Force Controlled Assembly of Emergency Stop Button

Stolt, Andreas; Linderoth, Magnus; Robertsson, Anders; Johansson, Rolf

Published in:
Proc. 2011 IEEE International Conference on Robotics and Automation (ICRA2011)

2011

2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Force Controlled Assembly of Emergency Stop Button

Andreas Stolt, Magnus Linderoth, Anders Robertsson, and Rolf Johansson

Abstract— **Modern industrial robots are fast and have very good repetitional accuracy, which have made them indispensable in many manufacturing applications. However, they are usually programmed to follow desired trajectories and only get feedback from position sensors. This works fine as long as the environment is very well structured, but does not give good robustness to objects not being positioned or gripped accurately. A solution is to use additional sensing, such as force sensors and vision. How to combine the data from the different sensors and use it in a good way to control the robot is still an area of research.**

This paper describes an assembly scenario where a switch should be snapped into place in a box. Force sensing is used to resolve the uncertain position of the parts and detect the snap at the end of the operation. During the assembly an uncertain distance is estimated to improve the performance. By performing the assembly several times, learning is used to generate feed-forward data, which is used to speed up the assembly.

I. INTRODUCTION

The traditional way of using industrial robots is to program them to follow position trajectories and only use feedback from position sensors, and this is something that is handled very well by modern robot controllers. They are fast and have good repetitional accuracy. This approach is however not that good to use when uncertainties are introduced in the task specification. Additional sensors are then necessary to resolve the uncertainties and complete the task.

The incorporation of additional sensors in this paper is done using the the constraint-based task specification methodology [3]. It is a general framework that makes it easy to algorithmically incorporate multiple sensors, geometric uncertainties and to handle both redundant manipulators and redundant tasks. The task specification is also easy to use and to reuse.

An earlier framework for specifying end-effector based motion tasks is the operational space formulation [9], where generalized task specification matrices are used. A survey of how robotic assembly should be performed is proposed in [2].

Examples of sensor fusion of vision and force sensing is given in [7], which describes an assembly scenario where cylinders are to be inserted into a rotating engine. Another example is [4], where an unknown surface is tracked. In [10] assembly was performed using sensorimotor primitives.

Previous work in robotic assembly can be found in [1], where optimization of force control parameters with respect to cycle time is made in assembly of a clutch. An example from the automotive industry is [6], which describes powertrain assembly. In [14] synchronized Petri nets are used to model the assembly process and an experimental evaluation is made with a peg-in-hole assembly. An example of assembly from the construction industry where position control is used is described in [5].

A survey of robotic manipulators in space with focus on dynamics and control is [12]. A general framework for tool planning in assembly tasks is given in [13]. An application of force control in robotics other than assembly is [11], where force control is used to avoid sliding movements when drilling.

The assembly scenario in this paper is to assemble the internals of an emergency stop button, see Fig. 1. A switch should be placed in one of five available slots and has to be snapped into the right position. The location and the orientation of the bottom box is assumed to be uncertain, as well as the grasping of the switch. Force sensing is used to resolve these uncertainties.

The objective is to create a framework that enables the operator to specify assembly tasks in an easy way, as a step-by-step procedure. No exact modelling of the involved parts is necessary, since finding contact points and following surfaces can be handled by the force sensing

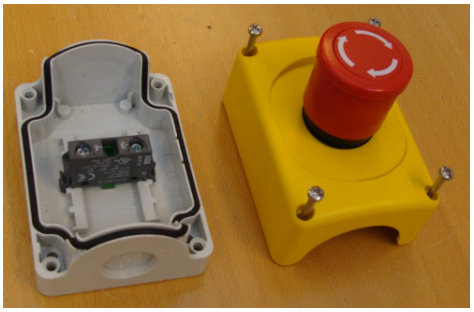

Fig. 1. The emergency stop button that is used as an experimental case.

II. TASK MODELING

A. Constraint-based task specification

A thorough explanation of the constraint-based task specification methodology is given in [3]. A summary of the parts that are relevant for this paper are given below.

Andreas Stolt, Magnus Linderoth, Anders Robertsson, and Rolf Johansson are with the Department of Automatic Control, LTH, Lund University, Sweden, andreas.stolt@control.lth.se

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 – Challenge 2 – Cognitive Systems, Interaction, Robotics – under grant agreement No 230902 - ROSETTA. This document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained herein.

The constraint-based task specification framework specifies the relative motion of objects by imposing constraints. To be able to specify these constraints a so called kinematic chain is needed, and it consists of two object frames and two feature frames. The first object frame is usually attached to the object one wants to manipulate and the second object frame is usually attached to the robot. The feature frames should be attached to features on the object to manipulate and on the robot. They should be chosen in such a way that the task constraints become as easy as possible to specify. A kinematic chain should have 6 degrees of freedom, and they are distributed over the transformations between the feature and the object frames. These six degrees of freedom are represented by χ_f, the so called feature coordinates.

Aside from the feature coordinates there might also be uncertainties in the pose between the previously defined coordinate frames. To handle this problem an extra transformation between each of the the previously mentioned coordinate frames is introduced, and the degrees of freedom in these transformations are χ_u, the uncertainty coordinates.

The variables one wants to constrain are chosen by specifying outputs y. In general, each output can be a function of the feature and the robot joint coordinates, but if the kinematic chains have been chosen properly the outputs will in most cases directly correspond to some of the feature coordinates.

B. Kinematic chain in the specific task

One kinematic chain is used in the assembly task and the object and feature frames related to it are shown in Figs. 2 and 3. An overview of how the frames are connected by the feature coordinates is given in Fig. 4.

- Object frame $o1$ is attached to the box. It is related to the world coordinate frame by a constant transformation.
- Feature frame $f1$ is attached to one end of the switch. The orientation is the same as $o1$.
- Feature frame $f2$ has its origin in the same position as $f1$, but the orientation is the same as the robot flange frame.
- Object frame $o2$ coincides with the robot flange frame.

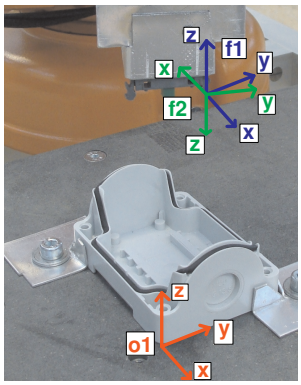

Fig. 2. Illustration of the different coordinate frames in the assembly task.

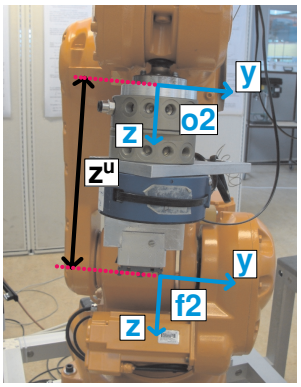

Fig. 3. Illustration of the uncertainty coordinate z^u between the two frames $f2$ and $o2$.

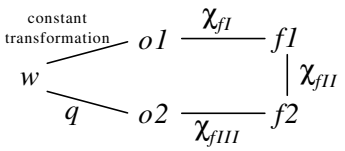

Fig. 4. Schematic description of the kinematic chain in the assembly task. w denotes the world coordinate frame and q denotes the robot joint coordinates.

The feature coordinates χ_f are divided into three groups depending on which frames they relate. The coordinates used are

$$\chi_{fI} = (x, y, z) \qquad o1 \rightarrow f1$$
$$\chi_{fII} = (\varphi, \theta, \psi) \qquad f1 \rightarrow f2$$
$$\chi_{fIII} = (-) \qquad f2 \rightarrow o2$$

The coordinates χ_{fI} give the position of the origin of $f1$ using Cartesian coordinates in $o1$. χ_{fII} describe the rotation from $f1$ to $f2$ using Euler ZYX-angles. χ_{fIII} has no feature coordinates, since the transformation between $f2$ and $o2$ is fix.

In this task all feature coordinates are chosen as outputs, according to

$$y_1 = x \quad y_2 = y \quad y_3 = z$$
$$y_4 = \varphi \quad y_5 = \theta \quad y_6 = \psi$$

Uncertainties in the task include the exact location of the box and its orientation. They are however resolved using guarded search motions, i.e., the motion is velocity controlled in the search direction and stopped when a contact force is detected. Once contact is made, it is maintained by using force control, and hence no explicit uncertainty coordinates are used to model this uncertainty. The exact position of the grasp is also assumed to be uncertain, and the z-distance from $f2$ to $o2$ (Fig. 3) is therefore modeled as an uncertainty coordinate z^u. The distance in x and y are also uncertain, but they are small compared to the z-distance and therefore considered to be known with sufficient accuracy.

C. Uncertainty estimation

An illustration of the uncertainty coordinate z^u is given in Fig. 3. It can be estimated by performing a rotation in ψ (rotation around $f2$ x-axis, see Fig. 5) while keeping the switch in contact with the box. If z^u is known exactly, the contact forces at the origin of $f2$ will remain constant during the rotation without any force control. In practice the contact forces will change and force controllers will modify the velocity references in the y and z directions to maintain the forces.

Let us assume that there is an estimation error $\tilde{z}^u = z^u - \hat{z}^u$, illustrated in Fig. 5, where $\hat{z}^u$ is an estimate of z^u. This estimation error gives rise to attempted rotations around the origin of frame $f2'$ instead of around the origin of $f2$. Since the contact is force controlled the actual rotation will, however, be made around the origin of $f2$, and the velocity of $o2$ in its y-direction will be $v = -\dot{\psi} z^u$.

This can be used to set up a dynamical model for z^u, according to (1), where the state $s = z^u$ and the measurement $m = v$ is the velocity of $o2$ in its y-direction, g and n are Gaussian noise.

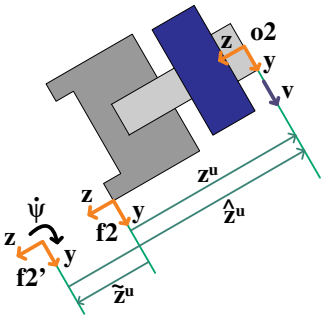

Fig. 5. Illustration of the uncertainty coordinate z^u.

$$\begin{cases} \dot{s} = g(t) \\ m = -\dot{\psi}s + n(t) \end{cases} \tag{1}$$

A Kalman filter [8] can be used to estimate z^u. A discretized model of (1) is (2), with system matrices defined in (3). The noise covariances are assumed to be given by (4).

$$\begin{cases} s(k+1) = As(k) + w(k) \\ m(k) = C(k)s(k) + e(k) \end{cases} \tag{2}$$

$$A = 1 \quad , \quad C = -\dot{\psi} \tag{3}$$

$$\begin{aligned} E\left[w(k)w^T(k)\right] &= Q(k) \\ E\left[w(k)e^T(k)\right] &= 0 \\ E\left[e(k)e^T(k)\right] &= R(k) \end{aligned} \tag{4}$$

A Kalman filter for the model (2) is given by (5)-(11).

$$\hat{s}(k|k-1) = A\hat{s}(k-1|k-1) \tag{5}$$

$$P(k|k-1) = AP(k-1|k-1)A^T + Q(k) \tag{6}$$

$$\tilde{m}(k) = m(k) - C(k)\hat{s}(k|k-1) \tag{7}$$

$$S(k) = C(k)P(k|k-1)C^T(k) + R(k) \tag{8}$$

$$K(k) = P(k|k-1)C^T(k)S^{-1}(k) \tag{9}$$

$$\hat{s}(k|k) = \hat{s}(k|k-1) + K(k)\tilde{m}(k) \tag{10}$$

$$P(k|k) = \left(I - K(k)C(k)\right)P(k|k-1) \tag{11}$$

D. Transient detection

During the final stage of the assembly the switch snaps in place on the box. Empirical studies of the transients originating from the snap show that a distinct signature is present. Sharp dips can be seen both in the z-force and the ψ-torque (around $f2$ x-axis). A way to detect the snap is to assume that the expected snap signature and its probability distribution are known. A measured sequence of force and torque samples can then be seen as an outcome of a multivariate random variable. By further assuming that this random variable is Gaussian it is possible to calculate the probability for a certain sequence of force and torque samples to be the outcome of a snap.

The probability density distribution can be learned by manually marking where the snap has occurred for a number of snaps. This makes it possible to calculate the expectation value and the covariance matrix. The probability density function for a measured sequence $x \in \mathbb{R}^n$ is given by (12), where μ is the expectation value and Σ the covariance matrix.

$$f(x) = \frac{1}{(2\pi)^{n/2}\left|\Sigma\right|^{1/2}} \exp\left(-\frac{1}{2}\left(x-\mu\right)^T \Sigma^{-1}\left(x-\mu\right)\right) \tag{12}$$

A way to detect a snap is to check whether x lies inside the hyper-ellipsoid defined by (13).

$$\left(x-\mu\right)^T \Sigma^{-1}\left(x-\mu\right) < c^2 \tag{13}$$

Using this condition can be motivated by the fact that all points inside the hyper-ellipsoid have higher probability density than all points outside the hyper-ellipsoid, which can easily be verified by inserting (13) into (12). For even dimensions n the probability p of a sample x resulting from a snap to be inside the hyper-ellipsoid (13) is given by

$$\begin{aligned} p &= \int_{(x-\mu)^T\Sigma^{-1}(x-\mu)<c^2} f(x)dx \\ &= 1 - \left(\sum_{i=0}^{n/2-1} \frac{(c^2/2)^i}{i!}\right) \exp\left(-\frac{c^2}{2}\right) \end{aligned} \tag{14}$$

A good feature of this detection method is that it is well suited for a real-time implementation. Since the sample size n and the desired probability p are the same for all samples during execution, the constant c in (13) can be calculated in advance using (14). Furthermore, μ and Σ are calculated offline using training data. So all that has to be done in real-time is to insert x in (13) and check whether the inequality is satisfied. Sequences of nine force samples and nine torque samples were considered. Before the method was applied to the data the offset was removed by considering a weighted average value.

E. Learning

When searching for contact during assembly, there is a delay between the moment when the contact is first sensed by the force sensor and the time when the robot actually stops. If the contacting material is stiff, then the contact force increases very rapidly during this delay and the components may break. Hence, the approach phases have to be made quite slow.

A learning strategy can be used to speed up the assembly. During a number of assemblies the positions where the components make contact are recorded. When the distribution of the contact points is known the speed profile of the approach can be changed so the speed is high in the beginning and then decreases when it comes to the region where the contact is expected to be made.

III. ASSEMBLY SCENARIO

The assembly strategy chosen was to methodically eliminate all uncertainties by guarded search motions. A state machine describing the assembly operation is given in Fig. 6. In Fig. 7 snapshots from the execution are shown with red arrows indicating the direction of movement.

It was assumed that the position and orientation of the box was known well enough for an initial search operation (state 2) to hit the bottom of the box in front of the slot to mount the switch in, see the red area in Fig. 8. Once contact was made, linear searches were performed to find the slot (states 3 and 4). Next, the switch was rotated to make contact on the opposite side of the switch (state 5) and then turned until it fell into the slot on that side (state 6). The initial pose should be such that it was known which way to turn to find the slot. Once this was done the remaining step was to push the switch down and wait for the snap (state 7).

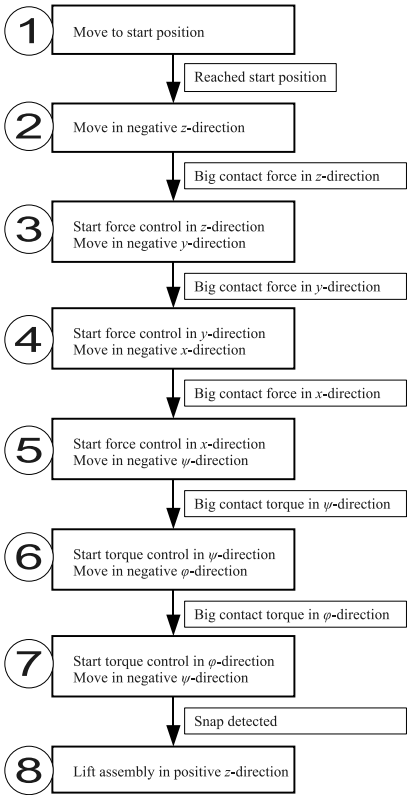

Fig. 6. State machine describing the assembly sequence.

IV. EXPERIMENTAL RESULTS

A. Uncertainty estimation

The estimation scheme described in Section II-C was implemented in the assembly scenario. The time spent in the estimation state was rather short though, which made it hard to draw any conclusions. Another experiment was therefore performed to try out the estimation. Initial contact was searched for in z-direction and then in y-direction. The contact forces were controlled and a rotation in ψ (around the x-axis) was started. Two different initial guesses were used and the result is found in Fig. 9.

The y-force was controlled to 3 N and the z-force to 5 N. The estimator converged to about the same distance in both experiments. The forces approached the references and the velocities, the force control modifications due to an incorrect estimate, approached zero during the experiments.

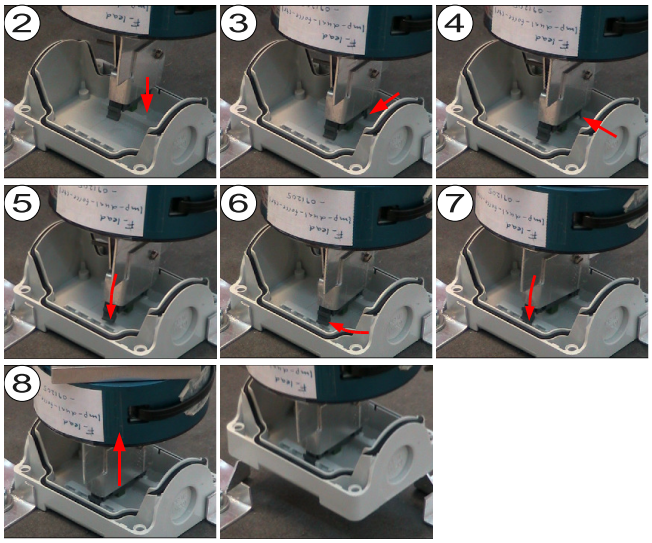

Fig. 7. Snapshots of the assembly sequence. The search directions in the different states of the assembly are indicated with red arrows.

B. Transient detection

A typical snap can be seen in Fig. 10 at $t = 1.65$ s. Snaps from two different switches have been logged, ten snaps from each switch. The training data is presented in Fig. 11, where also the point-wise mean snap is given. It is clearly seen in the torque diagram that two different signatures are present, originating from the different switches. The result of the snap detection method when applying it to three new snaps from each switch is shown in Fig. 12. The inverse of the test value (left hand side of (13)) is plotted over time for the six snaps. The percentages on the horizontal lines indicate the ratio of the snaps that will be detected if that line is used as the threshold.

As Fig. 12 indicates the method is well suited for the snap detection. As the influence from the noise is low it is easy to find a suitable threshold that decides when there has been a snap. One choice could be the 99 % threshold, which means that it is chosen so that 99 out of 100 snaps will be detected by the method.

Different structures of the covariance matrix were evaluated. In Fig. 12 it has been tested to let the random variables be independent of each other, i.e., the covariance matrix was diagonal. Experiments were performed using both band-diagonal and full covariance matrices, but both variants caused a degradation of the results. The reason for this might be that the estimated covariance is not accurate enough, when only relying on twenty measured snaps.

C. Learning

The learning strategy described in Section II-E was implemented on the system. The controllers were already tuned to give the fastest possible assembly without risking to damage the components. When the system had been trained to know the distribution of the contact points, the speed of the approach was increased by a factor of 4 outside the expected contact region. This decreased the assembly time from 8.6 s

to 3.6 s. Diagrams of the states over time with and without training can be seen in Fig. 13.

D. Assembly sequence

Force data from an experimental execution is given in Fig. 14, together with the corresponding state in the assembly sequence. The first three states shown are linear search motions in z, y and x (state 2 has been divided into two substates, where 2 is a fast z-search and 2.5 a slow z-search). The transition conditions are large contact forces in the corresponding search directions, which is easily seen in the force plot. States 5 and 6 are rotational searches, the transition conditions are large corresponding torques. Notice how the ψ-torque (around $f2$ x-axis) drops around $t = 10$ s. This is because the switch slides down into the slot in the box. State 7 is the push-down state, and when the snap occurs (at $t = 11.7$ s) a transition to the last state is made, where the robot lifts the switch and the box to show that it has finished the assembly.

More insight is given in Fig. 15, where velocity data from the experiment is shown. Measured data is given when the corresponding coordinates are position or velocity controlled, but the control signal (the desired velocity) is given when the coordinates are force controlled. The search motions are easy to recognize on the non-zero velocity, and also the push down state can be interpreted as a search (in ψ).

V. DISCUSSION

The uncertainty estimation performed in Section IV-A did not converge to a distinct value, which probably was caused by that the rotation was made around a rounded edge, rather than a point contact. The deviation from zero in the velocity diagram was probably caused by a small error in the y-distance, which was assumed to be known. The estimation was corrupted in the end of the experiments when the switch made contact at a second point, which was not covered by the model.

The forces and torques are controlled to a constant value once contact is made in the assembly task. The diagrams in Section IV-D, however, show something else. The reason for this is probably the relatively large search speeds, that introduce disturbances for the orthogonal force controlled directions.

The assembly speed can be substantially increased when

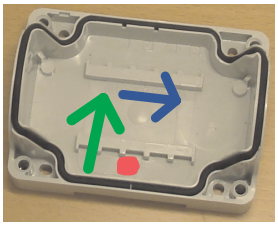

Fig. 8. Illustration of where the initial search should make contact (red area) and the following search directions, green first and then blue.

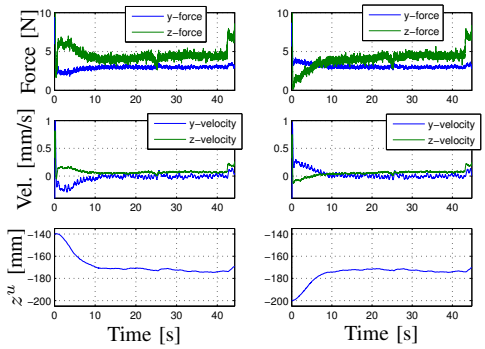

Fig. 9. Data from the two uncertainty estimation experiments. The uppermost diagrams show the y- and z-forces, the middle diagrams the y- and z-velocities, which equal the control signals. The lowermost diagrams finally show the time evolution of the estimation of the uncertainty coordinate.

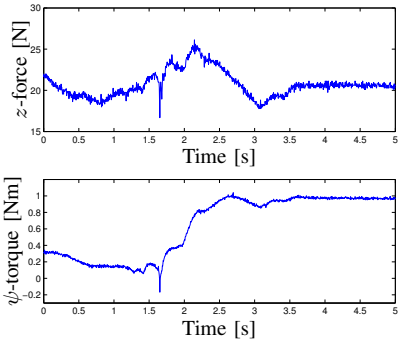

Fig. 10. Illustration of experimental data for a typical snap over time. The upper diagram shows the z-force that is controlled to 20 N and the lower diagram shows the ψ-torque that is controlled to 1 Nm. The snap signature can be seen at $t = 1.65$ s.

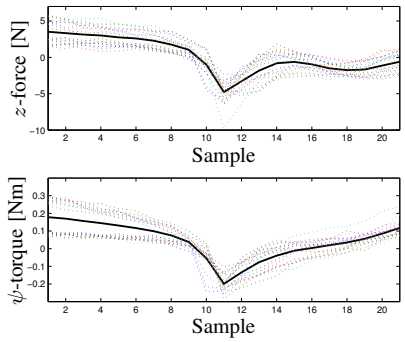

Fig. 11. The data used for training of the snap detection method. The dotted curves are measured force and torque sequences in the vicinity of the snap, which occurs att sample 11. The thick black line is the point-wise mean value of the data.

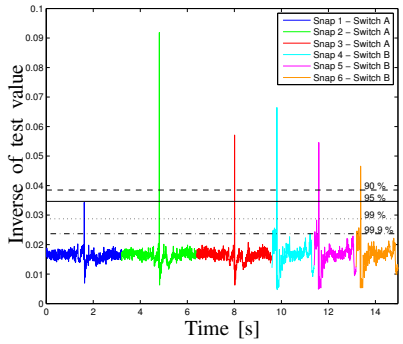

Fig. 12. Results of snap detection method. The test value is the left hand side of (13). Three snaps from each of the two used switches are used, and data from respective snap are arranged sequentially. The snaps are clearly seen as the spikes in the diagram. The horizontal black lines correspond to confidence thresholds.

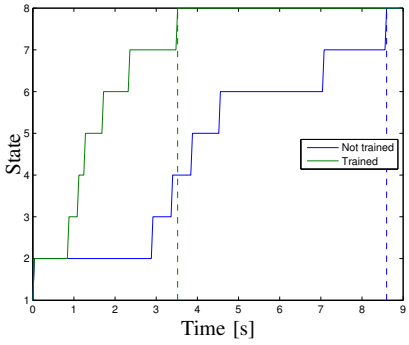

Fig. 13. Time evolution of the state sequence from the assembly task with and without training. The vertical dashed lines indicate when the assembly operations finish.

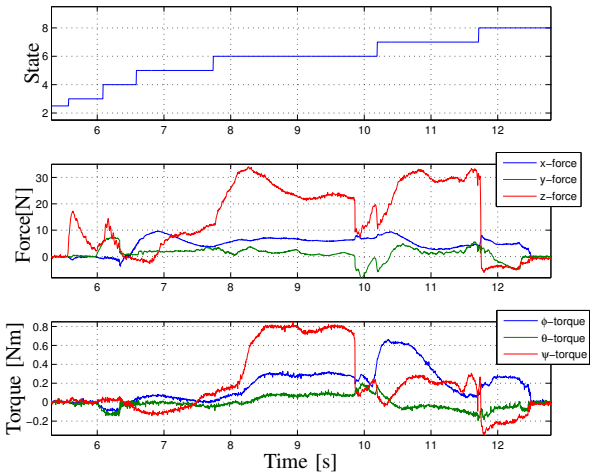

Fig. 14. Force data from an assembly sequence vs. time. The uppermost diagram shows the state sequence, the middle the forces and the lowermost the torques.

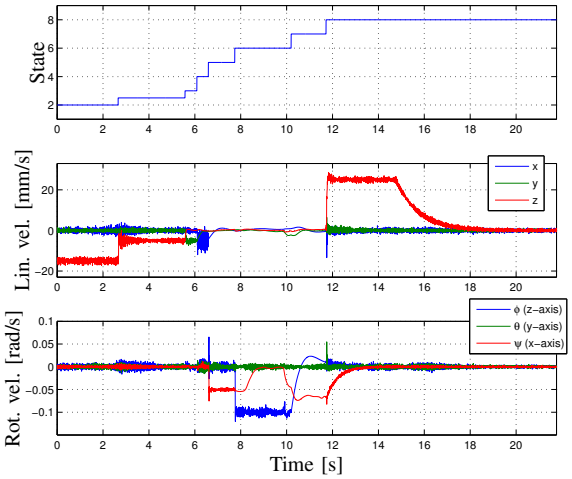

Fig. 15. Velocity data from an assembly sequence vs. time. The uppermost diagram shows the state sequence, the middle the linear velocities and the lowermost the rotational velocities.

using the learning approach described in Section II-E. This strategy will however only speed up the assembly to a certain point, because it does not include the actual search motions that make contact. The contact stiffness together with unavoidable time delays in the control system requires search speeds to be slow and force controllers can not be too aggressive. There are however ways to use learning to speed up the sequence even more. The positional and orientational uncertainty of the box can be learned, which can then be used to choose an appropriate initial pose. This approach can also be implemented using a vision system.

The feature coordinate system chosen in this paper contains a representation singularity, and in the current implementation it is left to the user to avoid problems with this, by choosing a representation such that the singularity is never reached in the specific task. This is however something that an implementation should be able to take care of. An alternative solution would be to use an internal singularity-free representation.

The snap detection algorithm has only been trained using two different switches. These have shown slightly different signatures, and there is therefore a risk that the algorithm be overtrained. In a real assembly scenario each switch would be a new one, so the training data would probably be more representative if a new switch was used for each training snap.

VI. CONCLUSIONS

Force control was used to successfully perform the assembly of two components with uncertain positions. The task was conveniently specified using constraint-based task specification methodology. Online estimation of an uncertain parameter was used to improve performance. A learning strategy was used to speed up the assembly significantly by performing the assembly many times and generating better feed-forward data.

REFERENCES

[1] T. Arai, N. Yamanobe, Y. Maeda, H. Fujii, T. Kato, and T. Sato. Increasing Efficiency of Force-Controlled Robotic Assembly -Design of Damping Control Parameters Considering Cycle Time. *CIRP Annals-Manufacturing Technology*, 55(1):7–10, 2006.

[2] H. Bruyninckx, T. Lefebvre, L. Mihaylova, E. Staffetti, J. De Schutter, and J. Xiao. A roadmap for autonomous robotic assembly. In *Proc. of the Int. Symp. on Assembly and Task Planning*. Fukuoka, Japan, 2001.

[3] J. De Schutter, T. De Laet, J. Rutgeerts, W. Decré, R. Smits, E. Aertbeliën, K. Claes, and H. Bruyninckx. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *Int. J. Robotics Research*, 26(5):433, 2007.

[4] B.K.G. Di Xiao, N. Xi, and TJ Tarn. Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment. *IEEE Trans. Control Systems Technology*, 8(4):635, 2000.

[5] E. Gambao, C. Balaguer, and F. Gebhart. Robot assembly system for computer-integrated construction. *Automation in Construction*, 9(5-6):479–487, 2000.

[6] D. Gravel, F. Maslar, G. Zhang, S. Nidamarthi, H. Chen, and T. Fuhlbrigge. Toward robotizing powertrain assembly. In *7th World Congress Int. Control and Automation*, pages 541–546, 2008.

[7] S. Jörg, J. Langwald, C. Natale, J. Stelter, and G. Hirzinger. Flexible robot-assembly using a multi-sensory approach. In *IEEE Int. Conf. Robotics and Automation*, volume 4, pages 3687–3694. San Francisco, USA, 2000.

[8] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME–J. Basic Engineering*, 82(Series D):35–45, 1960.

[9] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. Robotics and Automation*, 3(1):43–53, 1987.

[10] J.D. Morrow, B.J. Nelson, and P.K. Khosla. Vision and force driven sensorimotor primitives for robotic assembly skills. In *Proc. of the Int. Conf. on Intelligent Robots and Systems*. Pittsburgh, PA, USA, 1995.

[11] T. Olsson, M. Haage, H. Kihlman, R. Johansson, K. Nilsson, A. Robertsson, M. Björkman, R. Isaksson, and G. Ossbahr. Cost-efficient drilling using industrial robots with high-bandwidth force feedback. *Robotics and Computer-Integrated Manufacturing*, 26(1):24–38, 2010.

[12] P. Van Woerkom and AK Misra. Robotic manipulators in space: a dynamics and control perspective. *Acta Astronautica*, 38(4-8):411–421, 1996.

[13] R.H. Wilson. Geometric reasoning about assembly tools. *Artificial Intelligence*, 98(1-2):237–279, 1998.

[14] W. Zhang, T. Mao, and R. Yang. A new robotic assembly modeling and trajectory planning method using synchronized Petri nets. *The International J. Advanced Manufacturing Techn.*, 26(4):420–426, 2005.