Reconfigurable Robots, Part II

March 4, 2003

Class Meeting 15
Announcements

• Assignment #4 due this Thursday (March 6)

• Exam #2 next Thursday (March 13)
  – Covers material from Feb. 11 through March 6
Recall Last Time: Chain-Type Reconfigurable Robots
Today: Lattice-Type Reconfigurable Robots
Characteristics of Lattice Robotics

- Can aggregate with identical modules into a variety of shapes
- All robots identical ("unit modular")
- Challenge: motion planning and coordination
Example of Lattice-Type Reconfigurable Robots: Robotic Molecule: Rus (Dartmouth)

• The Molecule:
  – 4 degrees-of-freedom
  – Can aggregate with other identical modules to form 3D dynamic structures.
Movies of Crystalline Robot Simulation
Rus et al., Dartmouth

See also web site: http://www.cs.dartmouth.edu/~robotlab/robotlab/robots/crystal/version2.html

The Crystalline Module: Atoms and Bonds

Presented by Matthew Aldridge
Primitive Operations for Crystal Modules

- (expand <atom> <dimension>)
- (contract <atom> <dimension>)
- (bond <atom> <dimension>)
- (free <atom> <dimension>)
Definition of “Self-Reconfigurable”

• Theorem 1: Unit-modular robotic system is self-reconfiguring if unit-modules have following properties:

  – Structure formation: Groups of unit modules can be assembled into arbitrarily shaped rigid structures.

  – Module relocation: In every structure composed of unit modules, some unit module can be relocated to each location on the surface of the structure without human intervention.
Relocating Module on a Crystal

- Unlike many reconfigurable robot systems, Crystalline module can move through volume:
Relocating Modules

- **Scrunch**: two adjacent and connected Atoms contracted in dimension normal to their connected face

- **Axis**: connected string of 2 or more Atoms along one dimension.

- **Atom Connectivity Graph (ACG(C))**: undirected graph whose vertices represent Atoms and edges represent bonded inter-Atomic interfaces

- **Theorem**: If an axis contains a scrunch, that scrunch can be moved to any position on axis by inchworm propagation. If intersecting axes contains scrunch, that scrunch can be transferred to other axis, provided there is sufficient structure to maintain connectedness.
• C: Crystal
• S: Start location on surface of C
• G: goal location on surface of C

• Atom Relocation Algorithm:
  – Build ACG(C)
  – Find path $p$ from S to G in ACG(C)
  – Create scrunch s along supporting axis for S by pulling in Atom that was in the starting location
  – Drive s along $p$ using transitions from each segment of the path to the next
  – Relax s along the supporting axis for G, popping an Atom out into G
Movies of Atom Relocation
Planning for Shape Metamorphosis

• Grain Crystals:
  – Grain(n) contains all Crystals that can be tiled by cubic blocks of Atoms of side-length n, so that the set of planes that coincide with all sides of all blocks intersect only at block edges and corners

• Grain motion primitives:
  – (scrunch <grain> <dimension> <sense>): create planar compression at one of faces
  – (relax <grain> <dimension> <sense>): expand a compression at one face of grain
  – (transfer <grain> <dimension> <sense>): move compression at one face of a grain into adjacent neighbor grain
  – (propagate <grain> <dimension> <sense>): move compression at one face of grain to opposing face of grain
  – (convert <grain> <dimension1> <sense1> <dimension2> <sense2>): relocate a compression at one face of a grain to one of the orthogonal faces of that grain
Melt-Grow Planner for Shape Metamorphosis

• S is staring crystal
• G is goal crystal
• I is intermediate crystal

• Melt-Grow:
  – Melt S into I
  – Grow G out of I
Melt-Grow Planner Details

- **Mobile grain**: a grain that can be removed without disconnecting the crystal

- **Grain Connectivity Graph (GCG(C))**: undirected graph whose vertices represent grains and whose edges represent active connections between neighboring grains

- **Parent grain**: grains adjacent to yet-to-be-filled vacancies in I (if melting) or in G (if growing)
“Paraphrased” Melt-Grow Planner

• Locate “stem” to grow I

• While still melting:
  – Select mobile grain (still in S)
  – Select parent grain (in I)
  – Move mobile grain to parent grain

• While still growing:
  – Select mobile grain (still in I)
  – Select parent grain (in G)
  – Move mobile grain to parent grain
Another Look at the Reconfiguration
Summary of Self-Reconfigurable Robots

• 4 primary types:
  – Chain-type modules
  – Lattice modules
  – Connectable large-scale robots
  – Modular plug-and-play robots

• Algorithms for self-reconfigurability
  – CONRO hormone-based reconfiguration
  – Melt-Grow planning for compressible unit modules

• Be sure you understand the details of how these reconfiguration algorithms work!
Next Time…

• Topic:
  – Multi-Robot Path Planning / Traffic Management