Learning of Temporal Sequences for Motor Control of a Robot System in Complex Manipulation Tasks

Karlheinz Hohm, Yanming Liu, and Hans G. Boetselaars

Darmstadt University of Technology
Institute of Control Engineering, Control Systems Theory & Robotics Lab.
Landgraf-Georg-Strasse 4, D-64283 Darmstadt, Germany
E-Mail: hohm@rt.e-technik.tu-darmstadt.de
URL: http://www.rt.e-technik.tu-darmstadt.de
Phone: ++49-6151-167402, Fax: ++49-6151-293604

Abstract. Learning of temporal sequences is a topic of research in such different areas as speech and other temporal pattern recognition as well as motor control. On the other hand there is very much research effort in the area of rule based control, mainly fuzzy control, to achieve flexible systems with high autonomy. This paper concentrates on the question how a transfer of information from rule based behavior towards fast stepwise feed forward controlled motion trajectories can be achieved, so that an artificial system is able to learn from itself by generating motion sequences out of rule based behavior on its own. This allows to learn suitable trajectories, which lead to solutions of problems, where they were originally unknown, but where rules exist, e.g. stored in fuzzy controllers, which specify a desired behavior of the system in that context.

1 Introduction

When examining the nervous system, responsible for the human motor control, one can recognize a hierarchical structure, physiologically as well as functionally. In the highest level the motor cortex is the center of motor control. It connects the output of cortical and subcortical centers and is the initial point for motor commands going to the brain stem as well as directly to the spinal cord. The translation of motion planning, based on visual feedback or heuristic rules for at least partially known problems, into action is done directly as well as including the basal ganglia and the cerebellum who have to coordinate and control the motor activities. The physically and functionally subsequent level is the brain stem. There the sensorial information from the skin, the muscles and the joints as well as from position of the head gather. It is a kind of intermediate station of most sensorial signals going up to the brain as for the control signals going down. Finally in the lowest level the spinal cord provides simple motion sequences commonly summarized as reflex actions. They are triggered either by sensory neurons or by signals from the higher level control of the central nervous system.

This work concentrates on the intermediate level of the motor system. The motor cortex is responsible for planning and organization of complex motions performed consciously, while the cerebellum is the motor controller for timing and precision and therefore also responsible for motions performed unconsciously but started intentionally and thus are no reflex actions
(for a detailed model of the human motor system see e.g. [1]). A typical example for this is a tennis player learning the temporal sequence of a forehand drive. It appears that at first during a training phase the motion is a conscious motion controlled by the cerebellum but supervised by the motor cortex all the time. Later an initial signal given by the motor cortex is enough to recall the whole sequence (with slight variations to hit the ball even if it approaches on different trajectories) while the motion itself is performed unconsciously only controlled by lower levels. However, conscious interactions by the motor cortex are still possible but due to the time needed for e.g. visual information processing they are restricted by a considerable time delay. From a control theory point of view this corresponds to a stepwise feed-forward control where in between a period of time needed for feed-back control a couple of feed-forward control actions are performed.

Learning of temporal sequences is a topic of research in such different areas as speech and other temporal pattern recognition as well as motor control (for a survey see e.g. [7]). This paper concentrates on the question how a transfer of information from the conscious to the unconscious level can be imitated so that an artificial system is able to learn from itself by generating motion sequences out of conscious, i.e. rule based, behavior. This would allow to learn fast motion sequences for problems, where the trajectories leading to a solution are originally unknown, but where rules exist, e.g. stored in fuzzy controllers, which specify a appropriate behavior of the system in that context. The application we consider for these kind of motion sequences is in the field of non destructive, flexible robot assisted disassembly of electronic devices for re-use of valuable modules.

The next section discusses the disassembly problem and in section three our approach for autonomous learning of temporal sequences for motor control of a robot system is described. Section four shows experimental results and the paper ends with conclusions and a brief outlook on future work in section five.

2 The Disassembly Problem

With regard to the already very high but still increasing number of end-of-life products especially in the area of electronic devices, non-destructive disassembly and re-use of valuable modules is one possibility to face this problem in an environmentally sensible way. For economical reasons, due to the resulting quantity of products to be recycled as well as for safe working conditions for human workers, increasing automation of disassembly seems to be indispensable.

Therefore we already use a system based on fuzzy controllers, which represents a conscious level in the above mentioned hierarchy using heuristic rules (for details see e.g. [8] or [9]). The disadvantage is, that the system is not able to learn ”of its experiences” by repeating experiments. Integration of an architecture for learning parameterized motion sequences should allow the system to learn a suitable trajectory until it is able to perform tasks by using stepwise feed-forward control and thus decrease the time or the costs needed.

In this paper, we discuss the robotic disassembly of the objective block including the lens system out of a camcorder. Figure 1 shows the camcorder and the initial state at the beginning of the disassembly operations in a simplified schematic representation as ”block-in-box problem”. The objective module is located inside the casing only allowing a disassembly motion upwards, which, however, is constrained by an undercut of the casing.

The problem can be discussed as a two dimensional one, happening in a plane, e.g. the y-z-plane, of the three dimensional space. Thus for the disassembly of the lens module the external forces in y- and z-direction, and the external torque around x-direction with respect to the actual task coordinate frame $T$ are the information used by the fuzzy controllers for on-
line generation of evading movements by reacting accordingly. The only external directive is the global disassembly direction in which the robot system is supposed to disassemble the module until a final height is reached (for details about this control architecture see [9]). In figure 2 a typical sequence of some situations during the disassembly of the lens module is illustrated, while figure 3 shows examples of measurement data of the resulting trajectories generated online by the fuzzy controller. For each example the y- and z-position and the orientation around the x-axis at a certain z-position with respect to the task coordinate frame $T^{(0)}$ at starting point of the disassembly operation, which is identical to $T$ in figure 1, is illustrated. During the manipulation the task coordinate frame $T$ is moving with respect to the stationary base coordinate frame $B$. However, to be independent from absolute positions within the robots workspace the motion is considered with respect to the task coordinate frame $T^{(0)}$ at start of the manipulation.

The questions discussed in this context concern the generation of suitable motion trajectories by learning from the trajectories and the "behavior" generated online by the fuzzy controllers, which within the hierarchy described above means the transition of information from the conscious to the unconscious control level.

Figure 2: A typical sequence of some situations during the disassembly of the lens module is illustrated in a schematic representation.
3 Autonomous Learning of Temporal Sequences for Motor Control of a Robot System

The system we developed for learning temporal sequences is based on discrete Hidden Markov Models (HMMs). A HMM is a doubly stochastic process. It consists of an underlying stochastic process that is not directly observable (i.e., hidden). However, it can indirectly be observed through another set of stochastic processes which produce the sequence of observed symbols. HMMs have already been successfully applied to problems as speech recognition (e.g., [6]), force analysis (e.g., [4]), or mobile robot path planning (e.g., [11]), where they had to learn the structured knowledge from a set of observed sequences of symbols. Recently it has been used for skill learning problems by modeling human action as an HMM with its parameters learned from training data (e.g., [10]).

In our approach we use the discrete HMM to realize the above mentioned transition from the heuristic level to the level of motion sequences so that the system is now able to "learn from its experience" and apply this architecture to the problem of robot assisted disassembly. I.e. that the system memorizes the informations about successful motion trajectories, performed by using the fuzzy rules to solve a given problem, and uses this data for generating motion sequences.

3.1 Generating a Suitable Discrete Hidden Markov Model

A discrete HMM consists of finite number of states connected by transitions and a finite alphabet of symbols, which can be observed while the HMM is in a certain state. For each state exists on the one hand a discrete condition transition probability distribution, describing the probability for staying in the same state or changing from the actual state to any other state, and on the other hand a discrete condition output probability distribution, describing the probability of emitting a certain output symbol at the current state. Therefore, the number of states and the number of discrete symbols as well as the states and the symbols itself have to be chosen to specify a discrete HMM. Furthermore, then the discrete condition transition probability
matrix, and the discrete condition output probability matrix must be determined.

To obtain a finite number of states as well as a finite alphabet of observable symbols, vector quantization is used. In order to enable the system to determine the number as well as the location of the states and the symbols in their particular input space itself, self-organizing neural networks of the Growing Neural Gas (GNG) type were used for that (for details about GNG networks see e.g. [2], [3]). The position of the neurons in the particular input space represents the resulting codebook for the states respectively the symbols. Thus also the generation of the states and of the symbols as well as the determination of their specific number is done autonomously by the system and is automatically adapted to the training data.

Different experiments proofed that using the cartesian space is a good choice for the input space to determine the states of the HMM. However, to keep the system more flexible, this should not be done with respect to the base coordinate frame $B$ but to the task coordinate frame $T^{(0)}$ at start of the disassembly operation. This allows to use the same motion sequence at every point within the workspace of the robot system without changes. During the experimental evaluation of this approach we further found that as an additional dimension for the input space the time since the start of the manipulation should be used to enhance the results. This ensures that during slow as well as during fast manipulation phases a sufficient number of states can be defined to get a good approximation of the motion trajectory. For the observable symbols we use the velocity commands, so the three dimensional input space for the vector quantization of the symbols consists of the translational velocity in $y$- and $z$-direction and the rotational velocity around the $x$-axis of the task coordinate frame $T$. The training of the HMM is done with the Baum-Welch algorithm, where the training data for the states and the observable symbols consists of the originally stored data classified by their particular code book of the vector quantization.

Once the HMM has been trained, the robot disassembly operation is represented by the transition probability matrix and the output probability matrix. Then the learned HMM can be used to control the robot during the disassembly process according to the sequence with the maximum probability. I.e., the transition probability matrix of the learned HMM is used to obtain the best sequence of states as the desired sequence of states for the robot disassembly, and the discrete output probability matrix of the learned HMM can be used to obtain the best sequence of velocities as the desired sequence of velocity commands for motor control of the robot system for the disassembly process.

### 3.2 Generating the Final Motion Command Sequence

The problem now is, that it is not possible for the robot system to perform a whole disassembly operation within a few sampling steps, i.e. within the number of states in the optimal sequence of the HMM, which would be the number of commands to the robot. Therefore, in a first step the velocity command sequence was extended with respect to the probability of the underlying states of the HMM to stay in themself, to have a minimum falsification of the sequence. Besides the velocity command sequence there is also the sequence of states of the HMM, which offers also a possibility to command the robot with a position sequence, because the states are a direct model of the optimal motion trajectory. Here is also an extension in the number of commands necessary, which can easily be done be linear interpolation between the states. In both cases the final command sequence is stored as a prototype for the given problem in an interpolating neural net.

---

1 This is also an advantage of using the cartesian space in contrast to using the joint space, which basically would also be a possibility.

2 It is to note that position here means the cartesian position and orientation.
Figure 4: The architecture for recalling the final motion command sequence is illustrated. The system receives its input e.g. by a supervisory control level.

For storing and recalling the final command sequences we use an approach which is particularly suitable for motor control e.g. of a robot due to the fact that it does not only reproduce temporal sequences in exactly the way they where learned but it is able to generate slightly modified sequences according to given parameters, too. This is important to keep the system highly flexible. As illustrated in figure 4, it consists of two main elements, a short term memory to preserve context information within a sequence, which in our architecture is a virtual time base along which the sequence can be stored and recalled, and a following long term memory, where a prototype of the sequence itself is stored. The input consists of an increment $\Delta t_v$ to the virtual time base to get the next sequence element and a set of parameters to realize suitable modifications, e.g. geometric distortions, necessary to adapt the prototype motion sequence to the actual situation. Thus the system is able to learn and recall motion sequences also with slight variations (for details see [5]).

4 Experimental Results

For the experimental evaluation a set of twenty successful disassembly operations, which the system performed by using the fuzzy controllers, was used to generate the motion sequence. The four dimensional space for defining the states by the GNG networks consisted of the time elapsed since start of the manipulation and of the translational position in y- and z-direction and the rotational orientation around the x-axis with respect to the task coordinate frame $T^{(0)}$. The three dimensional space for defining the observable symbols consisted of the translational velocity in y- and z-direction and the rotational velocity around the x-axis of the actual task coordinate frame $T$ at each sampling unit. Out of this data the GNG networks generated about fifty states and also about fifty observable symbols. The optimal state sequence found after training the HMM consisted of about thirty states.

The experiments were performed using a six axes puma type robot equipped with a force/torque sensor at the wrist, providing the information for the motions with the fuzzy controller, and a parallel jaw gripper for grasping the object with a desired grasp force. The system works at a sampling time of about 50msec. All experiments were first tested on a wooden model imitating the situation of the lens modul inside the camcorder as block-in-box problem. In conjunction with these evaluations the resulting trajectory based on the motion sequence of the HMM as well as the arising contact forces with the other objects, i.e. here the wooden box around, were considered to avoid damage during the disassembly. A stop of the motion
was allowed by a supervisory control level which simply provides a threshold operation. It is interacting in critical situations, i.e. contact forces and/or torques are beyond a threshold, like the conscious control level in human motor control.

The experiments with the velocity command sequence were quite successful, although in some parts large contact forces were measured which means that there the modeled velocity course is not exact enough for the problem. Doing the experiments with the position command sequence leads to very good results. Figure 5 shows examples of measurement data of the resulting trajectories performed by the system during experiments with the real camcorder. Again the y- and z-position and the orientation around the x-axis at a certain z-position with respect to the task coordinate frame $T^{(0)}$ at starting point of the disassembly operation is illustrated. One can see the rather smooth course of the motion in contrast to the original trajectories found by the fuzzy controller as illustrated for some examples in figure 3. The robot now moves along a trajectory, which is very good adapted to the geometric constraints by the casing of the camcorder, providing a continuous rotation of the object to avoid collisions, and it does not need to "look for the right way and orientation" anymore.

Considering the time needed for the whole disassembly operation, even without any further optimization the system is about 10% faster by using the position command sequence than using the fuzzy rules due to the smoother trajectory.

5 Conclusions

We presented a method which enables an artificial system to autonomously transfer information from rule based behavior towards stepwise feed forward controlled motion trajectories, so that it is able to learn from itself by generating motor command sequences out of rule based behavior on its own. This allows to learn motion sequences for problems, where the trajectories leading to a solution are originally unknown, but where rules exist, e.g. stored in fuzzy controllers, which specify a appropriate behavior of the system in that context. The method is based on discrete HMM to generate an optimal command sequence for the robot system. For training the HMM the system memorizes motion data of successful trajectories for a given problem. This data is preprocessed by vector quantization with GNG networks which allow to define the number as well as the states and the observable symbols itself. The resulting command sequence is then stored as a prototyp solution for this problem in an interpolating neural net. The whole process of information transition is done autonomously by the system, without any external interactions.
The experiments with a real robot system proofed the effectivity of the method using position command sequences, but also velocity command sequences were successfully tested. Already without any further optimization the necessary manipulation time decreased. Besides more complex mechanisms for speed optimization, already by a simple variation of the increment given to the virtual time base during the recall phase of the motion trajectory an increasing of the motion speed is possible.

Of course the learned trajectory is only optimal for exactly the learned problem. To keep the flexibility of the system the additional input parameters of the recall subsystem for the motion sequences, shown in figure 4, can be used to adapt the motion sequence e.g. by geometric distortion. If the geometry of the problem is known in advance, this can be done before starting the manipulation. However, in the context of disassembly problems this is not very likely. Instead, the supervising level, which checks the arising forces and torques, can be used to adapt those parameters online according to the main contact directions. On the other hand it is possible to use the sequence information as an advanced input for the fuzzy controller. So no longer only a global disassembly direction is given as a priori information, but a servo control loop for the motion trajectory is possible with additional evading movements superposed by the fuzzy controller, if the contact forces/torques are too large, because the actual problem differs too widely from the problem during learning of the prototype sequence. In this context the less exact modelling of the velocity command sequence is a minor problem, so that these also could be used here. The evaluation of these interacting control loops to keep most of the flexibility of the system as well as the optimization of the learned prototype sequences is part of our current work and will further be examined.

References