Qualitative Reconstruction of Control Skill

Ivan Bratko
Faculty of Computer and Information Sc., University of Ljubljana
Ljubljana, Slovenia
Ivan.Bratko@fri.uni-lj.si

Abstract

Controlling complex dynamic systems requires skills that operators often cannot completely describe, but can demonstrate. This paper is concerned with the problem of transfer of human control skill into an automatic controller. The process of reconstructing a skill from an operator's behavioural traces by means of Machine Learning (ML) techniques is called behavioural cloning. This paper focusses on the questions of appropriate representation for behavioural cloning when the main goal of cloning is to provide insight into operators' subcognitive skills, and not only to induce a successful controller. Disadvantages of traditional, numerical representations are discussed, and benefits of qualitative representations are demonstrated in a case study.

Introduction

Controlling a complex dynamic system, such as an aircraft or a crane, requires operator's skill acquired through experience. In this paper we are interested in the question of designing automatic controllers by transfer of operators' skill into a controller. One approach would be to attempt to extract the skill from the operator in a dialogue fashion whereby the operator would be expected to describe his or her skill. This description would then be appropriately formalised and built into an automatic controller. The problem with this approach is that the skill is subcognitive and the operator is usually only capable of describing it incompletely and approximately. Such descriptions can only be used as basic guidelines for constructing automatic controllers, because as discussed for example in (Urbančič and Bratko, 1994), the operator's descriptions are not operational in the sense of being directly translatable into an automatic controller. Given the difficulties of skill transfer through introspection, an alternative approach to skill reconstruction is to start from the *manifestation* of the skill. Although an operational description of the skill is not available, the manifestation of the skill *is* available in the form of traces of the operator's actions. One idea is to use these traces as examples and extract operational descriptions of the skill by Machine Learning techniques. Extracting models of a real-time skill from operators' behaviour traces by Machine Learning (ML) is also known as *behavioural cloning*, a phrase coined by Donald Michie (1993).

In general there are two goals of behavioural cloning:

1. To generate *good performance* clones, that is those that can reliably carry out the control task.
2. To generate *transparent* clones, that is those that would help making the human operator's skill symbolicly explicit.

The second goal is important for several reasons:

- Operationalising human operator's instructions for controlling a system. Such instructions are a useful source of information, but are normally too incomplete and imprecise to be directly translatable into a control program.
- Flexibly modifying and optimising the induced clones to prevent some undesired patterns in their behaviour.
- Understanding what exactly a human operator is doing and why. This is of great practi-

cal importance regarding the capture of exceptional operators' skill and its transfer to less gifted operators. The operator's control strategy would ideally be understood in terms of goals, subgoals, plans, feedback loops, causal relations between actions and state conditions etc. These conditions are to be stated in terms of information that is easily accessible to the operator, e.g. visually. It will be argued later that such information should be largely qualitative, as opposed to the prevailing current experimentl practice in behavioural cloning.

Behavioural cloning has been studied in various dynamic domains, including: pole balancing (Michie *et al.*, 1990), piloting (Sammut *et al.*, 1992; Cessna aircraft; Michie and Camacho, 1994; F16), operating crains (Urbančič and Bratko, 1994), electrical discharge machining (Karalič 1995). Bratko, Urbančič and Sammut (1995) give a review of experiments in behavioural cloning.

The conclusions from these studies can be summarised as follows. On the positive side, successful clones have been induced using standard ML techniques in all the domains. Also, the so-called clean-up effect, whereby the clone surpasses its original, has been occasionally observed in all the domains. However, the present approaches to behavioural cloning typically suffer from the following problems:

(a) They lack robustness in the sense that they do not provide any guarantee of inducing with high probability a successful clone from given data.

(b) Typically, the induced clones are not sufficiently robust with respect to changes in the control task.

(c) Although the clones do provide some insight into the control strategy, they in general lack conceptual structure and representation that would clearly capture structure and style of the operator's control strategy.

In particular the last deficiency is indicative of slow progress in generating clones which would help us to really *understand* the operator' subcognitive skill. In this paper we investigate reasons for this and propose that one of the reasons usually lies in the representation used in skill reconstruction. The usual representation is inherited from traditional control theory and is entirely numerical. It will be shown in this paper that often such representations do not correspond to representations that operators work with. It will be shown by an experimental study that more appropriate representations are largely qualitative and involve history (not just the current state of the system) and qualitative trends.

There seems to be no other work concerned directly with qualitative approaches to behavioural cloning. However, more broadly related work includes research on qualitative reasoning about control. Makarovič (1991) derived controllers by qualitative reasoning about differential equations models. Kuipers and Astrom (1994) applied qualitative simulation to the analysis of heterogenous fuzzy controllers. Bratko (1996) derived a qualitative control rule from a qualitative differential equations model of the controlled system. De Jong (1996) based his "explanation-based control" on a qualitative model of the controlled system.

Problem formulation for machine learning

The following is the usual procedure of applying Machine Learning (ML) to recovering control rules from example execution traces. A continuous trace is normally sampled so that we have a sequence of pairs $(SystemState_i, Action_i)$ ordered according to time. Then, usually, the *sequence* of these pairs is viewed as a *set* of examples, thereby ignoring the time order. This simplification can be justified by the formal argument that the control action is fully determined by the state of the controlled system. Many Machine Learning programs can be viewed as reconstructors of functions $y = f(x_1, x_2, ...)$ from sets of given pairs of the form:

$$((x_{1i}, x_{2i}, ...), y_i)$$

The arguments $x_1, x_2, ...$ are usually called *attributes*, and function value y is called the *class* value. So the usual formulation of behavioural cloning as a ML task is as follows: the system state variables x_1, x_2 etc. are considered as attributes, and the action as the class variable.

However, this representational decision of viewing sequences as sets is debatable: the human operator's decisions almost always depend on the history and not only on the current state of the system. In controlling the system, the operator pursues certain goals over a time period and then switches to other goals, etc. So both the state *and* the current goal determine the action, although the goal is not part of the system's state but only exists in the operator's brain. In spite of these reservations, all the known studies in behavioural cloning basically treat example traces as sets.

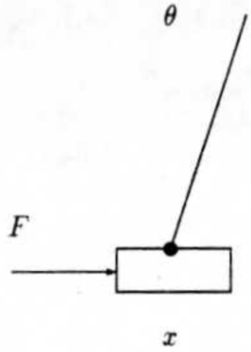
Another questionable representational decision is the choice of (numerical) system state variables as attributes. An operator, executing his skill on an actual physical system, or on a dynamic graphical animation of a simulated system, is hardly using the numerical state variables. First, some of these variables values are hard to observe precisely (as the classical control theory would assume), for example the angular velocity of the pole in pole balancing. Also, the operator surely does not online evaluate an arithmetic formula to decide on his next action. Instead, he has to base his control decisions on other, easily recognizable patterns in terms of qualitative features of the current state and recent previous behaviour of the system. Such qualitative features include the following ones, well known in qualitative physics: the sign of a variable (the pole leaning left or right), the direction of change (pole rotating left), variable crossing a landmark (pole upright), variable just reached local extreme. Small continuing oscillations cannot be precisely perceived (neither amplitudes nor exact times of extremes), but they are perceived qualitatively as a "stuttering" motion with, say, an overal trend to increasing.

A study (Urbančič and Bratko 1994) of the correspondence between induced clones and the operators' own understanding was quite revealing of the qualitative features actually used by the operators in the crane domain. The operators were asked to describe their skill for the purpose of teaching a novice operator or translating the skill into a control program which would emulate the skill. As it turned out, such "operalisations" of the operators' instructions were never successful because the operators did not know exactly what they were doing. However, the operator's instructions were very indicative of suitable, largely qualitative representations that should be used for faithful cloning. The following fragments from the instructions of one of the operators about crane control are quite illuminating: "When $\dot{x} \approx 0.30$ and $\dot{x}$ increasing or $\dot{x}$ decreasing and $\dot{x} \approx 0.25$ make force F_x equal to 6000. (...). Now $\dot{x}$ is decreasing oscillatorily towards 0 – one minimum is at $\approx 0.30 \pm 0.02$. A little before the next minimum, change F_x to 6000. In the same "swing", a little before the maximum change F_x to 5500. Wait until steady. (...). Now $\dot{x}$ oscillates slightly. When $\dot{x}$ is increasing, a little before maximum change F_x to 5500."

To repeat, in this paper we are particularly interested in behavioural cloning with the aim of *understanding* the operator's skill. The next sections present a detailed experiment in how prevailingly qualitative representation made of the above indicated elements can be used in ML-based reconstruction of pole balancing skill.

Initial experiment in human skill acquisition and its reconstruction

Eight students were asked to learn the task of controlling a simulated pole-cart system (Fig. 1), using the instrument representation. That is, they did not know at all what the controlled system was. Instead, they could only see the current values of four variablesA, B, C and D displayed as bar chart. These variables were actually the pole-cart's state variables x (position of the cart),$\dot{x}$ (velocity of the cart), θ (angle of the pole),and $\dot{\theta}$ (angular velocity). The parameters of the simulated pole and cart were the same as used by most of the researchers in previous studies in this



$$\ddot{x} = \frac{4F + 2lm\dot{\theta}^2 \sin\theta - 1.5mg\sin 2\theta}{4M + 4m - 3m\cos^2\theta}$$

$$\ddot{\theta} = 6\frac{(m+M)g\sin\theta - F\cos\theta - 0.5ml\dot{\theta}^2\sin\theta\cos\theta}{(4M + 4m - 3m\cos^2\theta)l}$$

Figure 1: The pole-and-cart system. $\theta = 0$ for pole upright, $x = 0$ for cart centred. M is the mass of the cart, m is the mass of the pole, and l is its length.

domain (e.g. Michie and Chambers 1968): mass of the cart $M = 1kg$, mass of the pole $m = 0.1kg$, length of the pole $l = 1m$. The start state was: $x = -2m, \dot{x} = \theta = \dot{\theta} = 0$. The control task was to bring the cart into the middle of the track ($x = 0$) and stay inside a small neighbourhood of $x = 0$ for 2 sec.The control regime was that of bang-bang with switching the control force between +10N and -10N, pushing the cart to the right or left.

The subjects were allowed 150 trials to experiment with the system and acquire the control skill. To make the task easier, the simulator was slowed down by a factor of 8. A trial either ended successfully, or was terminated with failure because of one of the following reasons: the cart went outside the track limits (-2.4m, +2.4m), or the angle became too large (± 45 deg), or time limit was reached (15 simulated seconds, that is 120 actual seconds after the slow-down).

Two subjects (M.M. and A.Z.) who had at least one successful trial were then asked to translate their skills into a control algorithm. More precisely, they were asked to write a Pascal function E = E(A,B,C,D) where E is the control force. Their prevailing method was introspection and experimentation with the simulator. They wrote several control programs and numerous versions of their refinements. Their typical program consisted of some 30 lines of Pascal code. None of these programs could complete the control task. After a total of about one week of trying, they produced a final version that was close to success. It moved the cart to the center of the track, but then, trying to reduce $\dot{x}$, the controller quickly collapsed because of pole crash. This inability to handcraft a controller on the basis of skill appears paradoxical in the view of the fact that there exists a controller that can be written as one line of Pascal. Namely, the classical control theory solution (obtainable after linear approximation of the corresponding differential equations) establishes that control force is a linear combination of the state variables. An appropriate combination of the coefficients is:

$$F = 0.109x + 2.17\dot{x} + 26.53\theta + 6.78\dot{\theta}$$

This control rule can be successfully executed in the bang-bang regime at, say, 50 Hz by simply taking the sign of F above. Fig. 2 shows an example control trace. It should be noted that the students at that time still did not know what the controlled system was.

M.M. and A.Z. stated in their final report the difficulties they felt contributed to their failure to reconstruct a successful controller: "We found it difficult to translate our control strategy into a program. Some of our control decisions are reflex and we are probably not aware of them. Another difficulty is that the control program can only use the current values of the four state variables, but cannot, for example, take into account how long a variable has not changed its direction. In manual control we found this information quite significant."

Later one of the two students (Andrej Zalar) was given a machine learning program RETIS (Karalič 1992) as a tool to help him in cloning his own skill. Zalar took the usual approach to behavioural cloning. He used sequences of state-action pairs from his own traces as sources of

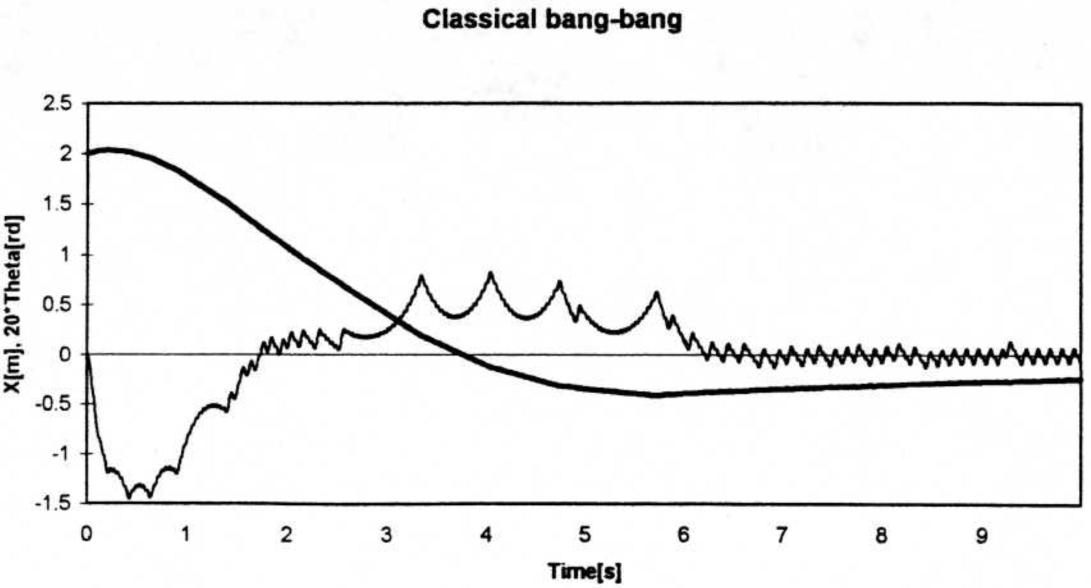
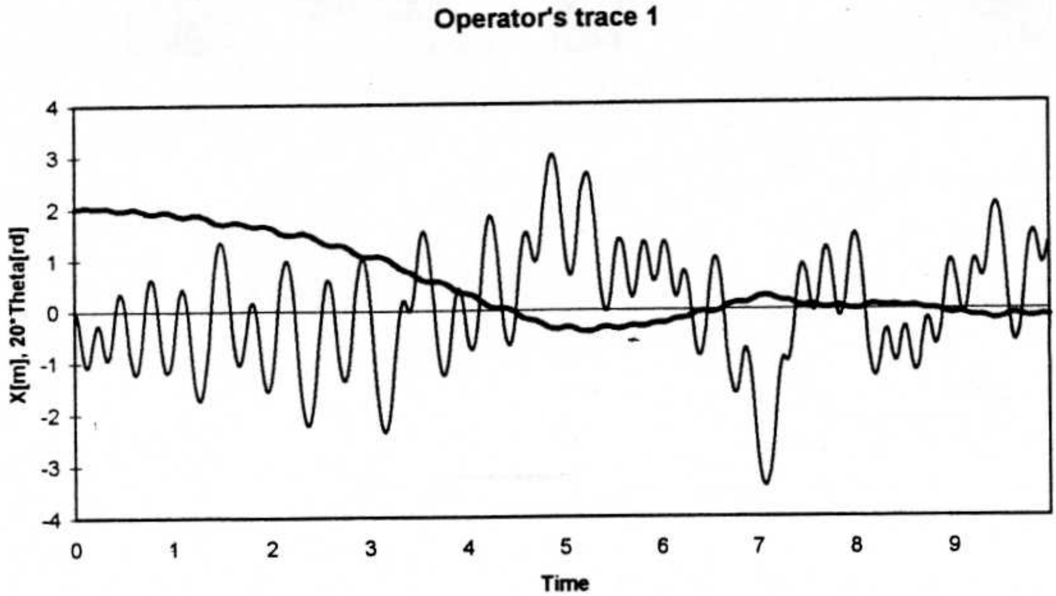


Figure 2: Example control trace produced by the "classical" control rule executed under the bang-bang regime.

examples to induce rules for action as a function of state variables. He thus induced several successful controllers which were typically much shorter than the control programs he handcrafted previously (Zalar 1994). He formed a learning set by concatenating two of his control traces, starting from the opposite sides on the track. He produced these two traces with special care, this time using an animated graphics simulator and already knowing that the controlled system was pole-and-cart. He tried to demonstrate the skill cleanly so that the traces would be a tutorial demonstration for a learner (either human or machine). He chose the first 10 seconds of simulated time of each of the traces (Fig. 3). These two traces provided enough variety for reliable induction of good control rules. However, some care was necessary in setting the parameters of RETIS. In particular, it was necessary to handle rather high level of noise (small inconsistencies) in the traces by rather high degree of tree pruning (done through setting the m-parameter of the m-pruning technique). Eventually, with appropriate settings many different successful controllers were induced in the form of regression trees. Fig. 4 shows one of them obtained after drastic pruning to cope with noise.

This controller, executed under the bang-bang regime performs rather well. However, a closer inspection reveals that the resulting behaviour is qualitatively quite different from the operator's original style of driving the system (compare the traces in Figs. 3 and 4). The resulting behaviour is in fact qualitatively more similar to the one produced by the classical controller (trace in Fig. 2). This confirms a suspicion that the controller of Fig. 4, although successful, does not provide a good insight into the essential mechanism of the operator's skill. Similarly, the state variables representation and the formulas in Fig. 4 do not seem to adequately reflect the operator's control strategy. This can be concluded simply on the grounds of the mismatch between this representation and the representations used in operators' descriptions of their skill. There is also a big difference in the rates at which the human and his clone change the direction of the control force. Typically, the human keeps the direction unchanged for five to eight sample times , that is time points at which he is allowed to change the direction. On the contrary, the clone tends to change the direction of force almost as frequently as possible, that is at about five times higher rate than the human. In the next section we use a



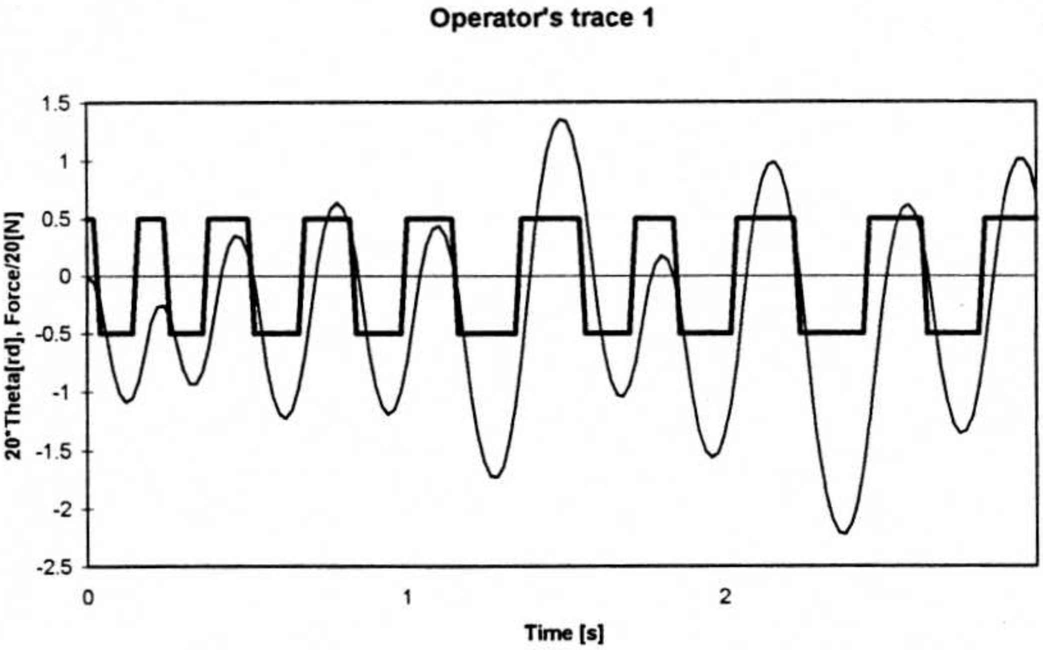
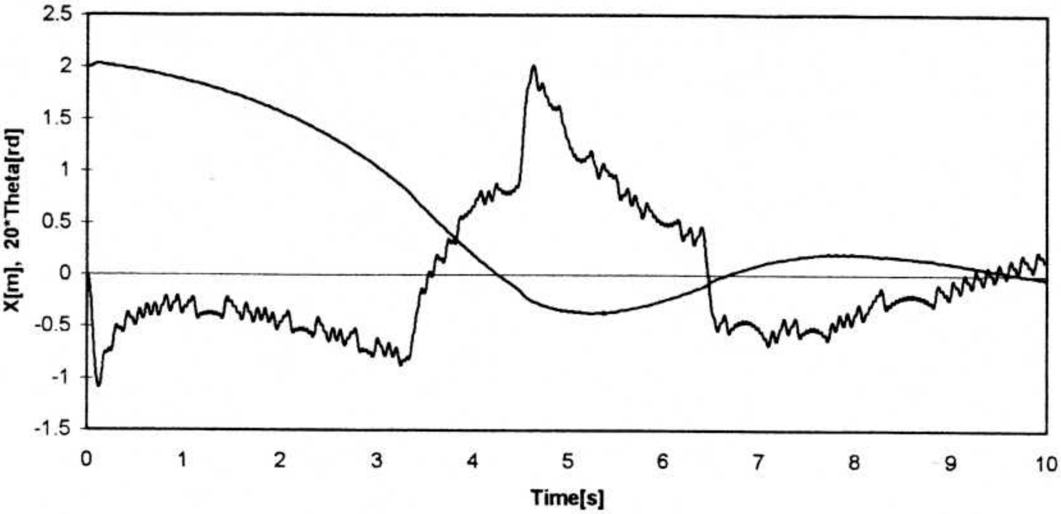


Figure 3: One of the operator's traces used for the reconstruction of his skill. The top diagram shows the position of the cart and the angle of the pole over 10 seconds of simulated time. The bottom diagram shows the first three seconds of the angle and the control force.

```
Created with m = 0.00, pruned  with m = 40.00

NODE: Weight=999.00,  y = -0.151+2*x1+4.4743*x2+112*x3+ 11*x4,  Error = 6.334
[ X  <=  -0.125940 ]
     LEAF: Weight=367.00,  y = -6.919 -0*x1+12.8902*x2+142*x3+16*x4,  Error = 5.236
[    -0.125940 < X ]
     NODE: Weight=632.00,  y = 1.059+2*x1+4.7957*x2+123*x3+12*x4,  Error = 6.103
     [ X <=  0.800450 ]
          LEAF: Weight=464.00,  y = 0.328+11*x1+9.1461*x2+132*x3+15*x4,  Error= 5.988
     [ 0.800450 < X ]
          LEAF: Weight=168.00,  y = 48.911-21*x1+31.5710*x2+148*x3+29*x4,  Error= 4.176
     ENDNODE
ENDNODE
```



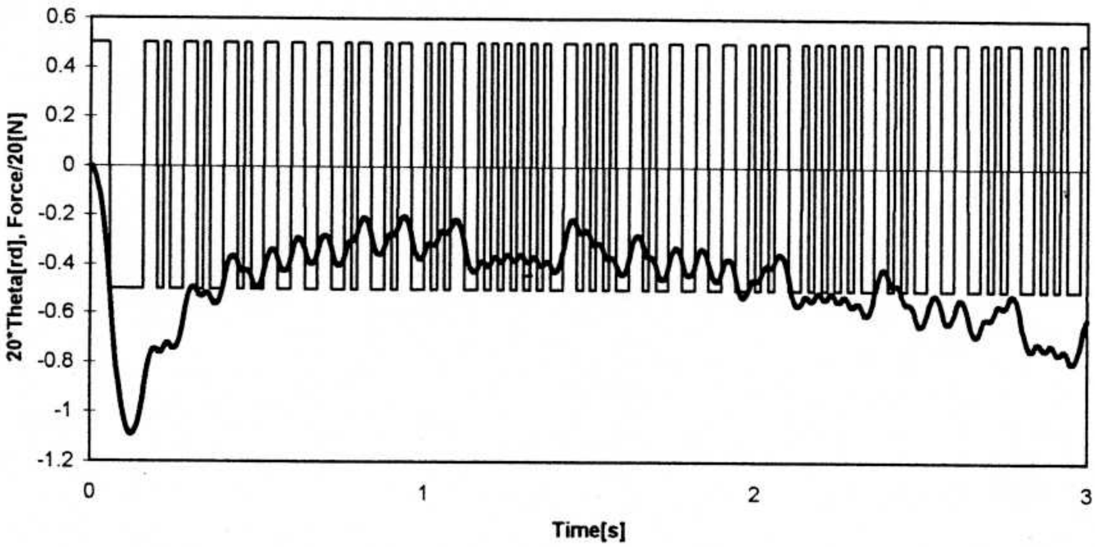


Figure 4: Top: a regression tree controller induced from the two operator's traces. The variables y, x1, x2, x3, and x4 in the regression formulas correspond to Force, X, DX, Theta and DTheta respectively. Bottom: a control trace produced by this rule; the lower diagram shows fast switching rate of Force.

more plausible, largely qualitative representation as a basis for skill reconstruction by means of Machine Learning from operator's traces.

Skill reconstruction with qualitative representation

The original numerical operator's traces were converted into qualitative as follows. For each time point we have the values of the four state variables and the force determined by the operator at this time point. First, to introduce some notion of history, the previous value of force was added. Now, each numerical variable was converted into a qualitative value *pos*, *zero* or *neg* in the usual way. Also, time points at which there was no qualitative change w.r.t. the previous time point were eliminated. This resulted in 400 time points out of 999 in the original two traces. The learning problem was then: determine the value of control force F as a function of five (qualitative) attributes X, DX, TH, DTH, PREVIOUSF (cart position and its velocity, pole's angle and its velocity, and the previous control force respectively).

A decision tree learning program Magnus Assistant (Mladenić 1990) was used this time. This program belongs to the large family of tree learners that includes the well known programs CART (Breiman et al. 1984) and C4.5 (Quinlan 1993). A distinguishing feature of Magnus Assistant is its method for coping with noisy data called minimal error pruning with m-estimate (Cestnik and Bratko 1991). Such a noise-handling technique was necessary to use in our case because of noise in the data.

By varying the pruning level (parameter m), various decision trees were induced. They show similar relationships, however the one in Fig. 5 seems to capture the operator's control strategy in the neatest way. This tree was obtained with quite severe pruning that corresponds to $m = 5$. The tree in Fig. 5 is as output by Magnus Assistant, just some of the statistical information was removed from the output for simplicity. The numbers in the leaves of the tree indicate the relative frequences of the two decision classes *pos* and *neg*. Obviously, most of the decisions are qualified by high probabilities whereas in some cases the decisions are less clear. Let us consider the likely decisions as "hard" constraints on the control strategy. Studying the tree in Fig. 5 reveals a very simple qualitative control strategy that can be expressed by the following constraints regarding the controller's actions:

1. When Force is negative (pushing left) and angle decreasing (pole rotating to left), keep Force negative.

2. When Force is negative and angle negative and increasing, keep Force negative until some time before (or just the time when) the angle crosses zero (pole upright).

3. There are two similar analogous constraints with the signs of force, angle and angular velocity reversed w.r.t. the situations above.

Notice that the tree does not mention at all the attributes X and DX. Their influence was not sufficiently regular in the learning data and conseqently they were pruned away as spurious (noise). There is one decision left unclear in the above control constraints: when exactly the switch of Force from left to right and vice versa occurs. The induced constraints only specify intervals at which this happens. Regardless of this, however, the qualitative behaviour of the angle is determined: the pole oscillates about the vertical position. But the extreme values of the angle and the oscillation rate depend on the exact values of the angle at which the force changes direction. In particular, asymmetrical switching points between left and right cause asymmetrical oscillation. Experiments with the simulator quickly show that by varying the switch points within the specified intervals for the angle, the possibly asymmetrical oscillations cause bias in the horizontal acceleration of the cart. So by adjusting the angular switching points, the operator may (indirectly) control the horizontal acceleration.

Using the Retis learning program on the operator's traces again, it was found that a useful parameter to determine the switch angle value is the ratio between the switch angle and the previous local extreme value of the angle. This extreme is

```
Tree construction parameters:
Informativity pruning factor:    0.00
Class frequency threshold:     100.00%
Minimal weight threshold:       10.00%
Post pruning:                     YES
Post pruning using m-estimate: 5.00
tree was reduced to 40.74% of previous size
```

```
|   PREVIOUSF [400]
|   [neg]
|   |   DTH [194]
|   |   [pos]
|   |   |   TH [109]
|   |   |   [pos]
|   |   |   |   pos                  [9, 75.0]
|   |   |   |   neg                  [3, 25.0]
|   |   |   [neg, zero]
|   |   |   |   neg                  [62, 63.9]
|   |   |   |   pos                  [35, 36.1]
|   |   [neg, zero]
|   |   |   neg                  [85, 100.0]
|   |   |   pos                  [0, 0.0]
|   [pos]
|   |   DTH [206]
|   |   [neg]
|   |   |   TH [100]
|   |   |   [pos]
|   |   |   |   pos                  [42, 67.7]
|   |   |   |   neg                  [20, 32.3]
|   |   |   [neg, zero]
|   |   |   |   neg                  [24, 63.2]
|   |   |   |   pos                  [14, 36.8]
|   |   [pos, zero]
|   |   |   pos                  [105, 99.1]
|   |   |   neg                  [1, 0.9]
```

Figure 5: Decision tree induced from qualitative operator's traces. The class at the leaves is the direction of control force.

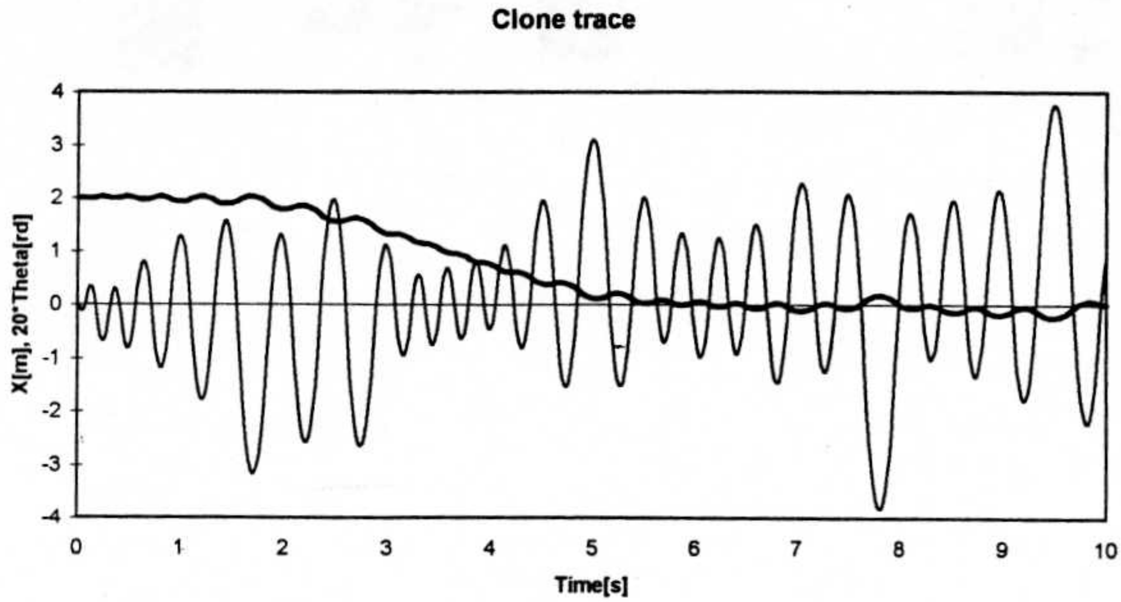
either minimal (negative), or maximal (positive), depending on the type of switch: if positive then switch to left, if negative then switch to right. Experimentally it was found that the behaviour of the system is rather robust with respect to the exact choice of this angle ratio. However, by qualitatively comparing the clone's control traces and the operator's traces, the following setting seemed to correspond to the operator very well: when angle is decreasing and Force is positive, switch Force to negative when angle is at 40% of the last local maximum; if at the same time switch from left to right occurs at angle crossing zero (from negative to positive) then the resulting oscillatory behaviour also causes a positive acceleration in the horizontal direction. Analogously, a negative horizontal acceleration can be forced. "Symmetrical" switching angles cause no horizontal acceleration.

It should be noted that this horizontal acceleration is associated with pole's oscillations. Therefore X and DX also behave oscillatorily. Due to bang-bang regime when force changes are only allowed at discrete times, these oscillations are quite irregular, but they produce clear overall trends of either accelerating to left or right.

Some simple rules were experimentally determined about when to accelerate right or left, or when not to accelerate at all. Figure 6 shows a control trace of this semi-qualitative clone. The qualitative appearance of this trace is very similar to the original operator's trace in Fig. 3. Experiments also show that the so obtained clone, although being far from the optimal controller, is very robust with respect to changes in the control task, such as change of initial state or change in the parameters of the controlled system.

Conclusion

Some problems of behavioural clonig were discussed in the paper. In particular, disadvantages of using, as usual in behavioural cloning, numerical state variables representations were demonstrated. Such representations, borrowed from classical control theory, although very useful in the design of controllers, are inadequate for skill reconstruction when the goal is to understand the



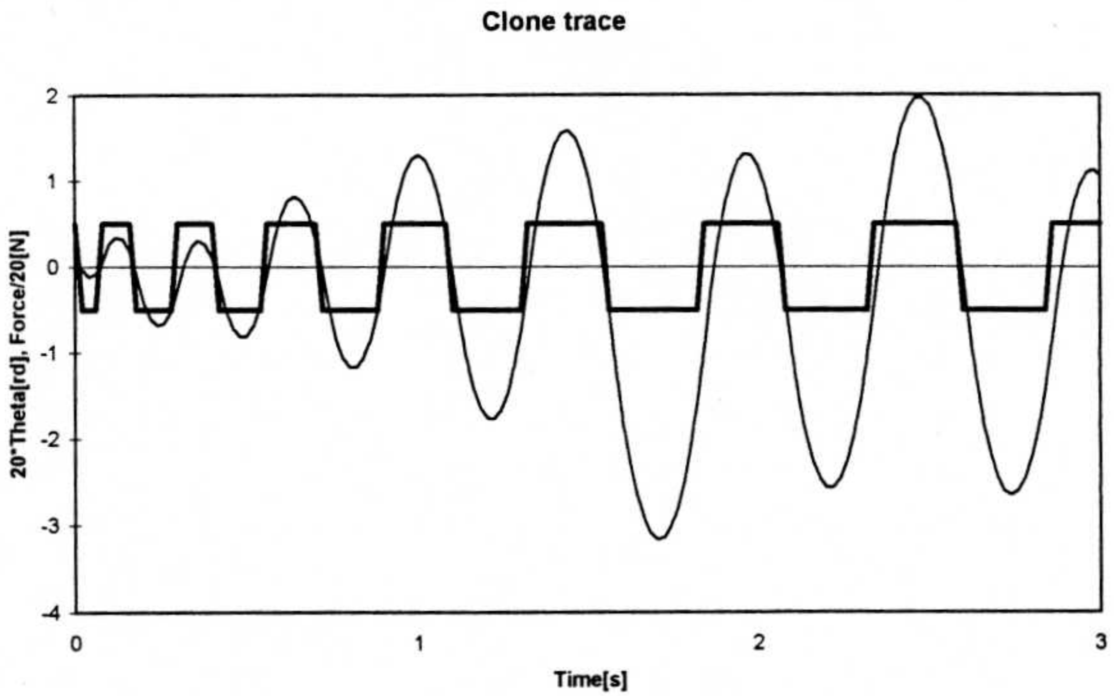


Figure 6: A control trace produced by the "qualitative" controller of Fig. 5.

essence of the operator's control strategy.

It was shown in the paper by a detailed case study that qualitative representations for machine learning are more appropriate. The resulting induced controller seems to be a likely correct reconstruction of the operator's control skill. It reveals a very simple control strategy that can be easily executed under the perceptional limitations imposed on the operator. This strategy can also be easily communicated to a novice operator, who should be able to quickly learn to execute it effectively.

Acknowledgements

This work was partly supported by the Slovenian Ministry of Science and Technology. I would like to thank several people who contributed by carrying out experiments, providing learning software, or through discussion, including: Aram Karalič, Marko Grobelnik, Donald Michie, Dunja Mladenič, Claude Sammut, Tanja Urbančič, Andrej Zalar.

References

Bratko, I. (1996) Deriving qualitative control for dynamic systems. *Machine Intelligence 14* (eds. K. Furukawa, D. Michie, S. Muggleton) Oxford: Clarendon Press.

Bratko, I., Urbančič, T, Sammut, C. (1995) Behavioural cloning: phenomena, results and problems. *Automated Systems Based on Human Skills*, IFAC Symposium, Berlin, sept. 1995.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984) *Classification and Regression Trees.* Belmont, CA: Wadsworth.

Cestnik, B., Bratko, I.. On estimating probabilities in tree pruning. *Machine Learning - EWSL 91* (ed. Y. Kodratoff) Springer-Verlag 1991 (Proc. EWSL 91, Porto, Portugal, March 1991)

De Jong, G. (1995) Explanation Based Control. *Proc. Int. Conf. Machine Learning 95*, Tahoe City, Ca., July 1995.

Karalič, A. (1992) Employing Linear Regression in Regression Tree Leaves. *Proc. 10th European Conference on Artificial Intelligence*, pp. 440-441, Vienna, 1992.

Karalič, A. (1995) *First-Order Regression.* Ph. D. Thesis: Ljubljana University, Faculty of Electr. Eng. and Computer Sc., Ljubljana.

Kuipers, B., Astrom, K. (1994) The composition and validation of heterogeneous control laws. *Automatica*, Vol. 30, 233-249.

Makarovič, A. (1991) A qualitative way of solving the pole balancing problem. In: *Machine Intelligence 12* (J. Hayes, D. Michie, E. Tyugu). Oxford University Press.

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.

Mladenić, D. (1990) *Machine Learning System Magnus Assistant*, B.Sc. Thesis, University of Ljubljana, Faculty of Computer and Info. Sc. (in Slovenian).

Quinlan, J.R. (1993a) *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, Ca.

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. (1994) 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.

Zalar, A. (1994) *Learning to Control a Dynamic System.* Univ. of Ljubljana, Faculty of Electr. and Computer Sc. (BSc Thesis, in Slovenian).