Improved Adaptive-Reinforcement Learning Control for Morphing Unmanned Air Vehicles

James Doebbler, Monish D. Tandale, John Valasek
Texas A&M University

Andrew J. Meade
Rice University

AIAA Infotech@Aerospace Conference
26-29 Sep 2005
Washington, DC
Overview

- Brief Introduction to Morphing Air Vehicles
- Adaptive-Reinforcement Learning Control Architecture
- Simplified Model of a Morphing Vehicle
- The Old Way
  - Reinforcement Learning Module
  - Structured Adaptive Model Inversion Control
  - Numerical Example
- The New Way
  - Numerical Example
- Results & Conclusions
- Future Work
Student Research Team

2005 - 2006
Advanced Concept Evolution

SOA Metal Supercritical Wing

AE Tailored Wet Composite Wing

Composite Blended Wing

Composite Wing & Fuselage VLA

Intelligent Personal Air Vehicles

Low-Boom Supersonic Transport(s)

Active Aero & Structures

Bio/Nano Self-Optimizing Aircraft

Aerospace Vehicle Systems Technology Program
Today’s Challenges:  

- Develop light, strong, and structurally efficient air vehicles.
- Improved aerodynamic efficiency.
- Design fuel-efficient, low-emission propulsion systems.
- Develop safe, fault-tolerant vehicle systems.

Technology Solutions:

- Nanostructures: 100 times stronger than steel at 1/6 the weight
- Active flow control
- Distributed propulsion
- Electric propulsion, advanced fuel cells, high-efficiency electric motors
- Integrated advanced control systems and information technology
- Central “nervous system” and adaptive vehicle control
Big Picture Research Goals

- WHEN to reconfigure
- HOW to reconfigure
- LEARNING to reconfigure
Which Morphing?

- **Morphing for Mission Adaptation**
  - Large scale, relatively slow, in-flight shape change to enable a single vehicle to perform multiple diverse mission profiles

as opposed to:

- **Morphing for Control**
  - In-flight physical or virtual shape change to achieve multiple control objectives (maneuvering, flutter suppression, load alleviation, active separation control)

John Davidson, NASA Langley, AFRL Morphing Controls Workshop – Feb 2004

Doebbler, Tandale, Valasek, & Meade

AIAA-2005-7159-7
Our Approach

- Make Full Use Of The Physical Knowledge Of The Problem
- Learn When to Reconfigure With Machine Learning

Adaptive Control

Adapting

Parameters in a Known Functional Relationship

Machine Learning

Learning

Reconfiguration Policy
Control Architecture

Reconfiguration Command Generation

Knowledge Base

System Performance Evaluation

Adaptive Controller

Control Information Distribution

Sensed Information Aggregation

Environment

Synthetic Jets for Virtual Shaping and Separation Control

MultiSensor MEMS Arrays for Flow Control Feedback

Control Architecture

Doebbler, Tandale, Valasek, & Meade

AIAA-2005-7159-9
Adaptive-Reinforcement Learning Control (A-RLC)

Conceptual Control Architecture for Reconfigurable Aircraft

SAMI
Structured Adaptive Model Inversion
(Traditional Control)

Flight controller to handle wide variation in dynamic properties due to shape change

RL
Reinforcement Learning
(Intelligent Control)

Learn the morphing dynamics and the optimal shape at every flight condition in real-time
Morphing Vehicle Model

Lockheed Martin

NextGen

Doebbler, Tandale, Valasek, & Meade

AIAA-2005-7159-11
Morphing Vehicle Evolution

<table>
<thead>
<tr>
<th>Year</th>
<th>Aircraft Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003</td>
<td>2-D Plate</td>
</tr>
<tr>
<td>2004</td>
<td>Rectangular Block</td>
</tr>
<tr>
<td>2005</td>
<td>Ellipsoid</td>
</tr>
<tr>
<td>2006</td>
<td>Delta Wing</td>
</tr>
<tr>
<td></td>
<td>Final Objective</td>
</tr>
</tbody>
</table>

Doebbler, Tandale, Valasek, & Meade

AIAA-2005-7159-12
Morphing Vehicle - TiiMY

**Shape**
- Ellipsoidal shape with varying axis dimensions.
- Constant volume (V) during morphing
- 2 independent variables: y and z, dependent dimension \( x = \frac{6V}{\pi yz} \)

**Morphing Dynamics**
- Smart material: carbon nano-tubes or shape memory alloy
- Morphing Dynamics: Simple Nonlinear Differential Equations

\[
\begin{align*}
\ddot{y} + 2y\dot{y} &= V_y \\
\ddot{z} + 3z\dot{z} &= V_z
\end{align*}
\]
Shape Morphing Animation

TiiMY
Morphing Time Histories
Optimal Shapes at Various Flight Conditions

- Optimality is defined by identifying a cost function.
  \[ J = J(\text{Current shape, Flight condition}) \]

\[ J = J_y + J_z = (y - S_y(F))^2 + (z - S_z(F))^2 \]

\[ S_y = 3 + \cos\left(\frac{\pi}{2} F\right) \quad \text{and} \quad S_z = 2 + 2e^{-0.5F} \]
6-DOF Mathematical Model for Dynamic Behavior

- Variables
  \[ p_c = [d_x \ d_y \ d_z]^T \]
  \[ v_c = [u \ v \ w]^T \]
  \[ \sigma = [\phi \ \theta \ \psi]^T \]
  \[ \omega = [p \ q \ r]^T \]

- Nonlinear 6–DOF Equations
  - Kinematic level:
    \[ \dot{p}_c = J_I \nu_c \quad \dot{\sigma} = J_a \omega \]
  - Acceleration level:
    \[ m\dot{\nu}_c + \ddot{\omega}mv_c = F + F_d \]
    \[ I\dot{\omega} + \dot{I}\omega + \ddot{\omega}I\omega = M + M_d \]

- Drag Force
  - Function of air density, square of velocity along axis, and projected area of the vehicle perpendicular to the axis

additional dynamics due to morphing
Reinforcement Learning
Reinforcement Learning

1. **Actor** takes **action** based upon states and preference function
2. **Critic** updates **state value function**, and evaluates action
3. Actor updates preference function

Learning is done repetitively, by subjecting to different scenarios
Reinforcement Learning

- Actor-critic method
  - On policy, TD method
  - The actor selects actions
    \[
    \pi_t(s, a) = \arg\max_a p(s, a)
    \]
    **Preference of an action:**
    **Greedy Policy:**
    \[
    \pi_t(s, a) = \Pr\{a_t = a | s_t = s\} = \frac{e^{p(s, a)}}{\sum_a e^{p(s, a)}}
    \]
    **Gibbs softmax policy:**
    - The critic criticize the actions
      **State value functions:**
      \[
      V(s_t) = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)
      \]
      **TD error:**
      \[
      \delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)
      \]
      - Strengthen or weaken the tendency to select one action
      \[
      p(s_t, a_t) \leftarrow p(s_t, a_t) + \beta \delta_t
      \]
Q-Learning

- Only has to learn the action value function $Q(s,a)$
  - How well agent performs action $a$ in state $s$ under policy $\pi$
- Q learning is an off-policy temporal difference method
- Proven convergence

$Q(s,a)$

$\gamma$ \quad $\alpha$ \quad $\varepsilon$

$\epsilon$-greedy policy

```
Q-Learning()
Initialize $Q(s,a)$ arbitrarily
Repeat (for each episode):
    Initialize $s$
    Repeat (for each step of episode):
        Choose $a$ from $s$ using policy derived from $Q(s,a)$
        // (e.g., $\epsilon$-greedy policy)
        Take action $a$, observe $r, s'$
        $Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
        $s \leftarrow s'$
    until $s$ is terminal
return $Q(s,a)$
```
Function Approximation
K Nearest Neighbor Policy Iteration

- The shape of the vehicle is on continuous domains
- Use K-nearest neighbors method to approximate the action-value function $Q(s,a)$
  - Collect a set of state-action pair samples
  - **Compute optimal action-values of these samples using Q-learning**
  - The action-value of a new state-action pair is the interpolation of those of its K nearest neighbors.
  - Takes a weighted average of the K nearest neighbors in the sampled state space
- Susceptible to absence of accurate information near the desired point

<table>
<thead>
<tr>
<th>KNNPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect Sample</td>
</tr>
<tr>
<td>Learn $Q_{sample}(s_i,a_i)$</td>
</tr>
<tr>
<td>For $Q(s_0,a_0)$, find ${s_i}$, the set of K nearest neighboring $s$ to $s_0$</td>
</tr>
<tr>
<td>$Q(s_0,a_0) = \sum_{n=1}^{K} \frac{Q_{sample}(s_i,a_i)}{Distance(s_i,a_i,s_0,a_0)}$</td>
</tr>
</tbody>
</table>
Structured Adaptive Model Inversion Control
Adaptive Control

model reference adaptive control

Reference model

Controller

Plant

Estimated parameters

Adaptation Law

$r$ $u$ $y_m$

$y$ $e$

$y$ $e$

Estimated parameters $ightarrow$ Adaptation Law

Controller

Plant

Reference model

$y_m$

$e$
Structured Model Reference Adaptive Control

Akella, Schaub, Junkins (Texas A&M)

Dynamics

2\textsuperscript{nd} order differential equations

\[ \dot{x} = v \]

\[ F = ma \]

\[ \dot{v} = a = \frac{F}{m} \]

Exact kinematic relationship between position and velocity

Acceleration level relationships between forces and system parameters

Doebbler, Tandale, Valasek, & Meade

AIAA-2005-7159-25
Structured Adaptive Model Inversion

Subbarao, Junkins (Texas A&M)

Features

- Dynamic inversion inner-loop, with an MRAC outer-loop to handle system uncertainties.
- Controls are solved for explicitly:

\[ \dot{x} = Af(x) + Bu \]

Reference Trajectory \( x_r, \dot{x}_r \)

Control Law \( u = B^{-1}(\dot{x}_r - Af(x) - \lambda e) \) so that the error dynamics becomes \( \dot{e} = -\lambda e \)

- Undesirable dynamics are cancelled and replaced with user specified desired dynamics.
- Easily applicable to nonlinear systems.
- Error dynamics can be specified.
- Shown to be very effective for a wide variety of systems.
Structured Adaptive Model Inversion Features

Trajectory Tracking for Dynamic Systems

- **Plant:**
  - Nonlinear in states, affine in control, uncertain parameters appear linearly.

- **Control:**
  - Dynamic Inversion and Sliding Mode Control.
  - Dynamic Inversion requires knowledge of system parameters, which are inherently uncertain.

- **Adaptive Learning Parameters:**
  - Updated in real-time, and used for the Dynamic Inversion

- **Adaptation Mechanism:**
  - Driven by the error between the actual plant trajectory and the reference trajectory

- **Stability Analysis:**
  - Guarantees that the plant trajectory, asymptotically converges to the reference trajectory in the presence of Parametric Uncertainties, and initial condition errors.
Structured Model & Minimal Parameterization

Kinematic Level Model

\[ \dot{p}_c = J_I v_c \]
\[ \dot{\sigma} = J_a \omega \]

Acceleration level Model

\[ m \dot{v}_c + \tilde{\sigma} m v_c = F + F_d \]
\[ I \dot{\omega} + \dot{\tilde{\omega}} I \omega + \tilde{\omega} I \omega = M + M_d \]

Attitude Control

\[ I_a^* (\sigma) \ddot{\sigma} + C_a^* (\sigma, \dot{\sigma}) \dot{\sigma} = P_a^T (\sigma) M \]

Minimal Parametrization of the Inertia Matrix

\[
\begin{bmatrix}
I_{11} & I_{12} & I_{13} \\
I_{12} & I_{22} & I_{23} \\
I_{13} & I_{23} & I_{33}
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
a_3
\end{bmatrix} =
\begin{bmatrix}
a_1 & 0 & 0 & a_2 & a_3 & 0 \\
0 & a_2 & 0 & a_1 & 0 & a_3 \\
0 & 0 & a_3 & 0 & a_1 & a_2
\end{bmatrix}
\begin{bmatrix}
I_{11} \\
I_{22} \\
I_{33} \\
I_{12} \\
I_{13} \\
I_{23}
\end{bmatrix}
\]
Control Law & Update Law

Using Minimal Parametrization of the Inertia Matrix

\[ I^*_a(\sigma)\ddot{\sigma} + C^*_a(\sigma, \dot{\sigma})\dot{\sigma} = Y_a(\sigma, \dot{\sigma}, \ddot{\sigma})\dot{\theta} \]

unknown parameters

With the control law

\[ M = P_a^{-T}\{Y_a(\sigma, \dot{\sigma}, \dot{\sigma}_r, \ddot{\sigma}_r)\dot{\theta} - C_{da} \dot{\varepsilon} - K_{da} \varepsilon\} \]

the closed loop dynamics take the form

\[ I^*_a\dddot{\varepsilon} + \{C_{da} + C^*_a(\sigma, \dot{\sigma})\} \dot{\varepsilon} + K_{da} \varepsilon = Y_a(\sigma, \dot{\sigma}, \ddot{\sigma})\tilde{\theta} \]

and, along the adaptive law

\[ \dot{\theta} = -\Gamma Y_a(\sigma, \dot{\sigma}, \dot{\sigma}_r, \ddot{\sigma}_r)^T \dot{\varepsilon} \]

guarantees asymptotic stability of the tracking errors
Adaptive–Reinforcement Learning Control
Numerical Example

- **Objective**
  - Demonstrate *optimal* shape morphing for *multiple* specified *flight conditions*

- **Method**
  - For every flight condition, **learn optimal policy** that commands voltage producing the optimal shape
  - **Minimize total cost** over the entire flight trajectory
  - Evaluate the learning performance after 200 learning episodes

---

**RL Module is Completely Ignorant of Optimality Relations and Morphing Control Functions:**

*It Must Learn On Its Own, From Scratch*
Numerical Example

reinforcement learning definitions

- **Agent**: Morphing Air Vehicle Reinforcement Learning Module

- **Environment**: Various flight conditions

- **Goal**: Fly in optimal shape that minimizes cost

- **States**: Flight condition; shape of vehicle

- **Actions**: Discrete voltages applied to change shape of vehicle
  - **Action set**: \( A = \{(V_y, V_z) \mid [0, 0.5, 1, \ldots , 5]; [0, 0.5, 1, \ldots , 4] \}\)

- **Rewards**: Determined by cost functions

- **Optimal control policy**: Mapping of the state to the voltage leading to the optimal shape
Learning Process

Error in the Action Preference Function after:

- 20 Episodes
- 60 Episodes
- 100 Episodes
- 200 Episodes
Comparison of True Optimal Shape and Learned Shape

KNN learns poorly for several flight conditions
Time Histories of Angular States

\[
\begin{align*}
\phi \quad &\text{(deg)} \\
\theta \quad &\text{(deg)} \\
\psi \quad &\text{(deg)} \\
\rho \quad &\text{(deg/sec)} \\
\theta \quad &\text{(deg)} \\
\rho \quad &\text{(deg/sec)} \\
\end{align*}
\]

\[0 \quad 50 \quad 100 \quad 150 \quad 200\]

\[100 \quad 0 \quad -100\]

\[2 \quad 1 \quad 0 \quad -1 \quad -2\]

\[1 \quad 0 \quad -1\]

\[0.5 \quad 0 \quad -0.5 \quad -1\]

\[0.5 \quad 0 \quad -0.5 \quad -1\]
Time Histories of Adaptive Parameters

**Graph 1:**
- **Y-axis:** Norm of Inertia Vector (||\textit{inertia vector}||)
- **X-axis:** Time (sec)
- **Legend:**
  - Blue line: Estimated
  - Red dashed line: True

**Graph 2:**
- **Y-axis:** Mass
- **X-axis:** Time (sec)

Trajectory Tracking Controls

\begin{align*}
T_x (N) & \quad M_x (Nm) \\
T_y (N) & \quad M_y (Nm) \\
T_z (N) & \quad M_z (Nm)
\end{align*}

\begin{align*}
\text{Time (sec)} & \quad \text{Time (sec)}
\end{align*}
Morphing Control Voltages

Voltage - V_y

Voltage - V_z

Time (sec)
What Happened? (1)

- F Still Equals d(mv)/dt
  - Stiffness, frequency, damping ratio, and cost function have a major effect on learning times and learning performance.
  - **Slowest element in system is critical**

- Some Assembly Required: Tuning
  - Flight condition transitions, morphing dynamics, learning performance, and adaptive control must be balanced to achieve good performance.
  - **Coordination and timing is everything**
What Happened? (2)

- Function Approximation
  - Errors remained which could not be eliminated with additional training.

- Use **Galerkin-based Sequential Function Approximation (SFA)** to approximate the action-value function $Q(s,a)$
Galerkin-Based Sequential Function Approximation (SFA)

- Ideal for approximating sparse multi-dimensional scattered data
- Adaptive, no ad-hoc user parameters to adjust
- Provides information on the sensitivity to the inputs
- Matrix construction and evaluations are avoided
- Computational cost is reduced while efficiency is enhanced by the low-dimensional unconstrained optimization
Example: New Way

- SFA learns optimal shape well
## Results

Normalised RMS error

<table>
<thead>
<tr>
<th></th>
<th>Y dimension</th>
<th>Z dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>K N N</td>
<td>1.42</td>
<td>0.821</td>
</tr>
<tr>
<td>S F A</td>
<td>1.27</td>
<td>0.661</td>
</tr>
</tbody>
</table>

10% reduction 20% reduction
Conclusions

- Improved function approximation provided large improvement in performance
  - Galerkin-based Sequential Function Approximation method learns optimal shape well

- Shape Changes for Mission Morphing can be treated as piecewise constant parameter changes
  - SAMI is a favorable method for trajectory tracking control

- Morphing for Control will require different control strategy
  - Piecewise constant approximation no longer valid
Future Research 1

- Modify the morphing dynamics to represent SMA actuators.
  - Hysteretic behavior

- Investigate control methodologies to handle faster shape changes
  - Linear Parameter Varying (LPV) control

- Investigate other function approximation methods
  - Radial Basis Functions (GLO-MAP)

- Modify the simulation to include a more advanced aircraft model
  - Wing-Body, Wing-Body-Empennage, etc.
Future Research 2

Morphing Space Based Antenna

- **STATE OF THE ART:**
  - Multiple antennas must be mounted on spacecraft to accommodate various ground station signals

- **RESEARCH GOALS:**
  Demonstrate feasibility of a reconfigurable antenna design that utilizes Reinforcement Learning to independently achieve optimal shape through use of SMA actuators

- **BENEFITS:**
  A single antenna capable of altering its geometry to achieve world-wide compatibility between receivers and transmitters
Future Research 3

- Directly learn voltage inputs required to achieve certain position states with time dependency.
  - Skips mathematical modeling and computer simulation steps
  - Avoids modeling and simulation errors
Questions?
Parameterized functional form

\[ V^\pi_t(s) = \sum_{j=1}^{N} \theta_{vtj} \Phi_j(s) \]

\[ p_{at}(s) = \sum_{j=1}^{N} \theta_{patj} \Phi_j(s) \]

Gradient descent learning

\[ \theta_{vt} = \theta_{vt-1} + \alpha(\tilde{y} - H_t \theta_{vt-1})H_t^T = \theta_{vt-1} + \alpha \delta_t H_t^T \]

The Temporal Displacement error \( \delta_t \) drives all learning in both actor and critic

\[ \theta_{pat} = \theta_{pat-1} + \alpha(p_{at}(s_t) + \beta \delta_t - H_t \theta_{pat-1})H_t^T = \theta_{pat-1} + \alpha \beta \delta_t H_t^T \]