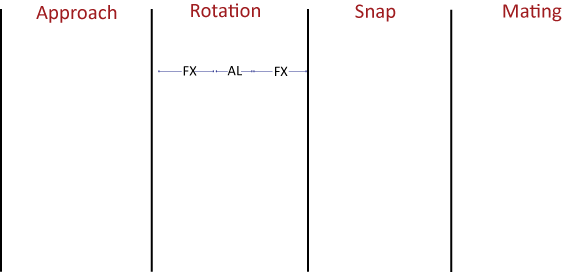
Probabilistic Approach for  
Change-Based Hierarchical Taxonomy (CBHT)   
for Snap Verification

# 1. Simple Probabilistic Model for a Snapping TASK Verification using CBHT.

* Definition of probability at the Low-Level Behavior Layer.
  + Consider that in each state we may have one or more LLB’s.



* + Assumptions:
    - * Assume we only are looking for the presence of a single low-level behavior in the state.
      * There may be several ways of representing the likelihood that that LLB is present in the state, two ways are:
        + Linear: compute the probability of a given LLB as a fraction of its total duration compared to the duration of the state.

In the above example, FX may occupy about 66% of the time of the state and hence we attribute that likelihood to it.

How often a behavior appears as a fraction of the total number of LLB’s that appear.

* Once these are computed, they can be used along with a “Condition’s Table” that establishes what LLB’s must be present to generate an HLB:

|  |  |  |  |
| --- | --- | --- | --- |
| *Axis/HLB* | Rotation | Insertion | Mating |
| Fx | FX | CT | FX |
| Fy | - | union(AL,FX) | FX |
| Fz | FX | AL | FX |
| Mx | - | union(AL,FX) | FX |
| My | FX | CT | FX |
| Mz | - | union(AL,FX) | FX |

* Then, the probability that Rotation is the HLB for that state would be the joint probability of all the elements:

).

* This can be done for each state.
* Weights can be included in the product to weight some features more heavily than others.
  + This may play more of a role in the Snap/Mating states.

# 2. Snap and Fault Verification System

* It’s important that we do not only find conditions for what works, but also to classify behaviors that lead to failure.
  + We may want to incorporate position information in the whole system.
  + The system basically plays the role of vision but at the sensing level.
  + Vision could be used to reinforce a behavior.
* For a faulty system:
  + Identify basic faults by state and their corresponding low-level behaviors.
  + They could be further organized in an organic graph.
    1. Rotation State:
       - No Docking
         * Manipulator moves too far or too short.
       - Stuck
         * PS behaviors show up predominantly in the first three axis.
    2. Insertion Stage
       - No contact at the end of the task
         * Need more pushing
       - Jamming/Wedging
         * PS behaviors show up
    3. Mating
       - Sliding
       - Separation

## Basic Verification System

* Master Table could be generated that summarize the verification mechanisms that the system would look for:

Layer 5: Verification Layer

Layer 4: HLB’s

Layer 3: LLB’s

Layer 2: Composites  
Layer 1: Primitives

# 3. Controller Policies

* Once this policy is defined, controller policies can be generated to fix potential mistakes.
  + This could come as an adjustment to desired values, or
  + As the establishment of a whole new strategy.

# Bayesian Inference

* Bayesian inference is to be computed separately for each of the three states that contain force-torque data: Rotation/Snap/Mating.
* Automata State Dependent Probabilities:
  1. Initial prior probabilities.
  2. System model (state transition) probabilities can be updated.
* We also will compute the duration average of key LLBs to be used to determine the presence of HLBs.
* The computation of HLBs will be through a weighted product of LLB’s probabilities.
* Initial Prior Probabilities given an automata state, :
  + It indicates how likely it is that the LLB is found in the state.

|  |  |  |
| --- | --- | --- |
| Rotation | Snap | Mating |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### Possible values for initial prior probabilities for the state of LLB’s:

* Use trial data, to find a mean value of the duration of the LLB per force axis per automata state. Render that duration a probability as a function of the duration of the automata state.

### Measurement Model

* For the measurement model, we can use trial data to estimate the mean duration of LLBs per force axis per automata state and then use a Gaussian distribution to estimate the probability of certainty that what the LLB that we have right now is correct.
* Some examples would be:
  + Measurement Model (36 possibilities),
    1. …
    2. …
    3. …
  + System Model (36 possibilities),
    1. …
    2. …
    3. …
  + Prediction Step (6 equations with 6 terms each), p. 30:

;

First we try , for all equations, then we try: , for all equations, and so on… until .

* + Correction step:
    1. Includes the measurement update:

Where, first, then , all the way until .  
 was given a single choice. The measurement was “is\_open”.­­

* + 1. All of these probabilities, after being normalized will add to 1.
* Summary:
  + Calculating Bayes theorem requires **three** **probability distributions**:
    1. Initial belief
    2. Measurement Probability
    3. State transition probability
  + There is also for a representation for the belief