A Morphogenetically Assisted Design Variation Tool

Aaron Adler, Fusun Yaman, Jacob Beal, Jeffrey Cleveland, Hala Mostafa
BBN Technologies, Cambridge, MA
aadler. fusun, jakebeal, jcleveland, hmostafa @bbn.com

Annan Mozeika
iRobot Corporation, Bedford, MA
amozeika@irobot.com

Work sponsored by DARPA; the views and conclusions contained in this document are those of the authors and not DARPA or the U.S. Government.
Brittleness of Electromechanical Systems

Even “simple” robots require careful design of many interacting components…

Once a system is constructed, difficult to modify design
Design Variation Tool

• Goal: design adaptability in engineered systems. When a design element is modified, the rest of the design automatically adjusts to compensate.
  – Enable users who are novices at electromechanical design to create functional design variants

• Idea: inspired by biology where animals adapt gracefully as they grow using feedback loops to make changes that maintain the integration of the organism as a whole
Functional Blueprints: Stress Functions

• Idea: keep the design always working, navigate through viable space

• Stress functions define viable and non-viable space

Copyright 2013 Raytheon BBN Technologies
Functional Blueprints (FBs)

Inspired by regulatory processes in natural morphogenesis

FB requirements:

• Initial viable design
  – always maintain design viability

• Behavior that degrades gracefully
  – detect when design elements start to become problematic

• Metric for stress degree & direction
  – Stress used as a coordination signal among different subsystems

• Incremental adjustment program

Examples:

Step Climbing (via ascent angle)

Flipper/Body Ratio

Self-Righting (via torque/mass)
Types of Functional Blueprints

- **Closed-Form FBs**: stress evaluated analytically using equations provided by domain expert
  - FB constraining robot body’s length-to-width ratio

- **Quantized-Component FBs**: for choosing from libraries of fixed component variants
  - *Servo* FB constrains the torque and mass of servo motors

- **Simulation-Driven FBs**: for evaluating complex non-closed-form functions
  - *Climb* FB maintains a miniDroid’s ability to climb and is evaluated through a ROS simulation

- **User-Command FBs**: temporary FBs that incrementally shifts a design or evaluator parameter towards the specified value
Evaluators measure how well a design accomplishes its goals. Input: design parameters. Output: evaluation metrics

- FBs blend evaluation metrics from multiple evaluators and convert them to stresses.
- FBs incremental programs calculate update for design parameters, resulting in a new design variant that is still viable.
miniDroid Case Study

Original miniDroid climbs over a 10cm step

Goal: a variant to climb over a 55cm step

- Selected subset of critical miniDroid functions
  - e.g., climbing over obstacles, flipping over
- Identified 7 key design parameters affecting these functions
- 8 FBs for evaluating these functions
miniDroid Case Study

Stress Inputs

- Climb Angle
- Step Height
- Body Length
- Body Width
- Flipper Length
- Motor Torque
- Motor Mass

FBs

- User-1
- Climb
- Proportion
- Body Shape
- Flipping
- Servo-Torque
- Servo-Mass
- Servo-MT

Simulation computed metric
Environment variable
Model parameter

Simulation out
FB stress in
FB update out

Copyright 2013 Raytheon BBN Technologies
MADV achieves goal in 75 design iterations
All intermediate designs are functional

Stress for each FB during 5.5x variation

Parameter changes during 5.5x variation
Generalizability and Scalability

- Scalability: more detailed miniDroid model developed in 3 hours: 112 parameters and 111 FBs
- Generated random networks of abstract FBs and design attributes:

![Graphs showing stress, convergence time, and partial stress for different perturbation sizes and number of iterations.](image)
Comparison vs. Genetic Algorithms

- Two GA fitness functions, max 1000 iterations:
  - FF1: function of distance between desired and initial solution and sum of constraint violations
  - FF2: functions of stress inputs used by FBs

- Results:
  - FBs more focused, fewer iterations, closer to initial design, lower final stress
  - Solved problems 280x faster, 100% convergence rate
  - FBs explore less of the design space

<table>
<thead>
<tr>
<th>Perturbation size 1</th>
<th>Initial - Final</th>
<th># Iterations</th>
<th>Time (s)</th>
<th>Stress</th>
<th>Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Blueprints</td>
<td>22</td>
<td>310</td>
<td>0.01</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>GA w/o stress (FF1), pop 50</td>
<td>48</td>
<td>756</td>
<td>0.12</td>
<td>0.65</td>
<td>35%</td>
</tr>
<tr>
<td>GA w/ stress (FF2), pop 50</td>
<td>80</td>
<td>714</td>
<td>0.52</td>
<td>0.2</td>
<td>40%</td>
</tr>
<tr>
<td>GA w/o stress (FF1), pop 500</td>
<td>55</td>
<td>714</td>
<td>2.65</td>
<td>0.14</td>
<td>40%</td>
</tr>
<tr>
<td>GA w/ stress (FF2), pop 500</td>
<td>88</td>
<td>323</td>
<td>2.8</td>
<td>0.05</td>
<td>85%</td>
</tr>
</tbody>
</table>
Usability & Extensibility

Our implementation of the MADV architecture is reusable and extensible

• Code base is highly modular
• FBs, parameters, and evaluator settings are specified in XML files
• Evaluators are Java objects allowing problem-specific evaluation (e.g., via RPC to rigid body simulators)
• Reused blueprints for Mars lander
Summary

• Design variant tool created that uses Functional Blueprints
• miniDroid case study: 5.5x variant
• Functional blueprints are generalizable, scalable, and reusable
  – Random FB networks
  – Mars lander
Project Team:

Raytheon BBN Technologies
Jacob Beal (PI)
Aaron Adler (co-PI)
Susan Katz
Brett Benyo
Jeff Cleveland
Jessica Lowell
Hala Mostafa
Kyle Usbeck
Fusun Yaman

Interns:
• Katie McGuire
• Taylor Campbell

iRobot
Annan Mozeika
Benjamin Axelrod

Intern:
• Gretchen Markiewicz

Team website:
http://madv.bbn.com
Code posted