BEHAVIOURAL CLONING: PHENOMENA, RESULTS AND PROBLEMS

Ivan Bratko[1,2], Tanja Urbančič[1], Claude Sammut[3]

[1] *Jožef Stefan Institute, Ljubljana, Slovenia*
[2] *Faculty of Electrical Eng. and Computer Sc., University of Ljubljana*
[3] *University of New South Wales, Sydney, Australia*

Abstract: Controlling complex dynamic systems requires skills that operators often cannot completely describe but can demonstrate. Behavioural cloning is the process of reconstructing a skill from an operator's behavioural traces by means of Machine Learning techniques. In this paper we analyse various phenomena and problems observed in experiments in behavioural cloning in several domains: piloting, driving a container crane, production scheduling and pole-balancing. The analysis includes the "clean-up" effect and the time delay between state and action. We derive from this analysis some elements of an emerging methodology for behavioural cloning.

Keywords: Machine learning, control system synthesis, human-centered design

1. INTRODUCTION

Controlling a complex dynamic system, such as an aircraft or a crane, requires a skilled operator who has acquired the skill through experience. The skill is subcognitive and the operator is usually only capable of describing it incompletely and approximately. Such descriptions can be used as basic guidelines for constructing automatic controllers but, as discussed for example in (Urbančič and Bratko, 1994a) the operator's descriptions are not operational in the sense of being directly translatable into an automatic controller.

If we are interested in designing an automatic controller based on an operator's skill, we have the following situation. An operational description of the skill is not available, but the manifestation of the skill is available as traces of the operator's actions. One idea, explored in several projects, is to use these traces as examples and extract operational descriptions of the skill by Machine Learning techniques. Extracting symbolic models of a real-time skill from traces of the operator's behaviour was termed *behavioural cloning* by Donald Michie (1993). To our knowledge, behavioural cloning has been applied in the following dynamic domains: piloting a simulated aircraft (Sammut *et al.*, 1992; Michie and Camacho, 1993), operating a crane (Urbančič and Bratko, 1994b), pole-balancing (Michie *et al.*, 1990) and production scheduling (Kibira, 1993).

Behavioural cloning is normally performed by applying standard machine learning (ML) techniques. Of course, the cloning problem needs a formulation and representation that fits these techniques. The usual ML-based approach to behavioural cloning, employed in all the domains above, has been as follows. The "behaviour trace", that is, a sequence of the states of the controlled system and the operator's control actions is viewed as a set of examples of correct control decisions. Each example consists of a pair $(State, Action)$ where $State$ is an attribute-value vector and $Action$ is a "class value" for a learning program. In such a formulation, attribute-based learning techniques can be applied to induce a functional relation

$$Action = f(State)$$

Both the attributes of *State* and the class value *Action* can be discrete or continuous. The learned function f then represents an artificial controller, or behavioural clone, which is supposed to mimic the original operator.

Sometimes a time delay is considered between *State* and *Action*. The reason is that the current *Action* is not viewed as the operator's response to the current *State*, but to some *previous* state. This delay between the state and the action is assumed to be due to the time needed for state recognition and physical manipulation of the controls by the operator. The functional relation to be learned in such a case is

$$Action(Time) = f(State(Time - Delay))$$

In some domains there are several control variables. In piloting, for example, the throttle, flaps, ailerons etc. can be manipulated simultaneously. In such domains the learning problem is partitioned into several sub-problems, each of them dealing with one control variable. These may be dependent, so the decision regarding one control variable may be used as an attribute for a decision regarding another control variable.

A way of structuring the learning problem, applied typically in piloting, is the division of the behaviour traces into phases. For example, a flight may be divided into take-off, straight-and-level flight, turn to a specified heading, etc. Separate controllers are induced from the traces for each phase of the flight. To carry out the control task, these controllers are invoked according to the particular phase of the flight plan. The phases, the plan, and the phase recognition conditions are hand crafted and not learned automatically.

In this paper we give a comparative analysis of some of the phenomena observed in behavioural cloning, taking into account the results in all the domains mentioned above. Among other aspects we discuss the "clean-up effect", time delay between state and action, robustness of induced clones and representation for inducing human-like control strategies. Another important issue is the comprehensibility of induced clones. This was addressed in (Urbančič and Bratko, 1994b).

2. PROBLEM DOMAINS

Before comparing and analysing the phenomena, successes and problems experienced in behavioural cloning, we present the main characteristics of the problem domains considered in this paper:

- *Pole balancing* (Michie *et al.*, 1990)
  The well-known problem of balancing a pole on a cart moving on a track of limited length has often been used to demonstrate new approaches in control synthesis, e.g. in (Michie and Chambers, 1968; Barto *et al.*, 1983; Anderson, 1987; Varšek *et al.*, 1993). For cloning, a "line-crossing" variant was used by Michie and coworkers (Michie *et al.*, 1990). Here, subjects are asked to make as many crossings of the mid-line of the track as possible, within the test period, without crashing.

- *Piloting* (Sammut *et al.*, 1992; Michie and Camacho, 1993)
  Two studies have been reported: piloting a Cessna (Sammut *et al.*, 1992) and piloting an F-16 (Michie and Camacho, 1993). In both cases, the task is to fly according to predefined flight plan, including take-off, climbing to a specified altitude, straight and level flight, turning and landing.

- *Operating cranes* (Urbančič and Bratko, 1994b)
  The task is to transport a container from an initial position to a target position. Performance requirements include basic safety constraints, stop-gap accuracy and as high capacity as possible. Here, capacity is measured as the total load transported within the allotted time.

- *Production line scheduling* (Kibira, 1993)
  The task is to schedule and control a serial manufacturing system. The problem is to determine an optimum or near optimum allocation of labor for a period of time on a production line at any time during a shift.

All the domains, except pole-balancing, are of interest for potential applications. On the other hand, pole balancing is useful as experimental domain because it allows clearer study of separate phenomena in cloning. Production line scheduling is atypical in that it requires control decisions at time points separated by several hours. Therefore, this domain does not have many of the characteristics common to most dynamic control problems and will be discussed in less detail than other domains.

In all the experiments, simulators of the dynamic systems were used. The flight simulator used in (Sammut *et al.*, 1992) was provided by Silicon Graphics Incorporated. In (Michie and Camacho, 1993), the authors used the ACM public-domain simulation of an F-16 combat aircraft. For pole

balancing and crane control, the simulators were developed specifically for these experiments. The crane simulator was assessed by a specialist crane designer as very realistic. It runs on an IBM compatible PC and provides real-time performance on a 33 MHz 386 or faster.

In Table 1 we compare the characteristics of the problem domains, along with the parameters of human and machine learning. The table gives some idea of the complexity of the tasks. However, some qualifying comments are required. One of informative characteristic for a domain is the time a human operator requires to master the task. Since individual differences can be very large (see e.g. (Urbančič and Bratko, 1994b), this should be treated only as a rough approximation. For pole-balancing, human learning time was around one hour. For operating the crane, approximately ten hours of training were needed. The F-16 piloting problem was more demanding than the Cessna variant. Also the meaning of event should be clarified. In the flying domains, an event corresponds to change in control action, while in the crane problem, events are actually snapshots, recorded at regular time intervals.

3. CLEAN-UP EFFECT

Michie and Camacho (1993) described the *clean-up effect* as follows. "When induction-extracted rules were installed in the computer as an 'autopilot', performance on the task was similar to that of the trained human who had generated the original behaviour trace, but more dependable ..." They continue with an explanation of this effect. "A trained human skill, ..., is obliged to execute via an error-prone sensory-motor system. Inconsistency and moments of inattention would then be stripped away by the averaging effect implicit in inductive generalisation, thus restoring to the experimenters a cleaned-up version of the original production rules."

Michie, Bain and Hayes-Michie (1990) were the first to report the clean-up effect which they observed in the pole-balancing domain. They also gave a quantitative assessment of clean-up. When a clone induced from an operator's traces is used as a *predictor* of the operator's actions, the prediction error rate often exceeds 20 %. Michie and Camacho interpret this error rate simply as indicative of the cumulative sum of human perceptual and execution errors. These errors are presumably filtered out by the induction program. (They used Quinlan's C4.5 (Quinlan, 1987). As a result, the clone's behaviour is much smoother

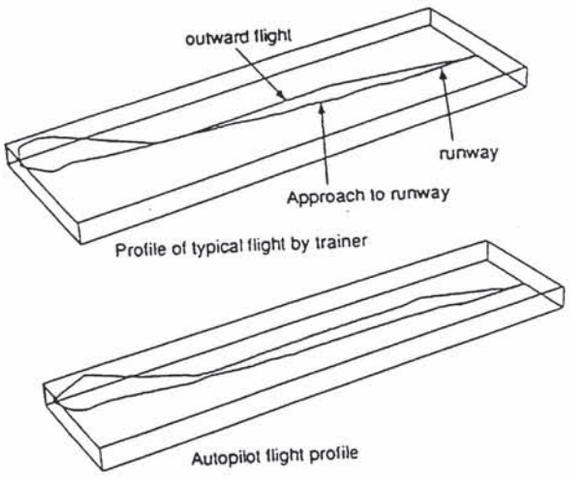

Figure 1: Human performance (upper trajectory) and the clone's execution of the same flight plan (lower trajectory).

than that of the human operator. Michie and colleagues (1990) measured this *performance* (not prediction!) clean-up in terms of ranges visited by the four system variables: position, angle, and their velocities. In general, better control results in smaller ranges. The ranges achieved by the clone were much tighter than those achieved by the human trainer. They were reduced by the clone to between 17 % and 45 % of the original human's ranges (depending on the system variable). This result is very illustrative although the simple measure of clean-up is debatable as it considers each of the state variables separately. The *fitness* function introduced in (Varšek *et al.*, 1993) is probably a better measure of performance and is more in the spirit of traditional control engineering.

Michie and Camacho (1993) also report on clean-up in flying the simulated F-16 aircraft. They considered the error in straight and level flight measured as the plane's deviation from a straight line. The clone's deviation was only about 15 % of the human's deviation.

In learning to fly the Cessna (Sammut *et al.*, 1992) the clean-up effect was also observed. Figure 1 shows the trajectories in three dimensions of two flights, one by a human operator and the other one induced from the same operator's traces. Both flights accomplish the same flight plan. Clean-up is easily noticeable in the approach to the runway.

Table 1: Characteristics of the domains and parameters of learning

	Pole and cart	Cessna	F-16	Crane
# state variables	4	15	15	6
# control variables	1	4	9	2
# types of control variables	boolean	real, integer	real, integer	integer
# subjects	10	3	1	6
# traces	1	90	20	450
# events in data set	3500	90.000	25.000	450.000
length of a trace	5 min	5 min	18 min	1 - 3 min
# phases	1	7	8	1
learning program	C4.5	C4.5	C4.5	Retis, M5
delay [seconds]	0.4 - 0.5	1 - 3	1	0 - 0.1
preprocessing of data needed	no	yes	yes	no
predefined plan needed	no	yes	yes	no

Similarly, clean-up was also observed in the crane domain. In the container crane, there are six system variables: position of the trolley and its velocity, rope length and its velocity, and rope inclination angle and its velocity. The task is to move the load from a start position to a given goal position. When the goal position is reached, the state variables must be kept sufficiently close to the goal values for some minimum time interval. Figure 2 shows two time behaviours. One is by a human operator. This contains precisely those events that were used to induce the clone that produced the other trajectory. The clone was induced by Quinlan's M5 (Quinlan, 1993) which generates regression trees that are by default drastically pruned. As in the previous cases, it can be seen in Figure 2 that the clone carries out the task in a style very similar to the original. The clean-up here is reflected in that the clone is noticeably more successful than the original with respect to the time required to complete the task (75 seconds for the clone compared to 90 seconds for the original).

In the production scheduling domain (Kibira 1993) the problem is to allocate production resources to tasks in the assembly line. The goal there is that at the end of an 8.5 hour shift, the queue sizes of currently available subassemblies at various stages of the line are as close as possible to specified goal levels (500 in Kibira's experiments). The queue sizes at the start of the shift deviated grossly from the target levels. Kibira (1993) gives the time behaviour of the queue sizes at various points in the assembly line for both human expert scheduler and the clone. The clean-up is here reflected in the fact that the clone's final level (at 8.5 hours) is always closer to 500 than the human's final level.

4. OTHER ISSUES

4.1 Reaction time delay

Some researchers strongly believe that there should be a time delay between the system's state and the control action. Such a delay seems necessary because an operator cannot react to a stimulus (that is system state) instantaneously. Sammut *et al.* (1992) in their experiments used a delay that varied between 1 and 3 seconds. Their choice of the delay was pragmatic and determined experimentally. The choice was not made in a principled way although they paid considerable attention to this question in their discussion. They believed that the delay was important, but did not have a firm theoretical basis for determining appropriate delays. Although a delay appeared to be critical, slightly altering the length of the delay for recording a subject's behaviour did not critically affect the clone's performance. The clone also had to implement a delay in order to accurately mimic the human subject. The length of this delay was critical since it had to match the recording delay.

Experiments in the crane domain were also carried out with various delays (Urbančič and Bratko, 1994b). It seems that in this domain zero delay does not produce inferior results compared to other delays. Some discussions lead to the belief that the operator's delay in fact varies and depends on the situation. There are quick, purely reactive decisions, and there are also strategic, longer term decisions that have to do with setting new intermediate goals, for example start accelerating until a goal speed is attained. Also it seems that a skilled operator is capable of compensating for delays in reaction time by predicting

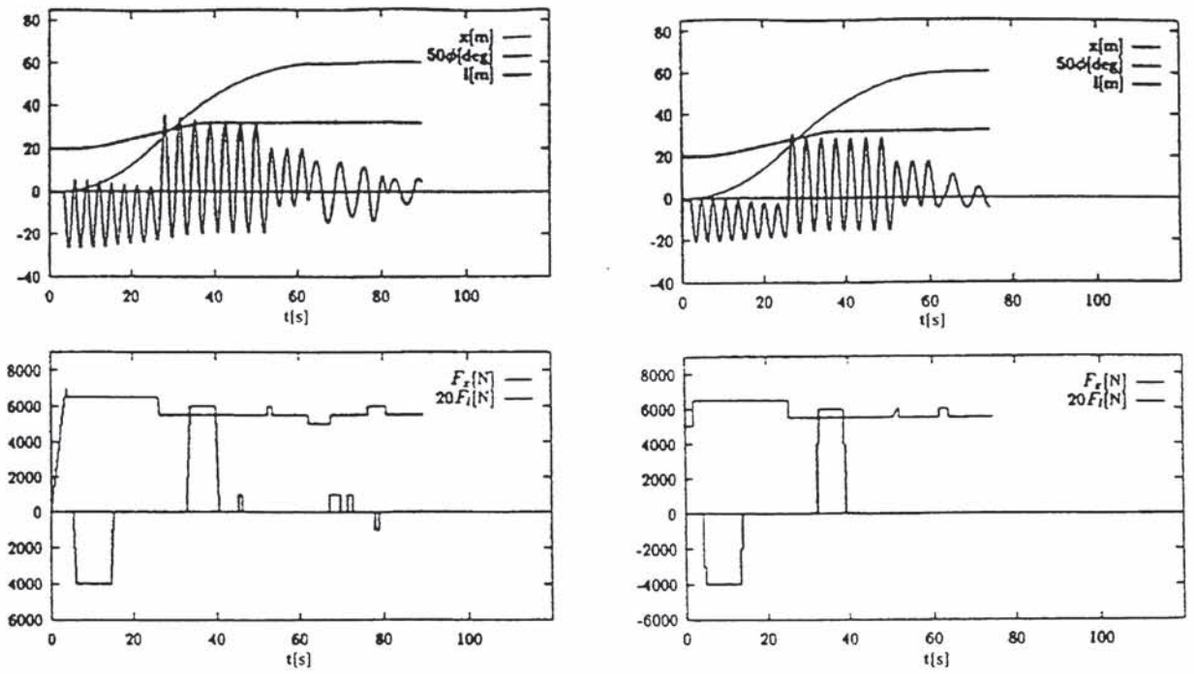

Figure 2: Diagrams on the left: one of the most "tutorial" traces by a human controlling the crane; on the right: the trace of a clone induced from the human's trace.

short term future states of the system. In extreme case this would even indicate that a "negative delay" would make sense. So straight-forward adoption of reaction time delays to unpredicted events, known from psychology, does not seem to be appropriate. To conclude, there seems to be no clear indication, either theoretical or experimental, of what would be an appropriate delay.

4.2 Choice of example traces for learning

One question that must be answered when designing a behvaioural cloning system is which traces to choose for learning among all the available example traces. The style and control strategies obviously vary significantly from operator to operator. or example in the crane domain, some operators tended towards fast and less reliable operation, others were slower, more conservative, and more reliable. Some operators were avoiding large angular accelerations at the expense of time. Such strategies produce reliable, but slow performance. This is in contrast with some operators strategies that tend to achieve faster times, but require higher accelerations of the trolley which causes large angles and requires very delicate balancing of the load at the end of the trace. There were also differences between the operators in the order of attaining the subgoals. Similar differences between individuals were observed in the learning to fly experiments (Sammut *et al.*, 1992).

To avoid mixing individual styles, the commonly agreed practice in behavioural cloning has been to combine training examples from the same subject only. However, even the example trajectories of the same subject may vary considerably. For example in the crane domain, the same subject using the same control style will produce trajectories whose finishing times are quite different. According to this, when trying to induce the most "tutorial" and "unadventurous" clones, the most conservative example trajectories have been found to be by far the most useful.

4.3 Brittleness

Experiments have shown that the generation of behavioural clones is feasible and that the clean-up effect can be observed in all the domains investigated so far. Successful clones perform similarly to the human subjects, although the clones' trajectories are of course not literal reproductions of the original trajectories. However, in the more

complex domains, such as flying and crane driving, the original experiments produced clones that were usually very brittle with respect to changes and were only successful within a fixed plan. They were sensitive to small changes in the task or the parameters of the problem domain.

These problems can be cured by training in a noisy environment. For example, the first piloting clones were built using a flight simulator that did not include turbulence or wind disturbances. Therefore, when flying straight and level, it was sufficient to leave the controls alone and the aircraft would continue along its original altitude and heading. A trace of such a flight provides no examples of what to do when the aircraft either begins off its desired course or what to do if it is pushed off course. This can be corrected by introducing turbulence and wind drift. The human pilot must now generate examples of corrections which can be used to train a more robust clone. Arentz (1994) performed just such experiments and found that clones can be constructed that are quite robust to substantial disturbances. Clearly, if the disturbances encountered by the clone are greater than those encountered by the trainer, we can expect loss of control since circumstances have been created that are outside the clone's range of experience.

4.4 Inducing human-like strategies

In all the work until now the clones have taken the form of decision or regression trees or rule sets. These clones are purely reactive and inadequately structured as conceptualisations of the human skill (unless embedded in a hand-crafted fixed plan). They lack the conceptual structure typical in human control strategies: goals and subgoals, phases and causality. The simple form of the clones as mappings from system states to actions does not suffice to express such a conceptual structure. This conceptual difference between the clones and the humans' own descriptions of their skill is analysed in (Urbančič and Bratko, 1994b).

Here we note some requirements for the representation of human-like controllers. First, such a controller should have some internal memory to maintain the current goals and phase of task. Furthermore, to enable the learning program to discover conceptual structure in a behavioural trace, the program should have access to some background knowledge about the domain.

In his work on improving yield in process control, Leech (1986) developed a two-stage method in which variables critical to the yield were first identified by induction. A further inductive step was used to construct control rules to achieve desired values for the critical variables. Donald Michie has suggested that an analogous scheme might be used in behavioural cloning. This approach is currently under investigation in the piloting domain. Initial results suggest that it may, indeed, be possible to construct more goal-oriented clones. However, there is still much work to be done.

Regardless of the representation, we believe that the study of human skill should take into account the constraints that humans must live with. One constraint is that the human can only look at a small number of state variables a time. Here are comments from one of our crane operators: "At this stage I only look at x very little; I never look at $\dot{\theta}$. [Later:] Here I never look at $\dot{x}$; if I do I get very confused". This suggests that at any given time the operator's decision only depends on a very small number of attributes. An important part of human strategy is to know what instruments to look at at various stages of the task.

5. CONCLUSIONS

Experience in behavioural cloning described in this paper indicates some elements of an emerging methodology which we summarise in the following paragraphs.

1. *Choice of example traces for learning.* Style and control strategies vary significantly from operator to operator. To avoid mixing individual styles, the commonly agreed practice in behavioural cloning has been to use training examples from the same subject only. However, even the example trajectories of the same subject may vary considerably. When trying to induce the most reliable and "unadventurous" clones, the most conservative example trajectories were found to be by far the most useful.

2. *Time delay between state and action.* Human response times for sudden stimuli do not necessarily give any indication of an appropriate delay for behavioural cloning. A reasonable method is to try first with zero delay and increase the delay gradually, looking for the best performance.

3. When *designing the representation*, i.e. choosing attributes, it is useful to take into account the operator's verbal description of his/her skill. The introduction of such a descriptions into the "cloning cycle" is discussed in (Urbančič and Bratko, 1994b).

These observations are relevant mainly when the goal of cloning is to achieve good performance. However, in using clones as *conceptualisations* of human skill, there is much to be done. Earlier, we noted the large conceptual difference between the clones, represented by decision or regression trees, and the humans' own descriptions of their skill. In (Urbančič and Bratko, 1994b) it was possible to establish only a partial correspondence between the operator's instructions and the clone. ML-based analysis of the operator's instructions helped to reveal that the operators occasionally were not doing what they believed they were doing. In part, this is due to the fact that subcognitive skills are not available to introspection and therefore, the operator's instructions are *post hoc* justifications for his/her behaviour. However, a significant part of the difference between the clone and the human's conceptualisation appears to be due to the limited representational capabilities of pure attribute-based learning systems. Structured induction techniques of the sort described in section 4.4 may be more appropriate. Inductive Logic Programming (ILP) techniques also hold promise for behavioural cloning because of their ability to accommodate background knowledge more flexibly than other learning methods. However, as most of the variables in control systems are continuous, further research is required to allow ILP to handle numeric data more effectively.

ACKNOWLEDGEMENTS

We thank Donald Michie for the many helpful discussions and suggestions that have contributed to our experiments.

REFERENCES

Anderson, C.W. (1987) Strategy Learning with Multilayer Connectionist Representations. *Proceedings of the 4th International Workshop on Machine Learning*, Morgan Kaufmann, pp. 103–114.

Arentz, D. (1994) Experiments in learning to fly. Computer Engineering Thesis, School of Computer Science and Engineering, University of New South Wales.

Barto, A.G., Sutton, R.S., Anderson, C.W. (1983) Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-13, No.5, 834-846.

Kibira, A. (1993) Developing an expert controller of a black box simulation of a telephone line using machine induction. Unpublished technical report, University of New South Wales, AI Laboratory.

Leech, W.J. (1986) A rule-based process control method with feedback. In: *Proceedings of the ISA/86 International Conference and Exhibit*, Houston, Texas.

Michie, D., Chambers, R.A. (1968) BOXES: An experiment in adaptive control. In: Dale, E., Michie, D. (eds.) *Machine Intelligence 2*, Edinburgh University Press, pp. 137–152.

Michie, D. (1993) Knowledge, learning and machine intelligence. In: L.S.Sterling (ed.) *Intelligent Systems,* Plenum Press, New York.

Michie, D., Bain, M., Hayes-Michie, J. (1990) Cognitive models from subcognitive skills. In: Grimble, M., McGhee, J., Mowforth, P. (eds.) *Knowledge-Based Systems in Industrial Control,* Stevenage: Peter Peregrinus, pp. 71–99.

Michie, D., Camacho, R. (1994) Building symbolic representations of intuitive real-time skills from performance data. In: K.Furukawa, S.Muggleton (eds.) *Machine Intelligence and Inductive Learning*, Oxford: Oxford University Press.

Quinlan, R. (1987) Simplifying decision trees. *International Journal of Man-Machine Studies,* Vol. 27, No. 3, 221–234.

Quinlan, R. (1993) Combining instance-based and model-based learning. *Proceedings of the 10th International Conference on Machine Learning,* Morgan Kaufmann, 236–243.

Sammut, C., Hurst, S., Kedzier, D., Michie, D. (1992) Learning to Fly. Sleeman, D., Edwards, P. (eds.) *Proceedings of the Ninth International Workshop on Machine Learning,* Morgan Kaufmann, pp. 385–393.

Urbančič, T., Bratko, I. (1994a) Learning to Control Dynamic Systems. In: D.Michie, D.Spiegelhalter, C.Taylor (eds.) *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, pp. 246–261.

Urbančič, T., Bratko, I. (1994b) Reconstructing Human Skill with Machine Learning. In: A.Cohn (ed.) *Proceedings of the 11th European Conference on Artificial Intelligence*, John Wiley & Sons, Ltd., 498–502.

Varšek, A., Urbančič, T., Filipič, B. (1993) Genetic Algorithms in Controller Design and Tuning. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-23(6):1330–1339.