Snap Assembly Considerations

**October 7, 2011**

# Considerations

1. Number of snap parts
   1. How do they play a role?
2. Direction Constraints.
   1. Are there any?
   2. How do they simplify the task?
3. Partial or Incomplete Snapping
   1. How can we check and resolve for partial snaps?
   2. Also for failed snaps
4. Desnapping
   1. Does desnapping have a different force signature altogether for the same snap?
5. Controller
   1. It seems we need a push and hold model, so that when the snap takes place the robot effector does not snap with it.
6. How do humans do snap?
   1. Explore
   2. Experiment
   3. Learn by trying and continually get better (adjust parameters)

# Breaking Down Areas in Snap Assembly

* Are we trying to create a new controller?
* Strategy?
* Generalization system?
* Failure recup system?
* Classification of force signatures?

**By Stages:**

* Find a strategy
  + First simple
  + Then more complex
* Develop a basic controller that allows you to do the stuff (not contribution)
* Learn/classify force profiles
  + To classify parts
  + To classify success/failure
* Recup system
  + In case of poor assemblies, recup.

## Assembly Types or styles

* Things that people assemble:
  + Snaps, Grommets, Fastener’s, and
* Types of Snap Assembly:
  + Hip Impact Insertion
  + Cantilever snaps
  + Torsional Snap Joints
  + Annular Snap Joints

## Automation

* Assembly-Part automation is unique to a given product.
  + Each product has its own assembly sequence plan.
* Generalizing Snap-Assembly?
  + If a given plan was given, could we have a method by which any snap could be implemented?
  + This adaptable approach would be huge.
* Maybe you need to start from simple-to complicated:
  + Size: from one-size to many sizes,
  + Velocity: from one velocity to many velocities,
  + From one-part-type to many types.

## Reasoning

* How about classifying force profiles with words. As humans do in force tasks. Too stiff, too soft. Harder.
* Or in terms of change perhaps. +ve gradient, -ve gradient, +ve quadratic gradient… maybe we can connect a positive-negative-positive sequence of changes as an assembly.
* Decisions can be dictated using fuzzy-logic. Not optimal but achievable.
* Can it be optimized more on top of it?
* Can it learn? Yes, humans can learn even with these set of rules. Fine tune.
* We would have to segment the signals appropriately.
  + Then classify them.
    - Providing useful terms. How would you describe them.
  + Then create a rule-set that is useful for all situations. Generic Enough.
    - Insert, snap, desnap
    - Stiff, soft
    - Fast, slow
    - Growing force, decreasing force
    - Or composites, growing force and fast
  + Maybe it could use relative terms as well
    - Insert more
    - Stiffer, softer
    - Faster, slower
* Then think about automation. How to automate? Use goal?
* Then learn at the end.
  + Unsupervised/supervised learning?
* Then further optimization.
* Use voice commands?

## Change

* Parameters of distance:
* I.e. consider your object.
  + There may be the guarded move region,
  + There may be the about to insert region,
  + There may be the snap region.
    - Each of which may have different dimensionality.
* So within these regions we look for vector gradient information of different lengths.
  + That also needs a filter/average mechanism.

# Building the Road Map

## Stages of Snap Assembly

* Localization
* Approach
* Rotation
* Insertion (snap)
* Mating
* Desnap

Which techniques will you use for each stage?

Build from simple to complex, from singular to multiple kinds of snaps.