Autonomous Reconfiguration Planning In Modular Robots

Rowan McAllister
Supervisor: Dr Robert Fitch

Quick Demo
Native Kinematics: A 3-R Module

ACFR design: (picture courtesy Jin Jin)

3 Degrees of Freedom

The Challenge: The State Space is Large

An Example Robot

- No. modules (Superbot): 10
- degrees of freedom: 30
- actions available: 120
- No. possible states: > 3.5 x 10^{24}

State-Network Concept

State A -> State B

- Rowan McAlistier -
The Challenge: The State Space is Large

An Example Robot
- No. modules (Superbot): 10
- degrees of freedom: 30
- actions available: 120
- No. possible states: $> 3.5 \times 10^{24}$

Solution: Hierarchical Planning

---

Rowan McAllister
The Challenge: The State Space is Large

An Example Robot

- No. modules (Superbot): 10
- degrees of freedom: 30
- actions available: 120
- No. possible states: $> 3.5 \times 10^{24}$

Solution: Hierarchical Planning
The Challenge: The State Space is Large

An Example Robot

- No. modules (Superbot): 10
- degrees of freedom: 30
- actions available: 120
- No. possible states: $> 3.5 \times 10^{24}$

State A

Solution: Hierarchical Planning

State B

State A

- Rowan McAllister

State B

3
Algorithm

Upper Level: Selects modules to relocate someplace

Lower Level: Solves relocation optimally; a search in native-kinematic space using distributed dynamic programming

Goal: Transition to given module-connector pair

Exploits:
- Module Symmetry
- Lattice Structure

Routines Include:
transition (state, action) → (State) newState
collisionDetection (state, action) → (Boolean) willCollide

Challenge: Stuck States

Problem:
Detachable module cannot reach next available connector

Connectivity Concept:
Detachable
Modules cannot detach; this would disconnect structure
Challenge: Stuck States

Solution:
Detachable module asks another module for help

Connectivity Concept:
Detachable
Modules cannot detach; this would disconnect structure

Module State

State (or ‘pose’)
1) Position

Connected part’s position (X, Y, Z)
Module State

<table>
<thead>
<tr>
<th>State</th>
<th>(or ‘pose’)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Position</td>
<td>Connected part’s position ( (X, Y, Z) )</td>
</tr>
<tr>
<td>2) Direction</td>
<td>(+/-\ {X, Y, Z})</td>
</tr>
<tr>
<td>3) DOF angular positions</td>
<td>(2 \times {-90, 0, 90}) deg. + (1 \times {0, 90, 180, 270}) deg.</td>
</tr>
</tbody>
</table>
Module State

**State** (or ‘pose’)
1) **Position**  
   Connected part’s position \( (X, Y, Z) \)
2) **Direction**  
   \(+/- (X, Y, Z)\)
3) **DOF angular positions**  
   \(2 \times [-90, 0, 90] \) deg.  \(+\ 1 \times (0, 90, 180, 270) \) deg.

**SuperState** (Abstraction for cooperation between mobile and helper modules)
1) Mobile-module state
2) Helper-module state
Transition Model

\[ T(s, a) = s' \]

\( s \in S \) (set of all SuperStates)
\[ \text{size}(S) \leq 108 \times (\# \text{ connectable faces of static modules}) \]
\[ + 31104 \times (\# \text{ helper modules}) \]

\( a \in A \) (set of all Actions)
\[ \text{size}(A) = 17 \]
Future Work

Apply reconfiguration program to ‘Roombot’ modules. These are manufactured by our international collaborators at EPFL (Switzerland)