Robotic Assembly
by
Qualitative Contact Models

Speaker: Fan Yang
10/24/2003
Contents

- Introduction
- Discrete event controller
- Motion planner
- Robot controller
- Conclusions and future work
Automatic Assembly

- Position-based control
  - Easy to program
  - Expensive for precision position control
- Compliant motion control
  - Reduce the uncertainty
  - Requires less accurate and less expensive equipment
  - Hard to program
Automatic Assembly System
Discrete Event Controller

- Representation
- Hypotheses generation
- Hypotheses verification
- Spatial adjacency
- Contact information extraction
- Planning
Qualitative Models

- Qualitative state adjacency graph
  - Expression: $G = (\Gamma, T)$
  - $\Gamma$ : space of qualitative configuration representations
  - $T$ : set of arcs connecting adjacent configuration states

- Augmented qualitative state adjacency graph
  - Expression: $G_a = (\Gamma, T, T_d)$
  - $T_d$ : set of all desired transitions
Sets of Elementary Contacts
– Contact Between Two Polyhedra

- A contact description is a set of elementary face-vertex, vertex-face and edge-edge contacts:

$$\gamma_{A,B} = \{f_i^A - v_j^B, ..., v_k^A - f_l^B, ..., e_m^A - e_n^B\}$$

The set of all syntactically correct contact description is denoted by $\Gamma_{A,B}$.

- A qualitative contact state is a geometrically feasible contact description $\gamma_{A,B}$. The set of all contact states is denoted by $\Gamma_{A,B}^c \subset \Gamma_{A,B}$.
Feature Interaction Matrix

- The convex decomposition of A and B are denoted as:
  \[ cv(A) = \{ A_1, A_2, \ldots, A_{m_{cv}}^{A} \} \quad cv(B) = \{ B_1, B_2, \ldots, B_{m_{cv}}^{B} \} \]

- Feature Interaction Matrix:

\[
A_{B} \Phi = \begin{bmatrix}
A_{1}B_{1} \Phi & A_{1}B_{2} \Phi & \cdots & A_{1}B_{m_{cv}}^{B} \Phi \\
A_{2}B_{1} \Phi & A_{2}B_{2} \Phi & \cdots & A_{2}B_{m_{cv}}^{B} \Phi \\
\vdots & \vdots & \ddots & \vdots \\
A_{m_{cv}}^{A}B_{1} \Phi & A_{m_{cv}}^{A}B_{2} \Phi & \cdots & A_{m_{cv}}^{A}B_{m_{cv}}^{B} \Phi
\end{bmatrix}
\]

\[
A_{B}J \Phi = \begin{bmatrix}
A_{B}J \Phi^{yf} & 0 & 0 \\
0 & A_{B}J \Phi^{fv} & 0 \\
0 & 0 & A_{B}J \Phi^{ee}
\end{bmatrix}
\]
Feature Interactions
- Vertex-Face Interaction

The interactions between all vertices of A and all faces of B is given by a vertex-face interaction matrix:

\[
\begin{pmatrix}
\phi_{A1}^{vf} & \phi_{A2}^{vf} & \ldots & \phi_{Am_A}^{vf} \\
\phi_{B1}^{vf} & \phi_{B2}^{vf} & \ldots & \phi_{Bm_B}^{vf} \\
\phi_{21}^{vf} & \phi_{22}^{vf} & \ldots & \phi_{2m_B}^{vf} \\
\vdots & \vdots & \ddots & \vdots \\
\phi_{m_A1}^{vf} & \phi_{m_A2}^{vf} & \ldots & \phi_{m_Am_B}^{vf}
\end{pmatrix}
\]

\[
\phi_{ij}^{vf} = \text{sign} \left( h_{ij}^{vf} \left( R_1^0, r_1^0 \right) \right) = \text{sign} \left( d_s^E \left( f_j^B, v_i^A \right) \right)
\]
Feature Interactions (cont’)
- Face-Vertex Interaction

- The interactions between all faces of A and all vertices of B is given by a face-vertex interaction matrix:

\[
A^B_{\Phi_{ji}^{fv}} = \left[ A^B_{\Phi_{ji}^{fv}} \right] = \\
\begin{array}{cccc}
\Phi_{j_1}^A & A^B_{\Phi_{j_1}^{fv}} & A^B_{\Phi_{j_2}^{fv}} & \cdots & A^B_{\Phi_{j_m}^{fv}} \\
\Phi_{j_2}^A & A^B_{\Phi_{j_1}^{fv}} & A^B_{\Phi_{j_2}^{fv}} & \cdots & A^B_{\Phi_{j_m}^{fv}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\Phi_{j_m}^A & A^B_{\Phi_{j_1}^{fv}} & A^B_{\Phi_{j_2}^{fv}} & \cdots & A^B_{\Phi_{j_m}^{fv}} \\
\end{array}
\]

\[
A^B_{\Phi_{ji}^{fv}} = \text{sign} \left( A^B_{h_{ji}^{fv}} (R_1^0, r_1^0) \right) = \text{sign} \left( d^E_s (v_i^B, f_j^A) \right)
\]
Feature Interactions (cont')
- Edge-Edge Interaction

- The interactions between all edges of A and all edges of B is given by a edge-edge interaction matrix:

\[
A^B \Phi_{ee} = \begin{bmatrix}
A^B \phi_{e1} \\
A^B \phi_{e2} \\
\vdots \\
A^B \phi_{en_b}
\end{bmatrix} =
\begin{bmatrix}
e^A_1 & e^B_1 & \cdots & e^B_{m_b} \\
e^A_2 & e^B_2 & \cdots & e^B_{m_b} \\
\vdots & \vdots & \ddots & \vdots \\
e^A_{m_a} & e^B_{m_a} & \cdots & e^B_{m_b}
\end{bmatrix}
\]

\[
A^B \phi_{kl} = \text{sign} (A^B h_{kl}^{ee} (R^0_1, r^0_1)) = \text{sign} ((\tilde{u}^0_{ek} u^0_{el})^t (r^0_{v_k l}, v^0_{v_k l}))
\]
Uniqueness: Many-to-One Mapping

- Let $C$ denote the configuration space of $A$.
- Let $\Gamma = \{A^B \Phi\}$ denote the set of all feature interaction matrices.
- A configuration $q = (r_1^0, r_i^0) \in C$ of $A$ corresponds to a feature interaction matrix $A^B \Phi \in \Gamma$.
- The mapping $g : C \to \Gamma$ is defined as: $A^B \Phi = g(q)$ where each element $A^B \Phi_{kl} = g_{kl}(q)$ is computed as:

$$g_{kl}(q) = \begin{cases}  
    A^B \Phi^f_{kl}(q) & \text{if } A^B \phi^f_{kl} \text{ corresponds to a vertex-face interaction} \\
    A^B \Phi^v_{kl}(q) & \text{if } A^B \phi^v_{kl} \text{ corresponds to a face-vertex interaction} \\
    A^B \Phi^{ef}_{kl}(q) & \text{if } A^B \phi^{ef}_{kl} \text{ corresponds to an edge-edge interaction} \\
    0 & \text{otherwise}
\end{cases}$$
Hypotheses Generation

- Hypotheses about less constrained neighbors of a configuration state $^{AB} \Phi_i$ can be generated by changing one or more zero elements of $^{AB} \Phi_j$ to $\pm 1$.
- Hypotheses about more constrained neighbors of a configuration state $^{AB} \Phi_i$ can be generated by changing one or more nonzero elements of $^{AB} \Phi_j$ to 0.
- Using local penetration information to rule out infeasible hypotheses in an early stage.
Hypotheses Verification

Problem statement: Given a hypothetical, arbitrary feature interaction matrix, does there exist a configuration of the objects for which

- the kinematic constraints imposed by the elements of the feature interaction matrix are satisfied
- the parts do not penetrate?
Falsify Hypotheses
- Penetration Information

Three different types of penetration exist. In Panel (a) vertex $v_1^A$ is inside object $B$. In Panel (b) the objects penetrate each other while vertex $v_1^A$ contacts face $f_1^B$. Panel (c) shows that the objects penetrate each other if an edge intersects with a face of the other object.

**Figure 8.1.** Three types of penetration of polyhedral objects.
Falsify Hypotheses
- Penetration Information (cont’)

- Vertex inside object
  Objects A and B penetrate each other if one of the vertex-face interaction matrices contains a row with only $-1$ entries or one of the face-vertex interaction matrices contains a column with only $-1$ entries:

$$(\exists i)(\exists j)(\exists k)(\forall l)\left( A_{iB_j} \phi_{kl}^{vf} = -1 \right)$$

$$(\exists i)(\exists j)(\exists l)(\forall k)\left( A_{iB_j} \phi_{kl}^{fv} = -1 \right)$$
Falsify Hypotheses
- Penetration Information (cont’)

- Vertex contacting a face

Objects A and B do penetrate each other if there exist i, j, k, l such that the following three conditions are satisfied:

\[ A_i B_j \phi_{kl}^{vf} = 0 \]

\[ (\forall f_{m}^{B_j} \in \text{adj}(f_{l}^{B_j}))(A_i B_j \phi_{km}^{vf} = -1) \]

\[ (\exists v_{n}^{A_i} \in \text{cobound}(v_{k}^{A_j}))(A_i B_j \phi_{nl}^{vf} = -1) \]
Falsify Hypotheses
- Penetration Information (cont’)

- Edge intersecting a face
  Let edge $e_k^{A_i}$ be bounded by vertices $v_{k_1}^{A_i}$ and $v_{k_2}^{A_i}$. Edge $e_k^{A_i}$ intersects face $f_l^{B_j}$ if the following two conditions are satisfied:

$$A_i B_j \phi_{k_1 l}^{vf} \times A_i B_j \phi_{k_2 l}^{vf} = -1$$

$$(\forall f_m^{B_j} \in \text{adj}(f_l^{B_j}))(A_i B_j \phi_{k_1 m}^{vf} = A_i B_j \phi_{k_2 m}^{vf} = -1)$$
Criteria for Geometric Feasibility – Non-penetration Condition

\[ g(A_k, B_l) = \min \sigma \]

subject to \[ A_k(\sigma) \cap B_l(\sigma) \neq 0 \]

\[ d^G_p(A_k, B_l) = \begin{cases} 
(\rho_A + \rho_B)(1 - g(A_k, B_l)) & g(A_k, B_l) \leq 1 \\
0 & g(A_k, B_l) > 1 
\end{cases} \]

\[ d^G_p(A, B) = \sum_{A_k \in cv(A)} \left[ \sum_{B_l \in cv(B)} d^G_p(A_k, B_l) \right] = 0 \]
Falsify Hypotheses
- Optimization Based Method

\[
\begin{align*}
\min \ G &= d_p^G (A, B) \\
\text{subject to :} \\
^{AB} h_{rs} (R_1^0, r_1^0) &= 0 \quad \forall (r, s) \in ZEROS \quad (^{AB} \Phi) \\
^{AB} h_{rs} (R_1^0, r_1^0) &> 0 \quad \forall (r, s) \in ONES \quad (^{AB} \Phi) \\
^{AB} h_{rs} (R_1^0, r_1^0) &< 0 \quad \forall (r, s) \in MINUS - ONES \quad (^{AB} \Phi)
\end{align*}
\]
Spatial Adjacency - Necessary Condition

Let $A^B\Phi_i$ and $A^B\Phi_j$ be two spatially adjacent qualitative configurations and let matrix $A^B\Phi_i$ be more constrained than $A^B\Phi_j$. Only elements that are zero in $A^B\Phi_i$ can change during a transition from $A^B\Phi_i$ to $A^B\Phi_j$. Elements that are nonzero in $A^B\Phi_i$ can not change during the transition.
Spatial Adjacency
- Necessary and Sufficient Conditions

Let $^{AB}\Phi_i$ be more constrained than $^{AB}\Phi_j$ and let and $^{AB}\Phi_i$ satisfy the necessary condition. Qualitative configurations $^{AB}\Phi_i$ and $^{AB}\Phi_j$ are adjacent at $q^* \in P_k(\{^{AB}\Phi_i\})$ if and only if there exists a twist $t_1$ such that:

$$(^{AB}d_{mn})^t \cdot t_1 = 0 \quad \forall (m, n) \in \{(m, n)\mid [^{AB}\Phi_j]_{mn} = 0\}$$

$$(^{AB}d_{mn})^t \cdot t_1 > 0 \quad \forall (m, n) \in \{(m, n)\mid [^{AB}\Phi_i]_{mn} = 0 \land [^{AB}\Phi_j]_{mn} = +1\}$$

$$(^{AB}d_{mn})^t \cdot t_1 < 0 \quad \forall (m, n) \in \{(m, n)\mid [^{AB}\Phi_i]_{mn} = 0 \land [^{AB}\Phi_j]_{mn} = -1\}$$
Contact Description

- A contact description between two non-convex polyhedra A and B is the union $\gamma^{A,B}$ of the contact descriptions between combinations of convex polyhedra $A_i$ and $B_j$ of the convex decompositions of A and B:

$$\gamma^{A,B} = \bigcup_{1 \leq i \leq m^A_{cv}} \bigcup_{1 \leq j \leq m^B_{cv}} \gamma^{A_i,B_j}$$
Vertex-face contacts

Let \( A_i \) and \( B_j \) be convex polyhedra. There is contact between vertex \( v_k^{A_i} \) and face \( f_l^{B_j} \) if and only if

\[
A_i B_j \phi_{kl}^{vf} = 0
\]

\[
(\forall f_m^{B_j} \in \text{adj}(f_l^{B_j}))(A_i B_j \phi_{km}^{vf} = -1) \lor (A_i B_j \phi_{km}^{vf} = 0)
\]
Face-vertex contacts

Let $A_i$ and $B_j$ be convex polyhedra. There is contact between face $f_k^A_i$ and vertex $v_l^B_j$ if and only if

$$A_iB_j \phi_{kl}^{f_v} = 0$$

$$(\forall f_m^A_i \in \text{adj}(f_k^A_i))((A_iB_j \phi_{ml}^{f_v} = -1) \lor (A_iB_j \phi_{ml}^{f_v} = 0))$$
Contact Information Extraction (cont’)

- Edge-edge contacts

Two types of edge-edge contacts are considered: edge-edge-cross contacts and edge-edge-touch contacts. An edge-edge-cross contact exist when two edges contact each other but are not parallel. This is shown in Frame (a). An edge-edge-touch contact exist when two edges contact each other while they are parallel. This is shown in Frame (b).
Edge-edge-cross contact

Let \( A_i \) and \( B_j \) be convex polyhedra. There exists and edge-edge cross contact between edge \( e_k^{A_i} \) and \( e_l^{B_j} \) if and only if the following three conditions are satisfied:

\[
A_i B_j \phi_{ee}^{kl} = 0
\]

\[
(A_i B_j \phi_{k_1 l_1}^{vf} \neq A_i B_j \phi_{k_2 l_1}^{vf}) \lor (A_i B_j \phi_{k_1 l_2}^{vf} \neq A_i B_j \phi_{k_2 l_2}^{vf})
\]

\[
(A_i B_j \phi_{k_1 l_1}^{fv} \neq A_i B_j \phi_{k_1 l_2}^{fv}) \lor (A_i B_j \phi_{k_2 l_1}^{fv} \neq A_i B_j \phi_{k_2 l_2}^{fv})
\]
Edge-edge-touch contact

Let $A_i$ and $B_j$ be convex polyhedra. There exists and edge-edge touch contact between edge $e_k^{A_i}$ and $e_l^{B_j}$ if and only if the following two conditions are satisfied:

\[
(A_i B_j \phi_{k_1 l_1}^{vf} = 0 \land A_i B_j \phi_{k_1 l_2}^{vf} = 0) \land (A_i B_j \phi_{k_2 l_1}^{vf} = 0 \land A_i B_j \phi_{k_2 l_2}^{vf} = 0)
\]

\[
(A_i B_j \phi_{k_3 l_1}^{f} = A_i B_j \phi_{k_4 l_1}^{f} = -1) \lor (A_i B_j \phi_{k_3 l_2}^{f} = A_i B_j \phi_{k_4 l_2}^{f} = -1)
\]

\[
(A_i B_j \phi_{k_1 l_3}^{vf} = A_i B_j \phi_{k_1 l_4}^{vf} = -1) \lor (A_i B_j \phi_{k_2 l_3}^{vf} = A_i B_j \phi_{k_2 l_4}^{vf} = -1)
\]
Define the *cost-to-go* from $^{AB} \phi_i$ to $^{AB} \phi_{goal}$ along trajectory $T_{tr} (^{AB} \phi_i, ^{AB} \phi_{goal})$ as:

$$CTG(T_{tr} (^{AB} \phi_i, ^{AB} \phi_{goal})) = n - \sum_{k=0}^{n-1} \beta N_c (^{AB} \phi_{i+k})$$
Summary: Discrete Event Controller

- Synthesize-Discrete-Event-Controller algorithm integrates the methods for hypothesis generation, hypotheses testing and contact information extracting into one algorithm that synthesizes a discrete even controller.

- The search should be terminated if an acceptable path from the goal configuration to the start configuration has been found or if all states have been expanded.
Motion Planner

- Assembly motion planning results from the consideration of uncertainty during the assembly process.
- Several works on this problem:
  - Local strategies vs global motion planning
  - LMT framework vs two phase approach
Local Strategies

- Related works:
  - A passive compliant device for peg-in-hole problem – by researchers at Charles Stark Draper lab.
  - A tilting strategy for convex pegs – by Strip.
  - A tilting strategy for inserting rectangular pegs – by Caine.
Local Strategies (cont’)

- Characteristic: purely local, help planning the path which the grasped part should move in response to the forces/torques of reaction.

- Problem: do not consider the problem on a more global scale.
Global Motion Planning

- Related works:
  - Using task geometry to place constraints on the compliant motions – by Mason.
  - A planner based on a contact-space representation of the parts – by Laugier.
  - Several works in the following LMT framework and two-phase planners.
LMT Framework

- **Related works:**
  - Computing motion strategies given bounded uncertainty using pre-images in configuration space (C-space) – by Lozano-Perez, Mason and Taylor.
  - Plan compliant motions in a finite state space for 3D translation without rotation – by S.J.Buckley.
LMT Framework (cont’)

- It attempts to form a one-step motion plan by finding a motion vector which moves the grasped part from the starting configuration to the goal configuration.
- The LMT models of uncertainty in the initial pose and in control have been followed by most other researchers.
- Limitations:
  - Computational methods for constructing pre-image have been found only for polyhedral C-space.
  - Poor computational efficiency.
Two Phase Approach

Idea:

- Phase 1: a nominal trajectory in free space is computed without considering uncertainty, only the geometric and spatial constraints are taken into account.
- Phase 2: the uncertainties are taken into account to refine the nominal trajectory and assembly motion plan is developed from it.
Two Phase Approach (cont’)

- Related works:
  - Design of nominal velocities and admittance matrices to guarantee force assembly – by Schimmels and Peshkin.
  - A fine motion planner for assembly where force/torque guided motions are incorporated into the plan as needed – by Gottschlich.
  - Find a nominal solution path in both free and contact configuration space – by Rosell, Basanez and Suarez.
  - Let a deviation from the nominal trajectory trigger an error recovery strategy or patch plan – by Xiao and Volz.
Two Phase Approach (cont’)

- Generating hypothetical contact formations – by Xiao and Zhang.
- Linearizing the configuration space around critical points – by Dakin and Popplestone.
Example: - by Gottschlich

**ASSEMBLY REPRESENTATION**

**PHASE 1: Ignore Uncertainties**
- **Stage 1:** Develop initial path
- **Stage 2:** Refine initial path into collision-free nominal path

**PHASE 2: Incorporate Uncertainties**
- **Stage 1:** Analyze uncertainties and characterize path segments.
- **Stage 2:** Develop assembly motion plan from segment characterization.

OFFLINE
Some related works are introduced.

Those works are dealt with the problem of generating motion planning for assembly by different representations, sensors and approaches.

Our work is to compute the motion planning by qualitative contact models. The motion planning is generated from the output of the discrete event controller, which will compute a qualitative contact state transition path for the assembly problem.
In order to implement robotic assembly, a robot controller will be designed to guide the grasped part to achieve the desired state transition that was specified by the discrete event controller.

- Inputs: velocity commands from motion planner.
- Outputs: commands to guide robot movements.
- Some existed compliant motion robot control laws will be considered to be implemented in the robot controller.
Conclusions and Future Work

- The whole system of robotic assembly is introduced.
- Design a motion planner to compute compliant motion parameters based on the desired contact transition generated by discrete event controller.
- Design/Find a suitable robot controller to execute the assembly tasks.
- Improve the discrete event controller by considering the hardware limitations and uncertainties.
- Synthesis above three parts to complete the whole robotic assembly system.