On-line Learning of B-Spline Fuzzy Controller To Acquire Sensor-Based Assembly Skills

Jianwei Zhang, Yorck von Collani and Alois Knoll
Technical Computer Science, Faculty of Technology,
University of Bielefeld, 33501 Bielefeld, Germany

Abstract

We present an on-line learning approach for developing sensor-based controllers and show how it can make the programming of assembly tasks easier. We suggest to combine the qualitative modelling of human expert skills and the self-tuning of control parameters so that a controller for a complex assembly task can be efficiently developed. It is then discussed how to construct a fuzzy controller with B-splines and why rapid convergence of its learning can be achieved. To apply the concept in a screwing operation, we propose several gradual steps of active on-line learning. Experiments were carried out with two independently controlled robot arms. Although general-purpose jaw-grippers are used and diverse uncertainties exist, the "elevator control" of a toy aircraft can be robustly built.

1 Introduction

Our research concentrates on using two robot arms to assemble complex aggregates. Among assembly operations, insertion and screwing are very important for investigating sensor-based control methods, [7]. In industrial applications, a screwing task is usually performed by position-based approaches using precise fixturing and special screwing tools with passive-compliance devices. However, in order to enhance the flexibility of a robotic system, approaches are necessary which are able to control a general-purpose hand/gripper based on sensor inputs. Only with sensors can the diverse uncertainties occuring during different screwing operations be detected and correctly handled.

Screwing is a robot skill to be acquired, either conventionally by manual programming or by self-learning of the robot. However, solutions to screwing are not as thoroughly discussed as the peg-in-hole problem. In [2] it was briefly discussed how to develop programs for screwing operations by using force feedback. [1] presented a telerobotic system for performing such an operation. Screwing with a multifingered hand was reported in [6]. To the best of our knowledge no experiments on screwing with two robot arms have been reported.

Robot learning aims at generating robot software in an evolutionary way. Off-line supervised learning must utilise data from human demonstration, and it cannot be guaranteed at all that the optimal controller is trained even if the human instructor demonstrated his best skills. Recently, serveral unsupervised learning methods have been presented. [3] discussed the training of a fuzzy-neural controller for position/force control through backpropagation and gave some simulation results. To train a controller for contour tracking based on force feedback, [5] used a neural network method.

Fuzzy logic is a suitable mechanism to model and summarise human knowledge for sensor-based control, [9]. In this paper, we introduce an on-line unsupervised learning approach based on fuzzy controllers. We try to tell the robot approximately "how-to", then let it practise in the real world and learn the optimal values of a number of control parameters. We also show how the parameters to be adjusted can be reduced to the fewest possible by constructing a fuzzy controller with B-splines. We show experimental results of screwing operations with two robot manipulators.

2 Screwing Control Problem

2.1 Experimental set-up

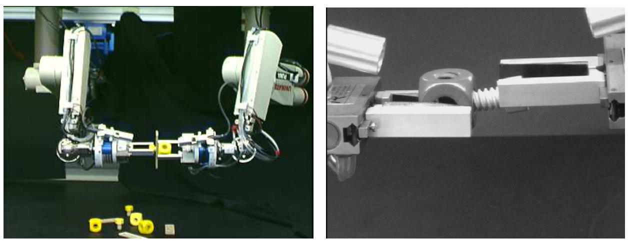
The problem of the screwing of a bolt into a nut originates from our collaborative project which aims at assembly of aggregates with wooden toy construction sets, [4]. The "elevator control" of a toy aircraft was selected as one aggregate to be built, Fig. 1.

(a) Bolt, bar and block-nut

(b) "elevator controls"

Figure 1: Parts and aggregates of the "Baufix" construction set.

The assembly agent in our experiment is a two-arm robotic system, Fig. 2(a). Two manipulators of type PUMA 260 are installed overhead. On the wrist of each manipulator, a parallel pneumatic jaw-gripper with integrated force/torque sensor and a hand-camera are mounted. The X-, Y-, and Z-axis in the coordinate system of the force/torque sensor coincide with the N-(normal), O-(orientation) and A-(approach) direction of the arm tool coordinate system, respectively.



(a) Two cooperating manipulators

(b) An inconvenient starting situation

Figure 2: The experimental set-up for fixtureless assembly.

We consider general screwing without using any fixture devices. Screwing with the two arms assumes the following starting situation: under the guidance of an overhead-camera and instructed by a command in natural language, one arm/gripper moves over a bolt, positions finely with the help of the hand-camera and grasps the bolt (this gripper is called "bolt-hand" in the following); the other arm/gripper similarly grasps a block-nut (that gripper is called "nut-hand"). After that, both hands move to a "rendezvous" point in their common working area, and they are ready for screwing[1].

[1] To emphasise the screwing problem, the "insertion of a bolt into the nut", which is actually a subproblem of screwing, is omitted here.

2.2 Motion sequence for screwing

Our experiments have shown that a set of sensor-based subtasks must be sequentially performed to complete a screwing operation:

1. **Find the contact** during approach of the bolt towards the block.
2. **Put the bolt in the nut** through either spiral search or visually-guided fine motion.
3. **Find the notch point** to fit the bolt-thread into the bolt-nut[2]. This can be realised by rotating the bolt-hand counterclockwise and is characterised by an impulse in the force-reading.
4. **Screw-in** by exerting a forward force to maintain the correct contact of the bolt with the block-nut while rotating clockwise. To finish a long-thread screwing using grippers with the limited rotation range, the bolt-hand has to open the gripper if a joint limit has reached, then rotate backward and **regrasp**.
5. **Terminate** when one hand detects a torque impulse in the approach direction.

2.3 Uncertainties

For a general-purpose arm/gripper system, the following three types of uncertainties must be taken into account:

Grasping precision. Although we have applied a hand-camera in a "self-viewing" configuration, which significantly improved the grasping precision in comparison with the open-loop positioning, regrasping still engenders deviation of the bolt from the rotation axis of the gripper.

Slippage of the part in the hand. Due to the effect of the resulting forces, the bolt grasped by a jaw gripper may easily slip during the screwing process.

Position limit and vibration. Even if the bolt and the nut are fitted quite well at the beginning, the intolerable forces can still be generated due to the limited positioning precision and the unavoidable vibrations during the hand rotation.

The uncertainties in a screwing process cause the following two concrete problems:

1. The bolt is not centrically grasped, i.e. the rotating axis of the bolt does not coincide with that of the gripper.

[2] Analogous to the *alignment* in the peg-in-hole problem.

2. The bolt is obliquely grasped, Fig. 2(b).

Without using sensors, such an operation can fail under each of the uncertainties discussed above. Therefore, sensor-based compensation motions become necessary. The resulting forces in case 1 and 2 in the normal and orientation directions should be minimised and stable. Additionally, to guarantee a successful screwing-in phase, a constant force in the approach direction should be exerted.

3 Learning of the B-Spline Fuzzy Controller

3.1 Combining design and learning

We adopt a two-phase scheme for developing a sensor-based motion controller. These two phases correspond to the qualitative design and the quantitative fine-tuning. Fuzzy logic is selected as the basic structure for modelling the qualitative rules extracted from expert knowledge. In the quantitative part of an assembly skill, the decisive parameters are fine-tuned.

In our B-spline fuzzy controller, values of the *control vertices* in the "THEN" part are the main parameters to adapt. To define the linguistic terms of the "IF" parts, it is unnecessary to specify each fuzzy set manually. A user simply selects the *granularity* for partitioning the universe of the input variable. The "optimal" parameters in these "IF-THEN" rules, especially the control vertices, cannot be determined only through design. They must be optimised in a natural "learning-by-doing" procedure through real operations with the robot.

3.2 Why B-spline fuzzy controllers?

The above idea can be effectively realised using the B-spline fuzzy controller proposed in our earlier work [8]. This type of controller is distinguished from the conventional fuzzy controller by the following features:

- B-spline basis functions are employed for specifying the linguistic terms (labels) of the input variables. By choosing the order k of the basis functions ($k \geq 2$), the output is C^{k-2}-continuous. However, too high an order will result in high computational burden and additional rules. In practice, orders 2, 3, 4 are suitable for modelling membership functions.
- The linguistic terms of all the IF-parts are generated automatically using the recursive definition of the basis functions. The human users only need to specify how fine each input variable should be partitioned.
- Each controller output is defined by a set of fuzzy-singletons, called control vertices in the B-spline terminology. The number of the control vertices is equal to the number of the rules. The values of the control vertices can be initialised approximately if *a priori* knowledge is available or simply as zero if this is not the case. The optimal values can be iteratively found through learning.
- If "product" is used as the fuzzy conjunction and "centroid" as the defuzzification method, computation of such a fuzzy controller is equivalent to that of a general B-spline hypersurface. Learning of such a controller is transformed to adjusting the control vertices for shaping an "ideal" curve or (hyper)surface.

3.3 The learning aspect

One problem with the learning of conventional fuzzy controllers is that too many parameters must be adjusted. This results in difficulties for the convergence of the learning algorithms and the learning speed. With B-spline fuzzy controllers, modification of control vertices effectively causes the changes of the control surface. Learning of the controller is then reduced to the learning of the positions of the fuzzy-singletons on the axis of the output variables[3]. Tuning a controller is equivalent to shaping the control surface. In CAD applications, the criterion for defining the "ideal" surface can be the visual appearance or some measures like length, curvature, energy, etc. For control applications, they should optimise certain cost functions, e.g. the action-value in the Q-learning paradigm.

We showed an important feature of the B-spline fuzzy controller: for supervised learning, if the squared error is selected as the action-value, its partial differentiation with respect to each control vertex is a convex function. Therefore, a learning algorithm employing simple gradient descent method can converge rapidly. For unsupervised learning, the learning process with the gradient descent will also show a stable asymptotic behaviour if the system possesses the following feature: given two controller outputs with different signs, the system changes away from the current state

[3] The changes of the knots of the B-spline basis function on the input will also influence the control surface. We have developed a self-adaptation module to find the best positions for them, based on the extreme points of the output.

in two directions ("better" or "worse" as measured by the action-value).

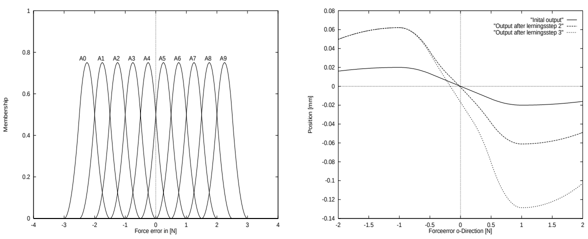
4 Implementation Issues

4.1 Initialising the controller

4.1.1 Input

The input information is provided by the force feedback during the motion. In the screwing operation, instead of absolute forces, the deviations of the real forces from the desired ones are used as the input variables, which can be restricted to $\pm 2N$ for our application. Greater force deviations occur when the bolt is placed on the thread-neck.

Firstly, the linguistic terms and their definition intervals are specified. At each of both ends of the input range $[-2N, +2N]$, two *virtual linguistic terms* are added to maintain the smooth controllability at the end of the interval [8]. If B-spline basis functions of order three are used, the generated linguistic terms can be seen in Fig. 3(a), where A_0 and A_9 are virtual linguistic terms, A_1 to A_8 are e.g. $HighNegForceError$, $LowNegForceError$, ..., etc.



(a) The input variables within the effective range $\pm 2N$

(b) The control curves during learning

Figure 3: Membership functions and the control curves.

Force deviations in all three directions are noted as F_{err} in *A*-, *O*- and *N*-direction. They are covered with these linguistic terms similarly.

4.1.2 Output

Linguistic terms of the output variables are defined in the following way:

```
def singleton: MoveVeryHighNeg 0.3
```

These fuzzy-singletons are the control vertices and noted as y_i in the following. They can be specified approximately if data for the control process are available, or initialised as zero if there is no *a priori* knowledge.

4.1.3 Rules

An example of a rule is:

"IF the deviation from the desired force is very high THEN the arm should move back in a big stretch"

In this way, the heuristic rules for acting in the real situations are extracted. They are called "core rules". To deal with the input combinations of the *virtual linguistic terms*, *marginal rules* have to be generated. That can be done by just copying the output values from the directly neighbouring input combinations. For details see [8].

4.2 Learning functions

There is no direct information on how the motion should be compensated to realise the desired forces in each control cycle. However, we know in which direction the compensation motion should go if the force errors are computed. In a finely partitioned local area, the force error is approximately proportional to the step of the last compensation motion if the time delay is omitted. Let

$F_{\text{err}}(z)$	the force error in control cycle z
η	the learning rate, transforming the force error to the compensation motion, with the unit [mm/N].

Then each control vertex y_i (i is the index of rules) can be modified similarly to gradient descent in supervised learning:

$$\delta_{y_i} = -\eta \cdot (F_{\text{err}})(z) \cdot N_i(F_{\text{err}}(z-1)) \tag{1}$$

where $z > 0$ and N_i is the i-th basis function of a selected order. Representing the "firing strength" of the i-th rule, $N_i(F_{\text{err}}(z-1))$ serves as the function to backpropagate the force error to the modification of y_i.

This algorithm works only if there is no time delay. Otherwise, the oscillation problem should be taken into account, which results from the time delay due to sensor data processing as well as limits of the control cycle. Through observation, an oscillation can be detected if $F_{\text{err}}(z) \cdot F_{\text{err}}(z-1) < 0$. The modification of the control vertices for this case has to be designed specially by summing up the force errors of the last several control cycles and using a bigger learning rate $\eta_O, \eta < \eta_O$:

$$\delta_{y_i} = \begin{cases} \eta \cdot F_{\mathrm{err}}(z) \cdot N_i(F_{\mathrm{err}}(z-1)) & \text{if } \alpha \geq 0 \\ \sum_{\nu=1}^{3} \eta_O \cdot F_{\mathrm{err}}(z) \cdot N_i(F_{\mathrm{err}}(z-\nu)) & \text{if } \alpha < 0 \end{cases} \quad (2)$$

with $\alpha = F_{\mathrm{err}}(z) \cdot F_{\mathrm{err}}(z-1)$

By using (2) instead of (1) for the system with time delays, the learning algorithm shows an asymptotic behaviour.

4.3 Learning step by step

Naturally, the whole screwing skill should be learned starting with easy motions and proceeding to more complicated ones, which are summarised as follows:

1. **Motions in the approach-direction.**

   The bolt-hand attempts to approach the nut and exerts a force along the Z-axis. This step is repeated many times until it is demonstrated that the screw can stay in the nut-thread by itself. The controller has learned how to apply a certain force in the approach direction by compensation motions. The condition of successful learning is that the bolt axis should not depart from the rotation axis too much, since cross forces cannot yet be compensated in this first learning step.

2. **Motions in the normal- and orientation-directions.**

   After the screw is sited in the nut-thread, the controller can learn analogously to step 1 along the X-, Y-axis, respectively.

3. **Screwing motions.**

   After step 1 and 2, the bolt-hand performs a slight rotation (about $10°$ in our experiment) repeatedly until the resulting forces along the X- and Y-axis are compensated to zero. Through these intentional abnormal motions, situations which can occur in the screwing operation are generated.

4.4 Experimental results

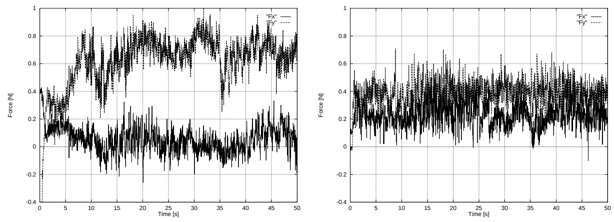
The force/torque sensor (type JR3-67M25A) was read at 8 KHz, and the force/torque data were computed at a rate of 125 Hz to filter out noise. The sampling time for joint position measurement of the two cooperating manipulators was 28 ms for the current experiment, which is on the order of the usual control cycle of most industrial robots.

This task was performed many times to judge its robustness. The screwing process can be successfully realised to build the "elevator control", Fig. 1(b). Experiments show that the controller can learn how to compensate the resulting forces after the above three steps are executed only once.

4.4.1 The control curve

We give an example of screwing with large positioning deviation of the bolt. Fig. 3(b) illustrates the control curves to compensate the force in Y-axis i) by approximate initialisation; ii) after learning step 2; and iii) after learning step 3.

The resulting forces with the controller before and after learning are shown in Fig. 4(a) and 5(b).



(a) Before learning (b) After learning step 3

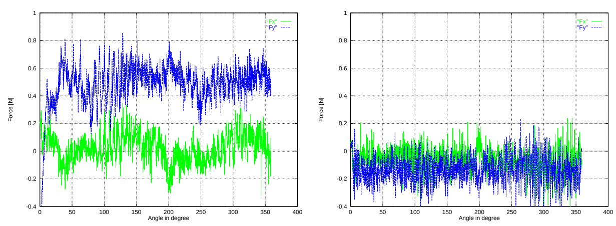
Figure 4: Forces along X- and Y-axis generated during screwing with large positioning deviations of the bolt from the rotation axis.

4.4.2 Screwing with both hands

We have successfully applied the learning procedure to control the two robot arms simultaneously. Both hands rotate during the "screwing-in" phase. Each arm controller can learn its own control curve after one complete screwing. Apparently, rotation of one gripper creates disturbances effecting the other gripper. Even in this case, the forces to be controlled can still be kept within the tolerance (Fig. 4(b)).

4.4.3 Comparison with PID

For comparison, a PID controller was also implemented. The adjustment of the PID coefficients needs experience, is tedious work with "trial-and-error" and needs a much longer time to achieve similar performance than our learning approach. Fig. 5 shows the comparative result of the learning fuzzy controller and a tuned PID controller.



(a) Tuned PID

(b) B-spline fuzzy controller after learning

Figure 5: Maintaining constant small values of F_X and F_Y - the comparison of PID and the learning B-spline fuzzy controller, both with large positioning deviations of the bolt from the rotation axis.

5 Conclusions

We developed a novel learning approach for realising screwing operations with a B-spline fuzzy controller and implemented the approach with a two-arm robotic system. The robot controller learns actively and on-line, instead of being trained passively and off-line. The learning process converges to a reasonable result after the three steps outlined in section 4.3. The advantages of our learning-controller approach are:

- The combined design/learning methodology. What must be done in the design phase is quite simple: select input variables, the granulation of partitioning the input space, and some approximate output values if they are available. The key work for learning is to find a suitable learning function and let the robot generate enough data for covering all possible situations.
- Rapid convergence of the learning process. This property benefits from the appropriately selected cost or error function.
- Smooth output. If a B-spline basis function of order 3 is used, the output is twice differentiable (Fig. 3(b)).

In this approach, the human-to-robot skill transfer is finally represented in form of "IF-THEN" rules with optimised parameters. No complex programming and control expertise are needed. Fine-tuning of the main controller parameters is done on-line automatically. No subjective evaluation is needed. The learning process is fast, the usual time-consuming "trial-and-error" is no longer necessary. The methodology of combining design and learning can be applied to acquire other assembly skills, such as peg-in-hole, handling flexible objects, etc. We have shown that this approach is very promising for realising efficient robot assembly skills based on sensorimotor coordination.

Ackowledgement

This research is partly funded by the DFG under grant SFB 360-D4.

References

[1] O. Fuentes and C. Nelson. The virtual tool approach to dextrous telemanipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1700–1705, 1996.

[2] H. Inoue. Force feedback in precise assembly tasks. In P. H. Winston and R. H. Brown, editors, *Artificial Intelligence: An MIT Perspective*, volume 2, pages 219–241, 1979.

[3] K. Kiguchi and T. Fukuda. Fuzzy neural friction compensation method of robot manipulation during position/force control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 372–377, 1996.

[4] A. Knoll, B. Hildebrandt, and J. Zhang. Instructing cooperating assembly robots through situated dialogues in natural language. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1997.

[5] F. Lange and G. Hirzinger. Learning force control with position controlled robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2282–2288, 1996.

[6] P. Michelman and P. Allen. Forming complex dextrous manipulations form task primitives. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3383–3388, 1994.

[7] K. Selke and P. Pugh. Sensor-guided generic assembly. In *Proceedings of the 6th International Conference on Robot Vision and Sensory Controls, Paris*, pages 11–19, 1986.

[8] J. Zhang and A. Knoll. Constructing fuzzy controllers with B-spline models. In *IEEE International Conference on Fuzzy Systems*, 1996.

[9] J. Zhang, F. Wille, and A. Knoll. Modular design of fuzzy controller integrating deliberative and reactive strategies. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1996.