr>
<tr>
<td>2548</td>
<td>318.30795</td>
<td>20</td>
</tr>
<tr>
<td>3059</td>
<td>322.06935</td>
<td>20</td>
</tr>
<tr>
<td>3577</td>
<td>318.30795</td>
<td>20</td>
</tr>
<tr>
<td>4078</td>
<td>318.30795</td>
<td>20</td>
</tr>
<tr>
<td>4594</td>
<td>318.30795</td>
<td>20</td>
</tr>
<tr>
<td>5111</td>
<td>322.06935</td>
<td>20</td>
</tr>
<tr>
<td>5629</td>
<td>322.06935</td>
<td>20</td>
</tr>
<tr>
<td>6009</td>
<td>322.06935</td>
<td>20</td>
</tr>
</tbody>
</table>

### % Random

<table>
<thead>
<tr>
<th>Samples</th>
<th>ticks</th>
<th>% Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>1011</td>
<td>28</td>
<td>90</td>
</tr>
<tr>
<td>2014</td>
<td>25</td>
<td>80</td>
</tr>
<tr>
<td>3035</td>
<td>28</td>
<td>70</td>
</tr>
<tr>
<td>4037</td>
<td>28</td>
<td>60</td>
</tr>
<tr>
<td>5070</td>
<td>28</td>
<td>50</td>
</tr>
<tr>
<td>6082</td>
<td>28</td>
<td>40</td>
</tr>
<tr>
<td>7107</td>
<td>28</td>
<td>30</td>
</tr>
<tr>
<td>8217</td>
<td>38</td>
<td>20</td>
</tr>
<tr>
<td>9276</td>
<td>38</td>
<td>20</td>
</tr>
<tr>
<td>10080</td>
<td>38</td>
<td>20</td>
</tr>
</tbody>
</table>

---

David Kortenkamp  
NASA Johnson Space Center/Metrica Inc.
Genetic Algorithms

- Bit string represents actions (or inputs) into the system

10101011010

- Simulation used to evaluate strings

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Results

<table>
<thead>
<tr>
<th>Exp</th>
<th>Ticks</th>
<th>Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
<td>147.7</td>
</tr>
<tr>
<td>2</td>
<td>31</td>
<td>147.5</td>
</tr>
<tr>
<td>3</td>
<td>31</td>
<td>49.2</td>
</tr>
<tr>
<td>4</td>
<td>31</td>
<td>49.2</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>49.2</td>
</tr>
<tr>
<td>6</td>
<td>31</td>
<td>51.9</td>
</tr>
<tr>
<td>7</td>
<td>31</td>
<td>52</td>
</tr>
<tr>
<td>8</td>
<td>31</td>
<td>147.5</td>
</tr>
<tr>
<td>9</td>
<td>31</td>
<td>140.9</td>
</tr>
<tr>
<td>10</td>
<td>31</td>
<td>147.5</td>
</tr>
</tbody>
</table>

evaluation = mission duration
i.e., ticks $^5$

<table>
<thead>
<tr>
<th>Exp</th>
<th>Science</th>
<th>Ticks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>183.9</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>184</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>183.9</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>184</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>184</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>184</td>
<td>26</td>
</tr>
<tr>
<td>7</td>
<td>183.8</td>
<td>26</td>
</tr>
<tr>
<td>8</td>
<td>183.8</td>
<td>26</td>
</tr>
<tr>
<td>9</td>
<td>184</td>
<td>26</td>
</tr>
<tr>
<td>10</td>
<td>184</td>
<td>26</td>
</tr>
</tbody>
</table>

evaluation = science cubed

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
What’s Wrong?

- We should be doing better, even simple control strategies achieve better results
- Hypothesis
  - Our abstraction of the state and action space is missing some key aspect of the problem
Multi-step **Genetic Algorithms**

- Bit string encodes a multi-step “plan”

```
10101011010 | 10100011011 | 00101111010 | 10111010110
```

- Bit strings are $n$ times 11 in length where $n$ is plan length before repeating.
Results

- Results were stunning

Compare to 31 ticks max using “normal” GA
Results (cont.)

Cumulative Science (Eval = [science]^3)

Compare to 184 total science max using “normal” GA

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
What’s Happening?

- GA is learning a “pulsing” strategy

![Graph showing store allocation and crew air and water output](image-url)
Eking out an Existence

Stores Levels for Pro-Ticks Policy

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Informing a GA

- How to use human intuition to help a GA along

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Results

Activity level constrained to 1

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Revisiting Q-Learning

- Our initial state space was impoverished and open loop (but GA does not need an explicitly designed state space and is also open loop)

- New state space
  - Water and air outputs from the crew (discretized to 0-2)
  - Air, water, food store status (0 or 1)
  - Ticks since last action for air and water stores

- New action space
  - To use or not use water, air or food store
Results still not as good as multi-step GA, but much better than original RL.

New RL finds same “pulsing” strategy.

<table>
<thead>
<tr>
<th>Samples</th>
<th>Science</th>
<th>Ticks</th>
</tr>
</thead>
<tbody>
<tr>
<td>647</td>
<td>350.166058</td>
<td>67</td>
</tr>
<tr>
<td>1295</td>
<td>362.099302</td>
<td>67</td>
</tr>
<tr>
<td>1951</td>
<td>411.486381</td>
<td>68</td>
</tr>
<tr>
<td>2622</td>
<td>355.082955</td>
<td>61</td>
</tr>
<tr>
<td>3267</td>
<td>264.029052</td>
<td>45</td>
</tr>
<tr>
<td>3925</td>
<td>379.153828</td>
<td>68</td>
</tr>
<tr>
<td>4554</td>
<td>361.391647</td>
<td>61</td>
</tr>
<tr>
<td>5213</td>
<td>909.616541</td>
<td>176</td>
</tr>
<tr>
<td>5865</td>
<td>496.283236</td>
<td>92</td>
</tr>
<tr>
<td>6500</td>
<td>319.51492</td>
<td>61</td>
</tr>
<tr>
<td>7108</td>
<td>284.349158</td>
<td>49</td>
</tr>
<tr>
<td>7718</td>
<td>505.187961</td>
<td>92</td>
</tr>
<tr>
<td>8361</td>
<td>672.872989</td>
<td>123</td>
</tr>
<tr>
<td>9015</td>
<td>482.361459</td>
<td>87</td>
</tr>
<tr>
<td>9681</td>
<td>398.32097</td>
<td>74</td>
</tr>
<tr>
<td>10068</td>
<td>423.490573</td>
<td>77</td>
</tr>
</tbody>
</table>
Other Experiments

- Optimizing both mission duration and science output at the same time
- Optimizing in the face of finite energy
- Optimizing in the face of finite time
- Feeding the results of GA back into RL
Open Questions

- GA is open-loop and has no guarantees, however, it does not require explicitly defining state.
- Can we use results of GA to design a state space that will let RL do better?
- Can we use results of GA to hand-code a better controller?
- Research question: How can different learning algorithms be used to compliment each other?
- How can human intuition be used to guide machine learning algorithms in their search?
On-going Work

- Developing a more challenging simulation
  - Stochastic processes
  - Cycles (plants, crew)
  - Model subsystems within water, air and plants
  - Portable (Java) and available to all

- Objective
  - Make simulation more realistic
  - Re-test machine learning techniques and see what changes are necessary
Future Work

- Begin working on other machine learning application areas
  - Sequence learning
    - Learning contexts as well as sequences
  - Integration with control systems
    - How good does initial control have to be for on-line learning to work?
    - How does control system decide when to devote resources to learning and when to use new knowledge?
- Investigate other ML techniques (memory-based, Samuel)
- Continue to explore theoretical issues of abstraction and model fidelity requirements
- Issue challenge to AI research community and make simulation available to all
- Begin applying techniques to real-world ALS testbeds

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Integrated Control of ALSS

- Distributed, integrated monitoring and control architecture
  - Dynamically reconfigurable
  - Integration of data from multiple sources
  - Distributed problem solving
- Software methodology for developing, debugging, verifying and validating such an architecture
  - Modeling tools and languages
  - Automated code generation from high-level specifications
- Coupled biological processes whose dynamics change over time
  - Dealing with randomness and implementing adaptation
- Hybrid discrete/continuous processes
  - Processes distributed over space and time
  - Processes that have widely varying time constants
- Failure modes that accumulate over time
- Optimization of finite consumables
  - Adaptation of standard operating procedures
  - Automated task planning and scheduling
  - Optimizing time, space and consumables

David Kortenkamp
NASA Johnson Space Center/Metrica Inc.
Integrated Control of ALSS (cont.)

- Integrating of human expertise (ground and crew) with autonomous control
  - Mechanisms and support for humans to adjust the level of autonomy and/or change the distribution of roles and responsibilities between autonomous control and humans
  - Mechanisms for the allocation of and control of initiative among humans and the autonomous control system
  - Circumstances and methods by which the autonomous control system notifies humans of environmental events (nominal and off-nominal) and accepts task inputs from human
### ALSS Autonomy Roadmap

<table>
<thead>
<tr>
<th>Planning/Scheduling</th>
<th>Executive</th>
<th>Machine Learning</th>
<th>Model-based Reasoning</th>
<th>Sensor Interpretation</th>
<th>Distributed Control</th>
<th>Human Interaction</th>
<th>Robotics</th>
<th>Intelligent Data Understanding</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple task planning for single subsystems</td>
<td>Single system procedures</td>
<td>Parameter tuning</td>
<td>Multi-step reconfiguration</td>
<td>Sensor fusion</td>
<td>System architecture</td>
<td>Natural language discourse with control system</td>
<td>Autonomous control of Traybot</td>
<td>Data models for storing data in database</td>
<td>Simple ground demonstration</td>
</tr>
<tr>
<td>Planning for different time scales</td>
<td>Probabilistic reasoning about task context</td>
<td>Learning/refining models from system data</td>
<td>Hybrid discrete/continuous models</td>
<td>Automatic event recognition</td>
<td>Communication protocols and APIs for distributed components</td>
<td>Situation views of control system status</td>
<td>Planning and control of simulated robots</td>
<td>Use of stored data to automatically refine models and simulations</td>
<td>Ground demo with crew</td>
</tr>
<tr>
<td>Mixed-initiative and crew activity planning</td>
<td>Distributed, cooperating executives</td>
<td>Learning cross-system optimal control policies</td>
<td>Hierarchies of models for reasoning across subsystems</td>
<td>Automatic sensor calibration</td>
<td>FDIR on control system components</td>
<td>Mixed-initiative planning interfaces</td>
<td>Shared control of EVA rovers and maintenance robots</td>
<td>Automated inventory control system</td>
<td>Station flight demonstration</td>
</tr>
<tr>
<td>Crop and menu planning</td>
<td>Reasoning about procedure execution</td>
<td>Optimizing control in changing environments</td>
<td>Modeling and reasoning about software procedures</td>
<td>Interpretation of sensor nets</td>
<td>Dense networks of distributed sensors</td>
<td>Mobile computing for control system</td>
<td>Plant chamber automation for food processing</td>
<td>Automatic identification of significant events</td>
<td>Long-term station deployment</td>
</tr>
<tr>
<td>Continuous planning and replanning</td>
<td>Procedure synthesis</td>
<td>Continuous learning</td>
<td>Procedure synthesis from models</td>
<td>Vision for crop inspection and crew tracking</td>
<td>Automated recovery from major control system failures</td>
<td>Crew tracking and plan recognition</td>
<td>Test of BIO-Plex IVA maintenance and inspection robot</td>
<td>Sophisticated analysis of trends and events</td>
<td>Beyond LEO deployment</td>
</tr>
</tbody>
</table>

**David Kortenkamp**  
**NASA Johnson Space Center/Metrica Inc.**