Agenda

- Background
- Existing solutions
- REACH
- Results Metrics
- Framework
- Plugins
- Example
- Future work
- Relevance
Background

- Industrial robotic systems designed to perform specified task(s)
  - Opposed to some robotics applications where new use-cases are researched on existing hardware
- Considerations for robotic system design
  - Workspace size
  - Workspace constraints
  - Workpiece geometry
  - Robot size
  - Robot configuration
- How to evaluate concepts to objectively?
Background

- **What do we want?**
  - A design that:
    1. Can reach an acceptable area on a workpiece with a given tool(s)
    2. Has the most flexibility for accommodating new parts/processes and/or changes to the environment
    3. Stays as far away from collision with the environment as possible
  - To understand:
    - How changes to system configuration affect the goals defined above
    - How the robot system will reach desired points
SwRI Examples

- Laser De-paint Robot
  - Must reach ~90% of area on mid-size aircraft (e.g. Boeing 737, Airbus A320)
  - Proposed configuration: 11+ DOF
    - 8+ DOF manipulator
    - 3 DOF mobile base
SwRI Examples

- Military Aircraft Maintenance Robot
  - Must service ~50% of area on C-17 aircraft
  - Tool Z-orientation free
  - Proposed configuration: 10+ DOF
    - 7+ DOF manipulator
    - 3 DOF mobile base
Existing Solutions

- **Brute force**
  - CAD environment
  - Offline programming software
  - 3D printed models
  - **Hard, time-consuming, and expensive**

- **Smarter Approach**
  - Automated robot base placement
    - Siemens Process Simulate
      - **Insufficient for high-DOF systems and mobile robots**
  - Inverse reachability
    - ROS-I Reuleaux package
      - **Lacks focus on the workpiece**

Adapted from [1]
REACH

- https://github.com/ros-industrial/reach

Core Process
- Generate desired reach points on a workpiece
- Solve inverse kinematics at each point
- Evaluate the reachability at each point
- Maximize the reachability values
- Report and visualize the results
REACH

- Maximize the reachability values
  - Infinite number of IK solutions for high-DOF systems
  - Gradient-based IK solver
  - Initial IK solution generally produces low score (if solution is even found)
  - At each target
    - Use neighbors as IK seed states
    - Re-solve IK at target
    - Re-evaluate reachability at target
  - Iterate until reachability stops improving
Framework

- Plugin-based architecture
  - Environment/inverse kinematics interface
  - Reachability evaluation criteria
  - Display interface
- Provides flexibility for different back-ends
- User-specifiable via YAML file
Results Metrics

- Percentage of targets reached
- Total reachability score of all points
- “Potential” total reachability score
  - What would the score be if the robot reached every target?
  - Total score / percentage reached
- Average number of reachable neighbors
Plugins

- Inverse Kinematics
  - 6-DOF constraint
  - Discretize about Z-axis

- Evaluation criteria
  - Manipulability
    - How easily the robot can move in any direction from a given pose
  - Nearest distance from collision
  - Distance from joint configuration
  - Combination of metrics (sum, product, etc.)

Adapted from [2]
Plugins

- Display plugin
  - Interactive markers at targets
    - Display robot state
    - Re-solve IK
    - Show seed state
  - Comparison between configurations
  - Results heat map
Example

- Laser De-paint Robot
  - C-17 aircraft
  - Results
    - Reach percentage: 93.6%
    - Score: 328,378
    - Normalized score: 350,832
Example

- Decide between several design concepts
  - Robot mounted on gantry
  - Multiple workpieces
  - Spherical wrist vs. offset wrist robot
- Use reach study data to narrow down concepts
  - % reachable: $R_1 \approx R_2$
  - Raw score: $R_2 > R_1$
  - Potential score: $R_2 >> R_1$
  - Use Design 1, Robot 2

<table>
<thead>
<tr>
<th>Design</th>
<th>Work-piece</th>
<th>Robot 1</th>
<th>Robot 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% Reached</td>
<td>Raw Score</td>
<td>Potential Score</td>
</tr>
<tr>
<td>D1</td>
<td>Object 1</td>
<td>91.90%</td>
<td>338.5</td>
</tr>
<tr>
<td></td>
<td>Object 2</td>
<td>73.60%</td>
<td>290.5</td>
</tr>
<tr>
<td>D2</td>
<td>Object 1</td>
<td>92.70%</td>
<td>357.5</td>
</tr>
<tr>
<td></td>
<td>Object 2</td>
<td>73.20%</td>
<td>287.2</td>
</tr>
<tr>
<td>D3</td>
<td>Object 1</td>
<td>74.80%</td>
<td>301.8</td>
</tr>
<tr>
<td></td>
<td>Object 2</td>
<td>57.40%</td>
<td>271.1</td>
</tr>
</tbody>
</table>
Future Work

- Reduce setup complexity
  - GUI
  - Improve mesh sampling to produce target points
  - Tighter integration of mesh sampling into application

- Visualization
  - Interpolate results to create heat map
  - Results by individual evaluation metric

- Non-linear optimization to maximize pose reachability
Relevance

- Makes analysis of robotic systems more feasible (especially high-DOF systems)
- Better analysis for single robot
  - Task/process oriented
  - Reach percentage
  - Visualize robot state at various target points
- Better analysis for multiple robot concepts
  - Compare reachability scores directly
  - Visualize reachable target “diffs” between various concepts
- Informs design decision more effectively than “gut feel”
Questions?

Michael Ripperger
Southwest Research Institute
6220 Culebra Rd.
San Antonio, TX, USA

+1 (210) 522-6292
michael.ripperger@swri.org

www.swri.org
www.rosindustrial.org
References