## Literature Review on

## Snap Assembly and Control Strategies for Assembly

## Snap Assembly

1. 1999ISEE-Sodhi-SnpFitDesign
   1. Understanding the snap-fit design.
      1. There is a methodology that follows sound engineering principles to produce fundamentally sound snap-fit interfaces:
         1. Understand the functionality
         2. Select basic shapes
         3. Use *constraint* and *enhancement* features in the interface.
         4. Select an *install direction*
         5. Select an assembly motion
   2. They can be useful to us in our work to first identify snaps, insertion direction, and assembly motion (linear, rotational).
2. 2003JAA-Chin-FrceGdBotAsmblyPhone
   1. Three force-guided robotic skills and sequences were identified for notch-lock snap assembly.
      1. Stopping, aligning, and sliding.
   2. The assembly process for a phone is not easily defined because
      1. Each phone generation is clouded with many exceptions.
3. 2005MD-Hoffman-Fund Snap Fit Jnts
   1. Three kinds of snap-fits:
      1. Cantilever
      2. Torsional Snap Joint
      3. Annular Snap Joints
   2. Deformation is determined by the strain of the material.
      1. Use 50% of strain break point for most plastics and 60-70% for polymers.
   3. Ridge geometry determines how easy it is to come in and out.
      1. 90% permanent snap.
      2. 45% typical of most applications that disassemble.
   4. 30% easily separates.
4. 2007ICASE-Popa-High Yield Autonomous MEMS Assembly
   1. Snap-Fastener advantages:
      1. Compensates for positional uncertainties
      2. Eliminates stiction problems
5. 2011TRO-Platt-BayFiltLocFlexMat
   1. Localized features in flexible material using Bayesian Filtering and particle filtering.
      1. Supported by Gaussian/Bayesian and multi-modal Gaussians.
      2. Gaussian models addressed uncertainty error.
         1. After testing with a single Gaussian fit, authors used a multi-modal Gaussian model.
   2. Evaluates localization performance by proprioceptive information/tactile information.
   3. Suggested a method to reduce the dimensionality of the problem by controlling the approach
   4. Limitations:
      1. Bayesian filtering is very sensitive on force-sensor calibration errors.
      2. Localization based on a single haptic map does not generalize well.
      3. The idea of generalization to other shapes is very important.
         1. I am thinking of finding something that measures changes
         2. Perhaps classifies the kinds of changes
      4. Particle filter did help a bit over the limitations of the haptic map
      5. Improvements:
         1. Class of features
            1. How to divide class? Size, area, style
6. 2011ICRA-Stolt-ForceCtrldAsmblyStpBttn
   1. Use a state-machine to direct linear position-force searches on the assembly device.
   2. Model the force-profiles as multi-variate Gaussian distributions.
   3. Use a hyper-ellipsoid to detect the snap.
   4. Optimize the assembly.

## Force Profile Feature Recognition Algorithms

1. 1999ICRA-Cervera-ElimAmbigRecurNN.pdf
   1. Self-organizing neural-networks and recurrent neural-networks.
   2. They observe that the force/moment information is inherently ambiguous.
   3. Ambiguity can be addressed by looking at the history of sensor readings.
   4. They use the recurrent neural-network to look at the history to make decisions about the future.
2. 1990ICRA-Asada-TchngLrnngNN4Cmplianc.pdf
   1. Learned to map associated forces with corrective motions along a nominal path using neural networks and used the mapping to effect non-linear compliance control in assembly tasks.
   2. Associated wrench data with velocity data by encoding a small set of contact states for a peg-in-a-hole assembly task.
   3. Used NN to learn representation of a FSM sequence.
   4. Then performed action learning through Q-learning.
3. 1996ICRA-Hovland-FreqDmainFrcMesurement4DscretEvntCntactRecgn.pdf
   1. Approach to recognize discrete events in an assembly task based on force measurement in the frequency domain.
   2. Discrete events modeled as Hidden Markov models.
   3. Used a training set for on-line recognition.
4. 1996ICRA-Sikka-MntrngCntctClstrng\_DscrmntFnctns
   1. Monitor signal
      1. Cluster training data based on a distance or some function
         1. Use discriminant function learning algorithms/qualitative reasoning methods to
            1. Identify discrete states/events
5. 2001ICRA-Newman-InterpForceProfileInsertions.pdf
   1. Guided assembly searches by finding a mapping between .
   2. Used a Backpropagation Neural net with k-means to find the inverse relationship between moment values , and relative displacement ().
   3. Used a quality map to distinguish between regions of low-value, medium-value, and high value.
   4. Switched from just considering one point at a time to considering the whole history of records.
   5. Learned that dependence on instantaneous measurements result in erratic performance.
      1. To reduce them, there are few recommendations:
      2. Move the robot slowly
      3. Take many samples over the course of x amount of time at a given location and then average the values in the buffer.
6. 2005ASME-Chhatpar-Lclztn4BotAsmblyPrtclFltrs
   1. Particle filters were selected as a tool to localize a key-lock assembly as it allows localization when:
      1. There are inaccuracies (even large ones) in the pose estimate of the robot,
      2. There may be more than one solution to the target position.
   2. Particle filters are a Monte Carlo method which maintain probability distributions for a varying number of particles/states in the system.

## Generalization

1. 2007ICRA-Thomas-MultSnsrFsnBotAsmblyPriclFltr.pdf
   1. Used particle filters to optimize the assembly motion.
      1. Combined visual and pose information obtained from CAD-generated FT maps and visual features.
2. 2008ICRB-Marvel-AutLrningParamOptimztnBotAsmblyUsngGentcAlgs
   1. This work seeks to provide adaptable solutions to the peg-in-whole assembly by integrating
      1. Generic assembly strategies with tunable parameters.
   2. The parameters are self-tuned empirically using a
      1. Genetic-Algorithm learning process
         1. That minimizes assembly time subject-to contact-force limits.
   3. Demonstrated approach on two industrial robots.