

Hidden Markov Models as a Process Monitor in Robotic Assembly*

GEIR E. HOVLAND† and BRENNAN J. McCARRAGHER‡

Keywords: *Discrete event systems, sensory perception, robotic assembly*

A process monitor for robotic assembly based on hidden Markov models (HMMs) is presented. The HMM process monitor is based on the dynamic force/torque signals arising from interaction between the workpiece and the environment. The HMMs represent a stochastic, knowledge-based system in which the models are trained off-line with the Baum-Welch reestimation algorithm. The assembly task is modeled as a discrete event dynamic system in which a discrete event is defined as a change in contact state between the workpiece and the environment. Our method (1) allows for dynamic motions of the workpiece, (2) accounts for sensor noise and friction, and (3) exploits the fact that the amount of force information is large when there is a sudden change of discrete state in robotic assembly. After the HMMs have been trained, the authors use them on-line in a 2D experimental setup to recognize discrete events as they occur. Successful event recognition with an accuracy as high as 97% was achieved in 0.5-0.6 s with a training set size of only 20 examples for each discrete event.

1. Introduction

Process plants must deal with changing states, multiple faults, unexpected situations, and unreliable measurements. Efficient process-monitoring techniques allow for robust control of such process plants. A control system with reliable process monitoring requires less stringent models of the process plant and its environment. Hence, an efficient process monitor is a natural first step in designing intelligent controllers for systems in uncertain environments. In robotic assembly, sophisticated task-level control schemes such as discrete event control have been developed in recent years. To allow for good decisions in such controllers, real-time process-monitoring techniques are needed to account for existing uncertainties in workpieces and the environment.

In this paper, we present a method for recognition of discrete events in robotic assembly by using force and torque measurements. A discrete event is defined as a change in contact state between the workpiece and the environment. Each discrete event in the assembly is described by a hidden Markov model (HMM). HMMs are well suited to event recognition in robotic assembly for two reasons. First, the state space is naturally discretized which is a basic requirement for any Markov modeling. Second, the sensory signals, particularly force sensing, have stochastic noise, which is a requirement for HMMs. The recognition of contact formations presented here has several advantages. First, we allow for dynamic motions of the workpiece (not limited to quasi-static as in Hirai (1989) and Trinkle and Zeng (1995)). Second, the method is model based and the models are trained on empirical data, which helps to ensure

*Reprinted by permission of Sage Publications, Inc. Hovland and MacCarragher, The International Journal of Robotics Research, pp. 153-168, Vol. 17, No. 2. © 1998 Sage Publications, Inc.

†ABB Corporate Research, Oslo, Norway.

‡Department of Engineering, The Australian National University, Canberra, Australia.

the validity of the models. Third, both sensor noise and friction are accounted for. Fourth, as in Eberman and Salisbury (1994), the method uses the fact that the frequency band is broad within a short period of time when a change in contact formation occurs, and hence a lot of information can be extracted.

2. Literature Survey

2.1. Process Monitoring of Assembly

One example of process monitoring in robotic assembly is found in Donald (1990), where a theory of planning multistep error detection and recovery strategies for compliant motion assemblies is described. The theory provides a technology for constructing plans that might work, but fail in a reasonable way when they cannot. However, the direct applicability of this theory is limited in robotic applications.

Eberman and Salisbury (1994) described a method for detecting state changes in a dynamic system. Four states are considered: impacts, sliding across a surface, no contact, and grasping contacts. For each of these states, a statistical force-signal model is built. The no contact model, for example, is chosen to be the Gaussian distribution $N(\mu_0, V_0)$. The validity of the models is checked against spectrograms of empirical data. Hypothesis testing is used to recognize different states, in which there is one stochastic representation of the force measurements for each state. Since the number of possible states is only four, the information given by the method is limited. For example, in robotic assembly, the state sliding across a surface covers a whole range of situations, since there may be many surfaces involved.

Bicchi, Salisbury, and Brock (1993) presented a method for resolving the location of a contact given force and torque measurements. They consider three contact types: (1) a point contact without friction, (2) a point contact with friction, and (3) a soft finger contact. Closed-form solutions for a static point contact with or without friction and a static soft finger contact for ellipsoidal surfaces are presented. An iterative solution is presented for a soft finger contact for any other surface. Multiple contact points are represented by a point called the contact centroid, through which resultant forces are applied. In robotic assembly, multiple contact points occur frequently. Although the contact centroid contains some geometric information about multiple contact points, in general it does not provide enough information to distinguish all possible contact formations between the workpiece and the environment.

In Hirai (1989), the highlight is on the recognition of the contact formation between the workpiece and the environment, not the change in contact. Assuming quasi-static motions of the workpiece, negligible friction, and noiseless sensing, they use the theory of polyhedral convex cones to find the set of statically admissible contact forces and moments between the workpiece and the environment. The possible contact formations are then the formations in which the measured forces/moments are contained in this set. This method only applies to systems with quasi-static motions, negligible friction, and noiseless sensing.

A different approach is used by McCarragher and Asada (1993). Assuming polygonal models of the workpiece and the environment, they highlight the discrete changes of state. A discrete change of state is defined as a change in the geometric constraint between the workpiece and the environment. This method uses qualitative force/torque measurements and hence is well suited for fast real-time monitoring in robotic assembly. However, by using qualitative measurements, a significant amount of information is ignored. It is therefore expected that more accurate, reliable, and robust

process monitoring in robotic assembly can be achieved by a more comprehensive treatment of the force/torque measurements.

Trinkle and Zeng (1995) describe a motion prediction method for a system with several bodies and multiple points of contact. Assuming quasi-static, planar rigid-body motions, they have developed a model to predict the motion of a single rigid workpiece contacted at multiple points by a general planar manipulator. The method allows for sliding and rolling contacts. Trinkle and Zeng plan to extend the formulation to 3D motions and multiple workpieces. This method applies only to systems with quasi-static motions.

Dutr , Bruyninckx, and De Schutter (1996) presented a different approach based on the energy of the manipulated object. The energy is the sum of the kinetic and potential energies, where quasi-static motions are assumed. A total energy error function is defined that depends on the geometrical uncertainty parameters ν . The values of ν for which the total energy error function reaches a minimum are found, and these values describe the contact situation. The contact model is validated by observing only one signal, the same total energy error function. Two advantages of the method are that it does not require training examples and the monitoring is based on only one signal. One limitation is the assumption of quasi-static motions.

In Badano *et al.* (1991), a different approach to robotic assembly is used. They show experiments for a planar peg-in-the-hole assembly in which the position error between the peg and the hole is compensated for by slight random movements. One limitation of this approach is that the random movements might change the contact state. Also, for a more complex assembly task, there is no guarantee that random movements will successfully insert the workpiece in the environment.

In the work presented in this paper, we allow for dynamic motions of the workpiece, friction, and sensor noise. Also, we do not require exact knowledge of the positions of the workpiece and the environment. Moreover, the method is model based and uses empirical data, which helps to ensure the validity of the models by training them using a maximum likelihood procedure. The method also exploits the broad frequency band when there is a sudden change in the discrete state in the assembly task. Hence, the solution presented here is a significant improvement over existing solutions.

2.2. Hidden Markov Models

Since the introduction of Markov models to speech processing in the 1970s, dramatic advances have been made in this area (Picone 1990). Hidden Markov models (HMM) have been used by several authors in speech recognition (for example, Huang, Ariki, and Jack 1990; Rabiner *et al.* 1985a), and in character recognition (Vlontzos and Kung 1992). The experimental results in these papers all show a high successful recognition rate. In speech recognition, Picone (1990) measured the recognition rate as high as 99.3% for a vocabulary of 2,000 words and 94.6% for a vocabulary of 20,000 words. In character recognition, Vlontzos and Kung (1992) achieved recognition rates from 97% to 99% with printed as well as handwritten characters. In the work presented here, we propose similar techniques for recognition of contact formations in robotic assembly using the continuous force/torque measurements.

The number of papers presented in which the theory of HMMs is used in robotics is small. One example is found in the paper by Hannaford and Lee (1991), in which an HMM is used as a process monitor for the teleoperation of a peg-in-the-hole

assembly. The HMM describes the task structure, and each state in the HMM describes a subtask. The different states in the HMM are characterized by different means and covariances of the force/torque measurements. Hannaford and Lee studied a peg-in-the-hole task with four subtasks; i.e., moving, tapping, inserting, and extracting. The moving subtask has a force/torque mean of zero, both the tapping and the inserting subtasks have a positive (but different) mean value, and the extracting subtask has a negative mean value, and these values distinguish the different states (subtasks). Each of these subtasks is described by a single mean value, and this number is dependent on the magnitude of the contact forces. These individual subtasks cover a whole range of contact formations. Hence, the information given by the process monitor has few details. In the work presented here, we make the process monitoring independent of the contact force magnitudes, and we also divide the assembly process into a larger number of subtasks, which makes the process monitoring more accurate and detailed. Instead of using only one HMM for all subtasks, we use one HMM for each subtask. Moreover, we get a more comprehensive treatment of the measurements by using frequency components of the force/torque signals.

Zhu (1991) used an HMM to describe and predict the motions of obstacles in a dynamic environment. Initially, an image of the scene is acquired by a visual sensing device. This image is processed, and any potential obstacles in the environment are detected. The HMM is then used to make predictions of the obstacle motions. A collision-free path is searched for, and finally maneuver commands are generated. As the robot proceeds, this cycle repeats. Although the HMM is not used as a process monitor here, and the application is not in robotic assembly, the work presented illustrates how an HMM can be used in a dynamic system.

Another example of HMMs applied to a different area in robotics is the work by Xu and Yang (1995) and Yang, Xu, and Chen (1994), in which the HMM approach is applied to skill learning in robotics. They use HMMs to represent two types of skill—action skill and reaction skill. To learn the action skill, they use a left-to-right HMM. The HMM characterizes two stochastic processes: the measurable action and the immeasurable mental states that are involved in the skill learning. The reaction skill is modeled as a set of HMMs, in which each HMM describes the mapping from sensory data to control action.

3. Discrete Event System

The process monitor that is the focus of this paper is part of a discrete event system. The process monitor cooperates with a discrete event controller (DEC) to achieve successful assembly. In this section, we give a brief overview of the control structure. The discrete event system is shown in Figure 1. The control structure is divided into a continuous and a discrete part. The continuous part controls the velocity of the workpiece in Cartesian space, where the velocity reference is set by the DEC. The process plant, consisting of the manipulator, the workpiece, and the environment, is nonlinear and time variant. Hence, different continuous controllers and velocity references are needed for different situations. For example, the DEC might vary the robot PI gains and the velocity commands depending on the discrete state γ in Figure 1. The purpose of the process monitor is to detect and classify the time-varying situations.

In Figure 1, γ is the recognized discrete state from the process monitor, where the definition of discrete state is given in Definition 1.

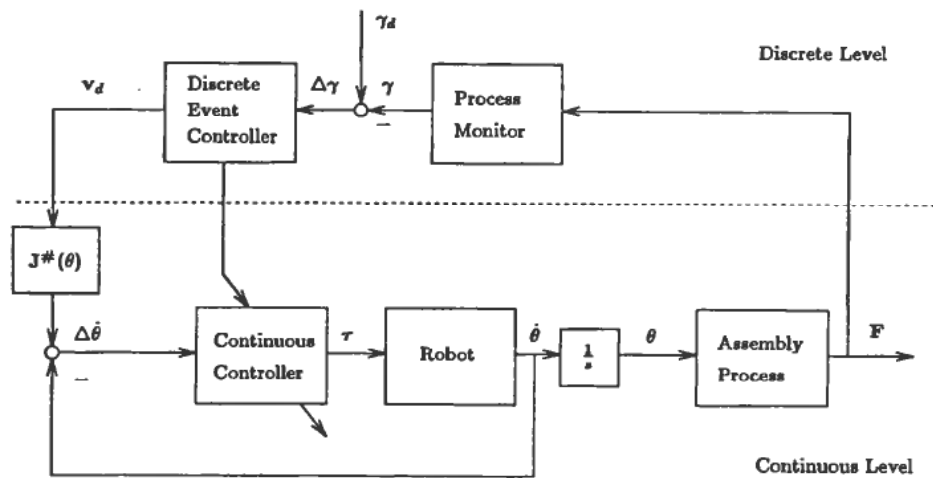


Figure 1. Block diagram of discrete event system for robotic assembly.

Definition 1 (Discrete State). A discrete state is defined as a contact formation consisting of zero, one, or more edge-surface contacts between the workpiece and the environment. The set of discrete states is then the set of all possible contact formations (McCarragher and Asada 1995a). \square

The control system is described briefly as follows. γ_d is the desired contact state, \mathbf{F} is the force/torque measurements from the interaction between the robot and the environment, \mathbf{v}_d is the Cartesian velocity reference from the DEC, θ and $\dot{\theta}$ are the position and velocity measurements from the robot and τ is the commanded motor torques from the continuous controllers. The function $\mathbf{J}^{\#}(\theta)$ is a transformation from Cartesian space velocities to joint space velocities; i.e., $\dot{\theta}_d = \mathbf{J}^{\#}(\theta)\mathbf{v}_d$. In the experiments, we use the closed-loop tracking solution presented in the paper by Chiacchio *et al.* (1991, 413). The closed-loop solution avoids the long-term drifts that occur with the simple inverse Jacobian approach. The manipulator joints are controlled individually by PI velocity controllers.

We consider a planar assembly task in which we have only one type of contact, the *edge-surface* contact. The edge or the surface involved can be a part of either the workpiece or the environment. Examples of this contact type are shown in Figure 2.

We do not model *edge-edge* contacts, since this type of contact consists of a point only. The moving workpiece causes the edge-edge contacts to exist for a very short time period. Accurate process monitoring of such contacts is difficult. Note that a *surface-surface* contact is allowed, and it is described by two edge-surface contacts.

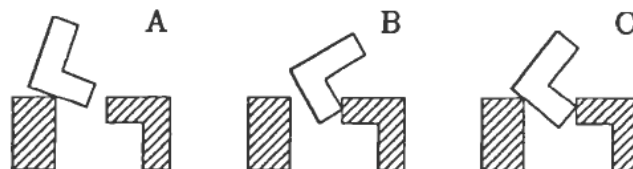


Figure 2. Examples of different edge-surface contacts between the workpiece and the environment. Note that contact state C is a combination of the two edge-surface contacts A and B.

The method presented here can be extended to a 3D assembly task (McCarragher 1996). For a 3D task, the HMMs would be trained on all 6-DOF force/torque measurements.

In addition to the discrete contact states, we define the discrete events as follows.

Definition 2 (Event). An event in robotic assembly is a change of discrete state. The events are discrete in time and describe a gain or a loss of a single edge-surface contact between the workpiece and the environment. \square

We use the notation from Astuti and McCarragher (1995), in which γ_i is used for contact states, $M = \{\gamma_1, \gamma_2, \dots, \gamma_m\}$ is a finite set of discrete states, and $E \subset M \times M$ is a finite set of discrete events. Any discrete event e_k is then given by an ordered pair $e_k = (\gamma_i, \gamma_j) \in E$, describing a change from the discrete state γ_i to γ_j . Given a contact state γ_i , only events from a subset of E are possible. We define this subset of admissible events as $E_{\gamma_i} \subset E$. The subset E_{γ_i} is determined from the contact state network of the assembly task. For a description of contact state networks for robotic assembly, see McCarragher and Asada (1995a).

The discrete event model and controller are described by McCarragher and Asada (1995a) and McCarragher and Asada (1995b). First, a discrete event model of the assembly process is developed. Based on this model, the DEC directs the assembly process through a series of events to the desired end state in the discrete event model. The DEC uses a discrete event trajectory that specifies the discrete controls. The discrete control vector is then mapped onto continuous velocity commands to be sent to the robot arm. The execution of these velocity commands will ideally cause the system to pass through the desired events. Since the workpiece is in contact with the environment, the dynamic equations of motion depend on the discrete contact state. Astuti and McCarragher (1995) described how to find the constraint equations for each contact state by using distance functions. The desired Cartesian velocity commands v_d are then found by solving an optimization problem with linear constraints in the commanded velocities.

It is extremely difficult to guarantee that the velocity commands will guide the system through the desired events. To overcome this problem, the process monitor is used to give the DEC feedback; i.e., to tell which events occur. The process monitor uses the force/torque measurements resulting from interaction between the robot arm and the assembly environment. The output of the process monitor is the recognized contact formation, and this output is compared with the desired contact formation. If there is a mismatch between these two, an error has occurred, and proper control actions must be given to correct the error. Ideally, the commanded velocities from the DEC will guide the system through the desired event trajectory. However, since it is difficult to achieve this, a process monitor plays a very important role in discrete event control. With a process monitor, the system is able to detect errors and unwanted situations, which is an important step toward an intelligent control structure with the ability to recover from errors.

4. Process Monitor

The process monitor developed in this paper is based on HMMs. An HMM is defined as a doubly stochastic process. The underlying stochastic process is *not* observable (it is hidden), but it can be observed through another set of stochastic processes

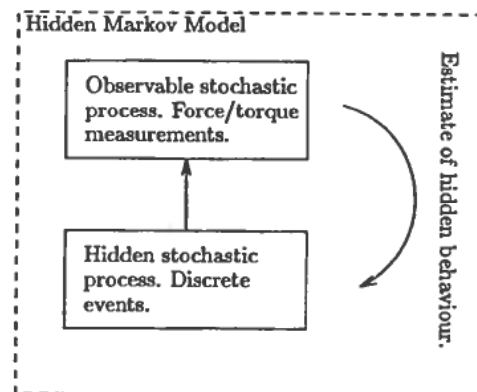


Figure 3. Illustration of an HMM. The underlying stochastic process can only be observed through another stochastic process.

that produce the sequence of observed symbols (Rabiner and Juang 1986). Stochastic force/torque measurements are used to obtain information about the discrete events, which are illustrated in Figure 3. Each state in the hidden process has a stochastic description of the force measurements. The observation symbols are compared to these stochastic descriptions (for example, Gaussian distributions) to find the probability of each hidden state.

In addition to the force/torque measurements being stochastic, the underlying process must also be stochastic. In our work, the hidden models represent the discrete events defined in the previous section. The underlying stochastic process corresponds to the internal behavior of the events, since we are using one HMM for each event. The internal behavior is influenced by several factors. In a robotic assembly task, for example, the internal stochastic behavior is caused by the position of the workpiece in the gripper, friction between the workpiece and the environment, or tolerancing errors in the geometries of the workpiece and the environment.

The stochastic behavior of the underlying process is very important because the whole theory of HMMs is based on the fact that both the underlying process and the observed process are stochastic. In the paper by Hannaford and Lee (1991), the underlying process is not the internal behavior of the events but rather the sequence of discrete events. In that case, the assumption has to be made that the sequence of discrete events is stochastic. However, when the hidden process is the sequence of discrete events, the HMM probabilities will be highly dependent on the event trajectories chosen in the training set. In the method presented here, the HMMs are independent of the actual event trajectory, and hence modifying the trajectory, on-line to recover from errors does not influence the recognition rate of the HMMs.

There are many possible HMM models for the events. One type of model that has been successfully applied to speech recognition and also works well for robotic assembly is the absorbing parallel left-to-right model illustrated in Figure 4. (See Picone (Figure 9) 1990, for other possible left-to-right models.)

For this model, an event always starts in the state q_1 and ends in q_6 or q_7 , but there are several possible paths through the states. The observed symbols proceed forward in time, which is the main reason for using a left-to-right model. The left part of the model is associated with the early measurements whereas the right part of the model is associated with measurements at some later time. Since these measurements

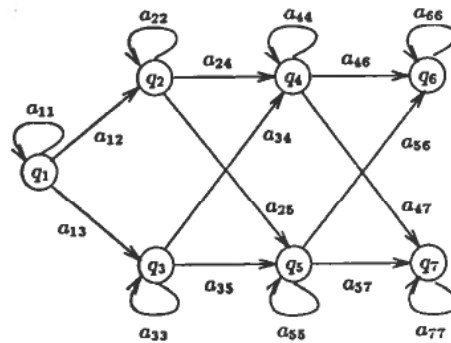


Figure 4. HMM λ^k for each event. q_i are the states and a_{ij} are the hidden stochastic process state change probabilities.

might be very different, a left-to-right model is ideal for extracting this information. Left-to-right models are a good tool for describing how signals evolve in time. This is an important property of the HMMs. With this property, we are able to describe dynamic motions with dynamic force/torque measurements of the workpiece, as opposed to quasi-static motions.

4.1. Observation Symbols

One key element in successful process monitoring is the observation symbols. In this paper, we use the frequency components of the force/torque signals as observation symbols. There are several reasons for using frequency components. First, when an event occurs in robotic assembly, as well as in any dynamic system where there is a sudden change in the constraint equations, the frequency band is broad and occurs within a short time scale of the event (see Figure 5).

Therefore, a lot of information can be extracted by using frequency components as observation symbols. Second, by normalizing the total energy in the measurements, we make the observation symbols independent of the signal magnitudes; i.e., a gentle contact (event) will have similar observation symbols as a brutal contact. It is also

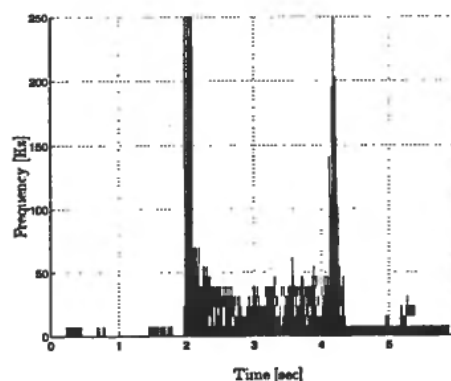


Figure 5. Spectrogram of the F_y force measurements for two events. The first event starts at $t = 2.0$ s followed by another event at $t = 4.2$ s. The measured force was sampled at 500 Hz. Notice that the frequency band is broad and occurs within a short time scale of the event.

worth noting that research on human tactile perception has shown that our skin force sensors are sensitive to force frequencies (Johansson 1978). By using frequency components, the training will estimate the mean and the covariance of the energy in different frequency components of the force/torque measurements. Hannaford and Lee (1991) used the mean and covariance of the force magnitude measurements directly in a robotic system where a HMM is used as a process monitor.

In this paper, we use continuous observation symbols represented by the continuous vector \mathbf{x}_t . The observation probability $b_j(\mathbf{x}_t)$ is then given by

$$b_j(\mathbf{x}_t) = \mathcal{N}[\mathbf{x}_t, \mu_j, \Sigma_j] \quad (1)$$

where \mathcal{N} is the normal density and μ_j and Σ_j are the mean vector and covariance matrix, respectively, associated with the internal state j in the HMM. It is possible to use discrete observation symbols, but better performance has been reported for continuous HMMs (Huang, Ariki, and Jack 1990), which is caused mainly by the vector quantization error in discrete observation symbols.

We now define the transform

$$\mathcal{C}_j(\omega_i) = \left| \log \frac{\mathcal{F}(\omega_i)_j}{\mathcal{F}(0)_j} \right| \quad (2)$$

where $j \in \{F_x, F_y, M_z\}$ are the 3-DOF force/torque measurements for a planar assembly task from time t to $t + W$. W is a constant design parameter representing the length of the observation window. \mathcal{F} is the Fourier transform, and ω_i are the observation frequencies. As mentioned above, we divide each frequency component by $\mathcal{F}(0)_j$ to make them independent of signal magnitudes. The logarithm function is used to make the covariance of the observation symbols smaller. Small covariances will make the observation probabilities larger relative to the hidden process state probabilities, which is desirable in the training and the operation of the HMMs. The reason we want the observation probabilities large compared to the hidden process state probabilities can be seen from Figure 3. The hidden process is observed via the observation probabilities.

With n observation frequencies, the continuous vector \mathbf{x}_t is then given by

$$\mathbf{x}_t = [\mathcal{C}_{F_x}(\omega_1), \dots, \mathcal{C}_{F_x}(\omega_n), \mathcal{C}_{F_y}(\omega_1), \dots, \mathcal{C}_{F_y}(\omega_n), \mathcal{C}_{M_z}(\omega_1), \dots, \mathcal{C}_{M_z}(\omega_n)]^T \quad (3)$$

4.2 Length of Observation Window

Once we have chosen our observation symbols, we have to choose the length of the observation window W . This choice is a compromise between the amount of information and the time spent waiting for the information. The additional information from time $t - 1$ to time t can be measured by looking at the average change in the observation vector \mathbf{x}_t for all discrete events. In this paper, we use a threshold test on the change of the observation vector, since the amount of information from an event will decrease with time. The threshold test is given by

$$\frac{\sum_{e_k \in E} \sum_{l_t^k \in L^k} |\mathbf{x}_t^{l_t^k} - \mathbf{x}_{t-1}^{l_t^k}|}{n_e n_l} \leq \Delta x \quad (4)$$

where $|\cdot|$ is the vector norm, n_e is the number of events $e_k \in E$, and n_t is the size of the training sets L^k . When the change in the observation symbol vector from time $t-1$ to t is less than a chosen threshold percentage Δx , we stop making observations. The data used are then x_0, \dots, x_t . The threshold test in eq. (4) is performed once and off-line on the entire training set, and we use the same observation length for all events. We also use the same observation length in the on-line recognition process. By using this threshold test, we assure that a certain amount of the information will be contained in the observation symbols x_0, \dots, x_t . For the experiments presented in this paper, the threshold test in eq. (4) was used on a training set containing examples for each discrete event. It was found that the average change in the observation vector was less than 25% with an observation sequence of length 4. This observation sequence length was used in the experiments; i.e., the observations are always x_0, x_1, x_2 , and x_3 .

4.3. Training

The first step of the HMM recognition method is to build an HMM for each event $e_k \in E$. The most difficult problem in HMMs is how to adjust the model parameters to maximize the probability of the observed data matching the model. There is no known way to solve this analytically for a maximum likelihood model (Huang, Ariki, and Jack 1990), so using a numerical method is desirable. The method we use is a standard estimation method called the Baum-Welch reestimation formula (see for example Rabiner and Juang 1986).

We use the following notation for a HMM based on continuous observations:

- T = length of observation sequence
 - N = number of hidden process states in the HMM
 - $Q = \{q_1, q_2, \dots, q_N\}$ hidden process states
 - $A = \{a_{ij}\}$ HMM hidden process probability matrix
 - π = initial state probability
 - δ = absorbing state probability
 - x_t = continuous observation vector at time t
 - $B = \{b_j(x_t)\}$ HMM observable process probability vector
- (5)

We use the compact notation $\lambda = (A, B, \pi)$ to represent an HMM. For the model in Figure 4, we have

$$\begin{aligned}
 N &= 7 \\
 Q &= \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\} \\
 \pi &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \\
 \delta &= [0 \ 0 \ 0 \ 0 \ 0 \ \delta_1 \ 1 - \delta_1]^T
 \end{aligned}$$

where $\delta_1 \in [0, 1]$ will be changed during the training. δ_1 is the probability of ending in hidden state q_6 , whereas $1 - \delta_1$ is the probability of ending in hidden state q_7 .

The Baum-Welch method starts with initial values of all the HMM parameters.

After applying the method, the probability of the observed data matching the HMM is increased. Applying the Baum–Welch formula once is called an iteration. For the probability to reach a (local) maximum, we have to run several training iterations.

The Baum–Welch reestimation formula is based heavily on two variables: the *forward* and *backward* variables. They are defined as follows.

Definition 3 (Forward Variable α).

$$\alpha_t(j) = Pr(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_t, j = q_j | \lambda^k) \quad (6)$$

where $\alpha_t(j)$ is the probability of the partial observation sequence (until time t) and hidden process state q_j at time t , matching the model λ^k . \square

The forward variable α is calculated for every hidden process state in the HMM λ^k at every discrete time step $t = 1, \dots, T$ as follows.

1. $\alpha_1(i) = \pi_i b_i(\mathbf{x}_0) \quad 1 \leq i \leq N$.
2. Calculate $\alpha(\cdot)$ along the time axis, for $t = 2, \dots, T$, and $1 \leq j \leq N$:

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(\mathbf{x}_t) \quad (7)$$

Definition 4 (Backward Variable β).

$$\beta_t(i) = Pr(\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \dots, \mathbf{x}_T | j = q_i, \lambda^k) \quad (8)$$

where $\beta_t(i)$ is the probability of the partial observation sequence from $t + 1$ to the end, given the hidden process state q_i at time t , matching the model λ^k . \square

The backward variable β is also calculated for every hidden process state in the HMM λ^k at every discrete time step $t = 1, \dots, T$.

1. $\beta_T = \sigma$, absorbing state probability.
2. Calculate $\beta(\cdot)$ along the time axis, for $t = T - 1, \dots, 1$, and $1 \leq i \leq N$; i.e.,

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(\mathbf{x}_{t+1}) \quad (9)$$

If the observation sequence \mathbf{x}_t is long, a scaling technique of α and β is required to avoid mathematical underflow, since all the probabilities are less than or equal to 1.0. In the experiments presented in this paper, the observation sequences are of length 4 only, and mathematical underflow does not occur.

Definition 5 (HMM Probability). The HMM probability is defined as the probability of the entire observation sequence $\mathbf{x}_0, \dots, \mathbf{x}_T$ matching the model λ^k . Finding this probability for a model λ^k is called scoring the model. This is done by using the forward variable α as follows:

$$Pr(\mathbf{x}_0, \dots, \mathbf{x}_T | \lambda^k) = \sum_{j=1}^N \alpha_T(j) \quad (10)$$

\square

In robotic assembly, the HMM probability for λ^k is the probability of the observed force/torque measurements matching the model λ^k for event $e_k \in E$.

From the forward and backward variables, we now define two new variables describing the HMM state probabilities. The difference between HMM state probabilities and hidden process state probabilities is that the HMM state probabilities take into account both the hidden process and the observable process. These new variables will be used in the final form of the Baum-Welch reestimation formula.

Definition 6 (HMM State Probability $\sigma(j)$).

$$\sigma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\Pr(\mathbf{x}_0, \dots, \mathbf{x}_T | \lambda^k)} \quad (11)$$

where $\sigma_t(j)$ is the probability of being in the hidden process state q_j at time t , given the observation sequence $\mathbf{x}_0, \dots, \mathbf{x}_T$ and the model λ^k . \square

Definition 7 (HMM State Transition Probability $\xi(i, j)$).

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(\mathbf{x}_{t+1})\beta_{t+1}(j)}{\Pr(\mathbf{x}_0, \dots, \mathbf{x}_T | \lambda^k)} \quad (12)$$

where $\xi_t(i, j)$ is the probability of a path being in state q_i at time t and making a transition to state q_j at time $t + 1$, given the observation sequence and the model. \square

We now use the definitions for $\sigma(j)$ and $\xi(i, j)$ to express the Baum-Welch reestimation formula. For each event $e_k = (\gamma_i, \gamma_j) \in E$, we have a training set $\mathbf{L}^k = \{\mathbf{L}_1^k, \mathbf{L}_2^k, \dots, \mathbf{L}_n^k\}$ in which individual \mathbf{L}^k are matrices containing force/torque samples describing event e_k . In the planar case, we have that $\mathbf{L}_i^k = \{\mathbf{l}_{Fxl}^k, \mathbf{l}_{Fyl}^k, \mathbf{l}_{Mzl}^k\}^T$, where \mathbf{l}_{Fxl}^k , \mathbf{l}_{Fyl}^k , and \mathbf{l}_{Mzl}^k are vectors containing the force measurements in the x- and y-direction and the torque measurements in the z-direction, respectively, describing event e_k . These measurements are transformed into observation symbols \mathbf{x}_t as described in Section 4.1. Given the model λ^k with initial parameters, the reestimation formula for the internal state transition parameters a_{ij}^k is then given by

$$\bar{a}_{ij}^k = \frac{\sum_{\mathbf{L}_i^k \in \mathbf{L}} \sum_{t=1}^{T-1} \xi_t^{\mathbf{L}_i^k}(i, j)}{\sum_{\mathbf{L}_i^k \in \mathbf{L}} \sum_{t=1}^{T-1} \sum_{j=1}^N \xi_t^{\mathbf{L}_i^k}(i, j)} \quad (13)$$

We see that the estimate \bar{a}_{ij}^k is the sum of all transition probabilities from the state q_i to the state q_j divided by the probability of being in state q_i for model λ^k . The new estimates \bar{a}_{ij}^k are based on the HMM state transition probabilities $\xi(i, j)$, so we see now how the hidden process is estimated through the observable process. Since $\mathbf{x}_t = \mathbf{x}_t^{\mathbf{L}_i^k}$ are different for each training sequence, we also have that ξ_t are different for each training sequence \mathbf{L}_i^k , and hence are written $\xi_t^{\mathbf{L}_i^k}$.

By using multivariate Gaussian distributions, the estimates for μ_j^k and Σ_j^k for the model λ^k in eq. (1) are given by

$$\bar{\mu}_j^k = \frac{\sum_{\mathbf{L}_i^k \in \mathbf{L}} \sum_{t=1}^T \sigma_t^{\mathbf{L}_i^k}(j) \mathbf{x}_t}{\sum_{\mathbf{L}_i^k \in \mathbf{L}} \sum_{t=1}^T \sigma_t^{\mathbf{L}_i^k}(j)} \quad (14)$$

$$\bar{\Sigma}_j^k = \frac{\sum_{\mathbf{L}_i^k \in \mathbf{L}} \sum_{t=1}^T \sigma_t^{\mathbf{L}_i^k}(j) (\mathbf{x}_t - \bar{\mu}_j^k)(\mathbf{x}_t - \bar{\mu}_j^k)^t}{\sum_{\mathbf{L}_i^k \in \mathbf{L}} \sum_{t=1}^T \sigma_t^{\mathbf{L}_i^k}(j)} \quad (15)$$

Theorem 1 (Maximum Likelihood Estimates). The Baum–Welch reestimation formula given in eqs. (13)–(15) is an iterative solution for a (local) maximum likelihood HMM.

Proof: See Huang, Ariki, and Jack (1990). ■

Define the scalar $S^k = \sum_{i^k \in \mathbf{L}} Pr(\mathbf{x}_0^k, \dots, \mathbf{x}_T^k | \lambda^k) / n_i$ to be the training set score for λ^k based on the entire training set, where n_i is the size of the training sets \mathbf{L}^k . The training procedure is then described as follows.

1. Choose an initial model λ^k for each event $e_k \in E$.
2. Get the new model parameters from eqs. (13)–(15). These new model parameters are based on the entire training set \mathbf{L}^k for that event. This is called *supervised training*, since we know that the training set \mathbf{L}^k belongs to the event e_k .
3. If the change in the training set score S^k is below a chosen threshold value, the training is completed. Otherwise, go to step 2.

4.4. Event Detection

As described earlier in this paper, the amount of force information available when there is a change in the constraint equations is large. This is illustrated in Figure 5, which is a similar result to Figure 1 in Eberman and Salisbury (1994). This large amount of information available when an event occurs is the main reason for letting the HMMs represent the events rather than the contact states. However, to be able to use this information, we need an event detector. A change in the constraint equations is characterized by a sudden change in at least one of the force measurements. The changes in the force measurements are found from the time derivatives of F_x and F_y .

$$H_x(t) = \frac{1}{\Delta T} (F_x(t) - F_x(t - \Delta T))$$

$$H_y(t) = \frac{1}{\Delta T} (F_y(t) - F_y(t - \Delta T))$$

ΔT is considered a filter design parameter. The frequency response of $H_x(t)$ is given by

$$\left| \frac{H_x(j\omega)}{F_x(j\omega)} \right| = |1 - e^{-j\omega\Delta T}| \quad (16)$$

We have that $\max |1 - e^{-j\omega\Delta T}| = 2$ when $\omega\Delta T = \pi + n2\pi$, $n = 1, 2, \dots$. The filter cutoff frequency is then given by $\omega_c = \pi/\Delta T$. A large value of ΔT will result in differentiation of the force signals for low frequencies only. The simplicity of the filters in the time domain, $H_x(t)$ and $H_y(t)$, is an advantage in fast on-line event detection.

The event detector used here for a planar assembly task is described by the following threshold test:

$$H_x(t)^2 + H_y(t)^2 \geq (\Delta F)^2 \quad (17)$$

As in the paper by McCarragher and Asada (1993), we determine the threshold by looking at the standard deviation of the measurements. We do not want the effects from the friction forces to trigger the threshold test, so the workpiece was slid across a surface and $(\Delta F)^2$ was chosen to be a 3σ deviation from the mean value of $H_x(t)^2 + H_y(t)^2$. In our experiments, ΔT was chosen 1/3 s which gives a filter cutoff

frequency $\omega_c = 3\pi \text{ rad/s} = 1.5 \text{ Hz}$. Note that the choice of ΔT is dependent on several factors such as the dynamics of the manipulator, the dynamics of the force sensor, and the surface properties. With the choice of $\Delta T = 1/3$ we found ΔF to be $9.5N$. The reason why we chose a slow filter and the large value of ΔF can be seen from the large friction forces in Figure 9 in the Experiments section.

4.5. Event Recognition

The event recognition method is described as follows.

4.5.1. Training

First we build an HMM for each event in the event set. We use the observations from the training set L to estimate the optimum parameters for each event e_k , represented by λ^k , for the k th event in the event set. We use well-known estimation techniques like the Baum-Welch reestimation formulas.

4.5.2. Operation

From on-line measurements, we have a single test sequence $Y = \{y_{F_x}, y_{F_y}, y_{M_z}\}^T$ containing the 3-DOF force/torque measurement vectors in the x -, y - and z -direction, respectively. For each event model, we calculate the model score for λ^k , $P_k = Pr(Y|\lambda^k)$. The model score is found using Definition 5.

4.5.3. Evaluation

We chose the event whose model score is highest; i.e.,

$$k^* = \operatorname{argmax}_{e_k \in E_{\gamma_i}} [P_k]$$

where E_{γ_i} is the set of admissible events for the given contact state γ_i .

By comparing the model scores for the different events, we are also able to calculate a confidence level of the decision. We define the confidence level of the decision as follows:

$$C = \frac{P_{k^*}}{\sum_{e_k \in E_{\gamma_i}} P_k} \in (0, 1] \quad (18)$$

Note that $\sum_{e_k \in E_{\gamma_i}} P_k$ is not necessarily equal to one, since the model scores P_k are independent. For example, e_1 , e_2 and e_3 might be the only events in the set E_{γ_i} with model scores $P_1 = 0.1$, $P_2 = 0.2$, and $P_3 = 0.5$. In this case, $k^* = 3$, and the confidence level $C = 0.5/0.8 = 62.5\%$.

5. Experiments

The experimental setup consists of an Eshed Scorbot 5-DOF manipulator and a JR3 force/torque sensor (see Figure 6). The dimensions of the experimental setup are given in Figure 7.

We study a planar assembly task with 11 contact states as shown in Figure 8. As mentioned before, we do not model edge-edge contacts, since this type of contact consists of a point only. For the contact states shown, we have 25 possible discrete events; i.e., changes of contact stage (see Table 1).

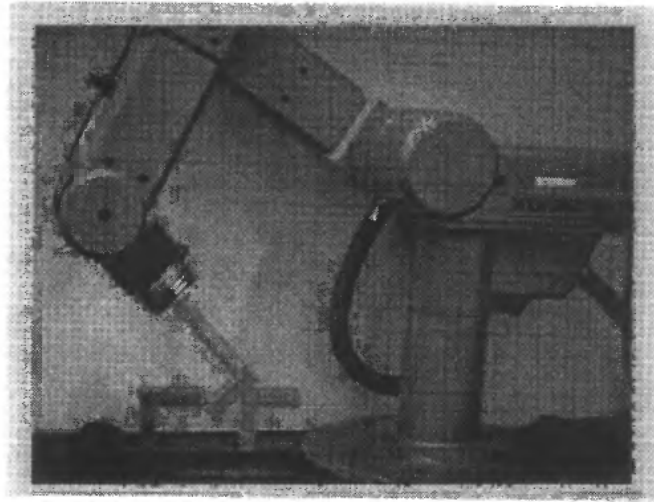


Figure 6. Eshed Scorbot 5-DOF manipulator and JR3 force/torque sensor used in the experiments.

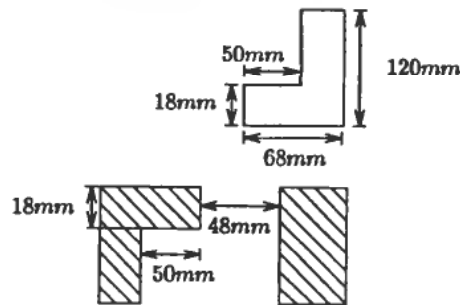


Figure 7. Dimensions of the experimental setup.

The HMMs were trained with varying sizes of the training set. Typical measurements in the training set for one of the events, e_5 , are given in Figure 9, where the force directions are as shown in Figure 8.

The performance of the HMM-based process monitor is shown in Figure 10. The recognition rates shown in the figure were obtained in on-line operation in a discrete event control framework. Also, the real-time measured forces were independent of the training set examples. Note that even with a training set size of only 4, the average recognition rate is as high as 81.5%. In our experiments, event e_{19} was among the most difficult events to recognize. This can be seen from Figure 10, in which the recognition rate for e_{19} was only 65% for a training set size of 4. Event e_5 is among the easy events to recognize. In our experiments, the method never failed to recognize e_5 , even with a training set size of only 4.

In Figure 11, we show the force measurements for the entire assembly task. The initial contact state in γ_1 , and γ_{10} is our desired final contact state. The robot guides the workpiece through the contact state sequence.

$$\gamma_1 \rightarrow \gamma_2 \rightarrow \gamma_3 \rightarrow \gamma_4 \rightarrow \gamma_5 \rightarrow \gamma_7 \rightarrow \gamma_9 \rightarrow \gamma_{10}$$

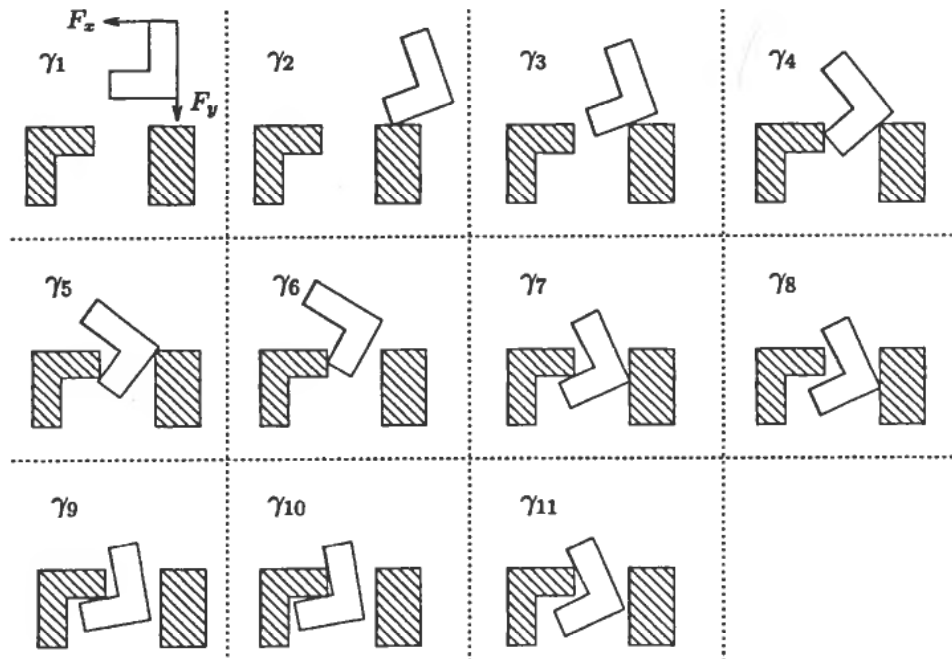


Figure 8. Model of contact states for a planar assembly task. We have not included every possible contact state in our model.

which is generated by the event trajectory

$$e_1 \rightarrow e_5 \rightarrow e_8 \rightarrow e_{10} \rightarrow e_{14} \rightarrow e_{19} \rightarrow e_{22}$$

The change in contact states (events) characterized by the sudden change in at least one of the force measurements is detected by the event detection method described in Section 4.4. After the event is detected, the method described in Section 4.5, is used to recognize the event. Once the recognition algorithm is completed, the information is sent to the discrete event controller, and the detection algorithm is started again to wait for the next event.

In the experiments, we used 10 observation frequencies chosen as shown in Table 2. The particular frequencies chosen are a result of sampling at 500 Hz and using windows of length 64 samples for the Fourier transform.

The observation frequencies are evenly distributed to allow the HMMs to describe both low-frequency and high-frequency behavior of the force measurements. With 10 observation frequencies, the length of the continuous observation vector \mathbf{x}_i is 30 and is given by eqs. (2) and (3).

For the HMM-based process monitor to work properly, good initial model parameters are needed for the maximum likelihood estimation algorithm. In the experiments presented here, we have used initial estimates as described as follows. When an event occurs in the training set for event model λ^k , the ensuing 256 samples are used by the HMMs. These 256 samples are divided into 4 windows of length 64 samples each. The first window is associated with state q_1 in Figure 4. The second window is associated with q_2 and q_3 , the third window with q_4 and q_5 and the fourth window with q_6 and q_7 , where q_i belong to λ^k (see Figure 12). The samples of F_x , F_y , and M_z are used to generate the observation symbols $\mathbf{x}_0, \dots, \mathbf{x}_3$. Samples from the

Table 1. Nomenclature

$\gamma_i \in \Gamma$	= set of possible contact formations between workpiece and environment
γ_d	= desired contact formation between workpiece and environment
$e_k \in E$	= set of all possible discrete events
n_e	= number of possible discrete events
E_{γ_i}	= set of possible events in contact state γ_i
$\theta, \dot{\theta}$	= robot joint coordinates and velocities
\mathbf{v}_d	= desired discrete Cartesian velocity command vector
$\mathbf{J}^*(0)$	= transformation from Cartesian velocities to joint velocities
\mathbf{L}^k	= training set for event e_k
n_l	= size of training set \mathbf{L}^k
λ^k	= hidden Markov model (HMM) for event e_k
$q_i \in Q$	= set of hidden process states for each HMM λ^k
N	= number of hidden process states
a_{ij}	= probability of a hidden process transition $q_i \rightarrow q_j$
π	= initial HMM hidden state probabilities
δ	= absorbing HMM hidden state probabilities
\mathbf{x}_t	= observation symbol at time t
T	= length of observation sequence
$b_j(\mathbf{x}_t)$	= observation probability for hidden state q_j at time t
μ_j	= vector of observation symbol mean values for hidden state q_j
Σ_j	= matrix of observation symbol covariances for hidden state q_j
$\mathcal{C}(\cdot)$	= transform from time-domain measurements to observation symbols
$\mathcal{F}(\cdot)$	= Fourier transform
W	= length of observation window
ω_i	= observation frequencies
$\alpha_i(j)$	= forward variable in hidden state q_j at time t
$\beta_i(j)$	= backward variable in hidden state q_j at time t
$\sigma_i(j)$	= HMM state probability in hidden state q_j at time t
$\xi_{i,j}$	= HMM state transition probability from hidden state q_i to q_j at time t
$F_x(t), F_y(t)$	= force measurement in the x - and y -direction at time t
$M_z(t)$	= torque measurement in the z -direction at time t
$H_x(t), H_y(t)$	= time derivatives of $F_x(t)$ and $F_y(t)$
S^k	= training set score for HMM λ^k

entire training set are used to find the means and the covariances of the observation symbols \mathbf{x}_i for each model λ^k , where \mathbf{x}_0 is used to initialize the observation probability densities in state q_1 , \mathbf{x}_1 is used in the states q_2 and q_3 , and so on.

Initializing the HMMs as illustrated in Figure 12 enables the HMM λ^k to describe the dynamic nature of the measurements. It is worth noting that the maximum likelihood training procedure might change the initial structure shown in Figure 12. If the measurements are static signals, for example, the entire observation sequence $\mathbf{x}_0, \dots, \mathbf{x}_3$ might be used to find the parameters for the state q_1 . We use the same initialization procedure for all the event models $\lambda^1, \dots, \lambda^{25}$.

The initial parameters for a_{ij} were chosen as follows:

$$\mathbf{A} = \begin{bmatrix} 0.04 & 0.48 & 0.48 & 0 & 0 & 0 & 0 \\ 0 & 0.04 & 0 & 0.8 & 0.16 & 0 & 0 \\ 0 & 0 & 0.04 & 0.16 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0.04 & 0 & 0.8 & 0.16 \\ 0 & 0 & 0 & 0 & 0.04 & 0.16 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

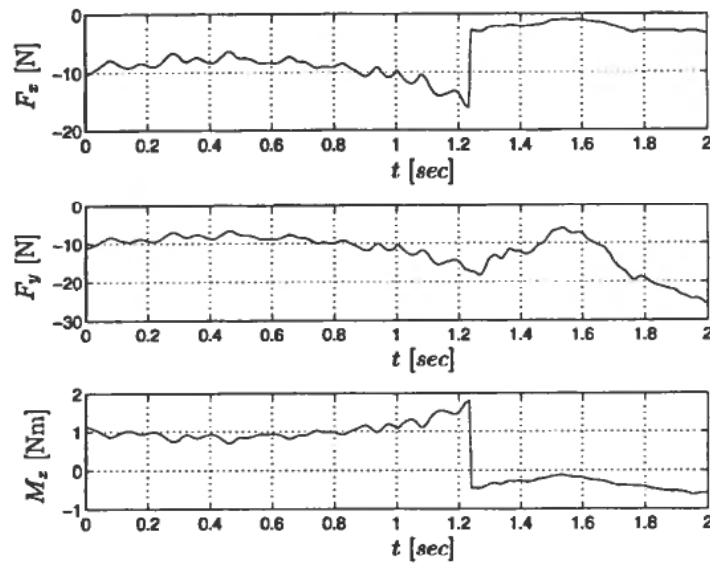


Figure 9. Typical measurements for event e_5 . The event occurs at $t = 1.23$ s, characterized by the sudden change in F_x . The time-varying force measurements before $t = 1.23$ s are caused by the friction when sliding along a surface. The process monitors log the data from $t = 1.23$ s to $t = 1.74$ s (256 samples at 500 Hz).

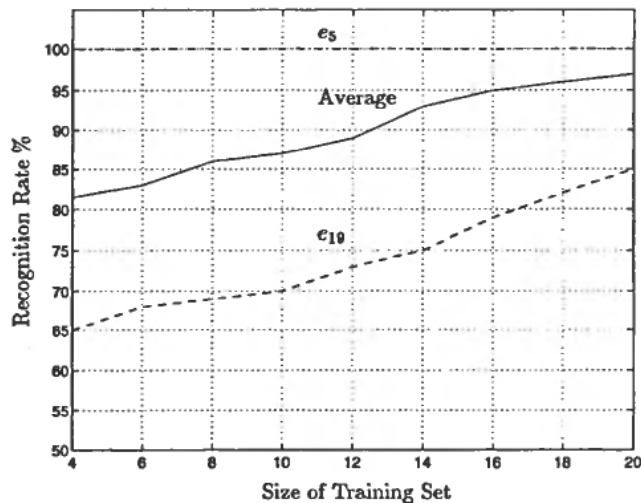


Figure 10. The performance rate versus the training set size.

The zeros and the ones in A are fixed probabilities because of the left-to-right structure of the HMMs. When choosing the initial HMM state transition probabilities, there are several possibilities. Since the final estimate is a local maximum, different initial parameters will in general produce different HMMs. However, the sensitivity of the recognition rates to errors in the initial hidden state probabilities is low (Rabiner *et al.* 1985b).

In our work, the training set contains measurements with two slightly different

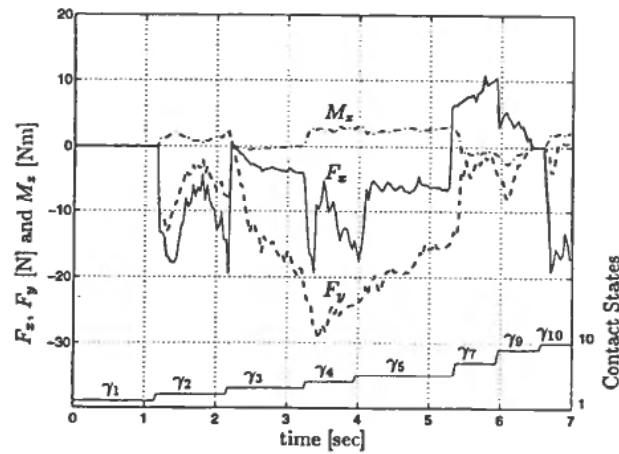


Figure 11. Force measurements for contact state sequence $\lambda_1 \rightarrow \lambda_2 \rightarrow \lambda_3 \rightarrow \lambda_4 \rightarrow \lambda_5 \rightarrow \lambda_7 \rightarrow \lambda_9 \rightarrow \lambda_{10}$.

Table 2. Possible Events for an Assembly Task with 11 Contact States

$e_1: \gamma_1 \rightarrow \gamma_2$	$e_6: \gamma_3 \rightarrow \gamma_1$	$e_{11}: \gamma_4 \rightarrow \gamma_6$	$e_{16}: \gamma_6 \rightarrow \gamma_4$	$e_{21}: \gamma_9 \rightarrow \gamma_7$
$e_2: \gamma_1 \rightarrow \gamma_3$	$e_7: \gamma_3 \rightarrow \gamma_2$	$e_{12}: \gamma_5 \rightarrow \gamma_3$	$e_{17}: \gamma_7 \rightarrow \gamma_5$	$e_{22}: \gamma_9 \rightarrow \gamma_{10}$
$e_3: \gamma_1 \rightarrow \gamma_6$	$e_8: \gamma_3 \rightarrow \gamma_4$	$e_{13}: \gamma_5 \rightarrow \gamma_4$	$e_{18}: \gamma_7 \rightarrow \gamma_8$	$e_{23}: \gamma_{10} \rightarrow \gamma_9$
$e_4: \gamma_2 \rightarrow \gamma_1$	$e_9: \gamma_4 \rightarrow \gamma_3$	$e_{14}: \gamma_5 \rightarrow \gamma_7$	$e_{19}: \gamma_7 \rightarrow \gamma_9$	$e_{24}: \gamma_{10} \rightarrow \gamma_{11}$
$e_5: \gamma_2 \rightarrow \gamma_3$	$e_{10}: \gamma_4 \rightarrow \gamma_5$	$e_{15}: \gamma_6 \rightarrow \gamma_1$	$e_{20}: \gamma_8 \rightarrow \gamma_7$	$e_{25}: \gamma_{11} \rightarrow \gamma_{10}$

Table 3. Observation Frequencies Used for Each of the Measurements F_x , F_y , and M_z

$\omega_1 = 7.8125$ Hz	$\omega_6 = 85.9375$ Hz
$\omega_2 = 23.4375$ Hz	$\omega_7 = 101.5625$ Hz
$\omega_3 = 39.0625$ Hz	$\omega_8 = 117.1875$ Hz
$\omega_4 = 54.6875$ Hz	$\omega_9 = 132.8125$ Hz
$\omega_5 = 70.3125$ Hz	$\omega_{10} = 148.4375$ Hz

angles between the workpiece and the environment. To make the HMMs represent these measurements effectively, we chose to have two dominant initial hidden state trajectories in the HMMs. The two dominant state trajectories are $q_1 \rightarrow q_2 \rightarrow q_4 \rightarrow q_6$ and $q_1 \rightarrow q_3 \rightarrow q_5 \rightarrow q_7$, which can be seen from the initial values of A .

As an example of what might happen after the training is completed, we consider the hidden process state probabilities for HMM λ^1 describing event e_1 :

$$A = \begin{bmatrix} 0 & 0.53 & 0.47 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.98 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 0.03 & 0.97 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.91 & 0.09 \\ 0 & 0 & 0 & 0 & 0 & 0.19 & 0.81 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This clearly shows the dynamic nature of the measurements, since all a_{ii} , $1 \leq i \leq 5$, are zero after the training. Also, we see that the two dominant hidden state trajectories

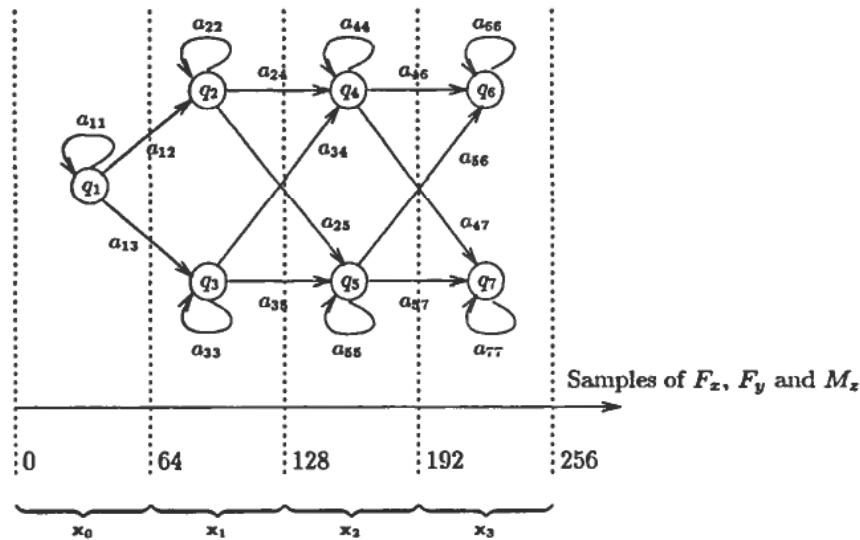


Figure 12. Initialization procedure of the HMMs.

are preserved. From the discussion above, we see that HMMs are a powerful tool to describe dynamic force signals. Also, with varying conditions in the training set, the HMM approach performs successful event recognition with high recognition rates. However, care must be taken when finding the initial probability density functions of the observation symbols.

6. Evaluation

The HMM force/torque based event recognizer has several advantages over existing methods, but the method also has its limitations. These are discussed in Sections 6.1 and 6.2. A discussion of how the method performs for more complex assembly tasks is given in Section 6.3.

6.1. Advantages

We allow for dynamic motions of the workpiece relative to the environment. There are no restrictions on the robot accelerations as there are for quasi-static motions. By allowing dynamic motions, the workpiece is no longer restricted to a certain kind of trajectory. We are now able to design continuous controllers with no restrictions on the accelerations. It should be noted, however, that the quasi-static approaches do not need training, and hence the implementation is less time consuming.

Our method is model based, where the models are trained on real measurements from the process plant. Hence, no assumptions on the measured forces/torques are made. Training the models on empirical data helps to ensure the validity of the models. After training, the model parameters can be used to extract meaningful information about the measured data. For example, if the force/torque measurements are static, the HMM probabilities a_{ii} , $1 \leq i \leq N$, will be large. On the other hand, if the measurements are dynamic, the HMM probabilities a_{ii} , $1 \leq i \leq N$, will be small, since the probability of staying in hidden state q_i is small when the observation symbols at time $t + 1$ are very different from the symbols at time t .

Since our method is trained on empirical data, the effects from friction between the workpiece and the environment and measurement noise are accounted for. In fact, any linear and nonlinear phenomena in the dynamic measurements are accounted for as long as the training data contains these phenomena. The HMM approach is based on a stochastic description of both the observable and the hidden process. Hence, the stochastic nature of phenomena such as friction is also taken into account. However, the method is not able to describe nonrepeatable phenomena. For example, wear and tear of the manipulator and the workpieces may cause the original training data to be an inaccurate description of the contact forces.

The frequency band of the force measurements is large when there is a sudden change in contact formation. By using the frequency components of the measured data, we are able to extract this information. Frequency information also eliminates the difference between hard and gentle contacts.

6.2. Limitations

One limitation of the HMM approach is the amount of training data required. In the work presented here, a training set size of 20 examples was used for each of the 25 discrete events, which gives a total of 500 examples. With this training set, a recognition rate of 97% was achieved. If one is willing to accept a lower performance, smaller training sets can be used, as shown in Figure 10. Also, the figure suggests that the training set sizes can be reduced for some of the discrete events, for example e_5 , without reducing the overall performance. Only events that are difficult to recognize require large training sets.

In our experiments, we are sampling at 500 Hz, and the observation window was chosen to be 256 samples. This means that the process monitor has to wait for approximately 0.5 s before starting to process the data. For time critical assembly tasks, 0.5 s might be too long. In such cases, the sample frequency should be increased such that the desired number of samples is collected in a shorter time.

Unless the initial model parameters are good, the estimated model parameters may be a local maximum. Although there exist other estimation algorithms, like the gradient projection method (Huo and Chan 1993), they all end up in a local maximum if starting with poor initial parameters. Rabiner *et al.* (1985b) simulated several 'toy' models in which the estimated parameters were compared to the true values. The conclusion in Rabiner *et al.* (1985b) was as follows: *With a good initial guess of the means (of the observation symbols), the parameter reestimation procedure is capable of yielding good models even if other model parameters have poor initial estimates.* In the Experiments section, we described how to get good initial estimates of the observation symbol distributions (see Figure 12) for successful event recognition. The procedure used to find the initial parameter estimates is not restricted to the work presented here. It can be applied to classification problems with left-to-right models and dynamic measurements.

6.3. Extension to More Complex Tasks

We have presented an HMM-based solution to the process monitoring of an assembly task with 11 contact states and 25 discrete events. For more complex tasks, the main effort will be the training of the HMMs. One HMM for each discrete event is required. Hence, for a more complex task, a training set larger than the 500 training examples used for the L-shaped workpiece is expected.

Once the HMMs are trained, however, we expect a similar high performance for more complex tasks as for the L-shaped assembly. In McCarragher (1996), the discrete event control of a 3D assembly task is studied. Naturally, the number of contact states and discrete events becomes large. However, given any contact state, only a limited number of discrete events are physically possible. An assembly task with a large number of contact states usually has a small number of admissible events for each contact state. Hence, for the process monitoring only, the HMMs corresponding to the admissible events have to be evaluated, and a high performance classification is still possible.

In general, there is no guarantee that the information in the force/torque signals is sufficient for successful process monitoring. For complex assembly tasks, it is desirable to make use of a wide range of sensory information; for example, force/torque, position/velocity, and vision. Moreover, it is desirable to combine the benefits from quasi-static and dynamic force based methods. Some of the discrete events are easy to recognize and only one monitoring technique is needed for accurate recognition. Other discrete events may be considerably more difficult to classify, and several different monitoring techniques are needed. Selecting different monitoring techniques in real time is a relatively new research area. One method for combining several process-monitoring techniques in a real-time discrete event control system is found in Hovland and McCarragher (1996). When using several monitoring techniques in real-time operation, the main goal is to achieve high performance monitoring while keeping the processing time low. The use of several techniques for improving the process monitoring becomes apparent as the degree of system uncertainty and ambiguity increases.

7. Conclusion

In this paper, we demonstrate how to use hidden Markov models (HMMs) and dynamic force measurements to recognize discrete events in robotic assembly. Extensive experimentation was conducted with a planar assembly task consisting of 11 contact states and 25 discrete events. For each discrete event, a training set of size 20 was generated. Based on this training set, the HMM parameters were estimated using a maximum likelihood algorithm—the Baum–Welch reestimation formula. Successful discrete event contact recognition was achieved in 0.5–0.6 s with an average recognition rate as high as 97%. The recognition rate of 97% was achieved in on-line operation in a discrete event control framework, where the real-time force measurements were independent of the training sets. Even with smaller training sets, successful event recognition was achieved with relatively large recognition rates. For example, with a training set size of only 4 examples, 81.5% of the discrete events in an independent test set were classified correctly. These results show that HMMs are a successful tool for process monitoring in robotic assembly. The HMMs are currently used in a discrete event control framework and effectively allow for on-line detection of errors in the discrete event trajectories. The use of sensory perception, as demonstrated in this paper for classifying contact states, is a first step toward an intelligent control structure with the ability to make task-level decisions in uncertain environments.

ACKNOWLEDGEMENTS

Geir E. Hovland was sponsored by the Research Council of Norway. This work was partially funded by a grant from the Australian Research Council.

REFERENCES

- ASTUTI, P. (1995). The convergence and control of a class of hybrid dynamic systems. PhD thesis, The Australian National University, Department of Engineering.
- BADANO, F. *et al.* (1991). Robotic assembly by slight random movements. *Robotica*, **9**, pp. 23–29.
- BICCHI, A., SALISBURY, J. K. and BROCK, D. L. (1993). Contact sensing from force measurements. *Int. J. Robotics Research*, **12**(3), pp. 249–262.
- CHIACCHIO, P., *et al.* (1991). Closed loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. *Int. J. Robotics Research*, **10**(4), pp. 410–425.
- DONALD, B. R. (1990). Planning multi-step error detection and recovery strategies. *Int. J. Robotics Research*, **9**(1), pp. 3–60.
- DUTRÉ, S., BRUYNINCKX, H. and DE SCHUTTER, J. (1996). (Minneapolis, MN, April 22–28). Contact identification and monitoring based on energy. *Proc. 1996 International Conference on Robotics and Automation*, pp. 1333–1338.
- EBERMAN, B. and SALISBURY, J. K. (1994). Application of change detection to dynamic contact sensing. *Int. J. Robotics Research*, **13**(5), pp. 369–394.
- HANNAFORD, B. and LEE, P. (1991). Hidden Markov model analysis of force/torque information in telemanipulation. *Int. J. Robotics Research*, **10**(5), pp. 528–539.
- HIRAI, S. (1989). Analysis and planning of manipulation using the theory of polyhedral convex cones. PhD thesis, Kyoto University, Department of Applied Mathematics and Physics.
- HOVLAND, G. E. and MCCARRAGHER, B. J. (1996). (Adelaide, November 21–22). Sensory perception and dynamic programming. *Proc. First Australian Data Fusion Symposium*.
- HUANG, X. D., ARIKI, Y. and JACK, M. A. (1990). *Hidden Markov Models for Speech Recognition*. Edinburgh: Edinburgh University Press.
- HUO, Q. and CHAN, C. (1993). The gradient projection method for the training of hidden Markov models. *Speech Comm.*, **13**, pp. 307–313.
- JOHANSSON, R. S. (1978). Tactile sensibility in the human hand: receptive field characteristics of mechanoreceptive units in the glabrous skin area. *J. Physiology*, **281**, pp. 101–123.
- MCCARRAGHER, B. J. (1996). Task primitives for the discrete event modeling and control of 6-DOF assembly tasks. *IEEE Trans. Robotics and Automation*, **12**(2), pp. 280–289.
- MCCARRAGHER, B. J. and ASADA, H. (1995a). The discrete event control of robotic assembly tasks. *ASME J. Dyn. Sys. Meas. Control*, **117**(3), pp. 384–393.
- MCCARRAGHER, B. J. and ASADA, H. (1995b). The discrete event modelling and trajectory planning of robotic assembly tasks. *ASME J. Dyn. Sys. Meas. Control*, **117**(3), pp. 394–400.
- MCCARRAGHER, B. J. and ASADA, H. (1993). Qualitative template matching using dynamic process models for state transition recognition of robotic assembly. *ASME J. Dyn. Sys. Meas. Control*, **115**(2A), pp. 261–269.
- PICONE, J. (1990). Continuous speech recognition using hidden Markov models. *IEEE ASSP Magazine*, **7**(3), pp. 26–41.
- RABINER, L. R. and JUANG, B. H. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, **3**(1), pp. 4–16.
- RABINER, L. R. *et al.* (1985a). Recognition of isolated digits using hidden Markov models with continuous mixture densities. *AT&T Technical J.*, **64**(6), pp. 1211–1234.
- RABINER, L. R. *et al.* (1985b). Some properties of continuous hidden Markov model representations. *AT&T Technical J.*, **64**(6), pp. 1251–1269.
- TRINKLE, J. C. and ZENG, D. C. (1995). Prediction of the quasistatic planar motion of a contacted rigid body. *IEEE Trans. Robotics and Automation*, **11**(2), pp. 229–246.
- VLONTZOS, J. A. and KUNG, S. Y. (1992). Hidden Markov models for character recognition. *IEEE Trans. Image Processing*, **1**(4), pp. 539–543.
- YANG, J., XU, Y. and CHEN, C. S. (1994). Hidden Markov model approach to skill learning and its application to telerobotics. *IEEE Trans. Robotics and Automation*, **10**(5), pp. 621–631.
- XU, Y. and YANG, J. (1995). (Nagoya, Japan, May 19–26). Towards Human–Robot Coordination: Skill Modeling and Transferring via Hidden Markov Model. *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, Japan, pp. 1906–1911.
- ZHU, Q. (1996). Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Trans. Robotics and Automation*, **7**(3), pp. 390–397.