2012 12th IEEE-RAS International Conference on Humanoid Robots
Nov.29-Dec.1, 2012. Business Innovation Center Osaka, Japan

Probabilistic State Verification for Snap Assemblies using the Relative-Change-Based Hierarchical Taxonomy.

J. Rojas, K. Harada, H. Onda, N. Yamanobe, E. Yoshida, K. Nagata, and Y. Kawai
Intelligent Sys. Research Institute, AIST
Tsukuba, Ibaraki, 305-8568, Japan

***Abstract*—Autonomous snap assemblies is a highly desirable robotic functionality. While much work has been done in active sensing for peg-in-hole assemblies and general compliant motions, snap assembly state estimation remains an open research problem. This work presents a probabilistic framework designed to account for uncertainties in assembly and yield more intuitive and robust outcome assessments. Simulation of an anthropomorphic robot HIRO performed a cantilever-snap assembly using the Pivot Approach control strategy and our snap verification system. The latter used a Bayesian Filter within its hierarchical taxonomy yielding belief states at two levels of the taxonomy. The last layer of the system, effectively assessed the outcomes of all test assemblies. The framework was effective in correctly assessing the outcome of all test assemblies.**

I. INTRODUCTION

Autonomous snap assemblies is a highly desirable functionality for robots. Snap assemblies are challenging due to their varied and complex configurations and their elastic components. For the last two decades, the closely related task of peg-in-hole assembly has benefited from active sensing. The latter refers to the robot's ability to reason about its state and make decisions to efficiently execute a desired task. A number of methods have been used to estimate the state of a task: including qualitative reasoning [1], learning contacts and state transitions [2], and fuzzy logic [3]. More recently, probabilistic methods have been applied to generate more robust manipulation behaviors in the presence of uncertainty [4], [5].

With regard to snap assemblies, the latter can be categorized into three assembly types [6], of which, the cantilever-snap type can vary in complexity by increasing the snap number (typically 1, 2, or 4). To date, state estimation has been implemented for a single-snap cantilever part [7]. However, there are no frameworks that generalize to snaps of varying complexity to perform state estimation and corrective motions for snap assemblies (we will refer to *Snap Sensing* as an equivalent to active sensing in this domain). Our focus is to design such a framework for both industrial use and personal robots. Previously in [6], we developed the Pivot Approach (PA) control strategy for cantilever-snap type assemblies (for all abbreviations please refer to Sec. **??**). The PA exploits snap parts' hardware design to constraint the task's motion and generate similar sensory-signal patterns across trials and systematically discretize the assembly into intuitive states (see Sec. III). In [8] we designed the Relative-Change Based Hierarchical Taxonomy (RCBHT) snap verification system. The latter worked in concert with the PA and was built on the premise that relative-change patterns can be classified through a small category set while aided by contextual information (see Sec. IV). While the RCBHT yielded promising results, it did not deal with uncertainty sources such as: noisy force-torque (FT) signals and non-deterministic predictions by the RCBHT. In this work we studied whether rendering the RCBHT probabilistic would yield a more robust, intuitive, and granular representations for the task's state.

To this end, a Bayesian Filter (BF) algorithm was designed and embedded in the system's taxonomy yielding temporal belief representations for low-level behaviors (LLB). Furthermore, the higher-level behavior (HLB) layer, computed the joint probabilities of LLBs that uniquely represent HLB's associated with the snap assembly automata states[1]. The joint probabilities yielded HLB beliefs for each automata state. The last layer, the Snap Verification Layer (SVL), used HLB beliefs across automata states to derive thresholds to determine test trial's outcomes. The effectiveness of the system was tested by simulating the snap assembly of a 4-snap cantilever camera part through Kawada's HIRO robot. Training trials were used to develop probabilistic models for the BF and derive thresholds for the SVL layer. The probabilistic RCBHT (pRCBHT) yielded intuitive temporal state representations for both LLBs and HLBs. The BF effectively replaced sequences of LLBs and HLBs for temporal belief signals in both layers. The pRCBHT correctly assessed the outcome of seven test snap assembly trials performed in simulation indicating the effectiveness of the probabilistic approach for cantilever snap assemblies.

The paper is organized as follows: Sec. II introduces the experimental setup. Sec. III describes the Pivot Approach control strategy. Sec. IV presents the non-probabilistic snap verification system. Sec. V introduces this work's contribution, the Bayesian Filter implementation within the RCBHT. Sec. VII discusses the significance and limitations of our work while Sec. VIII summarizes the key aspects of the paper.

II. EXPERIMENTAL SETUP

In this work the Pivot Approach [6] and the RCBHT system [8] were applied to a dual-arm 6 DoF anthropo-

[1]Automata states should not be confused with the task's state. The former refers the way in which the snap assembly task is divided, the latter refers to the behavior enacted by the robot for which the posterior will be computed

978-1-4673-1369-8/12/$31.00 ©2012 IEEE

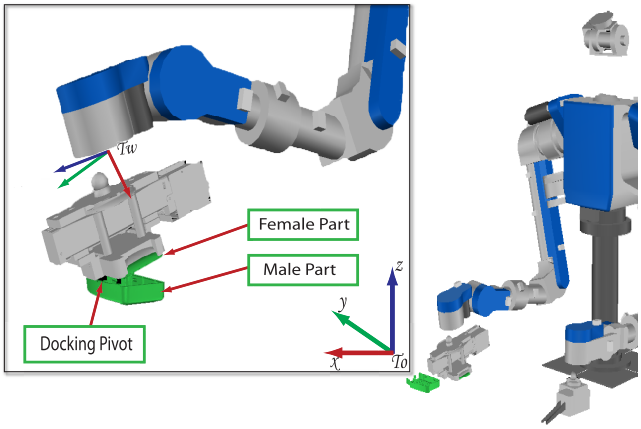

Fig. 1. Experimental Setup - The HIRO robot performs assembles a camera mold consisting of a female part and a 4-snap male part.

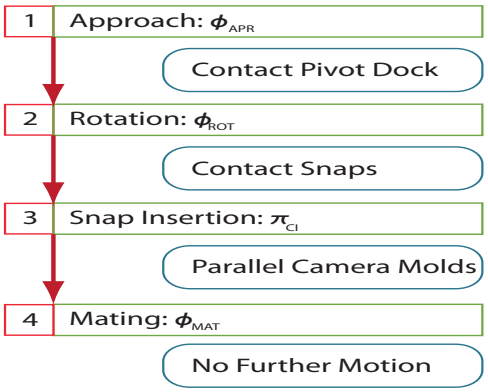

Fig. 3. Each of the four automata states (red) is accompanies by a controller template (green). Transition conditions are specified in blue.

morph HIRO robot that was simulated using the OpenHRP environment [9] (our original work used a Mitsubishi 7 DoF industrial manipulator). A CAD derived camera part consisting of both male and female parts was used. The female part was rigidly held by a specially designed tool mounted on the robot's wrist, while the male mold part, which consisted of four cantilever-snaps (two snaps were used in our previous work) was rigidly fixed to the ground. Cantilever-snap assemblies were executed through the PA and assessed through the pRCBHT. The experimental setup along with coordinate frames and reference points are shown in Fig. 1.

III. Pivot Approach

The PA exploits snap parts' hardware design to constraint the task's motion and generate similar sensory-signal patterns across trials and systematically discretize the assembly into intuitive states [6].

In our previous work, the PA was applied to a two-snap male part and decomposed the assembly into five automata states: Approach, Rotation, Alignment, Snap, and Mating. The male part consisted of a vertically-offset pivoting dock from the camera's top edge wall. This offset affected the optimality of the entry points between male and female parts requiring an "Alignment" stage for its completion. In our current work, a four-snap camera part without offset was used. Having the pivot at the wall edge allows for an optimal entry of male and female parts eliminating the need for an alignment. The assembly used four states to complete the task as seen in Fig. 3 and used the action transitions seen in Fig. 2 (see [6] for more details).

A. Controller Templates

In concert with the PA, each automata state is executed by a controller template. Controller templates were implemented under the Control Basis Framework [6]. The framework is designed to flexibly but systematically build and modify controllers. This framework is useful in environments where change is common. As part of our ongoing work in automating the snap assembly problem, the framework is beneficial in allowing us to flexibly implement controllers for different snap type configurations (cantilever, annular, and torsional), where each configuration may have varying degrees of complexity in their geometry, and to implement such strategies with different robots.

All controller templates, except for Rotation, are the same as in our previous work. Namely, the Approach, Φ_{APR}, Compliant Insertion, π_{CI}, and Mating, π_{MAT} controllers. Previously, the Rotation controller was a position controller unable to respond to force disturbances predominant during this task's stage. The controller was changed into a compound FT controller π_{ROT} with a dominant force controller ϕ_{fr} and a subordinate moment controller ϕ_{mr}. The controller's subordinate update commands are projected into the nullspace of the dominant controller's update space to optimize both objectives. The force controller uses two reference parameters to push the female part down and against the male part's wall, while the y-direction reference is set to zero as no horizontal push is necessary: $\mathbf{f}_{ref} = \{10, 0, 0.25\}$N. As for ϕ_{mr}, its reference parameter applies a torque in the y-direction $\mathbf{m}_{ref} = \{0, 20, 0\}$Nm. All reference parameters are applied in world coordinates.

The coordinate frame references used here will affect the selection of key LLB used in both the RCBHT and the pRCBHT (Sec. IV and Sec. Sec. VII). The Approach state transitions to the Rotation state when the x-force exceeds $9N$. The Rotation state transitions to the Snap state when the y-moment exceeds $0.60Nm$. Establishing transition thresholds is not trivial and some difficulties will be discussed in Sec. VII. The reference frames will influence what axis we focus on to characterize HLBs in the RCBHT. The PA automata is shown in Fig. 3.

IV. Relative Change-Based Hierarchical Taxonomy

The RCBHT is a state estimation scheme for used in snap assemblies. Over the last two decades much work has been

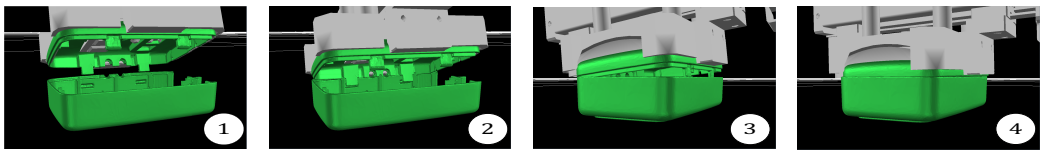

Fig. 2. The Pivot Approach is composed of four states: Approach, Rotation, Snap, and Mating.

done in active sensing for compliant motion tasks; initially most work was applied to peg-in-hole tasks [2], [1]. Recently, the concept of contact-state graphs has been used for general compliant motion tasks for simple geometrical parts as in [4], [5]. Even with simple geometrical parts, the contact state number can explode. The approach becomes unfeasible for geometrically complex parts as is typically the case with snap contacts.

With this in mind, the RCBHT yields state representations by hierarchically abstracting snap assembly FT data to generate intuitive HLBs [8]. The hierarchical taxonomy is composed of five increasingly abstract layers that encode relative-change in the task's force signatures. The taxonomy is built on the premise that relative-change patterns can be classified through a small set of categoric labels and aided by contextual information. The RCBHT analyzes FT signatures from all force axes independently and contextualizes the state according to automata state participation (the Approach stage is not considered as no FT data is gathered there).

The Bayesian Filtering method that will be further described in Sec. V is embedded within the RCBHT's 3rd layer and renders the verification system probabilistic. The system uses world coordinates and currently runs off-line, Fig. 4 for a system overview.

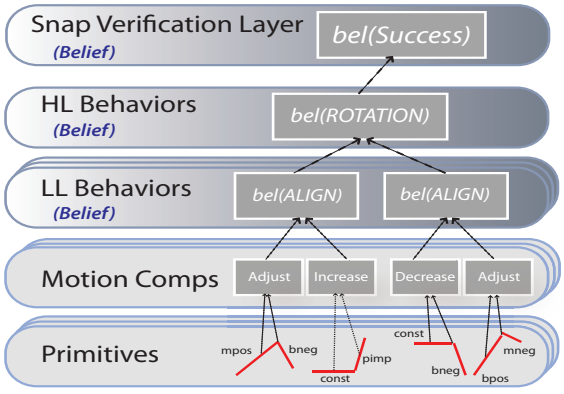

Fig. 4. The RCBHT abstracts FT sensor data to produce intuitive higher-level behavior representations that enable reasoning about the snap assembly's state.

A. Primitive Layer

The first layer is the primitives layer which partitions the FT data into linear data segments and classifies those segment according to its gradient magnitude(for more details see [8]). Linear regression is used in concert with a correlation measure to segment the data whenever a minimum correlation threshold is crossed.

The gradient classification separates data in which a contact or mating event occur. Contact phenomena is characterized by sudden and large changes in force signals, almost approximating an impulse. To this end, positive impulses ("pimp's") and negative ones ("nimp") were categorized for gradients $(\Delta force/\Delta time)$ with values greater or less than 1000. For mating events, there is little or no change in FT data, for this reason a constant label was assigned to signals with gradient less than the (absolute) value of 2. Currently we use the same gradient thresholds for force and moment axes. In between these two extremes, three gradient categories: big, medium, and small for both positive and negative signals were issued to inform about the relative gradient change. In total, 9 primitive classifications were derived.

B. Composites

The next layer was designed to extract action or motion compositions (MCs) by analyzing ordered-pair sequences of primitives. By studying the patterns in the ordered-pairs, action-level performance can be understood from the data. In total, seven basic MCs were derived: increase $(\uparrow force)$, decrease $(\downarrow force)$, constant (const. force), positive contact (pimp), negative contact (nimp), contact (pimp + nimp) and unstable motions. Positive gradients regardless of magnitude were paired as a single group 'Positive' gradients while the same was done with 'Negative' gradients.

Table I summarizes motion compositions classification based on primitive ordered-pairs. The table contains subtables representing five primitive groupings. The first primitive is in bold followed by a listing of possible primitives with corresponding motion compositions and labels to classify an MC. An example of the MC layer can be visualized in Fig. 5.

C. Low-Level Behaviors

The taxonomy's third layer considers MC ordered pairs along with contextual information such as signal duration and amplitude values to yield classifications. Eight LLB classifications were derived and labeled as: push, 'PS'; pull, 'PL'; contact, 'CT'; fixed, 'FX'; alignment, 'ALIGN'; shift, 'SH'; and noise, 'N'. The LLB formulation criteria is similar to those at the MC level. That is, for a pair of increase MCs labels, or decrease MCs labels, or constant MCs labels or adjust MCs labels; pull, push, fixed, or adjust LLBs are

assigned respectively. As for times in which the tool contacts the environment (whether pimp + nimp or vice-versa) a contact LLB is assigned. One major difference between the MC level and the LLB level is the introduction of a shifting behavior 'SH'. Shifts and alignments are similar but differ in that, whenever there are two contiguous adjustment compositions, if the second composite's amplitude is larger than the first, label it as 'SH' LLB, if smaller label it 'ALGN' LLB (see[8] for more details).

D. Refinement

Refinement stages are critical to the system acting as filters for both the system's MC and LLB layers. Each refinement stage cleans up MCs and LLBs generated during the initial algorithm pass. Each refinement stage analyzes each layer on the basis of three contexts: a composite's duration, a composite's amplitude value, and repeated composites:
- Time Context Filter: examines two contiguous composites (except for contacts). If one composite is much larger than another one, the smaller composite is considered trivial and is absorbed into the larger composite (see [8] for further details).
- Amplitude Context: the amplitude context compares two types of values (average and amplitude values) for contiguous composites and considers whether the composites can be merged. Similar average values indicate that both composites belong to the same region. Similar amplitude values increase the likelihood of representing the same event. The criteria for MCs and LLBs is defined below.

For the MCs: (i) 'i-d' pairs in either ordered are merged into adjustments, (ii) 'i-k' or 'd-k' pairs in either order are merged as increases or decreases respectively,

For the LLBs: (i) 'PS-PL' pairs in either order are merged into 'ALIGN,' (ii) 'SH-ALIGN' in either order where the second composite has a smaller amplitude, are merged as 'ALIGN,' and (iii) 'ALIGN-PS||PL' or 'SH-PS||PL' or in either order are merged as an alignment'ALIGN.'
- Repeated Context: If two contiguous composites have repeated labels, they are merged as one, and their data structures updated accordingly.

The refinement cycle thus finds patterns across composites that are not evident when primitive or composite ordered-pairs are initially analyzed. Furthermore, as the refinement cycles are run at the different abstraction levels, refinements at higher abstraction levels filter hidden patterns at lower abstraction levels. The refinement filter was run three times in each layer to merge most disjoint composites. Fig. 5 already shows post-refinement results.

E. High-Level Behaviors

The fifth layer contextualizes state reasoning by asking: "What LLBs **principally** describe the automata Rotation, Snap, and Mating states?" The HLB characterizes what LLB configuration, across all six force axes, define a given HLB. Currently we only focus on defining the HLBs that yield an assembly successful (we will later focus on failure cases). The selection of key LLBs is connected with the controller template and reference parameter selection in the Pivot Approach as well as the local task coordinate frame selection for the task (see Sec. III. The probabilistic aspects of the layer will be explained in Sec. V. Table II summarizes what LLBs characterize the above-stated desired HLBs:

TABLE II
COMBINATION OF NECESSARY LLB'S TO ASCERTAIN THE PRESENCE OF HLB'S.

HLB	LLB	Force Axis
Rotation	FX	Fx
	FX	Fz
	FX	My
Snap Insertion	CT	Fx,My
	Align||Fx	Fy-Mx,Mz
Mating	FX	Fx-Mz

F. Verification Layer

The fifth and last layer of the taxonomy assesses the outcome of a task. Previously the assessment was made purely by the presence of LLBs, in our current work it is performed based on threshold values that were derived from HLB beliefs. Sec. V-C will describe the process in detail.

TABLE I
MOTION COMPOSITIONS ACCORDING TO PRIMITIVE PAIRS.

Combination	Category	Label	Combination	Category	Label	Combination	Category	Label
Pos. Impulse			**Pimp**			**Constant**		
Negative	adjustment	a	Positive	pos contact	pc	Positive	increase	i
Positive	increase	i	Negative	pos contact	pc	Negative	decrease	d
Constant	increase	i	Constant	pos contact	pc	Constant	constant	k
Pimp	pos contact	pc	Pimp	unstable	u	Pimp	pos contact	pc
Nimp	neg contact	nc	Nimp	contact	c	Nimp	neg contact	nc
Neg. Impulse			**Nimp**					
Positive	adjustment	a	Positive	neg contact	nc			
Negative	decrease	d	Negative:	neg contact	nc			
Constant	decrease	d	Constant	neg contact	nc			
Pimp	pos contact	pc	Pimp	contact	c			
Nimp	neg contact	nc	Nimp	unstable	u			

V. BAYESIAN FILTERING

The RCBHT detected key LLB presence for all FT axes to characterize a given desired HLB. However, a number of factors render the assessment uncertain: (i) the assembly's sensory data is noisy, (ii) the RCBHT is limited to discriminate with certainty what LLBs are occurring, and (iii) key LLBs may only appear for a short amount of time during a state of the automata (see Fig. 5). For this reason, Bayesian filtering was used to deal with uncertainty at the RCBHT's third layer. BF's have been typically used to localize mobile robots and, more recently, features in manipulation tasks [10], [4]. We have opted to implement a BF algorithm in the context of the RCBHT to deal with the aforementioned uncertainties while easing computational complexity and express the task's high-level state as a probabilistic result. In our work, the Bayesian filter will model LLBs as the task's state. While LLBs are an indirect measurement of the state the selection reduces computational complexity and renders the BF as a viable approach. The latter will compute the posterior distribution (or belief) of a state x_t at time t, conditioned on all past measurements $z_{1:t}$ and all past motion commands $u_{1:t}$. The state belief is computed for each of the six LLBs presented in Sec. IV-C at every time step for each FT axis.

In BFs, a Markovian assumption is used which states that the knowledge about the current states and parameters suffices to make predictions about future states and measurements. The belief in a state x_t at time t is represented as: $bel(x_t) = p(x_t|z_{1:t})$. BFs use an update rule to recursively update the belief in the current state from the belief in the previous step: $\bar{bel}(x_t)$. The algorithm's first iteration requires an initial belief $bel(x_0)$ as a boundary condition. The update step can be better understood when decomposed in two steps: the prediction step and the correction step.

A. The Prediction Step

Predicts the state at time t by using a 'system model' in the previous time step $t-1$. The system model for our discrete system is expressed as:

$$P(x_t|z_{1:t-1}, u_{1:t-1}) = \sum_{x_{t-1}} P(x_t|x_{t-1}, u_{t-1}) P(x_{t-1}|z_{1:t-1}, u_{1:t-1}) \quad (1)$$

The system model is the sum of the products between state transition probabilities and priors probabilities for the previous state. While Bayesian filtering provides an optimal solution to estimating uncertainty, it does not explain how a probability model can be represented.

In terms of representations, LLB prior probabilities $P(llb)$ were defined as a function of their *cumulative duration* d with respect to the duration T of a single automata state s in a single FT axis a, such that at the completion of any automata state: $P(llb_{s,a}) = d_{s,a}/T_{s,a}$.

With respect to the state transition model, a 6x6 matrix of transition-counts between the six LLBs was computed for each automata state for each FT axis yielding 18 matrices. State transition probabilities were computed as a fraction of counts per LLB with respect to the total number of transition counts per automata state per FT axes. Seven training assembly trials were used to compute average values for: prior and state transition probabilities for each FT axis and each automata state.

The selection of *Cumulative Durations* as the measured feature in the pRCBHT reflects two aspects: (i) the longer an LLB lasts during an automata state, the more likely it is to dominate the behavior for that automata state, and (ii) due to disturbances or limitations in the RCBHT an LLB in an automata state may switch LLBs but return to the LLB of interest at a later time.

B. The Correction Step

The Correction Step updates the posterior by updating the prior belief (it corrects it) by incorporating the observed measurement, z_t, likelihood's and motor command's u_t such that:

$$P(x_t|z_{1:t}, u_{1:t-1}) = \eta P(z_t|x_t) P(x_t|z_{1:t-1}, u_{1:t-1}), \quad (2)$$

where, η is a normalization factor that guarantees that the probability sum does not exceed 1. Measurements represent the cumulative duration feature explained for prior probabilities. In the measurement's case, z_t measures the cumulative duration upto that point in time for a given state s for a given force axis a. The measurement likelihood was computed using a Gaussian distribution where z_t is the cumulative duration and x_t is the llb for which we are computing the belief: $P(z_t|x_t) = \mathcal{N}(z_t; \hat{x}_t, \sigma^2)$. The mean and the variance where calculated for each LLB for each automata state for each force axis by using seven trial assemblies.

In effect, when the cumulative duration of the selected LLB approaches the mean cumulative duration (for any of the existing six LLBs) computed from the training data, the more likely it is to be the correct measurement. For example, compare the Fy axis for Fig. 5 and Fig. 6. Notice, how in the former the FX LLB's duration in the Rotation state occupies almost the entire state. Then in the latter figure, the likelihood that FX is actually the measured LLB is 100% for most of the Rotation state's duration. The correction step and prediction step can be re-written in terms of a selected LLB llb_i posterior (or belief) as:

$$P(llb_{i,t}|z_{llb,1:t}, u_{1:t-1}) = \eta P(z_{llb,t}|llb_{i,t}) \sum_{j,t-1} P(llb_{i,t}|llb_{j,t-1}, u_{t-1}) \; P(llb_{j,t-1}|z_{1:t-1}, u_{1:t-1})). \quad (3)$$

Fig. 6 shows the belief over all LLBs per FT axis per automata state for a test assembly task. Note the correlation between this belief plot and the sequence of LLBs in Fig. 5.

C. Probabilistic HLB

As mentioned in Sec. IV-E, the HLB layer is a function of key selected LLBs (refer to Table II). Recall that Bayesian filtering was implemented to obtain a belief about the LLB states for each FT axis across the three automata states. The HLB layer then computes the joint probabilities of key LLBs that occur in all or any of the six FT axes.

Fig. 7 shows the HLB belief per each automata state. Five training trials were used to derive an average expectation of HLBs per automata state (thicker plot lines in magenta, blue, and black).

D. Probabilistic Snap Verification Layer

The Snap Verification layer (SVL) used a scheme in which three thresholds were generated to determine whether each automata state was successful. The thresholds work in a cumulative way such that the last threshold effectively determines whether the task was successful as a whole. The scheme in the SVL layer is designed to provide an intuitive assessment of the task's outcome such that with the success of each subsequent HLB, the likelihood of success of task (and with it the thresholds) also grow (see to Fig. 7 for reference).

The SVL thresholds were derived by using a weighted average function over training data HLB beliefs. The weighted function divides each of the automata states HLB beliefs by the total automata state number. The algorithm then adds the maximum value of the previous HLB belief over a *transition window* to the next HLB belief. A transitional window is necessary given that during automata state transitions, LLB beliefs change and decrease in likelihood. Taking the maximum value over the transitional window ensures that the drop in probability does not hurt the expected (next HLB's) likelihood (see dotted boxes in Fig. 7).

This algorithm shows increased likelihood success levels with each succeeding HLB state; *i.e.* if there was 100% probability that each automata state succeeded, the SVL layer would show the likelihood of Rotation at $1/3$, Snap at $2/3$, and Mat at 1. If, on the other hand, Rotation succeeds but the others do not, the SVL layer would show an success belief at $1/3$.

Fig. 7 presents three pieces of data: (i) the HLB Beliefs for five successful training examples, (ii) the averaged HLB belief values, and (iii) the weighted averaged values from which the SVL extracts threshold values at the automata state's end. Note that near transition areas between belief states, belief state's fluctuate drastically. Thus, a transition window is considered, in which the average of all belief state values is computed and used as the effective threshold. The Rotation state success threshold is marked at about 27%, the Snap's threshold is marked at about 51%, and the Mating's threshold is marked at about 74%.

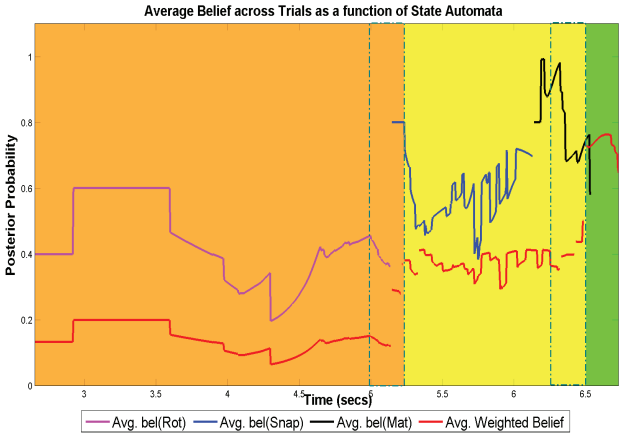

Fig. 7. The HLB and SVL layers: HLB belief per automata state is shown by three non-red lines averaged over five trials. The magenta line shows the Rotation belief, the blue line represents the Snap belief, and the black line represents the Mating belief. The weighted belief function used to extract success thresholds in the SVL layer is shown in red. The latter equals $1/3$ of the value of the automata state averaged beliefs. The dotted green lines represent transitional windows used to compute success thresholds for each automata state.

VI. Results

After the training phase, seven more test snap assembly trials were executed in simulation using the HIRO robot and the 4-snap camera mold. For each of the seven trials, the SVL correctly assessed the outcome of all seven assembly tasks as a function of the representative LLB belief. Out of the seven assembly tasks five succeeded and two failed.

VII. Discussion

The pRCBHT effectively computed the state belief for LLBs and HLBs. The SVL derived-thresholds effectively classified the outcomes of test assembly trials in our work. In our previous work, the RCBHT declared a task successful if the presence of key LLBs existed, even if that presence was very small. By introducing Bayesian filtering, the system computed likelihoods for belief states based on gaussian measurement models and averaged prior and transition probability models over history making the system more robust against the impact of noisy FT signals and the presence of short-lived LLBs.

The probabilistic outcome also yields more intuitive and granular state awareness than before. The LLB belief state conveys which behavior dominates a task at a specific time in a given automata state in a given axis. This level of granularity in conjunction with the SVL scheme, can allow us to classify defective assemblies contextualized by automata state and force axis, and fix them by issuing corrective commands as feedback to the controller. This will effectively close the loop between state estimation and corrective motion enabling snap sensing for parts of complex geometry.

The current work is significant as it establishes a framework by which to perform state reasoning on difficult manipulation tasks like snap assemblies. Our research has built

on the use of a motion-constraining strategy that yields: (a) similar-patterned signals over trials to facilitate the interpretation of FT data; and (b) intuitive automata states and transitions that work for cantilever-snap geometries of varying complexity. The control strategy works in concert with the RCBHT snap verification system, which is an alternative to the contact state graph approach for compliant motions. The RCBHT seems to be effective in reducing the problem's dimensionality while simultaneously offering a high-level of intuition about the task's state. Furthermore, by introducing a probabilistic nature to the system, uncertainties are mitigated with Bayesian calculus which yields temporal belief representations.

There are however, a number of limitations and challenges in our system. The first is the possibility of generating false-positives results. While the SVL layer correctly assessed of the outcome of all seven test trials; the two trials assessed as failures were in fact successful assemblies. The problem is not due to probabilistic miscalculations but on the fact that LLB compositions result from primitives and motion compositions classification as part of the first two taxonomy layers. On two occasions the RCBHT generated an 'AL' LLB tag for a task segment where an 'FX' corresponded. This limitation could be addressed by introducing probabilistic reasoning into the first two layers of the system and also by executing an optimization routine during training to minimize poor classifications.

Another limitation lies in that the probability models for priors, transition models and the measurement model's gaussian noise, depend on training data examples. In our work we used seven separate data samples to train and to test the system. The probability models (priors and transitions) can be improved by increasing the population size. Adjusting these values over time based on experience should improve statistical accuracy.

A very significant challenge, and one that requires further study, is how to work with transitions. With respect to the controllers and the control policy, transitions are fired when contact forces pass a threshold. Such methodology has the complication that if a threshold is set too low, noise could trigger the next state, if the threshold is too high, state estimation suffers near transition ranges since the estimation depends on behavior assumptions that begin to change at the edge of the transition. Learning how to deal with transition uncertainty will be an important aspect of snap sensing.

Another complication in this work is caused by the OpenHRP simulation environment. The latter was designed for walking humanoids [9]. The physics engine struggles to identify contact on the magnitude scale of assembly parts. On occasion, some penetration exists between parts thus affecting the expected LLB performance in different automata states.

As part of our future work, we will improve the probabilistic model by using the latest results in an assembly to update prior and transition probabilities in our model. We also hope to characterize error states and contextualize them by state. That is, we want to understand what errors are more likely for each automata state, classify them, and generate corrective motions that can render a defective assembly into a successful one. With this goal in mind, we will implement an online version of the system on the actual HIRO robot.

VIII. CONCLUSION

In this work a probabilistic snap state estimation system was presented. The system extended our previous system by embedding a bayesian filter in the taxonomy. The latter worked in concert with the motion constraining Pivot Approach as part of our ongoing work to design a framework for cantilever-snap assemblies of varying geometrical complexity. Probabilistic reasoning rendered our system more robust by computing outcome likelihoods instead of simply looking for the presence of key LLBs to assess the outcome of an assembly. Our system correctly evaluated the outcome of all simulated test snap assembly trials using Kawada's HIRO robot and a four-snap cantilever camera mold part. Our future work consists in the characterization of error states and generate corrective motions to generate the equivalent concept of active sensing to snap, what we call 'snap sensing'. We will also focus on implementing our system in real-time.

REFERENCES

[1] B. McCarragher, "Force sensing from human demonstration using a hybrid dynamical model and qualitative reasoning," in *IEEE Intl. Conf. on Robotics and Automation*, 1994, pp. 557–563.

[2] H. Asada, "Teaching and learning of compliance using neural nets: Representation and generation of non-linear compliance," in *IEEE Int'l Conf. on Robotics and Automation*, 1990.

[3] M. Skubic and R. Volz, "Learning force sensory patterns and skills from human demonstration," in *IEEE Intl. Conf. on Robotics and Automation*, 1997, pp. 284–290.

[4] W. Meeussen, J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. D. Schutter, "Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks," *IEEE Trans. on Robotics*, vol. 23.2, pp. 218–231, 2007.

[5] K. Gadeyne, T. Lefebvre, and H. Bruyninckx, "Bayesian hybrid model state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation," *Int. J. Robotics Research*, vol. 24(8), p. 615630, 2005.

[6] J. Rojas, K. Harada, H. Onda, N. Yamanobe, E. Yoshida, K. Nagata, and Y. Kawai, "Cantilever snap assembly automation using a constraint-based pivot approach," in *IEEE Intl. Conf. on Mechatr. & Automation*, 2012.

[7] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson, "Force controlled assembly of emergency stop button," in *IEEE Int'l Conf. on Robotics & Automation*, 2011.

[8] J. Rojas, K. Harada, H. Onda, N. Yamanobe, E. Yoshida, K. Nagata, and Y. Kawai, "A relative-change-based hierarchical taxonomy for cantilever-snap assembly verification (in print)," in *IEEE Intl Conf. on Robots and Systems*, 2012.

[9] F. Kanehiro, H. Hirukawana, and S. Kajita, "Openhrp: Open architecture humanoid robotics platform," *Intl. J. of Robotics Res.*, vol. 23-2, pp. 155–165, 2004.

[10] R. Platt, F. Permenter, and J. Pfeiffer, "Using bayesian filtering to localize flexible materials during manipulation," *IEEE Transaction on Robotics*, vol. 27, no. 3, pp. 586–598, 2011.

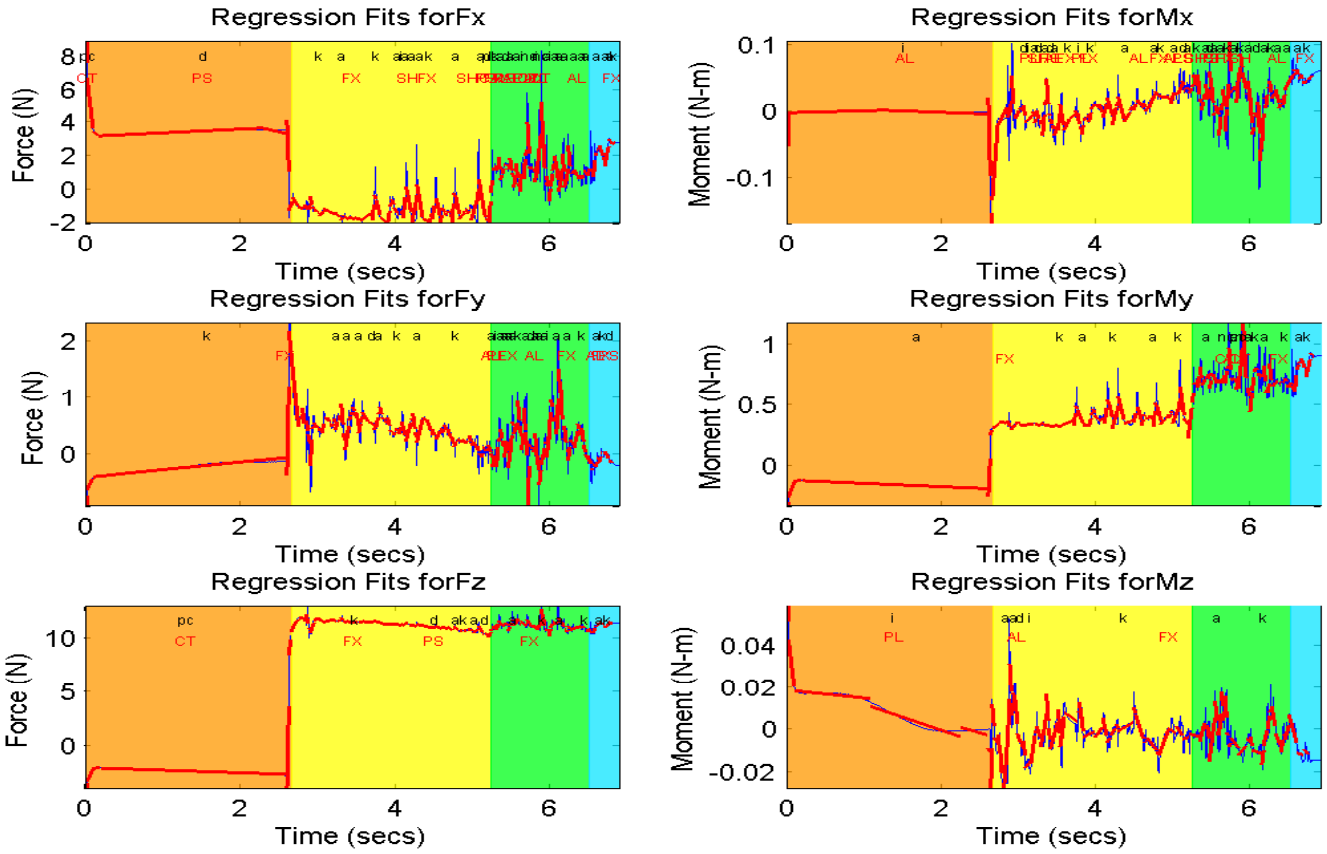

Fig. 5. The RCBHT's first three layers are presented in this figure. The first (Primitives) layer consists of linear fits shown by red line segments. The second (Motion Composition's) layer are shown in black text. The third (Low-Level Behaviors) layer are shown in red text labels. The colored boxes represent the Pivot Approach's stages: Rotation, Snap, and Mating. The figure shows the robot's state throughout the task.

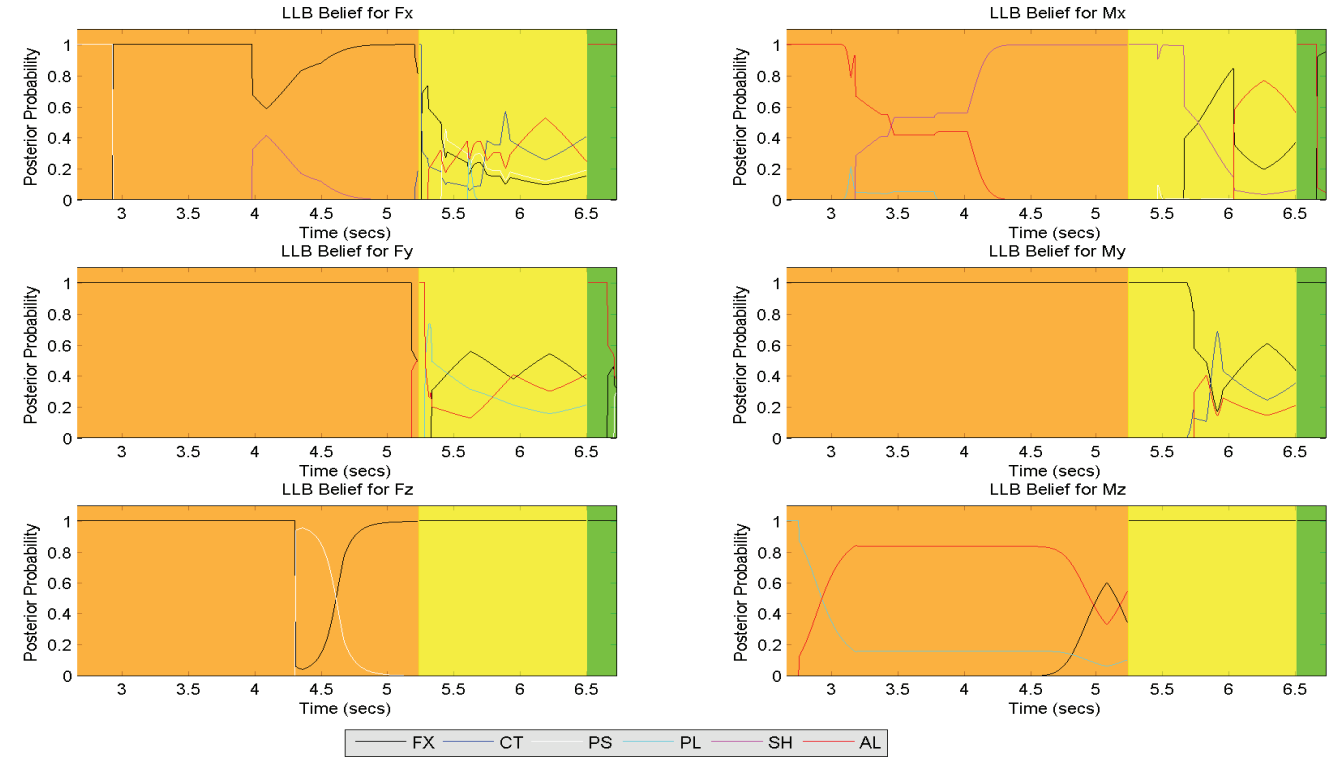

Fig. 6. LLB belief's are computed for each of the six behaviors in the LLB layer of the RCBHT. Each LLB's belief was computed with a Bayesian that used the behavior's average cumulative duration. The belief represents the likelihood that a behavior dominates the task at any moment in time.