Fuzzy predictive control based on human reasoning

R. Babuška J. Sousa H.B. Verbruggen

Delft University of Technology, Department of Electrical Engineering, Control Laboratory
Mekelweg 4, P.O. Box 5031, 2600 GA Delft, The Netherlands, tel: +31 15 783371
fax: +31 15 626738, e-mail: r.babuska@et.tudelft.nl

Abstract

Human knowledge is an important source of information for modeling and control of complex dynamic processes. Fuzzy sets proved to be suitable for dealing with subjective uncertainty encountered when incorporating human knowledge in the design of automatic control systems. Besides direct fuzzy control, in which the control law is explicitly described by If–Then rules, the knowledge-based approach can be applied at a higher level for formulating the control objectives and constraints. Appropriate control actions are then found by means of a multistage fuzzy decision making algorithm, using optimization over a finite horizon as in conventional predictive control. Compared to the standard quadratic objective function, the knowledge-based approach gives the designer more freedom in specifying the desired process behavior. By using fuzzy models, the uncertainty arising from the modeling of complex and partially unknown systems can be represented at the same conceptual level as is the uncertainty in the goals and constraints. Finally, a model-based search for an optimal control strategy can be combined with model-free reinforcement techniques inspired by human learning.

Keywords: Predictive control, fuzzy decision making, optimization, learning.

1 Introduction

Complex, nonlinear and partially unknown systems, encountered for instance in chemical process industry, biotechnology or climate control, present big challenges for automatic control. While the conventional linear control techniques often fail or can be applied only locally, human operators are able to control these systems across a wide range of operating conditions. *Knowledge-based* control tries to integrate the knowledge of human operators or process engineers into the controller design. *Fuzzy control*, one of the most popular techniques, has been successfully applied to a large number of consumer products and industrial processes [10, 12]. Most of the applications of fuzzy control use a *descriptive* approach introduced in the seventies by Mamdani [4]. The operator's knowledge is verbalized as a collection of If–Then control rules, that are directly translated into a control algorithm, as schematically depicted in Fig. 1.

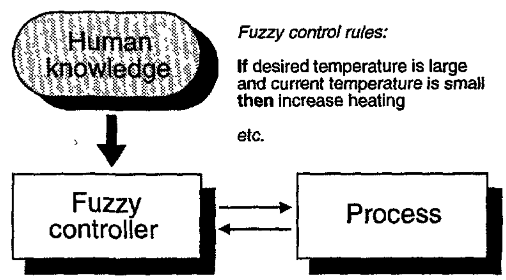

Figure 1: In conventional fuzzy control, operator's knowledge is verbalized as a collection of If–Then control rules.

With this methodology, no explicit model of the process is required, which can significantly reduce the development time if sufficient knowledge is available. If this is not the case, the design must rely on the tuning phase, which may be a tedious and time-consuming trial-and-error procedure. In industrial environments, an on-line experimental tuning is often not acceptable for e.g. safety, economical and environmental reasons. Moreover, it has been observed, that human control skills are sometimes difficult to verbalize since the operator's control strategy can be based on various control principles simultaneously, combining feedforward, feedback, predictive strategies in a complex, time-varying fashion. In that case, an operator may not be able to explain why he or she chooses a particular control action. Experience from knowledge acquisition also shows that opinions of different operators may be very different or even contradicting [5]. Being based on a human control strategy, the descriptive approach is also not suitable for control problems that go beyond the capabilities of the human operator, such as optimization. Process operators usually tend to react quite cautiously and do not want to force the system to the limits of the allowable regions.

In this paper we discuss an alternative approach, where human knowledge is used to specify the control objectives and constraints, not the control protocol itself. A decision making algorithm selects the control actions that best meet the specified criteria, see Fig. 2.

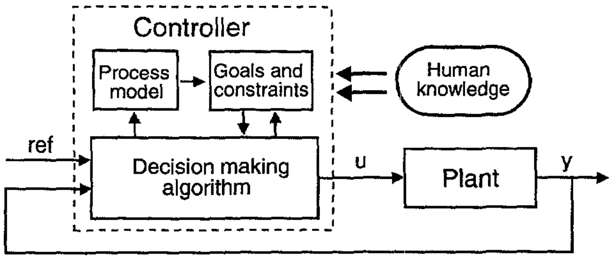

Figure 2: Controller based on objective evaluation and fuzzy decision making.

Since this *prescriptive* approach is closely related to predictive control, we first review the basic concepts of conventional predictive control, than we motivate the approach using fuzzy decision making. Finally, we discuss practical issues related to optimization, its computational complexity and we briefly describe a model-free optimization scheme using reinforcement learning.

2 Predictive control

Model-based predictive control has become an important research area of automatic control theory and it also has been widely applied in industry [7]. Reasons for this success are the ability of MBPC to control multivariable, nonlinear systems under various constraints in an optimal way (with respect to the specified objective function). The working of a MBPC is as follows. A model of the process predicts the process behavior over a specified (finite) prediction horizon, as shown in Fig. 3.

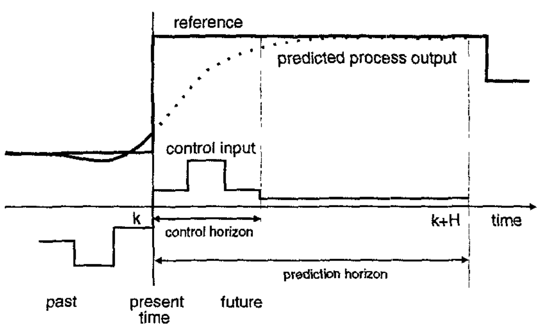

Figure 3: The principle of model based predictive control.

A predictive controller uses an optimization algorithm to calculate a sequence of future controller outputs over a control horizon, such that a specified objective (cost) function is minimized. Most of the objective functions are some modifications of the quadratic form [3]:

$$J = \sum_{i=1}^{H} \alpha_i (w(k+i) - \hat{y}(k+i))^2 + \sum_{i=1}^{H_c} \beta_i \Delta u(k+i-1))^2 \qquad (1)$$

where $\hat{y}$ denotes the predicted process output, w the desired process behavior (reference trajectory) and u the future control signal. H is the prediction horizon and H_c is the control horizon. Vectors α and β determine the weighting of the output error and the control effort with respect to each other and with respect to time. It is important to realize that the cost function is only a suitable mathematical approximation of the control objectives. While its quadratic nature is convenient for finding analytical solutions for linear models, it may be less suitable for achieving the "real" control goals, such as fast rise time, small overshoot, good damping, etc. Though many authors provide tuning rules that attempt to relate the desired performance to the setting of the individual parameters in (1), see e.g. [9], in practice, extra constraint (such as overshoot constraints, etc.) often must be imposed in order to meet the prescribed goals. The quadratic cost (1) minimizes the variance of the process output, which might not be always desirable. In many processes, it is sufficient to keep the controlled variables within certain limits and more accurate control is not desired since it makes the production more expensive. Too a tight control also reduces the information contents of the data that otherwise may be used for adapting the process model. Though optimal w.r.t. (1), the control response may have some undesirable features such as non-minimum phase closed-loop behavior for a minimum phase plant.

From practical reasons, it is desirable to have a direct control over the influence of the individual components of the objective function on the controller performance. It is advantageous if the degree of compensation among the partial goals and among the goals and constraints can be specified by the designer. This additional freedom can be achieved by choosing a different representation of the objective function, e.g. as a combination of fuzzy goals and constraints, as shown in the following section.

3 Objective function with fuzzy goals and fuzzy constraints

The idea of decision making in fuzzy environment was introduced in the beginning of seventies by Bellman and Zadeh [1]. In fuzzy decision making, the goals, constraints and also the systems under control can be fuzzy. An example of a fuzzy goal is "the product concentration should be *about* 80%", where concentration is the controlled variable and the vague expression *about 80%* is represented by a subjectively defined fuzzy set, for instance as shown in Fig. 4. For a crisp measurement x, the degree of satisfaction of a fuzzy goal G is determined by the membership degree of the measurement in the fuzzy set G, $\mu_G(x)$. For a value expressed as a fuzzy set [1] F, the degree of satisfaction of the goal G is computed as a degree of similarity between the two corresponding fuzzy

[1] Also the process values can be fuzzy, consider for instance notions based on human perception that can be expressed as rule-based combinations of measured variables (e.g. comfort may be defined using rules combining temperature and humidity).

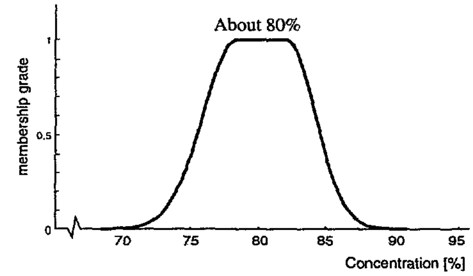

Figure 4: A membership function for the fuzzy goal about 80%.

sets, e.g. as $\max(\mu_F(x) \wedge \mu_G(x))$.

Simple fuzzy goals can be combined in more complex goals or can be refined by adding other conditions, e.g. "the product concentration should be *about* 80% but (and) not *substantially higher* than 82%". Here the goal "not substantially higher than 82%" can represent for instance a temporary restriction that can be added without removing or modifying the original goal. The logical connective *and* and the negation operator *not* are represented as intersection and complement of the fuzzy sets respectively, $G = G_1 \cap \bar{G}_2$, or in terms of membership degrees, $\mu_G(x) = \mu_{G_1}(x) \wedge (1 - \mu_{G_2}(x))$. Fuzzy constraints can be represented in a similar way as fuzzy sets C_i.

Fuzzy decision D for n fuzzy goals $G_1, G_2, \ldots, G_n$ and m fuzzy constraints $C_1, C_2, \ldots, C_m$ is a confluence of these goals and constraints. If we require simultaneous satisfaction of the goals and constraints, we may define D as intersection of the corresponding fuzzy sets:

$$D = G_1 \cap G_2 \cap \ldots \cap G_n \cap C_1 \cap C_2 \cap \ldots \cap C_m$$

or in terms of membership degrees

$$\mu_D(x) = \mu_{G_1}(x) \wedge \mu_{G_2}(x) \wedge \ldots \wedge \mu_{G_n}(x) \wedge \\ \wedge \mu_{C_1}(x) \wedge \mu_{C_2}(x) \wedge \ldots \wedge \mu_{C_m}(x)$$

The *maximizing decision* x^m is any $x \in X$ that maximizes $\mu_D(x)$, i.e.

$$\mu_D(x^m) = \bigvee_{x \in X} \mu_D(x)$$

Optimizing the system's performance over a finite horizon, as in predictive control, corresponds to finding an optimal sequence of decisions in a multistage decision making process. Assume that the system under control is described by a state transition equation

$$x(k+1) = f(x(k), u(k)) \tag{2}$$

Given the current state $x(k)$ we are interested in finding a sequence of actions $u_k, \ldots, u_{H-1}$, corresponding to the maximizing decision. This is a nonlinear optimization problem that can be solved e.g. by dynamic programming.

Example: Predictive control of a container crane. We give a simple example of predictive control with fuzzy goals and constraints of a container crane shown schematically in Fig. 5. A simulation model of a real container

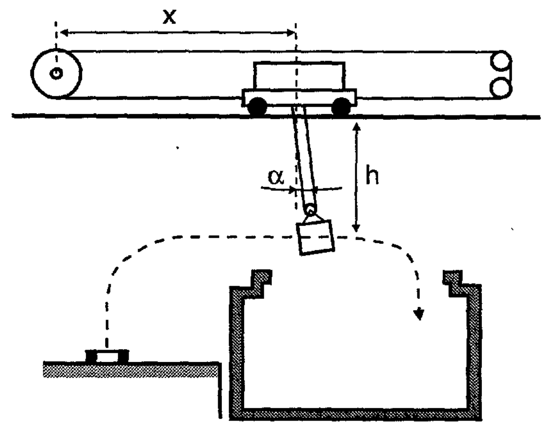

Figure 5: Schematic drawing of a container crane

crane of the port of Kobe was taken over from [8]. The system output variables are the trolley position x, the length of the rope h and its angle α. The torque of the trolley drive and the torque of the hoist motor are the manipulated inputs. In our example we only consider setpoint change of the trolley position from $x = 35$ m to $x = 45$ m on which we compare three different objective functions:

1. Sum of squared errors between the reference and the actual trolley position of x ($J = \sum_{i=k+1}^{k+H}(r(i) - x(i))^2$.
2. Minimization of the overshoot in the trolley position x, using a fuzzy goal with a membership function shown in Fig. 6 a.
3. A combination of 2) with a criterion for minimizing the variance of x around the setpoint, using membership function shown in Fig. 6 b.

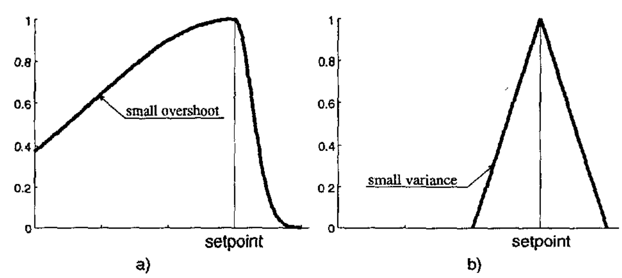

Figure 6: Membership functions for small overshoot (a) and small variance (b).

Note that the variance reduction term in 3) aims at a similar goal as 1). The two goals in 3) are combined using the logical *and* operator, i.e. the minimum of the membership degrees, to represent that both the goals should be satisfied simultaneously. Note that there is no compensation between the criteria involved, as opposed to (1). The prediction and control horizon were both 8 steps.

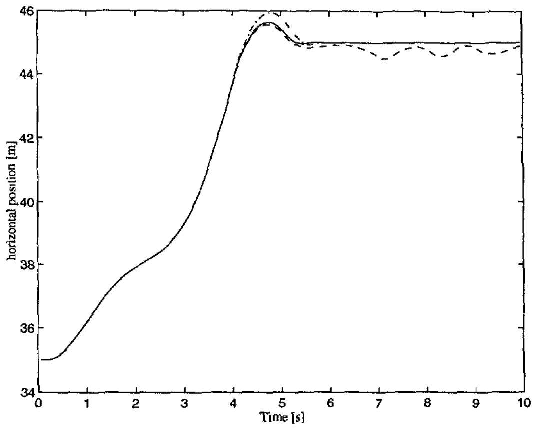

Figure 7: Simulation results for three different objective functions: minimum variance (dash-dotted line), overshoot (dashed line) and combination of the two using the minimum operator (solid line).

The simulation results shown in Fig. 7 clearly show that the oscillation obtained with the overshoot criterion alone can be eliminated by simply adding another criterion for variance minimization.

4 Approximating processes using fuzzy models

For engineering purposes, *mathematical models* are often constructed, based on, for instance, differential or difference equations, derived from physical laws and with parameters estimated via experimental identification. For well-defined systems, these standard mathematical tools lead to good models, even though the modeling process is often very tedious. There are, however, many systems where the underlying physical mechanisms are not known or are so complex that a mathematical model is difficult to obtain and to use. On the other hand, such systems can be described quite simply and with a sufficient accuracy in a verbal way, using fuzzy If–Then rules. Fuzzy sets are used for *partitioning* the continuous domains of the system input and output variables into a small number of overlapping regions labeled with linguistic terms such as LOW, HIGH, etc. A fuzzy model describes the system by establishing relations between the input and output labels. These relations can be expressed in the form of If–Then rules, mapping fuzzy regions from the premise space to other fuzzy regions in the consequent space. For instance, the following rule

If *voltage* is $HIGH$ then *speed* is $HIGH$

relates HIGH voltage on an electric motor to HIGH speed of its rotor. Fuzzy sets for linguistic terms HIGH are defined in their respective domains of voltage and rpm. Fuzzy inference mechanism ensures interpolation between the rules, providing answers to inputs that are not defined in the rule premises. This idea of linguistic fuzzy modeling was introduced in the pioneering papers of Zadeh and applied later on by Mamdani to fuzzy control of dynamical plants, see e.g. [4]. Instead of an explicit description by rules, the mapping can be defined via a *fuzzy relation*. The construction of this so-called fuzzy relational model is based on the theory of fuzzy relations and relational equations, see e.g. [6]. The output fuzzy set Y is computed from the input fuzzy set X via relational composition

$$Y = X \circ R$$

A dynamic system, such as (2) can be described as a composition of the input fuzzy set $U(k)$, state fuzzy set $X(k)$ and a relation R describing the system

$$X(k+1) = U(k) \circ X(k) \circ R$$

Fuzzy models have several useful properties. First, their belong to the class of general function approximators, e.g. a fuzzy model can approximate a smooth function to any degree of accuracy, as shown by Wang [13] (among others). Secondly, different kinds of information can be integrated for building fuzzy models, such as knowledge expressed as If–Then rules and numerical data (process measurements). Finally, the mathematical framework for representation of fuzzy models is convenient for analytical manipulations, such as analysis of the model, its inversion, etc. Fuzzy models also can be seamlessly integrated in a predictive control system based on the fuzzy decision making approach, as shown in the previous section.

5 Optimization based on reinforcement learning

The previously described approach requires a reliable model of the plant and if a significant model–plant mismatch appears, the controller performance rapidly degrades. Modeling and identification of complex systems is a difficult, time-consuming and expensive task, resulting in the fact that most of the design effort (sometimes as much as 80%) is spent on developing a good process model [7]. In many cases, an accurate model cannot be obtained which places the use of MBPC out of question. Therefore, techniques for optimizing the control policy without an explicit model of the plant are desirable.

Here again, it is useful to take inspiration from the way humans adapt their behavior in a particular environment without knowing an accurate model of that environment. Many learning tasks (think for instance of learning to play tennis) consist of repeated trials (attempts to hit the ball) followed by a reward (a nice shot) or punishment (picking the ball from the ground). Each trial can be a dynamic sequence of actions (run, taking a stand, hitting the ball) while the feedback (reinforcement) comes only at the end. Therefore, a large number of trials may be needed to figure out which particular actions were correct and which must be adapted.

A family of algorithms inspired by human and animal learning is known as *reinforcement learning*. Reinforcement learning assumes that there is no direct evaluation of the quality of the selected control action. Instead, an indirect evaluation is received, possibly after a sequence of control actions, in terms of (dis)satisfaction of the control objectives and/or violations of constraints. The reinforcement system learns how to predict the outcome of each particular action, using techniques like *temporal difference* [11] or *Q-learning* [14]. This prediction is used to adapt parameters of a suitable general function approximator (e.g. a neural network) to iteratively approximate the optimal control policy. This approach differs from other optimization techniques in two respects: i) no model of the system is needed, ii) the optimization is not done at once, it is distributed over a large number of small steps that gradually approximate the optimal policy.

Most of the approaches based on neural networks do not employ any prior information and learn the control policy from scratch. It is well known that *prior knowledge* can speed up the learning process (in our example, it might be useful to know how to grip the racket and how to stand for backhand and forehand). In more serious control tasks, prior knowledge is essential for a correct functioning of the system at the very start of the learning process (think of driving a car without knowing the function of the pedals, or stochastically exploring the effect of individual control variables in an unstable chemical process). The ability of fuzzy rule-based system to use prior knowledge on one hand and to approximate nonlinear functions on the other hand is advantageous for combining fuzzy models with reinforcement learning [2]. By using reinforcement learning, possibly approximate and imprecise prior knowledge expressed in terms of fuzzy rules can be refined on line, during the control process.

In order to briefly explain the principle of reinforcement learning, let us assume (without loss of generality) that the system to be controlled is represented by a state transition function (2). When the system is fuzzy, f is a rule base. The goal is to learn (or adapt) an associative mapping $\pi : X \rightarrow U$, by maximizing a (scalar) evaluation of the performance (reinforcement). Here π is so called *policy function* and is equivalent to a controller, that for a particular state $x \in X$ computes a control action $u \in U$, see Fig. 8. When controlling dynamic systems, the reinforcement evaluation is usually available only after a sequence of state–action pairs $\{(x(k), u(k))\}$.
For discrete control actions $U = \{u_1, u_2, \dots, u_m\}$, the policy π can be formed as a composition of two functions: a function approximator $g : X \rightarrow R^m$ that assigns

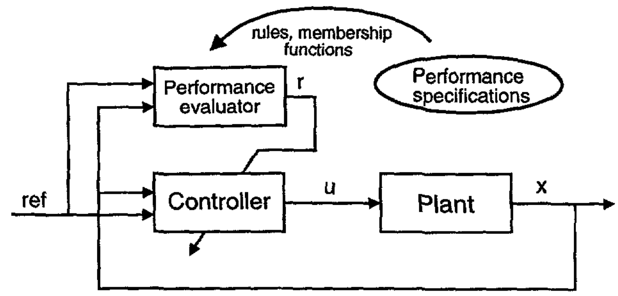

Figure 8: Control scheme with reinforcement learning.

each u_i a real value representing its merit (cf. degrees of fulfillment in fuzzy inference) and a fixed function $M : R^m \rightarrow A$ that selects a particular control action (cf. the aggregation/defuzzification block). Often, M is a maximum selector combined with a stochastic action modifier that ensures proper exploration of the search space.

The basic principle of reinforcement learning can be explained as follows. The goal is to maximize the total reinforcement over time, which can be expressed as a discounted sum of the immediate payoffs $r(x, u)$ computed by the performance evaluator (see Fig. 8). The sum, so called *value function* $V : X \rightarrow R$, is defined as:

$$V(x) = \lim_{N \rightarrow \infty} E\{\sum_{k=0}^{N-1} \gamma^k r(x(k), \pi(x_k)) | x_0 = x\} \quad (3)$$

where the constant $\gamma \in [0, 1)$ is a discount rate. $V(x)$ can be approximated, using the *temporal difference* operator

$$\Delta(x) = r(x(k), u(k)) + \gamma V(x(k+1)) - V(x(k))$$

which is a difference of predictions of $V(x)$ at two consecutive time steps. The estimate $\hat{V}(x)$ is updated, using $\Delta(x)$

$$\hat{V}^{k+1}(x) = \hat{V}^k(x) + \beta \Delta(x)$$

where β is a small positive constant. Finally, the learning rule for the merit function $g(x)$ is:

$$g^{k+1}(x) = g^k(x) + \alpha(\rho(x(k), u(k)) - V(x(k))) \quad (4)$$

where α is a small positive constant and $\rho(x(k), u(k))$ is the expected total reward obtained if u is applied to the system at state $x(k)$ and then policy π is followed. This estimate is not available, but it can be approximated as:

$$\rho(x(k), u(k)) \approx r(x(k), u(k)) + \gamma \hat{V}(x(k+1)) \quad (5)$$

Using (5) in (4) gives the update law for $g(x)$:

$$g^{k+1}(x) = g^k(x) + \alpha \Delta(x)$$

Obviously, the reinforcement learning (RL) methods solve the same optimization problem as the dynamic programming (DP) methods. The difference is that the DP methods are off-line, while RL techniques try to learn the optimal policy on-line, concurrently with the system operation. The DP methods search the entire space $X \times U$, while the RL methods operate on the a subset of states $\tilde{X}$ that occur during the system operation. Since $\tilde{X}$ may be significantly smaller than X, RL methods do not suffer from the curse of dimensionality as much as the DP methods. RL methods also do not require a system model and can be extended to continuous spaces of u.

6 Concluding remarks

In this paper we discussed some links between conventional predictive control, multistage fuzzy decision making and reinforcement learning. These three approaches, originating from different roots, nicely complement each other and in combination may be applied to control of complex dynamic systems that are difficult to deal with using standard methods. Let us summarize the main features of the proposed approach:

- The control objective function can be specified as any suitable combination of the terms representing degree of satisfaction of the goals and constraints, ranging from a simple conjunction to a problem-specific rule base. Such a rule base, for instance, can capture context-dependent importance of the control goals and constraints or different ways of their aggregation [15].
- With such an objective function, each control action can be explained in terms of partial decisions and degrees of satisfaction of the individual criteria, which is suitable for tuning and monitoring purposes.
- Subjective uncertainty can be easily incorporated in the specification of goals and constraints.
- The process itself can be represented as a fuzzy rule-based or a relational model. In this way, also ill-defined, partially unknown or highly nonlinear systems can be modelled in a transparent way.

For the higher flexibility one has to pay by increased computational costs, since the optimization problem is in general nonlinear and often non-convex. Reinforcement learning is considered as an on-line optimization method which can simultaneously add adaptivity features to the controller. Potential applications of the proposed methodology include such systems where humans have been or are part of the control structure or the environment, such as climate control, telemanipulation, crane control, etc.

References

[1] R. E. Bellman and Lotfi A. Zadeh. Decision making in a fuzzy environment. *Management Science*, 17(4):141–164, 1970.

[2] H.R. Berenji and P. Khedar. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Transactions on Neural Networks*, 3:724–740, 1992.

[3] D. Clarke. Advances in model-based predictive control. In D. Clarke, editor, *Advances in Models-Based Predictive Control*, pages 3–21. Oxford University Press, Oxford, 1994.

[4] E.H. Mamdani. Applications of fuzzy algorithms for control of simple dynamic plant. In *Proceedings IEE*, number 121, pages 1585–1588, 1974.

[5] Sander F.J. Ouwerkerk. *Development and Implementation of a Fuzzy Logic Controller for a Chemical Process*. Master's dissertation, Shell Nederland B.V., Delft University of Technology, Delft, The Netherlands, February 1995. (in Dutch).

[6] W. Pedrycz. An identification algorithm in fuzzy relational systems. *Fuzzy Sets and Systems*, 13:153–167, 1984.

[7] J. Richalet. Industrial applications of model based predictive control. *Automatica*, 29:1251–1274, 1993.

[8] Y. Sakawa and Y. Shindo. Optimal control of container cranes. *Automatica*, 18(3):257–266, 1982.

[9] A.R.M. Soeterboek. *Predictive Control - A Unified Approach*. PhD dissertation, Delft University of Technology, Delft, The Netherlands, November 1990.

[10] M. Sugeno, editor. *Industrial applications of fuzzy control*. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, The Netherlands, 1985.

[11] R.S. Sutton. Learning to predict by the method of temporal differences. *Machine learning*, 3:9–44, 1988.

[12] T. Terano, K. Asai, and M. Sugeno. *Applied Fuzzy Systems*. Academic Press, Inc., Boston, 1994.

[13] L.-X. Wang. Fuzzy systems are universal approximators. In *Proc. IEEE Int. Conf. on Fuzzy Systems 1992*, pages 1163–1170, San Diego, USA, March 1992.

[14] C.J.C.H. Watkins and P. Dayan. Technical note: Q-learning. *Machine learning*, 8:279–292, 1992.

[15] S. Yasonobu and S. Miymoto. Automatic train operation system by predictive fuzzy control. In M. Sugeno, editor, *Industrial Applications of Fuzzy Control*, pages 1–18. North-Holland, 1985.