On the Role of Machine Learning in Knowledge-Based Control

Werner Brockmann

University GH Paderborn
FB 14, FG Datentechnik
D-33095 Paderborn, Germany

Abstract. Knowledge-based methods gain increasing importance in automation systems. But many real applications are too complex or there is too little understanding to acquire useful knowledge. Therefore machine learning techniques like the directed self-learning which is used here may help to bridge this gap. In order to point out the advantages of machine learning in process automation, we applied the directed self-learning method to the control of an inverted pendulum. Through a comparison between a knowledge-based and a machine learning version of the controller, both based on the knowledge of the same expert, results were achieved which demonstrate the usefulness of machine learning in control applications.

1 Introduction

Knowledge-based control systems, hereinafter called KBC, offer an important alternative to conventional controllers especially because of their ability to implement nonlinear controllers in an intuitive and model-free way. A KBC therefore evaluates the state of the process by linguistic variables. The relation between concrete, numerical feature values and the corresponding terms of the linguistic variable may be described by fuzzy sets via membership functions, as in fuzzy controllers, or in a crisp manner, like the ARON-technique [1]. Both allow very complex nonlinearities, but knowledge based methods are normally limited by an overwhelming amount of rules in complex applications. So machine learning is a very promising approach to cope with complex applications. In this paper we therefore try to work out the advantages of the self-organizing process itself over a direct specification by expert knowledge. Thus we compared an online machine learning technique relying on a knowledge-based law of adaptation to a knowledge-based controller which is specified by control knowledge of the same expert. Here we use the ARON-technique (Alternatives Regularly Organized and Numbered) [1] in conjunction with the directed self-learning technique [2] because they are as simple as possible and offer a very intuitive way of learning. In order to work out the advantages of machine learning clearly, we used the simple and popular example of controlling an inverted pendulum.

2 Knowledge-Based Control of an Inverted Pendulum

Based on the experience of a human expert, a KBC was designed using the angle of distortion θ and its velocity θ' as features. Each feature is subdivided by 7 non-linear linguistic terms. On the output side, the motor voltage is specified directly in the control actions. The control rules were tuned to optimize the response to an initial distortion of 30°.

Fig. 2 shows the control results of the KBC. The first swing in the Fig. 2 is the control response to the initial distortion. The pole is raised with a small overshoot and is standing after 0.25 seconds. This result is optimal in the expert's notion because it reflects his individual idea of what the controller is capable of. The second swing additionally demonstrates the control response to a temporary (50 ms) force which tries to knock down the pole. Although the KBC was not tuned to meet disturbing forces, it keeps the pole upright with a satisfactory transient phase. The state plane representation of the KBC is shown in Fig. 6 in order to illustrate the control characteristics as an input-output relation. It clearly indicates the stepped, nonlinear control actions of the KBC which are typical for the ARON-technique.

3 Self-Learning Control of an Inverted Pendulum

The directed self-learning we used operates similar to self-organizing controllers and is described further in [2]. As Fig. 1 shows, it uses a second knowledge based system, the so-called hyper-system which is also implemented by the ARON-technique and incorporates a knowledge based law of adaption. In this case, the hyper-knowledge is also based on θ and θ′ for they are needed to rate the performance of the controller. They are differentiated by 10 and 11 linguistic terms. The rules describing the hyper-knowledge were spezified by the same expert as above. Such rule sound like *'IF θ is positive large AND θ′ is positive large THEN the responsible control action has to be increased by an amount of 3 Volts'*. The latency time we used to access a control actions of the basic-system was 2 sampling periods.

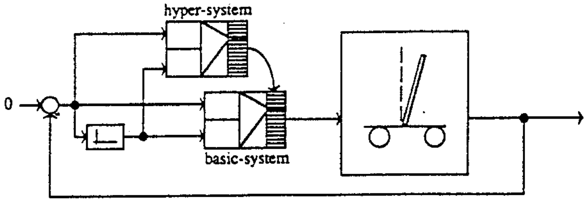

Fig. 1. Scheme of self-learning control of an inverted pendulum

The basic-system is the same as desribed in chapter 2, except that only 4 coarse rules were implemented to prevent the pole from falling down. They are only triggered when θ or θ′ is extremely large, as Fig. 6 illustrates. Due to these rules, the KBC only performs some strong control actions, making the pole to swing back and forth. This can be seen during the first 0.9 seconds in Fig. 3. At 0.9 seconds the hyper-system is activated. Its effect is clearly seen in Fig. 3 as well as in Fig. 8 because the control actions get more and more elaborated. Thus the pole is stabilized without any human interaction in the upper position within 1.3 seconds. The convergence is so quick because the basic-system uses a small amount of terms which therefore have a large range of applicability. Thus they are addressed and modified frequently.

The self-learning controller also had to cope with initial distortions. The results of distortions of ±30° are shown in Fig. 4 for applying the distortion for the first time in the first half of the figure and after 20 times in the second half. The corresponding state space representation is given in Fig. 9. It demonstrates that the hyper-system also improves these situations immediately although the hyper-knowledge was not designed for this case. Apparently a performance is yielded which is much better than that of the KBC. Only 0.13 seconds are needed to stabilize the pole, even without an overshoot Although the controller is based on a coarsely quantized feature space, this is the best result we obtained by investigating many different kinds of controllers, like linear, fuzzy and adaptive ones. The reason becomes evident when comparing Fig. 7 and Fig. 9.

The a priori given control knowledge is much more diversified, but doesn't reach the strength of the knowledge-based version if compared to Fig. 6. Thus, the reason for the improved control results is the elevation and the valley which occur within in the state space near to $\theta = 0$ and to $\theta' = 0$ (Fig. 9). This non-monotonic behavior is completely new and unecpexted at first. But considering the pole's moment of inertia it becomes evident that it is useful to apply a control action into the opposite direction in order to get a very fast stabilization, even though the pole is not yet in an upright position. Thus the machine learning version of the controller had detected a strategy which makes real sense.

The same effect is emphasized by applying an external force which tries to knock down the pole. So the first swing in Fig. 5 is based on the knowledge after the first trial of Fig. 3. The control response is still comparable to that of the knowledge-based control, see Fig. 2. The second swing shows the effect of the external force to the self-learned controller after training the intial distortions for 20 times. The tipping movement is reduced significantly and an overshoot is avoided although the disturbing force was not applied during the training phase so far. The third swing of Fig. 5 finally demonstrates the effect of considering the external force within the learning process for only 3 times. The tipping movement is reduced drastically so that it is even nearly negligible. The reason are the strong control actions which compensate the external force almost immediately which also can be seen in the state space representation of Fig. 10.

4 Concluding Remarks

In this paper we used the example of the self-learning control of an inverted pendulum to demonstrate the usefulness of machine learning techniques in process control. It turned out that techniques like the directed self-learning may lead to a significant improvement, even in applications for which a knowledge-based solution exists. The reason is that the machine learning version is not limited by the imagination of an expert, but is directly evaluating the process behavior instead. It further turned out that self-learning systems act flexible even in unforeseen situations. This may lead to a control behavior which differs from ordinary or knowledge-based control systems. Thus some unforeseen solutions may be derived automatically, as the effect of a disturbing force demonstrates.

References

1. Brockmann, W.: Combining Real-Time with Knowledge Processing Techniques. 5th Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA-AIE, Springer Verlag, Berlin, 1992, 594-603
2. Brockmann, W.: Online Machine Learning for Adaptive Control. IEEE Int. Workshop on Emerging Technologies and Factory Automation ETFA, CRL Publishing Ltd., London, 1992, 190-195
3. Mandic, N.J., Scharf, E.M. et. al.: Practical Application of a Heuristic Fuzzy Rule-Based Controller to the Dynamic Control of a Robot Arm. IEE Proc., Vol. 132, Pt. D, No. 4, 1985
4. Shao, S.: Fuzzy Self-Organizing Controller and its Application for Dynamic Processes. Fuzzy Sets and Systems, 26(1998), 151-164
5. Stender, J.: SOL - Second Order Learning. Brainware GmbH, Berlin, 1990
6. Yamakawa, T.: Stabilization of an Inverted Pendulum by a High-Speed Fuzzy Logic Controller Hardware System. Fuzzy Sets and Systems 32(1989), 161-180
7. Zhang, B., Grant, E.: Experiments in Adaptive Rule-Based Control. 3rd Int. Conf. on Industrial and Engineering Appl. of Art.Int. and Expert Systems, 1990, 563-568

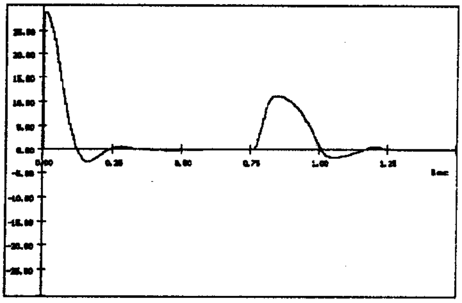

Fig. 2. Knowledge-based control of an inverted pendulum

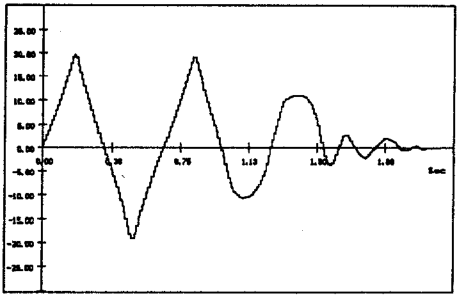

Fig. 3. Initial learning of the self-learning control of an inverted pendulum

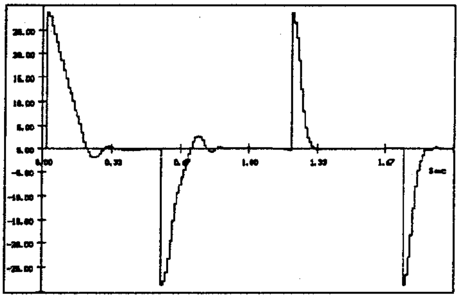

Fig. 4. Response to a distortion of ±30°

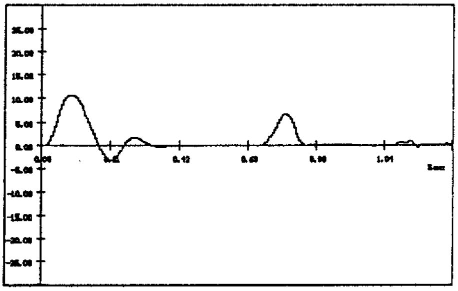

Fig. 5. Response to an external force

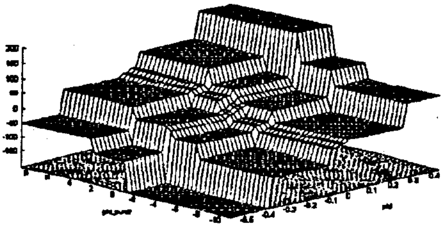

Fig. 6. State space representation of knowledge based controller

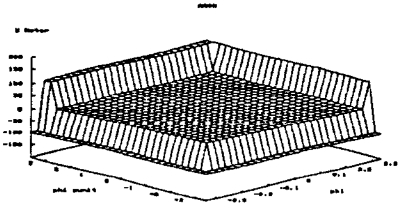

Fig. 7. State space representation of initial self-learning controller

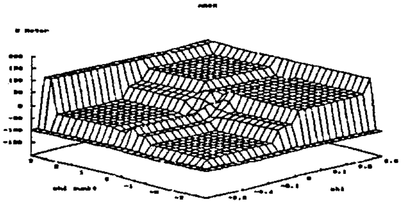

Fig. 8. Self-learning controller after first trial (Fig. 3)

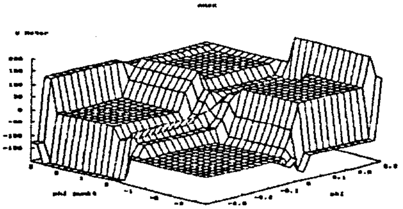

Fig. 9. Self-learning controller after 20 cycles of distortion

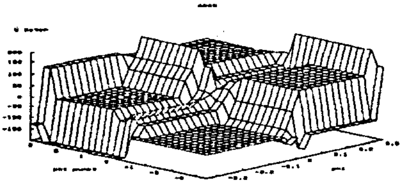

Fig. 10. Self-learning controller after applying an external force 3 times