Acquisition and Use of Internal Models of Stochastic Dynamic Systems for Human Decision-Making
(FA9550-06-1-0492)

PI: Robert Jacobs (University of Rochester)
Co-PI: David Knill (University of Rochester)

AFOSR Joint Program Review - Cognition and Decision Program and Human-System Interface Program (Jan 22-24, 2008, Arlington, VA)
# Dynamic Decision-Making (Jacobs)

**Objective:**
Develop new representations and algorithms for decision-making in dynamic environments guided by theories and data regarding human dynamic decision-making.

**DoD Benefit:**
Potential improvements in automated decision-making systems for stochastic, dynamic environments

**Technical Approach:**
- Statistical/Machine Learning methodologies
- Human experimentation

**Budget:**

<table>
<thead>
<tr>
<th></th>
<th>FY</th>
<th>FY</th>
<th>FY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual/Planned $K</td>
<td>116</td>
<td>119</td>
<td>122</td>
</tr>
</tbody>
</table>

**Annual Progress Report Submitted?**
yes

**Project End Date:**
November 30, 2008
List of Project Goals

1. Conduct mathematical analyses and computer simulations studying the acquisition and use of “global” representational primitives for dynamic decision-making.

2. Conduct mathematical analyses and computer simulations studying the acquisition and use of “local” representational primitives for dynamic decision-making.

3. Conduct experiments with people studying their acquisition of models of dynamics versus noise in decision-making environments.
Progress Towards Goals (or New Goals)

1. “Global” representational primitives:
   \( \rightarrow \) dimensionality reduction
   Articles:
   Cognitive Science conference (“best paper” award)
   *Neural Computation*

2. “Local” representational primitives:
   \( \rightarrow \) greedy additive regression
   Articles: in progress

3. Human experimentation:
   \( \rightarrow \) Near-optimal adaptation to different noise environments
   Article: *Journal of Neuroscience*
Dynamic Decision-Making

• Decision-making in environments with complex temporal dynamics
  – Decision-making at many moments in time
  – Temporal dependencies among decisions

• Examples:
  – Flying an airplane
  – Piloting a boat
  – Controlling an industrial process
  – Coordinating firefighters to fight a fire
Computational Complexity of Motor Control

- Task: Apply torques to a two-joint arm so that its endpoint moves from location $A$ to location $B$ in 100 time steps

- Assume: At each moment in time, torque at each joint can take on one of ten possible values

- Q: How many torque sequences are possible solutions?
  - A: $10^{200}$

- “Curse of dimensionality”
Motor Synergies

• Motor synergies: dependencies among degrees of freedom

• Motor synergies = motor primitives
  – Basic units of behavior that can be linearly combined to form complex units of behavior
  – To form complex behavior:
    
    only need to specify linear coefficients
Motor Synergies

- Task: Apply torques to a two-joint arm so that its endpoint moves from location $A$ to location $B$ in 100 time steps

- Without motor synergies: need to supply 200 numbers

- With $N$ synergies: need to supply $N$ linear coefficients
  - $N$ is typically a small number (e.g., 5-10)

Fig. 1. Muscle costimulation with nonredundant kinematics. *(Upper Left)* Muscle field of the Rectus Anticus (RA). *(Upper Right)* Muscle field of the Sartorius (SA). *(Lower Left)* Costimulation field. *(Lower Right)* Vector summation of RA and SA. This is one of the "worst" examples of summation in the nonredundant leg. Cos = 0.901; scaling = 0.71.
Local versus Global Strategies

- Global primitives:
  - Find solutions to tasks in training set
  - Perform dimensionality reduction (e.g., PCA) on set of solutions to obtain primitives

- Global linear combinations:
  - Find solutions to tasks in test set by linearly combining all primitives

- See Chhabra and Jacobs (2006)
Local versus Global Strategies

• Local primitives:
  – Primitives are solutions to individual tasks in training set

• Local linear combinations:
  – Find solutions to tasks in test set by linearly combining primitives via iterative procedure known as additive (sequential) regression
  – Sparse solutions: Linear combinations tend to contain very few terms
Greedy Additive Regression (GAR) Model

- Given new motor task:
  - Q: Does a linear combination of torque sequences from the library achieve good performance?
- Cost function:

\[
c(\tau; \theta^*) = \left\| \theta^* - \theta(\tau) \right\|^2
\]

- If yes, then done
GAR Model

• If not, then learn a new torque sequence
  – Feedback error learning
  – (Approximate) optimal control
  – Note: these methods are computationally expensive

• Add new torque sequence to library
  – Library sequences are solutions to individual tasks from training set
  – If library is full, remove an old torque sequence (remove sequence that is least used)
Greedy Additive Regression

- At iteration $t$

$$F^{(t)} = \sum_{j=1}^{t} \rho_j f_j$$

- Aggregate controller $F^{(t)}$ is a weighted sum of $t$ torque sequences from library
Greedy Additive Regression

- At each iteration

\[ F^{(t+1)} = F^{(t)} + \rho_{t+1} f_{t+1} \]

- A torque sequence from library is selected (with replacement)
- Weighted version of this sequence is added to \( F^{(t)} \)
Greedy Additive Regression

• Each torque sequence in library is associated with a trajectory of joint angles

• Evaluate each sequence by correlating its trajectory with the gradient of the cost function with respect to the elements of $F(t)$

• Sequence with largest correlation is selected

• Best linear coefficient for selected sequence is chosen
Greedy Additive Regression

- Relationships to other techniques:
  - Boosting, functional gradient descent
  - Matching pursuit
  - Local strategies in computer vision
    - E.g., Viola and Jones (2004)
Motor Tasks
GAR versus PCA

• GAR Model

  – Create library of torque sequences by running model on training tasks
    • 3000 tasks in training set
    • Library size was set to 100

  – Linearly combine sequences in library to perform each task in test set via greedy additive regression
    • 100 tasks in test set
GAR versus PCA

- **PCA Model**
  - Create library of torque sequences
    - Learn a sequence for each training task (via feedback error learning)
    - Perform dimensionality reduction via PCA
    - First 100 components explain 99% of variance in data
  - Linearly combine sequences in library to perform each test task via gradient descent (policy gradient)
GAR versus PCA

• GAR and PCA differ along two dimensions:
  – Local versus global primitives
  – Local versus global linear combinations
GAR Models with Libraries of Different Sizes

• If library is too small, then useful torque sequences may not be in library

• If library is too large, it may contain many sequences which are nearly never used

• Is there an optimal size?
Sparse representations
Summary

• Motor synergies = motor primitives
  – Computational motivation
  – Evidence from cognitive neuroscience

• Local versus global strategies
  – Local versus global representational primitives
  – Local versus global linear combinations

• GAR Model
Interaction with Other Groups and Organizations

• None
List of Publications Attributed to the Grant (2007-8)

