Towards snap sensing

Juan Rojas*, Kensuke Harada, Hiromu Onda,
Natsuki Yamanobe, Eiichi Yoshida, Kazuyuki Nagata
and Yoshihiro Kawai

Intelligent Systems Research Institute, AIST,
Tsukuba, Ibaraki, 305-8568, Japan
E-mail: rojas70@gmail.com
E-mail: kensuke.harada@aist.go.jp
E-mail: onda@ni.aist.go.jp
E-mail: n-yamanobe@aist.go.jp
E-mail: e.yoshida@aist.go.jp
E-mail: k-nagata@aist.go.jp
E-mail: y.kawai@aist.go.jp
*Corresponding author

Abstract: Automating snap assemblies is highly desirable but challenging due to their varied geometrical configurations and elastic components. A key aspect to automating snap assemblies is robot state estimation and corrective motion generation, here defined as snap sensing. While progress is being made, there are yet no robust systems that allow for snap sensing. To this end we have integrated a framework that consists of a control strategy and control framework that generalises to cantilever snaps of varying geometrical complexity. We have also integrated a robot state verification method (RCBHT) that encodes FT data to yield high-level intuitive behaviours and perform output verification. Optimisation procedures and Bayesian filtering have been included in the RCBHT to increase robustness and granularity. The system provides belief states for higher level behaviours allowing probabilistic state estimation and outcome verification. In this work, preliminary assembly failure characterisation has been conducted and provides insights into assembly failure dynamics. The results, though still in simulation, are promising as the framework has effectively executed cantilever snap assemblies and robust robot state estimation with parts of varying complexity in two different robotic systems.

Keywords: active sensing; assembly; snap assembly; force control; failure detection; signal processing; probabilistic assembly; Bayesian filtering; hierarchical taxonomy.

Reference to this paper should be made as follows: Rojas, J., Harada, K., Onda, H., Yamanobe, N., Yoshida, E., Nagata, K. and Kawai, Y. (2013) 'Towards snap sensing', *Int. J. Mechatronics and Automation*, Vol. 3, No. 2, pp.69–93.

Biographical notes: Juan Rojas is a Post-doctoral Associate at Japan's National Institute of Advanced Science and Technology (AIST) where he is currently researching robotic automation through signal interpretation and probabilistic learning. He received his BS, MS, and PhD in Electrical and Computer Engineering from Vanderbilt University in 2002, 2004, and 2009 respectively. From 2009–2011, he served as a Visiting Scholar at Sun Yat-Sen University in China. His research interests include long-term and short-term automation, cooperation and coordination of robot teams and human-robot interaction. He is also very interested in humanitarian applications and endeavours to find solutions to problems through technology and service.

Kensuke Harada received his BS, MS, and the Doctor degree in Mechanical Engineering from Kyoto University in 1992, 1994, and 1997, respectively. He worked as a Research Associate at Hiroshima University from 1997 to 2002. From 2002, he has been working as a Research Scientist at the National Institute of Advanced Industrial Science and Technology (AIST). His research interest includes mechanics and control of robot manipulators and robot hands, biped locomotion, and motion planning of robotic systems.

Hiromu Onda received his BEng and MEng from University of Tokyo in 1986 and 1988, respectively, and his PhD in Information Engineering, from University of Tokyo, Japan in 2001. From 1988 to 2000, he was a Research Scientist and then a Senior Research Scientist at the Electrotechnical Laboratory, MITI, Japan. Since 2000, he has been a Senior Research Scientist at the National Institute of Advanced Industrial Science and Technology (AIST), METI. His current research interests include programming by demonstration in VR, skill transfer, and ubiquitous robotics. He is a member of IEEE, RSJ, Japan SIAM, and JSME.

Natsuki Yamanobe received his ME and PhD on Precision Machinery Engineering from Graduate School of Engineering, the University of Tokyo in 2004 and 2007 respectively. Since 2007, she has been a Research Scientist at the Intelligent Systems Research Institute at Japan's National Institute of Advanced Industrial Science and Technology. Her main research interests are in robotic assembly. She is a member of IEEE, JSME, JSPE, and RSJ.

Eiichi Yoshida received his ME and PhD on Precision Machinery Engineering from Graduate School of Engineering, the University of Tokyo in 1993 and 1996 respectively. From 1990 to 1991, he was visiting research associate at Swiss Federal Institute of Technology at Lausanne (EPFL). In 1996, he joined former Mechanical Engineering Laboratory, later reorganised as National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. He served as a Co-Director of AIST/IS-CNRS/ST2I Joint French-Japanese Robotics Laboratory (JRL) at LAAS-CNRS, Toulouse, France, from 2004 to 2008. Since 2009, he is a Co-Director of CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan. His research interests include robot task and motion planning, modular robotic systems, and humanoid robots.

Kazuyuki Nagata received his BS and PhD in Engineering from Tohoku University, Japan in 1986, and 1999, respectively. He joined Tohoku National Industrial Research Institute (TNIRI) at former AIST of MITI, Japan in 1986, he assigned to Electrotechnical Laboratory (ETL) in 1991, and assigned to Planning Headquarters of AIST in 2001. He is currently a Senior Research Scientist with Intelligent Systems Research Institute of AIST. His current research interests include control of robotic manipulations, grasp planning, tactile sensing, and haptic interface. He is a member of the Japan Society of Mechanical Engineers (JSME), the Robotics Society of Japan (RSJ), and the Society of Instrument and Control Engineers (SICE).

Yoshihiro Kawai received his BE and ME from Nagoya University in 1987 and 1989. He was a Research Scientist at the Electrotechnical Laboratory, AIST, from 1989 to 1994, and a Senior Research Scientist from 1994 to 2001 (AIST was reorganised in 200l). He was a Senior Research Scientist at the Intelligent Systems Research Institute, AIST from 2001 to 2007. He has been a Research Group Leader in 2007. His research interests include computer vision and its application, e.g., factory automation systems, assistive systems for the visually disabled person, etc. He received a Best Video Award from the International Conference on Robotics and Automation in 2003. He is a member of the Robotics Society of Japan.

1 Introduction

An open research problem in manipulation and robotic assembly communities is the snap assembly automation. Snap assemblies are those that employ elastic snap parts. For snap assembly automation to be robust one must not only be able to implement a snapping mechanism but also understand potential failure patterns and recovery schemes. State estimation paired with corrective measures has been traditionally known as 'active-sensing' in the research community (Mihaylova et al., 2002).

For the last two decades, the closely related task of peg-in-hole assembly has benefited from active sensing. Active sensing is also desirable in snap assemblies and is defined here as 'snap sensing'. Progress is being made in sub-parts of the snap sensing problem; that is, in part's localisation (Platt et al., 2011), force controller and control strategy formulation (Stolt et al., 2011; Chin et al., 2003; Sayler and Dillmann, 2011); but not much literature yet exists for robot state reasoning for snap assembly.

In traditional peg-in-hole assemblies and manipulation, numerous methods have been used to estimate the robot's internal state including: qualitative reasoning (McCarragher, 1994); learning contacts and state transitions (Asada, 1990), and fuzzy logic (Skubic and Volz, 1997). More recently, probabilistic methods have been applied to generate more robust manipulation behaviours in the presence of uncertainty (Meeussen et al., 2007; Gadeyne et al., 2005).

With regard to snap assemblies, it is first important to acknowledge different types of snap fasteners classes, of which there are three main classes: cantilever, annular, and torsional (Soddhi and Sonnenberg, 1999). Each of them has their nuances but all of them are characterised by elastic components (Meitinger and Pfeiffer, 1995). Cantilever snaps, are the most common, and manufactured goods with this fastener can vary in the number of built-in snaps. Manufactured parts typically range from 1, to 2, to 4, or more cantilever snaps in their mechanism. For these reasons, there is a higher degree of complexity in applying traditional robot estimation methods across snap classes and various geometrical scenarios.

State estimation has been implemented for single-snap cantilever parts using probabilistic techniques in Stolt et al. (2011). However, there are yet no frameworks that generalise to snaps of varying complexity to perform snap sensing. Our focus is to design such a framework for both industrial use and personal robots. To this end, we have developed and integrated a series of processes composed of: a generalisable control strategy; flexible and scalable force controllers; and an intuitive signal processing system that aids in robot state estimation and that is aided by Bayesian filtering and optimisation routines to deal with uncertainty.

Previously in Rojas et al. (2012a), we developed the pivot approach (PA) control strategy for cantilever-snap type assemblies (for all abbreviations please refer to Section 14). The PA exploits snap parts' hardware design to constraint the task's motion and facilitate the assembly and signal interpretation processes (see Section 4). In Rojas et al. (2012d) 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 Section 6). In Rojas et al. (2012b), we implemented system calibration procedures that increase the robustness of the RCBHT system when used with new robots or snap parts. In Rojas et al. (2012c), we designed and embedded a Bayesian filter (BF) algorithm as part of the RCBHT to yield temporal belief representations for the top three layers of the taxonomy. The probabilistic RCBHT (pRCBHT) provided a more granular understanding of task behaviours, improving robot state estimation and facilitating assembly failure characterisation.

Each component of our framework has been shown to be a viable technology. We have sought to work towards the generalisability of the technology to different robotic systems and cantilever parts of varying complexity. Our methods have been applied to two robotic systems and two cantilever-snap mechanisms of varying complexity. More specifically, we have experimented with a Mitsubishi PA10 7 DoF manipulator and a two-snap cantilever-based camera pack, and an upper torso anthropomorphic robot HIRO with a four-snap cantilever-based camera pack. All results thus far have been in simulation. Current efforts are focused in realising a physical real-time implementation of the system.

The paper is organised as follows: Section 2 describes the experimental setup in detail. Section 4 describes the PA control strategy and Section 5 describes the control basis framework used in concert with the strategy to deploy a flexible and generalisable control scheme. Section 6 introduces the RCBHT snap assembly estimation system and Section 9 presents the probabilistic version of the latter. Section 10 presents preliminary findings on failure characterisation in snap assemblies. Finally, Section 11 presents relevant discussion points as well as future efforts and Section 12 summarises key aspects of our work. The Appendix includes a table with a summary of acronyms used in this work.

2 Experimental setup

For this work, two different robots and cantilever snap parts have been used to show the effectiveness of our framework in generalising snap assembly automation and robot state estimation. With all systems, robots were simulated using the OpenHRP environment (Kanehiro et al., 2004), CAD derived camera parts consisted of both male and female parts and 6 DoF FT sensors mounted on the robot wrist's were used.

Two experimental configurations were used. The first one consisted of a 7 DoF Mitsubishi PA-10 manipulator (Rojas et al., 2012a) with a female camera mold part rigidly held by a two-fingered gripper mounted on the wrist and a male part rigidly fixed to the ground. We name this experimental configuration the: female-active/male-passive FA-MP configuration. The male part consisted of two cantilever snaps and a vertically offset docking location as seen in Figure 1. In the second experimental configuration, the male-active/female-passive MA-FP configuration, a dual-arm 6 DoF anthropomorph HIRO robot was simulated. In this case, a male camera part was rigidly mounted on the robot's wrist, while the female part, was rigidly fixed to the ground as seen in Figure 2. Furthermore the male camera part consisted of four (not two) snaps. In the simulation, the world reference frame was located at both manipulator's bases. The tool point centre position and orientation, were determined with reference to the world coordinate frame *To*. On the other hand, the force and moment reference frames were determined with respect to the wrist's reference frame *Tw* as in Figures 1 and 2. Our approach currently assumes prior knowledge of the pivot dock location and orientation.

Figure 1 Experimental setup for a PA10 robot and a two-snap male camera part (see online version for colours)

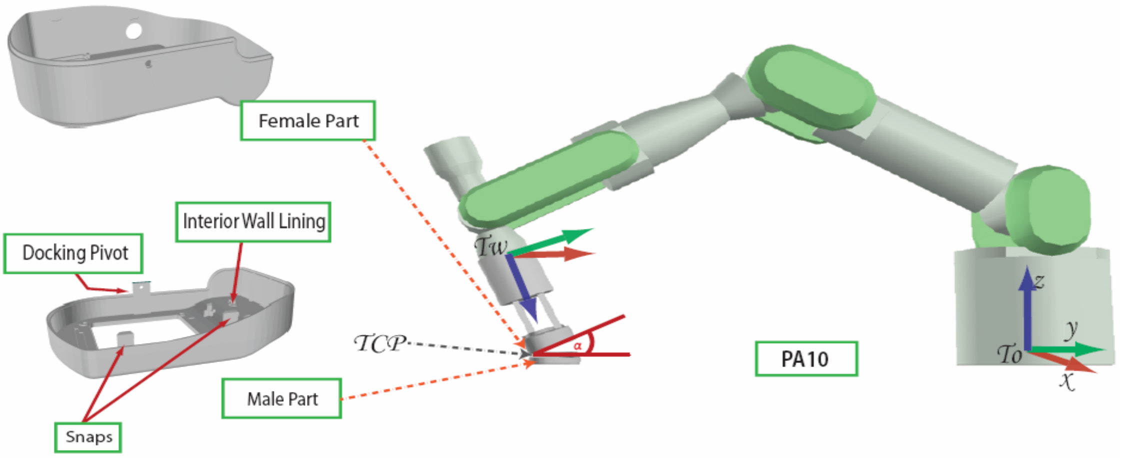
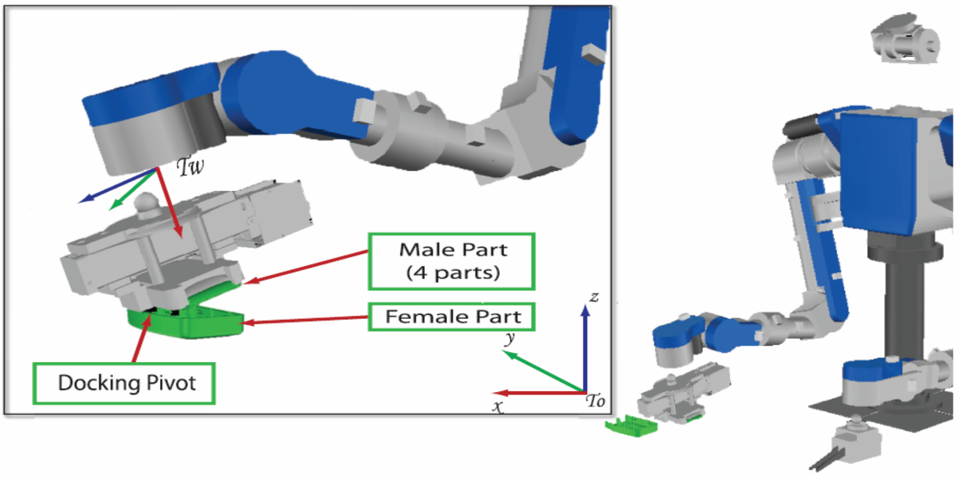


Figure 2 Experimental setup for a HIRO robot and a four-snap male camera part (see online version for colours)



3 Snap-parts classification

Industrial snap-fasteners use varying degrees of geometrical complexity. There are three types of snap joints: cantilever, annular, and torsional (Soddhi and Sonnenberg, 1999). The most common is the cantilever snap, which is an elastic protruding beam that snaps with a linear push. Cantilever-based parts often come in male and female counterparts, where the male part commonly consists of one, two, four, or even more beams (Hoffman, 2005). Cantilever-snap fastening mechanisms aims to mechanically constrain the assembly motion through the use of docking points and interior wall linings as seen in Figure 1. The constrained motion's goal is to facilitate parts entry by optimising the alignment of both male and female parts.

4 PA assembly strategy

The PA is an assembly control strategy with multiple goals:

1 to facilitate the interpretation of complex spatio-temporal FT signals by constraining the assembly motion and enabling similar-patterned signals to recur over assemblies

2 to yield high-level intuitive states that generalise to snap parts of differing geometry and whose state transitions can also generalise

3 to be deployed in conjunction with a flexible control framework that generalises for snap parts of differing geometrical complexities.

4.1 *Motion constrain design*

One of the most significant challenges in estimating the assembly task's robot state comes from the interpretation of the FT signals. Such signals are characterised by significant noise and in the case of cantilever snap assemblies, where there are more than one snap, complex spatio-temporal signal patterns are generated throughout the duration of the task in all six FT axes. For this reason, the snap assembly greatly benefits from a strategy that can minimise noise and complexity. To this end the PA strategy constraints the motion of the snap assembly to:

1 minimise spatio-temporal complexity

2 generate similar-patterned signals across assembly trials to facilitate their interpretation for robot state estimation.

Furthermore, the strategy conveniently exploits the built-in constraint motion mechanisms that characterise all manufactured snap-parts as discussed in Section 3. Manufactured cantilever-snaps of various complexities usually have a pivoting dock where male and female parts lock their position before attempting the parts insertion. Two and four-snap parts can be visualised in Figures 1 and 2 respectively.

4.2 PA states

The PA strategy decomposes the cantilever-snap assembly into five (or six) intuitive states that are generalisable for parts of varying complexities, that is: homing, approach, rotation, (alignment), insertion, and mating. The additional state is necessary when the docking pivot is not aligned with the wall ridge of the parts but vertically offset. That is, in Figure 1 the docking pivot is offset, while in Figure 2 the docking pivot is aligned with the wall ridge of both parts. The Alignment stage is necessary when the docking pivot is offset to optimise part's entry.

Recall that each of the PA control strategy states is to be connected to 'controller templates', to enable generalisations (Section 5). Our initial attempt at elaborating the strategy with the controller templates took place with the PA10 robot experimental configuration [i.e., with a 2-snaps part and offset pivoting dock (Rojas et al., 2012a)]. Later work refined the approach by replacing a force-position-based controller template in the rotation stage by a moment-force-based controller that was tested in the HIRO robot experimental configuration with a 4-snaps part, and a non-offset pivoting dock (Rojas et al., 2012c). Controller strategy details are described in the following two subsections.

4.2.1 PA10 control strategy configuration

The robot starts at the homing position and uses a position-based 'approach controller' ϕ_{APR}, to follow a curved trajectory which moves the tool-centre point to the docking position's neighbouring area to then slowly approach the pivoting dock. For the offset pivoting dock, contact is made at the top-edge of the docking position (see Figure 1) to achieve a height equal to that of the male snap beams. Such approach optimised parts' entry when the rotation state is finished. The female's part entry angle α ranges between $\alpha_1 \leq \alpha \leq \alpha_2$ where, $\alpha_2 < 90°$ and ensures that no joint limits are violated, and $\alpha_1 > 0°$ and ensures that the female part's does not immediately contact the male snaps. In our work, $\alpha = 22.5°$ as in Figure 1. After contact, state 2 uses a position-based 'rotation controller' π_{ROT}, to rotate the wrist in incremental steps until contact is made with the snaps. Upon contact, the reached location is near optimal and requires further alignment as provided by the moment-force-based 'alignment controller' π_{ALGN} in state 3. This controller sets its reference moment and force set points to < 0, 0, 0 > Newton and Newton-metres respectively (see Section 5). Such parameterisation drives the alignment of parts and sets the stage for the snap insertion. The latter was activated when a 1 mm dip was registered in the female snap upon alignment. The insertion also uses a moment-force-based 'insertion controller' π_{CI}. The difference between state 3 and 4 are the reference values for the controller. In this state, the desired forces are set to f_{ref} =< 0, 0, −0.5 > N to drive the insertion vertically downwards until no further motion is possible. Finally, state 5 activates a force-position 'mating controller' to maintain position and contact forces between the parts.

The PA10 state machine is shown in Figure 5, and the action sequence in Figure 3.

Figure 3 Sequential snapshots of the PA for the PA10 robot and two-snap parts are shown in this figure (see online version for colours)

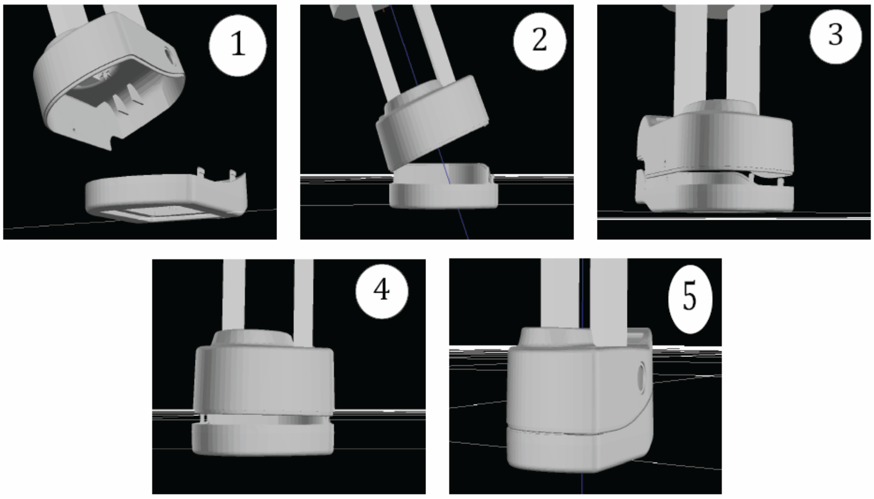
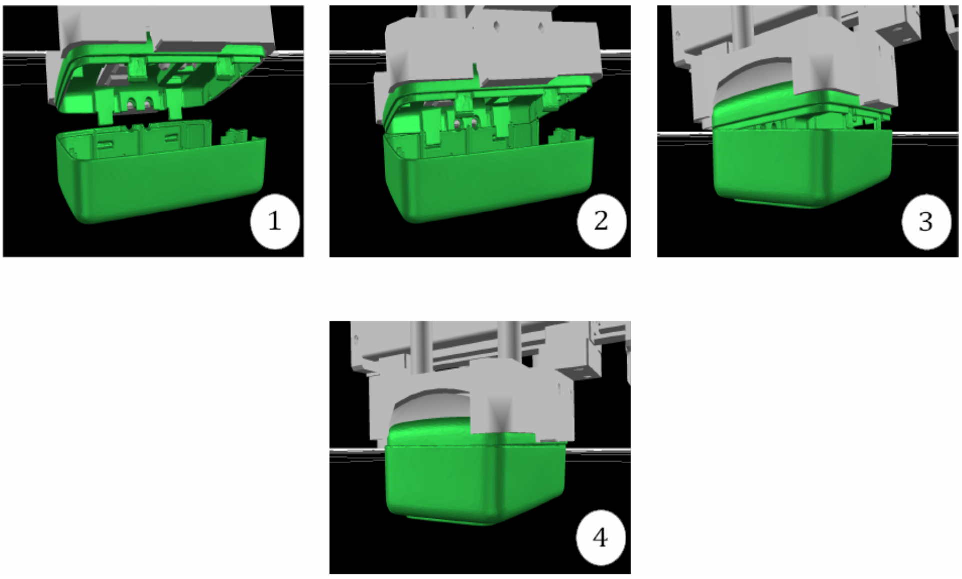


Figure 4 The PA with no offset docking position is composed of four non-trivial states: approach, rotation, snap, and mating (see online version for colours)



4.2.2 HIRO control strategy configuration

In our later work, we also attempted to use the PA for a camera part with four snaps and no offsetting docking pivot. In this work, the five standard PA states were used. All controller templates, except for rotation, were the same as in our previous work. The rotation controller was changed from a purely position controller to a compound moment-force controller π_{ROT} for three reasons:

1 We preferred to have a compound force-moment controller maintain the contact between the male and female parts during rotation instead of a purely positional controller. The dominant force controller ϕ_{mr} uses two reference parameters to push the female part vertically downwards (–z direction) and against the male part's wall (+x direction), while the y-direction reference is set to zero as no horizontal push is necessary: $\mathbf{f}_{ref}$ = {10, 0, 0.25} N.

2 Similarly, instead of commanding a position-based rotation, this controller drives the rotation of the male part as response to moment parameters along the y-direction for the subordinate moment controller ϕ_{mr}, such that $\mathbf{m}_{ref}$ = {0, 20, 0} Nm.

All reference parameters were applied in world coordinates.

Note that the coordinate frame references used in this experimental configuration have a direct effect on how our signal processing will work as explained in Sections 6, 9, and 11).

Finally, transition parameters are presented. The Approach state transitions to the rotation state when the x-force exceeds 9 N. The rotation state transitions to the snap state when the y-moment exceeds 0.60 Nm. Establishing transition thresholds is not trivial and challenges are discussed in Section 11.

5 Controller templates

Controller templates are control units that work in conjunction with each state in the PA. Such controller templates are required to be part of a flexible control framework that allows the templates to be modified easily to adapt to varying degrees of geometrical complexity within the cantilever-snap class. To this end the control basis framework was selected as a control paradigm to work with.

The control basis flexibly but systematically builds and modifies controllers. The approach decomposes a control problem into a set of asymptotically stable modular control elements that can be combined and flexibly rearranged to achieve desired strategies. The framework can combine controllers through the use of a null space operator (Rojas and Peters, 2012) and can sequentially combine primitive or compound controllers. Furthermore, the approach methodically determines which sensor inputs and motor actuators each controllers should use (not all sensor and motor resources need to be used every time) as well as the types of transformations necessary to transform incoming and outgoing data to the right space. In effect, the framework uses a small set of basis controllers and combines them in ways to effectively create sets of instructions or objectives to achieve a wide range of behaviours (Brock et al., 2005).

This approach was chosen as a foundational framework upon which we can develop a variety control strategies across industrial and humanoid robots to perform snap assemblies on different snap classes, each of which will require different approaches and controllers due to their differing geometrical properties.

5.1 Mathematical derivation

A basic mathematical definition is presented here for the control basis. For further details see Rojas and Peters (2012). Primitive controllers ϕ_i, where $i = 1 \sim m$, are elements in a basis set of controllers, Φ, such that $\phi_i \in \Phi$. A primitive controller optimises a partitioned portion of a designated control space (like joint angle space or Cartesian force space) and can be understood as the minimisation of a discrete basin of attraction. The basins of attraction are formulated through artificial potential functions defined over a typed domain, X_i, which are defined as the square of the error, where $\phi_i(\rho) = \rho^T\rho$ and ρ, is the difference between a reference input and a plant input, $\rho = \mathbf{q}_{ref} - \mathbf{q}_{des}$, at every time step.

Each controller reaches its objective by performing greedy descent, $\nabla\phi_i$, on the artificial potential function, while engaging selected sensor inputs and motor outputs. The surface potential minimisation in a specified domain space, X_i, over time is defined as:

$$\nabla_{x_i}\phi_i = \frac{\partial\phi_i}{\partial t}. \tag{1}$$

Each basis controller is bound to a selected subset of sensor input resources $\gamma_j \in \Gamma_j$ and output motor resources $\gamma_k \in \Gamma_k$ relevant to the task. Input and output signals are processed through sensor transforms (i.e., forward kinematics), s_j, and effector transforms (i.e., Jacobian), e_k to ensure that a task is guaranteed to operate within the region of a corresponding basis.

The closed-loop controller is implemented then, when the error between the incoming sensor information and the reference position is minimised within the discrete artificial potential basin, $\nabla_{x_i}\phi_i(\mathbf{x}_{ref} - s_j(\Gamma_j))$, and the gradient result is mapped to the output configuration space through an effector transform, $e_k(\Gamma_l)$. Given that the input data is of the same domain type as the artificial potential function, and the effector transform is of the same dimensions as the potential function, the controller's output, $\nabla_{y_k}\phi_i$, is defined as:

$$\nabla_{y_k}\phi_i = e_k\left(\Gamma_l\right)^T \nabla_{x_i}\phi_i\left(\mathbf{x}_{ref} - s_j\left(\Gamma_j\right)\right). \tag{2}$$

For convenience, the above expression is expressed in simplified notation as $\phi_i \,\big|_{e_k(\Gamma_l)}^{s_j(\Gamma_j)} (\mathbf{x}_{ref})$. If the controller has zero reference, then it can be omitted: $\phi_i \,\big|_{e_k(\Gamma_l)}^{s_j(\Gamma_j)}$. To concurrently optimise multiple goals, secondary control updates are projected onto the nullspace of primary control updates. This relationship is expressed in a compound controller π as having the secondary controller ϕ_2 be *subject-to* the primary controller ϕ_1, and is expressed as:

$$\nabla_y\left(\phi_2 \lhd \phi_1\right) = \nabla_y\phi_1 + \mathcal{N}\left(\nabla_y\phi_1^T\right)\nabla_y\phi_2, \tag{3}$$

where $\mathcal{N}(\nabla_y\phi_1^T) \equiv I - (\nabla_y\phi_1^T) + (\nabla_y\phi_1^T)$, and I, is the identity matrix, y is an n-dimensional space, and the nullspace of $\nabla_y\phi_1^T$ is a $(n - 1)$ dimensional space orthogonal to the direction of steepest descent (Platt, 2006). For convenience, equation (3) is written as $\pi_k : \phi_2 \lhd \phi_1$. Through nullspace composition techniques there is no need to specify how control resources will be shared across sub-controllers as long as the same control resources are used.

5.2 Assembly control basis set

The following primitive and compound controllers were implemented for both a Mitsubishi PA-10 serial link manipulator with a typical shoulder-elbow-wrist 7 DoF configuration and a HIRO anthropomorph with two 6 DoF arms. Position, force, and moment primitives are first introduced, and then references to how they are used in the assembly approach.

5.2.1 The position primitive

The position controller is based on the Jacobian transpose control method, where at each cycle, joint displacements are updated according to:

$$\Delta\mathbf{q} = J^T K_p \mathbf{e}, \tag{4}$$

where $\triangle\mathbf{q} \in \mathcal{R}^{7\times1}$ is a displacement of joint angles, $J^T \in \mathcal{R}^{7\times6}$ is the manipulator Jacobian, $K_p \in \mathcal{R}^{7\times7}$ is the position gain, and $\mathbf{e} \in \mathcal{R}^{6\times1}$ is the error in Cartesian positions.

For the position primitive, the sensor transform s_{pr} is the identity and operates on all joints and conveniently represented as: $s_{pr}(\gamma_{joint\_pos})$. The effector transform e_{pr} is the Jacobian transpose and effects torque updates in all joint motors: $e_{pr}(\gamma_{joint})$.

The square Cartesian error is used as the error function: $\phi_p = \frac{1}{2}K_p\mathbf{e}^T\mathbf{e}$, such that the gradient is $\nabla_{\mathrm{x}}\phi_p = K_p\mathbf{e}$. The basis controller can thus be defined as:

$$\nabla_q\phi_p = e_{pr}\left(\Gamma_{joint}\right)^T \nabla_{x_p}\phi_p\left(\mathbf{x}_{ref} - s_{pr}\left(\Gamma_I\right)\right), \tag{5}$$

or, more succinctly as $\phi_p \,\big|_{e_{pr}(\Gamma_{joint})}^{s_{pr}(\Gamma_I)} (\mathbf{x}_{ref})$.

Note that for our current work, a position controller was used in both the approach and rotation stages of the PA (see details in Section 4.2). Each controller is provided with a different trajectory point set and are referred to independently as the *approach controller* ϕ_{APR} and the rotation controller ϕ_{ROT}.

5.2.2 The moment and force primitives

Two controllers, force and moment primitives, update joint angle configurations so as to apply desired forces or moments. The force controller updates the end-effecter's location while the moment controller updates its pose.

$$\Delta\mathbf{q}_{1-7} = K_j^{-1} J^T K_f\left(\mathbf{f}_{ref} - \mathbf{f}\right), \tag{6}$$

$$\Delta \mathbf{q}_{1-7} = K_j^{-1} J^T K_m \left(\mathbf{m}_{ref} - \mathbf{m}\right), \tag{7}$$

where ($\mathbf{f}_{ref} - \mathbf{f}$) and ($\mathbf{m}_{ref} - \mathbf{m}$) are the force and moment errors conformed by the first three and last three elements in a $\mathcal{R}^{6\times1}$ vector respectively; and, K_f and K_m are the diagonal elements of a positive definite matrix $\mathcal{R}^{6\times6}$ that multiplied by the Jacobian transpose $J^T \in \mathcal{R}^{7\times6}$ generate torque updates for the appropriate joint configurations. The inverse of K_j and the other gains are precomputed to generate corresponding joint angle updates for each control cycle.

The force and moment residual controllers have sensor transforms $s_{fr}(\gamma_{\mathrm{force}})$ and $s_{mr}(\gamma_{moment})$ that return the F/T sensor data respectively. The artificial potential functions for the force and moment residual functions are proportional to the square of their errors:

$$\phi_{fr} = \frac{1}{2} K_f \left(\mathbf{f}_{ref} - \mathbf{f}\right)^2, \quad \phi_{mr} = \frac{1}{2} K_m \left(\mathbf{m}_{ref} - \mathbf{m}\right), \tag{8}$$

and they are differentiated with respect to their joint angle configurations to displace the trusses and minimise residuals:

$$\nabla_q \phi_{fr} = -K_f \left(\mathbf{f}_{ref} - \mathbf{f}\right)^2, \quad \nabla \phi_{mr} = -K_m \left(\mathbf{m}_{ref} - \mathbf{m}\right). \tag{9}$$

The controllers also have effector transforms $e_{fr}(\gamma_{torque})$ and $e_{mr}(\gamma_{torque})$ that converts appropriate force updates into joint torque updates by multiplying the inverse position gains and Jacobian transpose: $e_{fr}(\gamma_{torque}) = K_j^{-1}(J_{\gamma 1}, \ldots, J_{\gamma|\Gamma_k|})^T$ and $e_{mr}(\gamma_{torque}) = K_j^{-1}(J_{\gamma 1}, \ldots, J_{\gamma|\Gamma_l|})^T$ to produce the following primitive controllers:

$$\begin{aligned} &\phi_{mr}\Big|_{e_{mr}(\gamma_{torque})}^{s_{mr}(\gamma_{moment})}: \\ &\nabla_q \phi_{mr} = e_{mr}\left(\gamma_{torque}\right)^T \nabla_m \phi_{mr}\left(\mathbf{m}_{ref} - s_{mr}\left(\gamma_{moment}\right)\right). \end{aligned} \tag{10}$$

$$\begin{aligned} &\phi_{fr}\Big|_{e_{fr}(\gamma_{torque})}^{s_{fr}(\gamma_{force})}: \\ &\nabla_q \phi_{fr} = e_{fr}\left(\gamma_{torque}\right)^T \nabla_f \phi_{fr}\left(\mathbf{f}_{ref} - s_{fr}\left(\gamma_{force}\right)\right). \end{aligned} \tag{11}$$

In the subsequent subsections, compound controllers are introduced.

5.2.3 Rotation controller

The rotation controller π_{ROT} is composed of 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 optimise 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: $\mathbf{f}_{\mathrm{ref}} = \{10, 0, 0.25\}$ 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.

$$\pi_{ROT} = \phi_{fr}\Big|_{e_{fr}(\gamma_{torque})}^{s_{fr}(\gamma_{joint\_pos})}\left(\mathbf{f}_{ref}\right) \lhd \phi_{mr}\Big|_{e_{mr}(\gamma_{torque})}^{s_{mr}(\gamma_{moment})}\left(\mathbf{m}_{ref}\right). \tag{12}$$

5.2.4 Alignment controller

The alignment controller π_{ALGN} is a composite moment-force controller used in state 3 of the PA to align the parts. π_{ALGN} minimises residual moments and forces experienced when the female part has reached a near optimum insertion entry position with the male part after the rotation stage. The minimisation of residual moments with zero reference moment by the dominant controller aligns the wrist parallel to its mating part. The subordinate force residual controller ϕ_{fr} uses a small force reference in the negative z-direction to maintain a gentle contact through the alignment stage, where $f_d =< 0, 0, -0.1 >$ N. π_{ALGN} is defined in equation (13).

$$\pi_{ALGN} = \phi_{fr}\Big|_{e_{fr}(\gamma_{torque})}^{s_{fr}(\gamma_{force})}\left(\mathbf{f}_{ref}\right) \lhd \phi_{mr}\Big|_{e_{mr}(\gamma_{torque})}^{s_{mr}(\gamma_{moment})}. \tag{13}$$

5.2.5 Compliant insertion controller

The compliant insertion controller π_{CI} is similar to the alignment controller used in state 4 of the PA. This controller however has a higher magnitude value for the force reference in the z-direction such that $f_d =< 0, 0, -0.5 >$ N. This controller is activated once the alignment has taken place and the insertion can proceed. The larger reference value is designed to drive the insertion until an equal and opposite force is generated by contact with the male's part back wall.

5.2.6 Mating controller

The last controller is the mating controller π_{MAT} is a composite force-position controller used in state 5 of the PA. This controller uses the dominant force controller to maintain the mating position achieved upon contact during with the male's part back wall while the subordinate position controller seeks to maintain the position detected at contact. The mating controller is defined in equation (14).

$$\pi_{MAT} = \phi_{fr}\Big|_{e_{fr}(\gamma_{torque})}^{s_{fr}(\gamma_{joint\_pos})}\left(\mathbf{f}_{ref}\right) \lhd \phi_{p}\Big|_{e_{mr}(\gamma_{torque})}^{s_{mr}(\gamma_{force})}\left(\mathbf{p}_{ref}\right). \tag{14}$$

6 The relative change-based hierarchical taxonomy

The hierarchical taxonomy's goal was to connect human apropos actions like: 'approaching', 'rotating', 'aligning', 'snapped', and 'mated' with LLB's in a context-sensitive manner. One of the main challenges encountered in interpreting force signals is their inherent noise and spatio-temporal complexity. However, the force signals do inherently possess characteristics that describe the task at hand. The authors hypothesised that such characteristics could be extracted by looking at how temporal relative changes were associated to each other and contextualised by the state in which they occur. In so doing, intuitive behaviour sequence's can be extracted and their outcome examined. This level of discrimination is significant as it can be expanded to a real-time implementation and allow to reason about the state to perform corrective motions if necessary.

To bootstrap the approach, we partitioned the data into linear segments that approximate the data and classify the gradients according to magnitude per a small set of criteria. The next layer of abstraction examines at ordered-pair primitive sequences, and according to the gradient patterns presented and a small set of classification criteria, they are categorised into one of several types of motion compositions. The third layer abstracts sequences of motion compositions to identify LLB's, while the fourth layer looks at what LLB's are present in which states to determine if desired high-level behaviours are present. The final layer outputs the verification process results' according to whether or not the desired sequence of HLB's is present or not. A visualisation of the hierarchical taxonomy can be seen in Figure 6.

6.1 Primitive layer

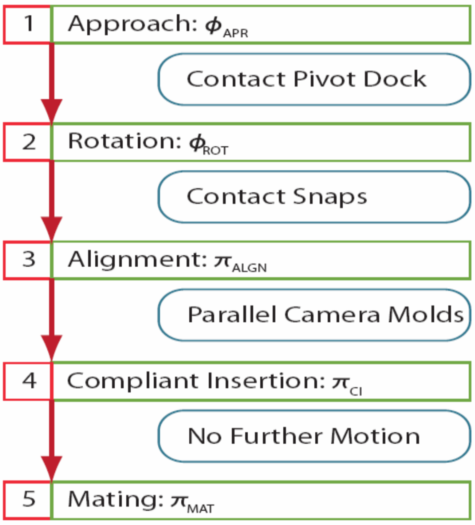
The primitives layer requires that each signal is partitioned into linear segments of data that closely approximate the original signal. Linear regression in concert with a correlation measure (the determination coefficient R^2) is used to partition the data whenever a minimum correlation threshold is crossed. If the determination coefficient drops under a given threshold the linear fit is partitioned and a new regression is started. The R^2 coefficient is a correlation measure that studies the ratio of the sum of the squares of the residual errors between the original data y and the fit data $\hat{y}$ to the sum of the variance σ_y^2 as shown in equation (15).

$$R^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sigma_y^2} \qquad (15)$$

The threshold used to partition the data was set at 0.70, such that if the correlation dropped to under 70%, a linear segment or 'partition' would be generated, and a new one would start at the next data point. The data was traversed by a window equal to five data points (the data was sampled at a frequency of 1 kHz by the simulation). The threshold values and the window length were empirically selected to partition the data sufficiently to capture relevant changes in the signals.

Each partition was accompanied by a data structure with seven types of information about itself: the average value across data points, the maximum value, the minimum value, the start time, the end time, the gradient value, and a gradient label. With respect to the latter, nine gradient labels (positive impulse, 'pimp'; big, medium, and small positive gradients, 'b/m/spos'; constant gradients, 'const'; and their negative equivalents, 'nimp', 'b/m/s/neg') were assigned according to ranges summarised in Figure 7. The classification first attempts to separate instances of data in which contact or mating takes place. On the one hand, contact phenomena is characterised by very rapid and large changes in force signals, almost approximating an impulse. To this end, positive and negative impulses were categorised for gradients with values greater or less than 70. On the other hand, for mating situations, there is little or no change in force, for this reason a constant label was assigned to signals with gradient values less than the absolute value of 1. In between these two extremes we chose to have three gradient categories for both positive and negative signals to give a general idea of the magnitude change registered for a signal. Figure 10 shows how the segmentation looks like across all five states (which are represented by five coloured boxes) for the force signal in the x direction for PA10 related-experiments.

Figure 5 The PA state machine for the PA10 robot and two-snap parts are shown in this figure (see online version for colours)



Notes: Each of the automata states (red) is accompanied by a controller template (green). Transition conditions are specified in blue.

Figure 6 The relative change-based hierarchical taxonomy (RCBHT) for cantilever-snap assembly verification (see online version for colours)

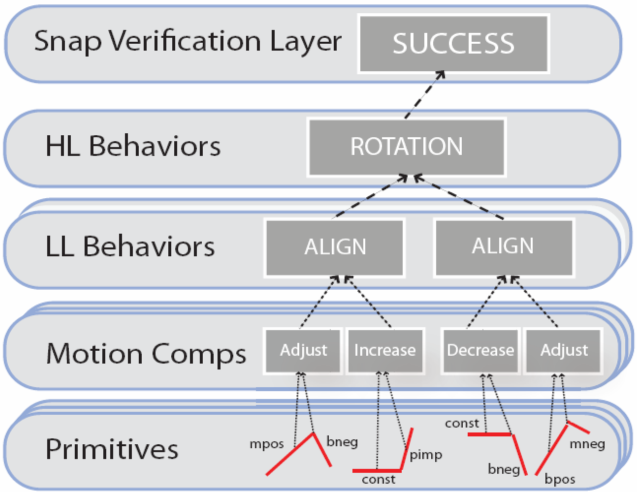
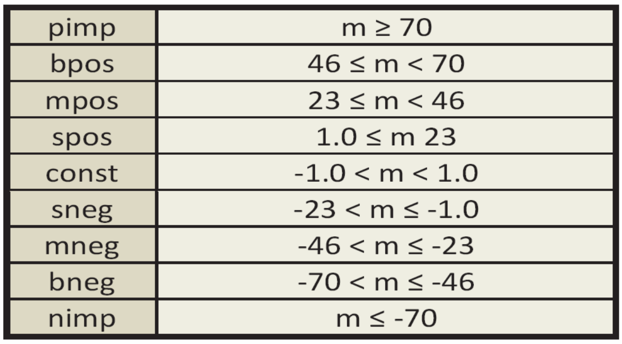


Figure 7 Gradient values classification for the primitives layers (see online version for colours)



pimp	m ≥ 70
bpos	46 ≤ m < 70
mpos	23 ≤ m < 46
spos	1.0 ≤ m 23
const	-1.0 < m < 1.0
sneg	-23 < m ≤ -1.0
mneg	-46 < m ≤ -23
bneg	-70 < m ≤ -46
nimp	m ≤ -70

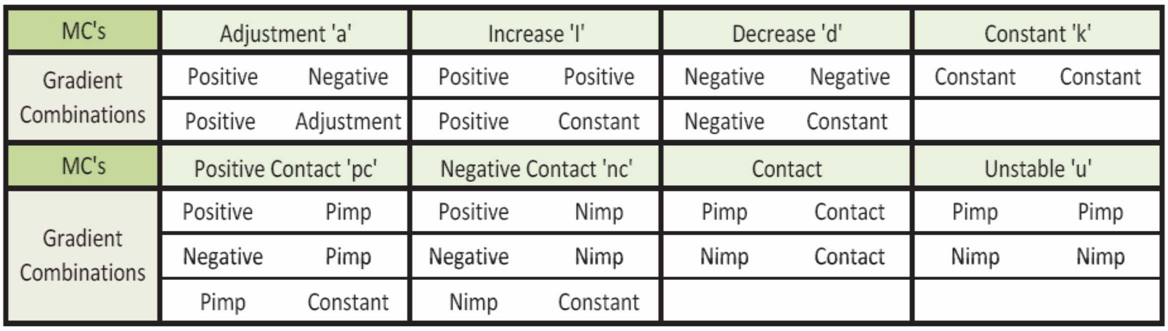
6.2 Composites

The next layer of abstraction identifies seven basic motions compositions (MC) by looking at ordered-pair sequences of primitives. The MC layer set is comprised of: adjustment, increase, decrease, constant, contact, positive contact, negative contact, and unstable motions. The positive and negative contacts, imply the sign of the (gradient) of the action.

A protocol was followed to minimise the effects of noise or erroneous segmentation. With respect to adjustments, primitives with big-small positive or negative gradients were considered as a positive or negative primitive category respectively. If a positive grouped primitive was followed by a negative grouped primitive an adjustment classification would be assigned to the ordered pair. Adjustments are motions in which the wrist records a quick ‘back-and-forth’ motion typically seen during alignment or insertion operations as the force controller tries to minimise residual errors. The reason to group positive and negative gradients is to maximise the likelihood of group adjustments even when the rate of change may be slightly different. Furthermore, for this particular category, we used a window of two data points instead of one to look for a matching pair (all other categories looked at the contiguous primitive). That is, if after finding a positive or negative gradient, and if the next data point was not negative or positive respectively, we would look at the next data point to look for a pair. Such procedure mitigates the presence of spurious signals that could prevent the proper grouping of an adjustment movement.

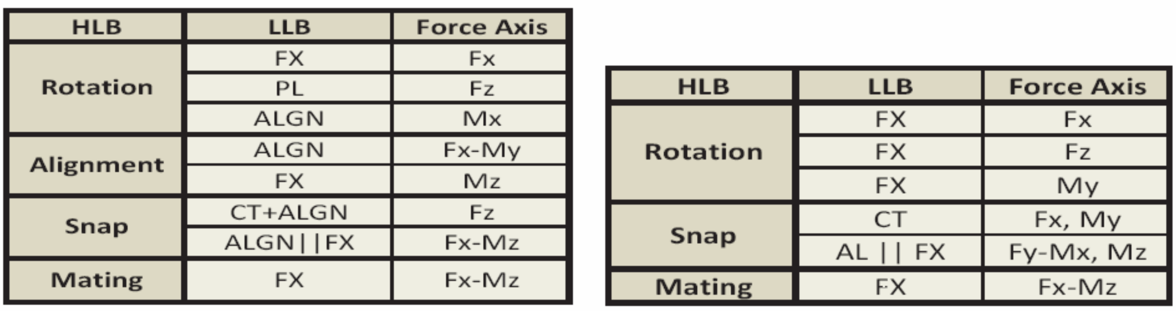
The ordered pair groupings for motion composition classification are summarised in Figure 8. Note that the table contains sub-tables representing five primitive groupings. The first primitive is in bold text followed by a listing of second primitives and the corresponding motion composition and label used in the plot as illustrated in Figure 8. As with the primitives layer, 11 pieces of information were collected for each MC: composition label, average value, root means square value, amplitude, the labels of the first and second primitives, the starting and ending times for both primitives, and the average time for both primitives.

Figure 8 Motion compositions according to primitive pairs in any order (see online version for colours)



MC's	Adjustment 'a'		Increase 'I'		Decrease 'd'		Constant 'k'	
Gradient Combinations	Positive	Negative	Positive	Positive	Negative	Negative	Constant	Constant
	Positive	Adjustment	Positive	Constant	Negative	Constant		
MC's	Positive Contact 'pc'		Negative Contact 'nc'		Contact		Unstable 'u'	
Gradient Combinations	Positive	Pimp	Positive	Nimp	Pimp	Contact	Pimp	Pimp
	Negative	Pimp	Negative	Nimp	Nimp	Contact	Nimp	Nimp
	Pimp	Constant	Nimp	Constant				

Figure 9 Comparison of the LLBs for both the PA10 and HIRO experimental configurations (see online version for colours)



HLB	LLB	Force Axis
Rotation	FX	Fx
	PL	Fz
	ALGN	Mx
Alignment	ALGN	Fx-My
	FX	Mz
Snap	CT+ALGN	Fz
	ALGN||FX	Fx-Mz
Mating	FX	Fx-Mz

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

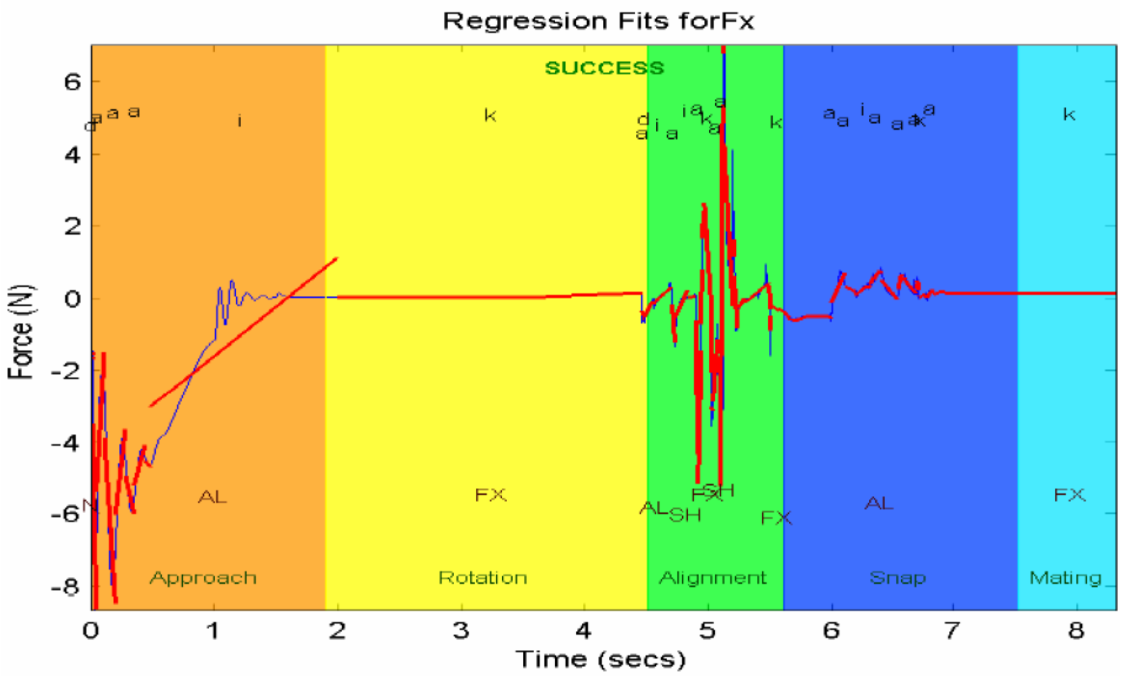
6.2.1 Refinement

After the MCs are generated, a refinement phase was used to filter less significant signals and augment more significant signals. To do so, the compositions were analysed under three contexts: a composition's time duration, a composition's amplitude magnitude, and composition repetition patterns.

- Time duration context: this filter examines two contiguous MCs. If either composition is seven times bigger than the other, the smaller composition is merged to the larger one and all data is updated correspondingly. The duration ratio was determined empirically.
- Amplitude value context: this filter pertains to the formation of adjustment signals and constant signals. We considered three possibilities:
  1. If there are contiguous primitives of types PC/NC or NC/PC, and if their amplitude is ten times smaller than the largest amplitude registered in the assembly, then treat them as an adjustment. This criteria seeks to disambiguate real contact signals and false ones by looking at their amplitude. Real contacts are characterised by large values.
  2. Similarly, if their is either an increase followed by a decrease and vice-versa, and both compositions have a similar amplitude (within (50%) of each other and they have a similar average value (100%) of each other, then merge as an adjustment and update the data correspondingly.
  3. If there is a sequence of an increase followed by a constant, or a decrease followed by a constant and vice-versa, and they have a similar amplitude (150%) and similar average value (100%), then merge them as a constant and update their information. This last filter targets small noisy signals that appear as increases or decreases but that in effect are constants. The amplitude threshold value is larger here to give more possibilities of catching increases or decreases within the narrow range of the constant's amplitude.
- Repeated compositions: the last filter takes signals that repeat and merges them as one. This filter is run iteratively until no more repetitions occur in the data.

The post-refinement composition layer results are shown in Figure 10 for a force signal sample in a PA trial.

Figure 10 This figure shows data related to the first four layers of the RCBHT (see online version for colours)



Notes: (1) The primitive layer: red line linear segments try to approximate original data and represent primitives. (2) The composites layer: composed by analysis of neighbouring primitives. Corresponding labels appear in black at the top-most part of screen. (3) The LLB layer: LLBs composed by analysis of neighbouring composites. Corresponding labels appear in uppercase red letters below the graph. (4) The HLB layer: HLBs derive from key LLBs. Corresponding labels appear in green at the bottom-most part of screen.

6.3 Low-level behaviours

The taxonomy's third layer considers motion composition ordered pairs along with signal duration and amplitude to yield classifications. Eight LLB classifications were derived and labelled as: push, 'PS', pull, 'PS', 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 contacts, if there is a positive contact followed by a negative one, or vice-versa, a contact LLB is assigned. One major difference between the MC level and the LLB level is introduction a shifting behaviour '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'.

With regards to the time duration context, if any motion composition lasts more that 100 milliseconds, it can by itself be a low-level behaviour, or if the contiguous composition is of the same classification they can also merge correspondingly. If any composition is less than the allotted duration and it does not have a matching pair, it is considered a noisy signal. With regards to the amplitude context, if there are two adjustments within a window of two data points, and their amplitudes decrease, render such a pair as an alignment, otherwise consider it a shift (or growing de-alignment). As for paired increase, decreases, and constants, they will yield pull, push, and fixed low-level behaviours correspondingly. Finally, as for contacts, if there is a positive contact followed by a negative one, or vice-versa, or even a stand-alone contact motion primitive, render this is a low-level contact behaviour.

6.3.1 Refinement

The LLB layer is also followed by a refinement phase. The latter filters based on the same three contexts as used before:

- Time context: this filter examines two contiguous behaviours (except for contacts). If either behaviour is five times bigger than the other, then merge towards the longer behaviour and update the data correspondingly. LLB's are longer than compositions, so this threshold value is to be smaller than the one used for the composition's time duration filtering.
- Amplitude context: the amplitude context pertains to alignments and shifts and there are four possible scenarios:
  1. If there is a push-pull pair in either order and they have similar amplitudes (150%) and similar average values (100%) render then an alignment.
  2. If there is a shift followed by an alignment, or an alignment followed by a shift, where the second behaviour has a smaller amplitude, then merge these as an alignment. This kind of merging is interesting because it can only be seen at this level of abstraction. While there may be a contiguous alignment-shift pair that was irreconcilable earlier, it can now be identified as an alignment. The same is done for a shift.
  3. Finally if there is an alignment followed by a pull or push or vice versa and they have similar amplitude (50%) and similar average value (100%), then merge as an alignment. In this case, after the previous refinement steps have been executed, if there are outstanding alignment-push or -pull pairs, the second behaviour is a considered a continuation of the alignment and is merged. Shifts are treated similarly.
- Repeated behaviours: as in the compositions layer, any two repeated behaviours can be merged as one. The post-refinement LLB layer is shown in Figure 10 for a sample signal in the PA.

6.4 High-level behaviours

The fifth layer contextualises the process monitoring by asking what low-level behaviours principally describe the high-level human apropos behaviours found in the PA: approach, rotation, (alignment), snap insertion, and mating[1].

Then, if key combinations of LLB's across the six force axes for a specific state are identified, then a certain HLB can be ascertained. For each state and corresponding HLB an LLB or sequence of LLB's are matched with a particular force axis as part of the selected criteria. The criteria is connected both to the PA states, to the controller templates, and to the coordinate frame assignments in local coordinates (see Figures 1 and 2). Given that we have two implementations of the PA and controller templates for the PA10-two snaps configuration and the HIRO-four snaps configuration we have two sets of key LLB criteria. They are presented in Figure 9.

6.4.1 Key LLB's for PA10-two snap experimental configuration

The reasoning behind the selection of LLB's and axis for the PA is intuitive. In state 2, the rotation state, the wrist maintains a constant force along the z-direction, while the force along the y-direction diminishes as the wrist aligns itself with male part. The rotation about the x-axis can be seen through a series of large alignments along the moment's x-direction. For state 3, all axes are aligning in some form. For force elements, there is an alignment in position, for moment elements there is an alignment in orientation. The only exception to this is the moment about the z-direction. A pattern emerges where the moment axis that corresponds to the wrist's direction of motion for the insertion (i.e., the z-axis for the PA) experiences little to no change throughout the assembly due to the nature of parts in the assembly task. For state 4, in the insertion state, Rusli studied typical force patterns for manually effected snap assemblies and states that initial resistance is characterises the insertion until the snap-catch slips behind the undercut in the mating part, at which time an interlock occurs. In other words, one a large increase in force is expected upon contact, followed by a large decrease in force. Hence, we expect to see a contact label followed by an alignment label. Other axis can expect to experience an alignment at this stage. Finally, for the mating state, all signals should present no motion change and thus be classified with a FX behaviour.

6.4.2 *Key LLB's for HIRO-four snap experimental configuration*

In this experimental configuration, the rotation controller is applying a constant force in both the x-and z-directions and a constant moment in the y-direction. For this reason we expect to find FX LLB tags in this state. Then, as for the insertion stage, experimental results consistently show that for successful assemblies there are CT LLB tags both in the force's x-direction and in the moment's y-direction. Both are correlated in that they represent the robot's downwards assembly motion. The rest of the axes are either aligning or fixed, that is, ALGN or FX tags should be seen in them. Finally, for the Mating state, there should be no movement and hence no change in gradient values if the structure is stable. FX labels are expected in all axis.

The fourth layer results are shown in Figure 10. If the high-level behaviours can be ascertained, they print on the plot in green colour. If they cannot be verified, they plot in red colour representing failure.

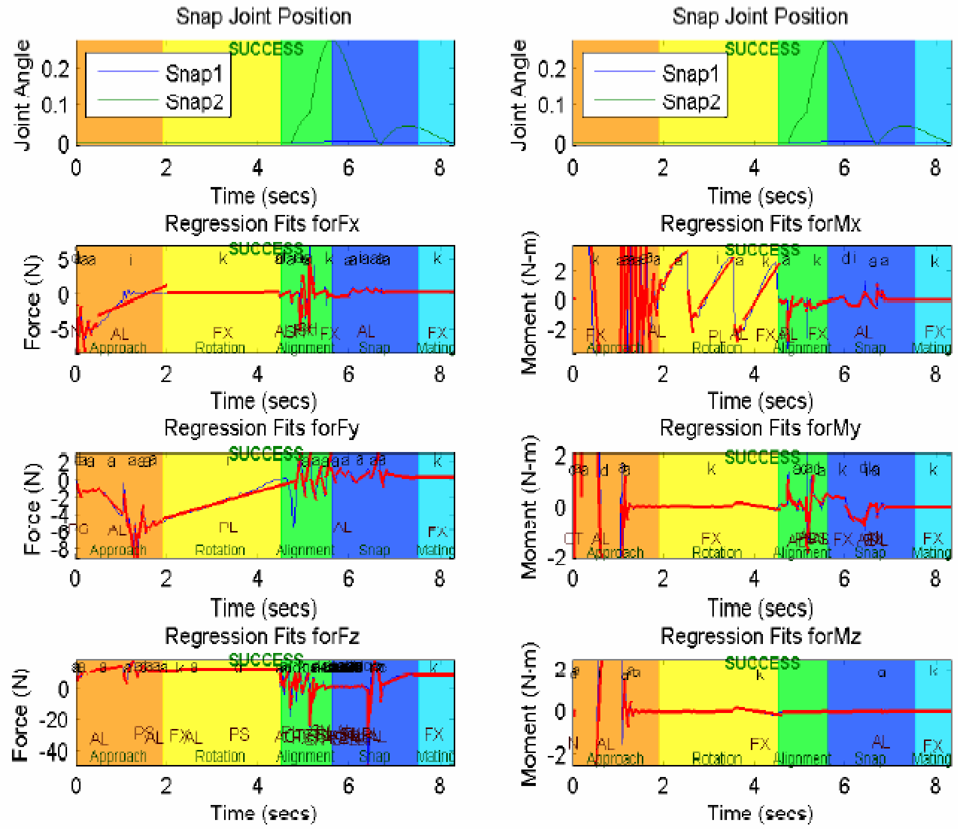
6.5 *Verification layer*

The fifth and last layer declares whether the assembly was entirely successful or not by posting a 'successful' or 'failure' label on each of the six axes of the FT plots.

7 PA10 and HIRO RCBHT experimental results

In this section we present the result of a trial in which each of the six force signals is analysed and combined to do a system verification. The results are visualised in Figure 11. The visualisation of all the results in Figure 11 contains very intuitive patterns between the LLB's and the HLB's for the PA. As noted in Section 6.4, the rotation behaviour is clearly identified by the pull in the F_y direction, the fixed position in the F_z direction, and by the alignment motion in the M_x direction. The align behaviour is characterised by alignment LLB's across all states. The snap insertion is distinguished by the high-force contact behaviour characteristic along the direction of insertion F_z. The last state is as expected characterised by fixed behaviours as at the mating stage no further motion should be experienced.

Figure 11 SV layer: if all HLB's are present, then the snap is successful (see online version for colours)



The system also predicted cases in which the assembly was a failure. One example is shown in Figure 12. During this attempt, the force controller encountered difficulties in properly aligning the parts as can be seen in the shifts in M_x and M_y. The failure to align not only result in a lack of contact behaviour in F_z but also a motion away from mating as visualised in the fifth state for M_x and M_y and as well as by the fact that the first snap did not converge to its home position. The verification system noted this by displaying the ‘snap’ and ‘mating’ states in red and by the ‘failure’ label at the top of each plot. The reason for failure may be due to unexpected force values that sometimes are rendered by our simulation programme upon contact. OpenHRP was originally designed for walking humanoid robots and its dynamic engine is not well suited for small contacts.

7.1 Analysis of PA10-HIRO RCBHT experimental results

For the PA10 experimental configuration, the system was run on a set of six simulation trials with 100% accuracy rate in its prediction. For the HIRO experimental configuration, the system was run on a set of fifteen simulations trials also with 100% accuracy rate. The results show that by limiting the way a cantilever snap assembly can be generated and that by using a small set of classification categories that encode relative change at different abstraction levels, force signals can be interpreted at a human apropos level of intuition that correspond to the assembly’s action states. The method effectively determined the status of successful and failed assemblies.

One limitation however was an issue of false-positives. While the SVL layer correctly assessed the outcome based on the LLB tags present, two trials assessed as failures were in fact successful. The problem was not due to errors in the HLB or SVL layers of the system but on the fact that LLB compositions result from primitives and motion compositions classification in the first two taxonomy layers. On two occasions the RCBHT generated an ‘AL’ LLB tag for a task segment where an ‘FX’ corresponded. This miscalculation is due to the hard-coded nature of gradient classification schemes in the primitives layer of the RCBHT system. To this end, we designed a Gradient Calibration method to optimise gradient classification values. The routine is described in Section 8.

Another limitation in the system is the uncertainty rendered by both the noise in FT signals and the interpretation schemes of the RCBHT in themselves. Furthermore in the HLB and SVL layer, outcome verification is decided based on the mere presence of LLB’s which in some cases may not be predominant. To this end, a Bayesian filtering mechanism was applied to the higher-level layers of the RCBHT system so as to reduce uncertainty in their predictions. The probabilistic filter is described in Section 9.

Figure 12 The snap verification system accurately detected the robot’s failure to snap and mate the parts in this task (see online version for colours)

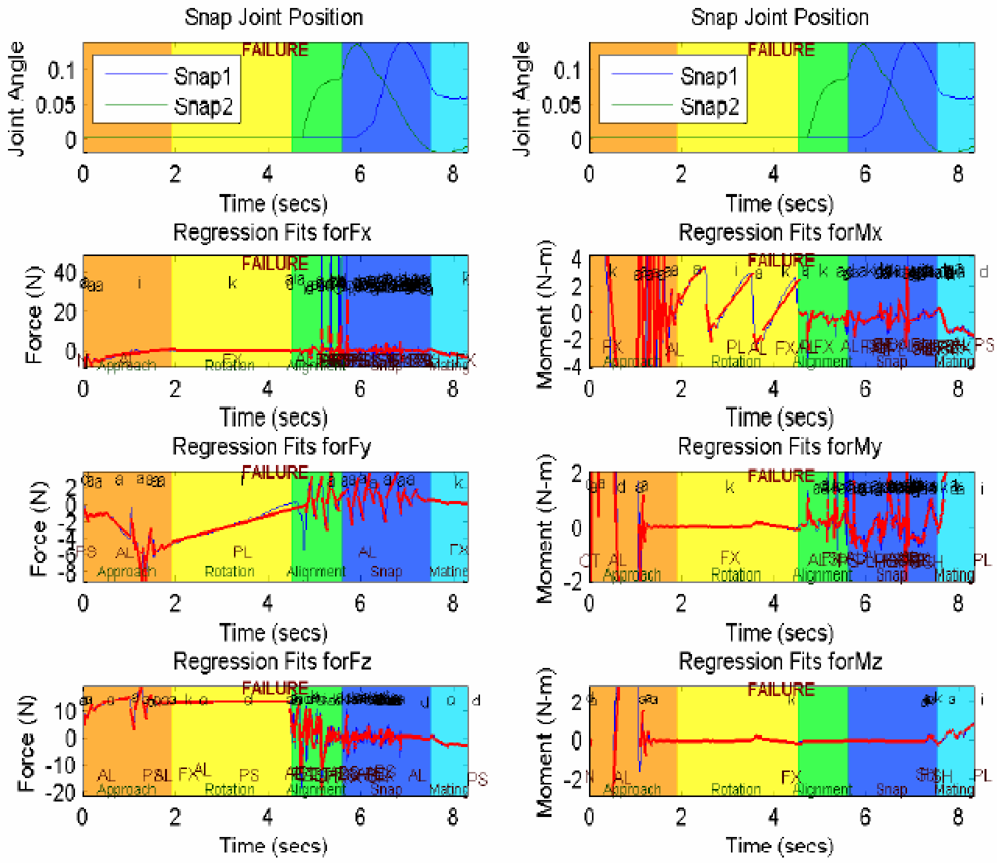
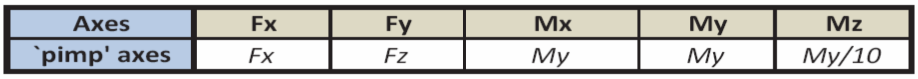


Figure 13 Summary of ‘pimp’ threshold assignments across force axes as part of gradient contextualisation (see online version for colours)



Axes	Fx	Fy	Mx	My	Mz
`pimp' axes	*Fx*	*Fz*	*My*	*My*	*My/10*

8 Gradient parameter optimisation

In the context of the PA and the RCBHT system, a gradient calibration routine was devised to acquire a set of gradient thresholds for the primitive layer that will be effective for a given robot-snap-part pair. That is, obtaining gradients that will optimally classify the FT signals into their appropriate labels. As mentioned in Section 6.1, the most important labels in our gradient classification scheme are the ones used to classify *contacts* (which yield large changes in FT data) and *constant* actions or *fixed* behaviours (where there is little to no change in FT data). Both of these labels have a positive and negative version. A total of eight labels are used to classify the gradient space as described in Section 6.1: *pimp*, *bpos*, *mpos*, *spos*, *sneg*, *mneg*, *bneg*, *nimp*. In our previous works, these values were obtained by trial and error. But in this work we devised a scheme to calibrate these values such that they will be effective in appropriately classifying FT signals as long as the same robot is working with the same part. This statement assumes that the PA is utilised. This is important as the PA constraints the snap assembly motion in the same way across trial thus enabling for similar patterns, in form and in magnitude, of FT signals to be generated across trials.

8.1 Gradient thresholds determination

The first step as part of gradient calibration consists in determining the values for the contact gradients (labelled ‘pimp’ or ‘nimp’) and the constant gradients (between ‘spos’ and ‘sneg’). Once these thresholds have been computed the space between ‘spos’ and ‘bpos’ and their negative counterpart can be divided into equally spaced segments, as in the table of Figure 7. In our work, we evaluated values of gradients within specific automata states (further detailed in the next section). A gradient was considered to exist within an automata state if the beginning of the primitive (or linear segment) started after the automata state began and finished before the automata state terminated.

8.1.1 Contacts

For contact LLBs, as per our PA key LLB selection criteria, typically occur in the Snap state of the *Fx* and *My* axes, and they only occur a small number of times. Statistical measures like the mean, median, or mode are not useful to extract an effective contact threshold. Instead, the absolute value of the maximum gradient is used. Once the maximum gradient is found, the ‘pimp’ label in the primitive’s layer is scaled by a constant *k*, such that: $pimp = k * (max(abs(grad_m)_{\text{state}}axis))$, $m \in M$, where *M* is the set of gradients in a given automata state in a given axis. Since this calibrated contact value will be used across trials, the value is scaled down to increase the likelihood of capturing contacts with similar values across trials. During the training phase an initial scaling factor of 0.90 was used but was later changed to 0.85 and will be described in the experiments.

8.1.2 Constants

For fixed LLBs, the mean, median, and mode values were compared to find which measure would yield the most effective threshold for constant signals. Experimental results (see Section 8.2) identified the mean to be the best measure. Hence, the ‘spos’ label in the primitive layer was set to the absolute value of the gradient mean as: $spos = abs(mean(grad_m)_{\text{state}axis})$.

8.1.3 Contextualisation

Another important consideration lied in whether we should compute such thresholds for each separate axes and in some cases for each automata stare or not. As per our previous findings, a successful assembly can be characterised by select LLBs which capture that main components of the task. Similarly, it is the gradients found in the same automata states as the key LLBs that dominate the classifications. By contextualising the gradient classification in this selective way, a more effective calibration can be attained. We begin with the contextualisation of the contact thresholds and then proceed with the constant thresholds.

The contact HLB is classified by two ‘CT’ LLBs in the snap automata state along the *Fx* and *My* axes. For this threshold, the ‘pimp’ value of the *Fz* axis was used to class the *Fz* and *Fy* axes and the ‘pimp’ value of the *Fx* axis was used for itself. The reason we separated the classification in this way was because the average value of gradients for the *Fy* and *Fz* axes was approximately 31 while that of *Fx* was 81. The *Fx* axis in world coordinates is the axis in which an initial vertical contact takes place between the camera mold parts upon snapping and represents the hardest contact. For the moment axes, the ‘pimp’ value of *My* was used for itself and for *Mx* but not for *Mz* we used one-tenth of the value of *My* as the mean of the latter is an order-of-magnitude smaller. A summary of gradient contextualisation can be seen in Figure 13. Once all gradient thresholds could be derived for a given trial, those values would be used to test whether or not the RCBHT system would classify successful assemblies (as observed by appropriate snapping and mating) across a number of training assemblies. This part of the experimentation further allowed us to make observations that were included in our calibration approach. The training experiments are described in the following section.

8.2 Gradient calibration experiments

In our experiments, five training and seven test assemblies were run under the HIRO experimental configuration. Each of the training and test trials were successful as supervised by an external user. In the training session, gradient thresholds were computed for each of the five trials. Then, each trial was assessed by each of the five different gradient sets of thresholds as part of the RCBHT system to determine whether the task was successful or not. In effect, 25 runs were attempted to assess the trial outcomes. As described in Rojas et al. (2012c, 2012d), the system considers the task to be successful if all key LLBs are present in the rotation, Snap, and Mating automata states. In this way, if the RCBHT declares the outcome to be successful we know that the gradient calibration was effective for that trial.

The results were organised in a table as shown in the table of Figure 14 in order of increasing magnitude for the 'pimp' threshold of the trials. From these experiments we can immediately note that it is those trials that yielded the contact thresholds of lowest magnitudes that yielded a successful outcome for other trials. This is so since the contact labels in trials with higher thresholds remain being contact labels in those trials. However, when the contact threshold of a trial is greater than a threshold in another trial, then the contacts there cannot be discerned and thus the RCBHT will not encounter contacts in the trial which are necessary in order for the Snap HLB to exist. The trend is that as the 'pimp' threshold increases per trial, the likelihood of assessing an outcome as success diminishes.

There are a few exceptions to this trend and they are due to a number of different issues. In the table of Figure 14, note that entries can be described by the *trial* number of the row and the *trial* number of the column. For entry (trial 1, trial 3), a failure might have been expected, however as noted in Section 8.1.1, the 'pimp' value was scaled by a factor of 0.90 originally and later by a factor of 0.85. By lowering the scaling factor by 0.05 points, it allows some trials with higher contact thresholds to still identify contacts in trials with lower contact thresholds (in fact 2 out of 5 trials behaved this way).

For entry (trial 4, trial 2), another failure might have been expected. In this case, a contact LLB appeared in what may be considered the transitional period between states. That is, the 'CT' LLB started within the Snap state but ended within the MAT state. As described in Section 8.1, we did not consider these kinds of transitional behaviours. This detected CT state for trial 2, had a high enough value that trials 4 and 5 detected it (along with other key LLBs) and rendered the outcome as successful. Dealing with transitional information is an important aspect that needs to be addressed. It was first identified in Rojas et al. (2012c) and will be addressed in the near future.

For entries (trial 2, trial 4) and (trial 5, trial 4), we have two failure assessments. The latter may have been expected but not the former. In fact, for entry (trial 5, trial 4) the scaling of the 'pimp' parameter would have allowed this trial to be assessed as successful, the problem lied elsewhere. No 'FX' behaviour was identified during the Mating automata state in the *Fy* and *Fz* axes as can be seen in Figure 15(a). The FT signal did indeed have a constant ('*k*') action at that stage, however, the latter had been absorbed by clean up process at the motion composition level. The latter used was absorbed into another action composition as part of a time context filtering after three clean up cycles. When, the filtering was reduced to two clean up cycles, the constant action appeared as well as the corresponding 'FX' LLB. This was not an isolated case as it also occurred in entry (trial 5, trial 4).

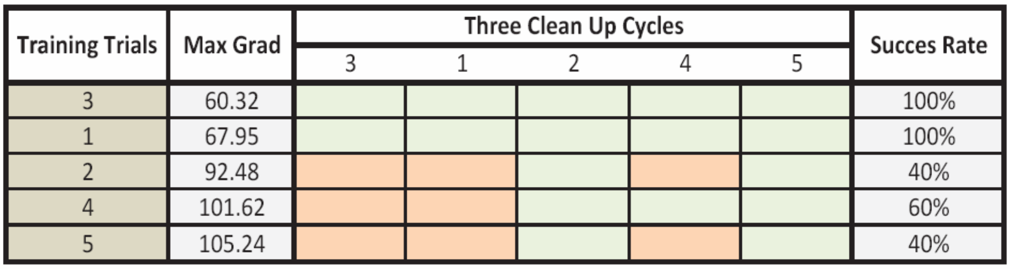
8.2.1 Minimum gradient selection

From the training experimentation, our calibration method was extended to consist of the following two aspects:

a run at least five successful trial assemblies, compute their 'pimp' thresholds according to Section 8.1, and choose the lowest magnitude threshold of the trial set

b utilise two clean up filtering cycles at the motion composition level rather than three to avoid excessive filtering.

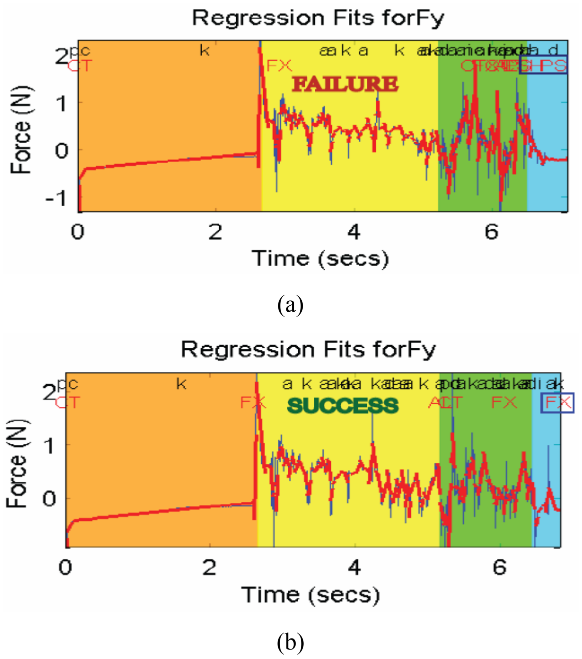
Figure 14 Five training trials are listed in order of increasing contact threshold value (see online version for colours)



Training Trials	Max Grad	Three Clean Up Cycles					Succes Rate
		3	1	2	4	5	
3	60.32	🟩	🟩	🟩	🟩	🟩	100%
1	67.95	🟩	🟩	🟩	🟩	🟩	100%
2	92.48	🟧	🟧	🟩	🟧	🟩	40%
4	101.62	🟧	🟧	🟩	🟩	🟩	60%
5	105.24	🟧	🟧	🟩	🟧	🟩	40%

Notes: The green colour indicates the RCBHT assessed the task as successful. The orange colour indicates the RCBHT assessed the task as failure.

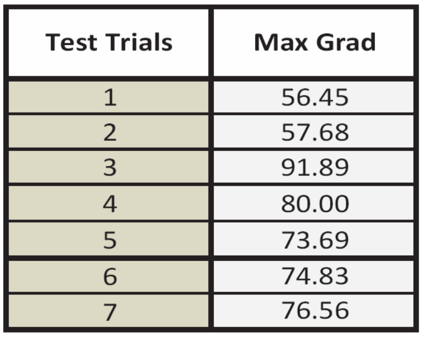
Figure 15 (a) Figure results used three clean up cycles as part of the filter at the motion composition level, the three cycles may filter important compositions at the second level of the system (b) Figure shows how when the clean up cycles are reduced to 2, the constant action (and correspondingly the 'FX' LLB) appears (see online version for colours)



(a)

(b)

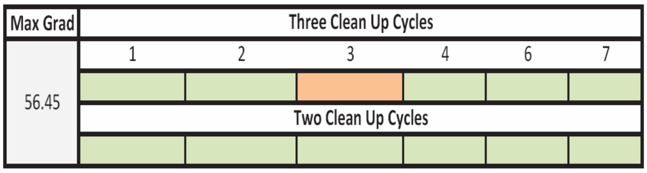
Figure 16 Seven test assemblies are listed with their contact threshold values (see online version for colours)



Test Trials	Max Grad
1	56.45
2	57.68
3	91.89
4	80.00
5	73.69
6	74.83
7	76.56

This gradient calibration methodology was run on seven test assemblies. The set of assemblies is also of interest because two of those trials presented false-positive results when using the RCHBT in Rojas et al. (2012c) (predicting a false outcome when in fact the assembly had been successful). The false-positive results had been a result of erroneous interpretation during the first two layers of the system. A summary of the contact gradient thresholds for the test assemblies is shown in the figure of Table 16. The first trial in the set yielded the smallest gradient and was used to generate the gradient classification thresholds that would be used for all test assemblies. For comparison the RCBHT was run once on the set of assemblies with two clean-up cycles and again another set with three cycles for comparison. The results are shown in the table of Figure 17. The calibration method worked 100% of the time for the test assemblies set when using two clean up cycles suggesting to be an effective calibration method. It also worked 85.7% of the time when using three clean-up cycles. Figure 18(a) shows how an uncalibrated system yielded a false-positive result in that a successful assembly did not posses a 'FX' LLB in the rotation state of the *My* axes. In Figure 18(b) on the other hand, the calibrated method was able to disambiguate the FT signal to show the presence of 'FX' LLBs. For further exploration, we tested the gradient thresholds from trial 1 in the training set with the two false-positive trials in the training set. In this case, when using two clean-up cycles both false-positive trials were correctly assessed as successful. When using three clean up cycles, it could only correctly interpret one of the two trials.

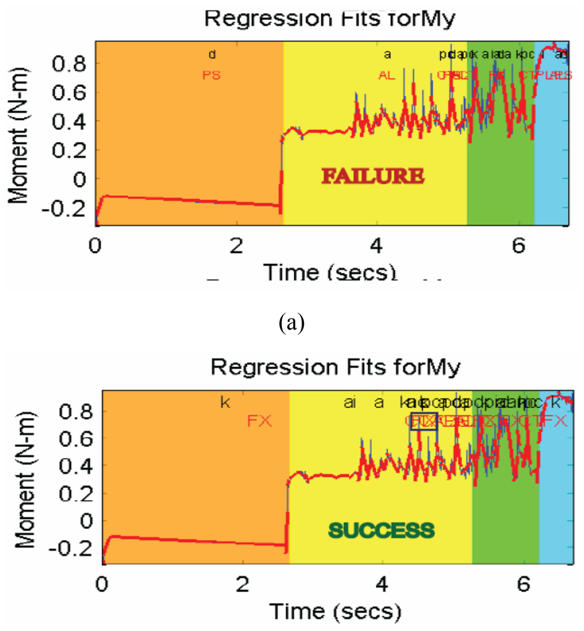
Figure 17 Our calibration method with 2 clean up cycles successfully interpreted all seven test assemblies (see online version for colours)



Max Grad	Three Clean Up Cycles					
	1	2	3	4	6	7
56.45	🟩	🟩	🟧	🟩	🟩	🟩
	Two Clean Up Cycles					
	🟩	🟩	🟩	🟩	🟩	🟩

Note: Including two assemblies that had yielded false-positive results when using un-calibrated gradient thresholds.

Figure 18 (a) Figure shows how the RCBHT assigned a false-positive result by assigning a successive trial as failure (b) Figure shows the difference in results after using the calibration method with two clean up cycles (see online version for colours)



(a)

(b)

8.3 *Gradient calibration analysis*

An effective gradient calibration method was implemented for the RCBHT system. Statistical measures were used to derive contact and constant gradient thresholds in contextually sensitive ways. During our experimentation we discovered that by running a number of training trials and selecting the gradient thresholds that belong to the trial whose contact threshold is the lowest, then the gradient calibration would be more effective. The experimentation also revealed a hidden fact prior to this work. That is, that when using three rounds of filtering at the motion composition level, some key motion composition actions are filtered away by the system. Two clean up cycles proved to be a very effective number for correctly assessing snap assembly outcomes.

One of the keys in effectively calibrating gradients is understanding the assembly stages properly. The PA is useful in dividing the snap assembly into automata states that are consistent across trials. Each of those automata states have similar signal-patterns across trials for the six different force-torque axis. Furthermore, by having the RCBHT system abstract relative change across increasing layers of intuition, we can identify actions and behaviours that consistently appear across trials when the task is successful. Herein lies the significance of our framework; moreover, in this work's context, the design of the gradient calibration method exploits such interpretations in order to identify what to look for in the FT signals and then implement statistical measures to identify relevant thresholds. Our calibration methods can be applied whenever a new task is implemented, whether a new robot or a new part. This is an important part of increasing the viability of this system as a way to automate assemblies.

It is also worth mentioning, that the calibration method assumed a constant value for the 'determination coefficient' described in Section 6.1. The calibration should not be affected by different correlation values as long as they are constant throughout all trials. Another aspect worth noting is that the threshold values produced by the calibration method are tied to the FT transition values employed in our controller as part of the PA. If the FT transition conditions were to change, the RCBHT system would have to be recalibrated. Currently the FT transition values of the PA are also experimentally deduced. FT transition conditions in the PA can be found in Rojas et al. (2012a, 2012c, 2012d).

9 Probabilistic RCBHT

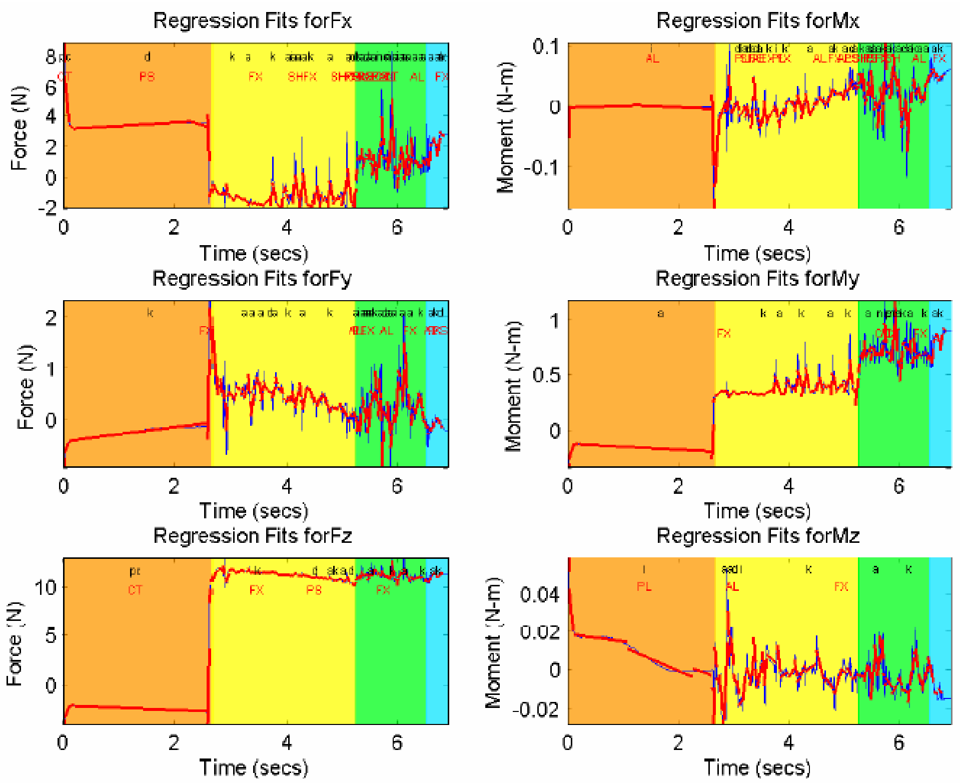
While the RCBHT detected key LLB presence for all FT axes to characterise a given desired HLB. However, a number of factors render the assessment uncertain:

1 the assembly's sensory data is noisy

2 the RCBHT is limited to discriminate with certainty what LLBs are occurring

3 key LLBs may only appear for a short amount of time during a state of the automata (see Figure 19).

Figure 19 The RCBHT's first three layers are presented in this figure (see online version for colours)



Notes: 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 behaviours) layer are shown in red text labels. The coloured boxes represent the PA's stages: rotation, snap, and mating. The figure shows the robot's state throughout the task.

For this reason, Bayesian filtering was used to deal with uncertainty at the RCBHT's third layer. BF's have been typically used to localise mobile robots and, more recently, features in manipulation tasks (Platt et al., 2011; Meeussen et al., 2007). 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 BF 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 Section 6.3 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: $bel(x_t) = p(x_t \mid z_{1:t})$.

BFs use an update rule to recursively update the belief in the current state from the belief in the previous step: $\overline{bel}(x_t)$. 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.

9.1 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:

$$P\left(x_t \mid z_{1:t-1}, u_{1:t-1}\right) = \sum_{x_{t-1}} P\left(x_t \mid x_{t-1}, u_{t-1}\right) P\left(x_{t-1} \mid z_{1:t-1}, u_{1:t-1}\right). \tag{16}$$

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 6 × 6 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:

1 the longer an LLB lasts during an automata state, the more likely it is to dominate the behaviour for that automata state

2 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.

9.2 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:

$$P\left(x_t \mid z_{1:t}, u_{1:t-1}\right) = \eta P\left(z_t \mid x_t\right) P\left(x_t \mid z_{1:t-1}, u_{1:t-1}\right), \tag{17}$$

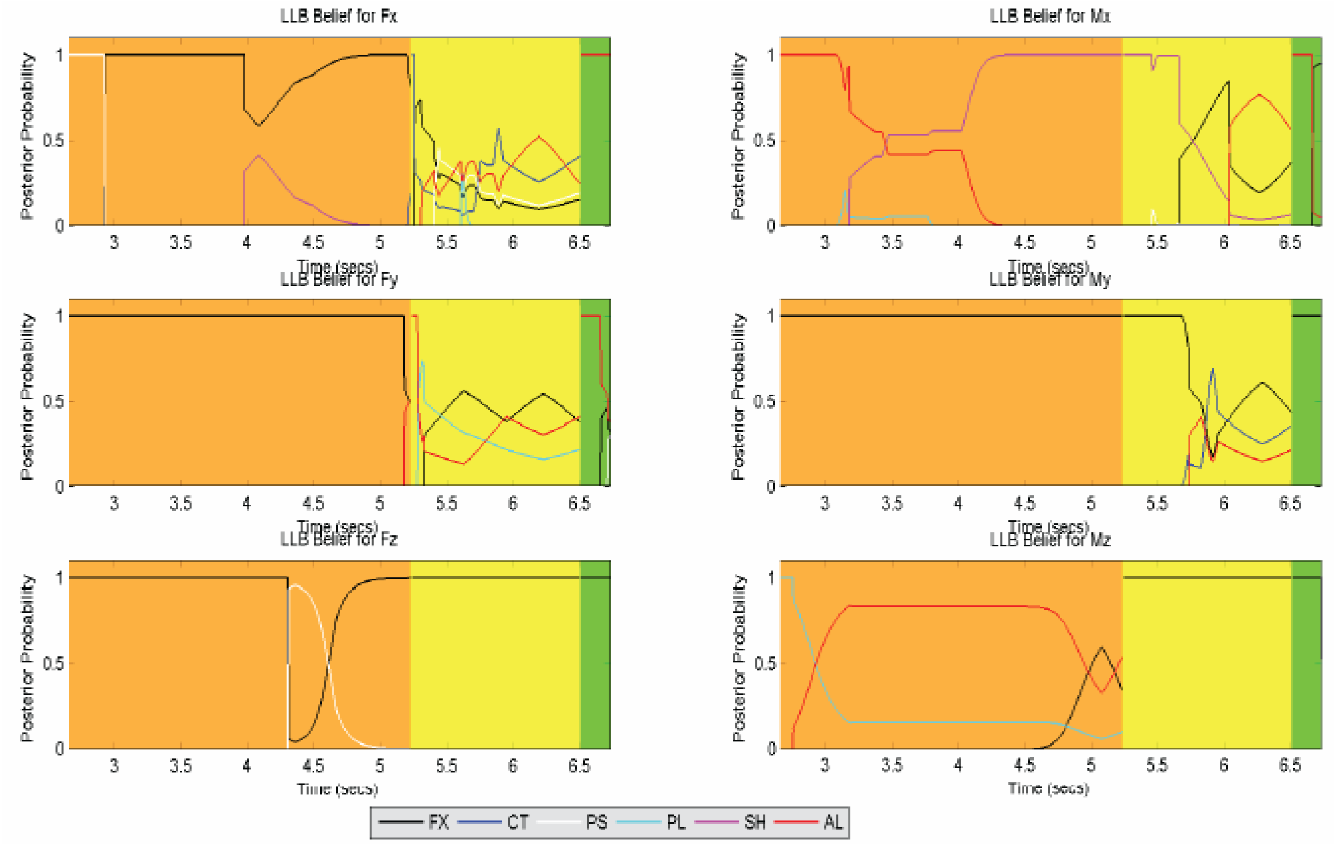
where η is a normalisation factor that guarantees that the probability sum does not exceed 1. Measurements represent the cumulative duration feature explained for prior probabilities. In the measurement's case, z_t measures the cumulative duration upto that point in time for a given state s for a given 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: $P(z_t \mid x_t) = \mathcal{N}(z_t; \hat{x}_t, \sigma^2)$. 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 Figures 19 and 20. 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:

$$P\left(llb_{i,t} \mid z_{llb,1:t}, u_{1:t-1}\right) = \eta P\left(z_{llb,t} \mid llb_{i,t}\right) \sum_{j,t-1} P\left(llb_{i,t} \mid llb_{j,t-1}, u_{t-1}\right) P\left(llb_{j,t-1} \mid z_{1:t-1}, u_{1:t-1}\right). \tag{18}$$

Figure 20 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 Figure 19.

Figure 20 LLB belief's are computed for each of the six behaviours in the LLB layer of the RCBHT (see online version for colours)



Notes: Each LLB's belief was computed with a Bayesian that used the behaviour's average cumulative duration. The belief represents the likelihood that a behaviour dominates the task at any moment in time.

9.3 *Probabilistic HLB*

As mentioned in Section 6.4, the HLB layer is a function of key selected LLBs (refer to Figure 9). 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.

Figure 21 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).

9.4 *Probabilistic SVL*

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 Figure 21 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 Figure 21).

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.

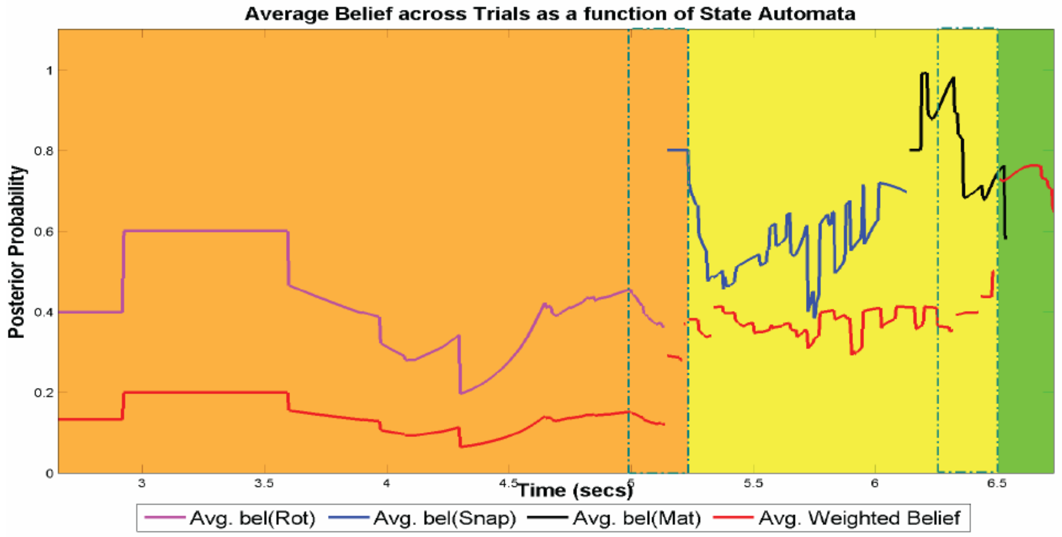
Figure 21 presents three pieces of data:

1 the HLB Beliefs for five successful training examples

2 the averaged HLB belief values

3 the weighted averaged values from which the SVL extracts threshold values at the automata state's end.

Figure 21 The HLB and SVL layers: HLB belief per automata state is shown by three non-red lines averaged over five trials (see online version for colours)



Notes: 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.

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%.

9.5 pRCBHT results

After the training phase, seven more test snap assembly trials were executed under the HIRO experimental configuration. 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.

9.6 pRCBHT experimental results analysis

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 behaviour 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 contextualised 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.

10 Preliminary Assembly failure analysis

The failure analysis that we have conducted concerns assemblies that succeeded in the Approach and rotation stages of the PA but failed at the Insertion point. That is to say, we have not focused on Assemblies that fail either initially in their approach and initial contact between parts, or during the rotation stage. Our failure analysis contemplates both experimental configurations introduced in this work.

Successful snap assemblies can be understood by extracting key LLBs. These LLBs, when studied contextually, that is, by studying what LLBs occur in what automata states of the PA and in what force axes we are able to identify their high-level representation. Figure 9 summarises such LLBs for both experimental configurations. As noted in Section 4.2.2, the selection of key LLBs also depends on our selection of task frames and controller templates. Yet in both experimental configurations, we note that the presence of a contact ('CT') LLB as part of the Insertion state in the force axis direction which corresponds to the vertical insertion direction (that is Fz for the two-snap part configuration and Fx for the four-snap part configuration) is necessary for the insertion to succeed.

It is not surprising then to find that all failed tasks were unable to experience a contact behaviour during the insertion stage in the aforementioned vertical direction. Additionally, all failed tasks also showed that whenever a CT LLB was missing from the vertical direction of motion, then an undesired CT, SH, OR PS and PL set of LLBs were present in the perpendicular direction of motion (that is the direction in which the robot moves forward to approach the mating part) at two stages:

1 the insertion stage

2 the rotation/alignment stage.

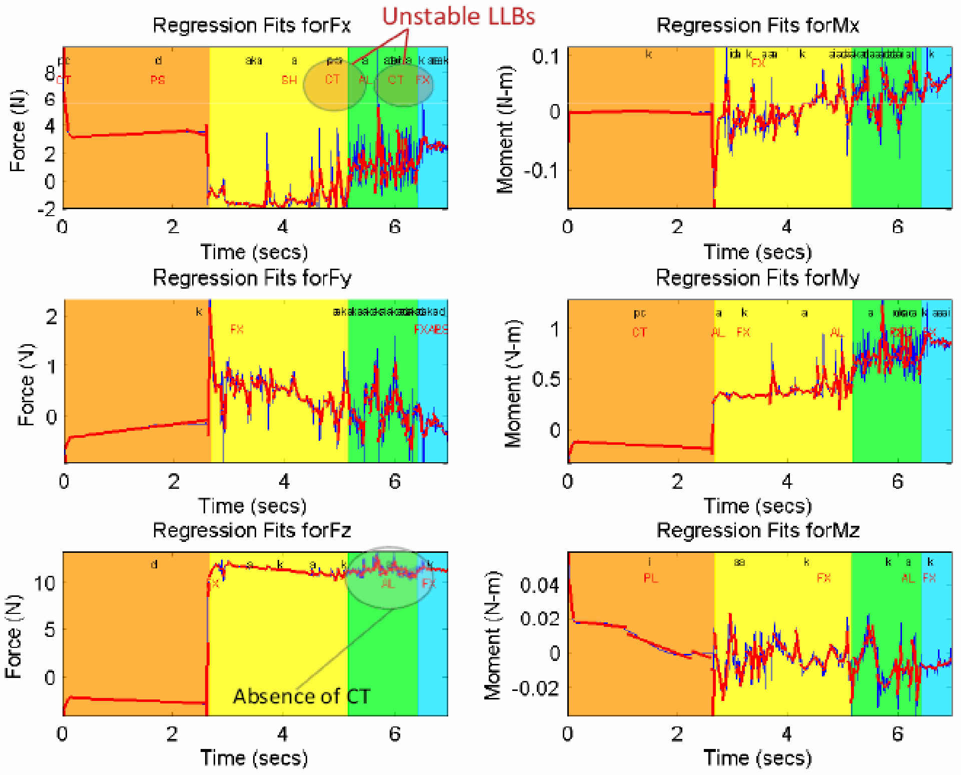
That is to say, that we can directly correlate the absence of a contact in the vertical insertion direction, to unstable lateral behaviours taking place during and prior to the insertion phase of the snap approach. For a visualisation of this dynamic see Figure 22. Another trend, though not with as strong a correlation, is that whenever there is instability in the lateral axis (say *Fx*), there will also be instability in the moment axis the perpendicularly bisects the force axis and that is in direction of the lateral motion (say *My*).

We can in a preliminary fashion indicate that if there is a presence of CT LLBs in the rotation/alignment/insertion stages of the lateral motion, that there is instability in the system. This instability, we infer, is most likely due to a problem in the initial docking. Further experimentation will be necessary to determine whether we can correct this motion in-situ or if it would be better to withdraw the approaching snap part entirely and try to reapproach the part again.

11 Discussion

With regards to the PA, the authors believe the latter was effective in exploiting the built-in mechanical design typical of snap parts to constrain the motion of the snap assembly and optimise entry points. By pivoting about the locking dock, better alignment existed not only at the initial contact point but along the camera parts' interior wall linings. Furthermore, due to the generation of similarly-patterned signals across trials, a small-set of gradient classification categories were sufficient to interpret the FT signals through increasingly abstract layers revealing the robot state throughout the task in human-apropos behaviours. Uncertainty in the signal processing was addressed by introducing Bayesian filtering and yielding belief-states at the LLB, HLB, and SVL level. By computed the likelihoods for belief states, more intuitive and granular state awareness was achieved as compared to the standard RCBHT system. This is significant because we can understand which behaviours dominate at any point in time and in any of the axes. This in turn allows us to better classify errors in the assembly process and generate more granular strategies for failure recovery.

Figure 22 This trial carried out with the HIRO robot and a 4-snap part shows a common failure characteristics when a contact behaviour cannot be established in *Fz*'s insertion state (see online version for colours)



Note: Labels indicate areas of interest in understanding failure phenomena.

The originality of our work is multi-faceted. The first aspect is identifying an assembly control strategy that can be applied to the cantilever-snap parts and that it can generalise to parts of varying geometrical complexity. It does so by working in concert with parallel controller templates that are also designed to optimally but flexibly meet control requirements for each state. Excellent work has been done in generating adaptable strategies in the industrial setting (Sayler and Dillmann, 2011), though these strategies have not extended for snap assembly procedures. With regards to the RCBHT system, is able to effectively abstract increasingly intuitive behaviour primarily due to two aspects:

1 Considering the classification of gradients not in an absolute way but in a relative way. Hypothesising that higher level behaviour require not exact understanding of change but that relative classifications are enough to encode behaviour.

2 That behaviour encoding requires appropriate contextualisation.

Contextualisation is present in multiple forms. We distinguish data across force axes, automata states, and level of abstraction in which data occurs, even in the way we do refinement cycles. There have been others works that have tried to understand change in FT signals (McCarragher, 1994; Eberman and Salisbury, 1994), but these have been limited to regular peg-in-hole assembly tasks. They also have not developed a framework upon which their application can extend to other configurations. In this context, others have also sought to build contact-state graphs to understand the robot state (Kwak et al., 2011; Meeussen et al., 2007). However, these methods become unwieldy with geometrical complexity as the number of contact possibilities grows exponentially. Additionally, it was important to identify sources of uncertainty in the RCBHT system and addressing them by introducing Bayesian calculus. Some researchers have applied Particle Filters to perform the estimation of contact formation as part of understanding robotic state through contact-state graphs in compliant motion tasks (Meeussen et al., 2007). In our work, by identifying Gaussian measurement models and prior and transition probabilities, we obtain high-level belief states from the snap assembly task through the RCHBT system. The latter is significant as they help us to understand which of all LLBs is dominating the task in a given force axis at a given point in time. This result will play an important role in helping us identify more precisely common failure patterns and generate corrective strategies to render the snap assemblies more robust. Our work in failure detection is preliminary but gives insight into the nature of failure dynamics. More statistically significant trials need to be systematically carried out to better characterise failure detection in its multiple forms and then devise systematic ways in which those failures can be corrected.

The use of simulation pose limitations to our study. However, the working principles of our framework should generalise well to physical experimentation as the only source of variation is the magnitude and form of FT signals. The gradient calibration process should also help to mitigate the effects of working with a real system instead of a simulated one. Limitations in the RCBHT are also concerned with other parametric values that have been chose such as: the determination coefficient threshold in the linear fit; and refinement phase values such as: the gradient margin values, time durations, amplitude values, and average value margins. Changes in these values could alter the verification result. There is still a need to implement methods that could extract these properties autonomously. Other limitations became apparent from our work with Bayesian filtering. A weakness in our system 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. Another aspect that we have identified is that we normally assume transitions in automata states happen instantaneously and not progressively. This assumption prevents us from addressing changing behaviours (gradient values, LLB presence, and LLB likelihoods) near transition windows. This phenomena requires that we address transitions as their own automata states and that we study how to interpret transition data to better understand robot state and error correction. Our current failure detection analysis is preliminary and more exhaustive and systematic testing is necessary to quantitatively understand failure patterns in cantilever-snap assemblies.

11.1 Future work

In terms of future work, we are working to implement the physical and online version of our system. This is imperative to corroborate our findings thus far. Additionally, there are number of pressing developments that we would like to implement. One of them is to develop autonomous parts localisation. By enabling the robot to identify the position of parts, the snap assembly can be executed completely autonomously. As mentioned in the Discussion, we also are looking towards systematically characterising assembly failure. Once all error possibilities are characterised, a system for corrective or regenerative motions can be designed and in this way close the loop between identifying the robot state and correcting failure-prone motions. This would enable us to achieve full 'snap sensing' functionality. Furthermore, once snap sensing is in place, we would also like to work towards executing desnapping behaviour and two-arm snap assemblies. Beyond this, there are many small improvements that can be included in our system as mentioned in Section 11.

12 Conclusions

This work presents undergoing efforts to develop a framework for Snap Sensing. Snap Sensing is related to the execution of snap assemblies and is defined as the ability for a robot to discern its state during a task and wherever necessary execute corrective motions. To this end we have integrated a framework that consists of a control strategy that generalises to cantilever snap parts of varying geometrical complexity and by virtue of its motion constraining design eases the interpretation of FT data. This strategy works in concert with a flexible control framework that supports the generalisation of the strategy. Furthermore, we have integrated a robot state estimation and snap assembly verification method. The latter encodes gradient relative-change in FT data and abstracts this information contextually to produce high-level intuitive behaviours that are used in turn to determine whether an assembly is successful or not. The robot state verification method has been made more robust by including gradient optimisation procedures and Bayesian filtering into the signal processing. The latter yields belief states for higher level behaviours which are used to predict the success likelihood of an assembly. Preliminary assembly failure characterisation was conducted and provided insight into assembly failure dynamics' for cantilever-snap parts. Ongoing work seeks not only to acquire the snap assembly's robot state but also to enable the identification of failure-prone actions to in turn generate corrective motions.

Acknowledgements

This paper is a revised and expanded version of the paper entitled 'A Constraint-Based Motion Control Strategy for Cantilever Snap Assemblies' presented at the IEEE ICMA 2012 Conference in Chengdu, China, on August 5-8, 2012.

References

Asada, H. (1990) 'Teaching and learning of compliance using neural nets: representation and generation of non-linear compliance', in *IEEE Int'l Conf. on Robotics and Automation*.

Brock, O., Fagg, A., Grupen, R., Platt, R., Rosenstein, M. and Sweeney, J. (2005) 'A framework for learning and control in intelligent humanoid robots', *Intl. J. of Humanoid Robots*, Vol. 2, No. 3, pp.301–336.

Chin, K.S., Ratnam, M.M. and Mandava, R. (2003) 'Force-guided robot in automated assembly of mobile phone', *J. of Assembly & Aut.*, Vol. 23, No. 1, pp.75–86.

Eberman, B. and Salisbury, J.K. (1994) 'Application of change detection to dynamic contact sensing', *International J. of Robotics Res.*, Vol. 13, No. 5, pp.369–394.

Gadeyne, K., Lefebvre, T. and Bruyninckx, H. (2005) 'Bayesian hybrid model state estimation applied to simultaneous contact formation recognition and geometrical parameter estimation', *Int. J. Robotics Research*, Vol. 24, No. 8, pp.615–630.

Hoffman, J.M. (2005) 'Fastening and joining fundamentals of annular snap-fit joints', *Machine Design*, Vol. 77, No. 1, pp.82–88.

Kanehiro, F., Hirukawana, H. and Kajita, S. (2004) 'Openhrp: open architecture humanoid robotics platform', *Intl. J. of Robotics Res.*, Vol. 23, No. 2, pp.155–165.

Kwak, S.J., Hasegawa, T. and Chung, S.Y. (2011) 'A framework on automatic generation of contact state graph for robotic assembly', *Advanced Robotics*, Vol. 25, Nos. 13–14, pp.1603–1625.

McCarragher, B.J. (1994) 'Force sensing from human demonstration using a hybrid dynamical model and qualitative reasoning', in *IEEE Intl. Conf. on Robotics and Automation*, pp.557–563.

Meeussen, W., Rutgeerts, J., Gadeyne, K., Bruyninckx, H. and De Schutter, J. (2007) 'Contact-state segmentation using particle filters for programming by human demonstration in compliant-motion tasks', *IEEE Trans. on Robotics*, Vol. 23, No.2 , pp.218–231.

Meitinger, T. and Pfeiffer, F. (1995) 'Modeling and simulation of the assembly of snap joints', in *Proc. of IEEE Int. Symposium on Assembly and Task Planning*.

Mihaylova, L., Lefebvre, T., Bruyninckx, H., Gadeyne, K. and De Schutter, J. (2002) 'Active sensing for robotics – a survey', in *Int'l Conf. on Numerical Methods and Applications*.

Platt, R., Permenter, F. and Pfeiffer, J. (2011) 'Using Bayesian filtering to localize flexible materials during manipulation', *IEEE Transaction on Robotics*, Vol. 27, No. 3, pp.586–598.

Platt, R.J. (2006) *Learning and Generalizing Control-based Grasping and Manipulations skills*, PhD thesis, U. of Mass. Amherst.

Rojas, J. and Peters II, R.A. (2012) 'Analysis of autonomous cooperative assembly using coordination schemes by heterogeneous robots using a control basis approach', *J. of Autonomous Robots*, Vol. 32, No. 2, pp.97–172.

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

Rojas, J., Harada, K., Onda, H., Yamanobe, N., Yoshida, E., Nagata, K. and Kawai, Y. (2012b) 'Gradient calibration for the RCBHT cantilever snap verification system', in *IEEE Int'l Conference on Robotics and Biomimetics*.

Rojas, J., Harada, K., Onda, H., Yamanobe, N., Yoshida, E., Nagata, K. and Kawai, Y. (2012c) 'Probabilistic state verification for snap assemblies using the relative-change-based hierarchical taxonomy (in-print)', in *IEEE-RAS Intl. Conf. on Humanoid Robots*.

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

Sayler, S. and Dillmann, R. (2011) 'Experience-based optimization of universal manipulation strategies for industrial assembly tasks', *Robotics and Autonomous Systems*, Vol. 59, No. 1, pp.882–898.

Skubic, M. and Volz, R.A. (1997) 'Learning force sensory patterns and skills from human demonstration', in *IEEE Intl. Conf. on Robotics and Automation*, pp.284–290.

Soddhi, R.S. and Sonnenberg, M. (1999) 'Use of snap-fit fasteners in the multi-life-cycle design of products', in *IEEE Intl. Symp. on Electr. & Env.*

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

Notes

[1] In actuality, we do not directly assess the approach stage given that there are no contact forces at this stage. But if the rotation analysis is successful we assume that the approach was too.

Appendix

Abbreviations in this work are found in the table below:

Systems	PA	Pivot approach
	MA-FP	Male-active/female-passive
	FA-MP	Female-active/male-passive
	RCBHT	Relative change-based hierarchical taxonomy
Layers	MC	Motion composition
	LLB	Low-level behaviour
	HLB	High-level behaviour
	SVL	Snap verification layer
LLBs	PS	Push
	PL	Pull
	CT	Contact
	FX	Fixed
	AL	Alignment
	SH	Shift
	N	Noise
Bayesian	pRCBHT	Probabilistic RCBHT
	BF	Bayesian filter