Applying Bayesian Networks for Intelligent Adaptable Printing Systems

Arjen Hommersom* , Peter J.F. Lucas*, René Waarsing†, and Pieter Koopman*
*Institute for Computing and Information Sciences
University of Nijmegen, The Netherlands
Email: {arjenh,peterl,pieter}@cs.ru.nl
†Océ Technologies BV
Venlo, The Netherlands
Email: rene.waarsing@oce.com

***Abstract*—Bayesian networks are around more than twenty years by now. During the past decade they became quite popular in the scientific community. Researchers from application areas like psychology, biomedicine and finance have applied these techniques successfully. In the area of control engineering however, little progress has been made in the application of Bayesian networks. We believe that these techniques are useful for systems that dynamically adapt themselves at runtime to a changing environment, which is usually uncertain. Moreover, there is uncertainty about the underlying physical model of the system, which poses a problem for modelling the system. In contrast, using a Bayesian network the needed model can be learned, or tuned, from data. In this paper we demonstrate the usefulness of Bayesian networks for control by case studies in the area of adaptable printing systems and compare the approach with a classic PID controller. We show that it is possible to design adaptive systems using Bayesian networks learned from data.**

I. Introduction

Many complex systems such as printers are required to make dynamic in-product trade-offs between various qualities of operation at the system level, which can be viewed as the capability to adapt. While many definitions of adaptability have appeared in literature (see [1] for a summary), we here define *adaptability* to be such system-wide trade-offs. In printing systems, system-wide qualities include the power division, the speed of printing, the power consumption, etc. Such trade-offs heavily depend on the system's environment, e.g., humidity, temperature, available power, etc. Failure to adapt adequately to the environment might result in faults or suboptimal behaviour.

The area of adaptive control has a long tradition of over 50 years. Several approaches in this field exist. First, model-reference adaptive control (MRAC) uses a reference model that reflects the desired behaviour of the system. On the basis of the observed output and of the reference model, the system is tuned. The second type of adaptive controllers are so called self-tuning controllers (STC), which estimate the correct parameters of the system based on observations and tunes the control accordingly. In the last few decades, also techniques from the area of artificial intelligence (AI), such as rule-based systems, fuzzy logic, neural networks, evolutionary algorithms, etc. have been used in order to determine optimal values for control parameters (see e.g. [2]).

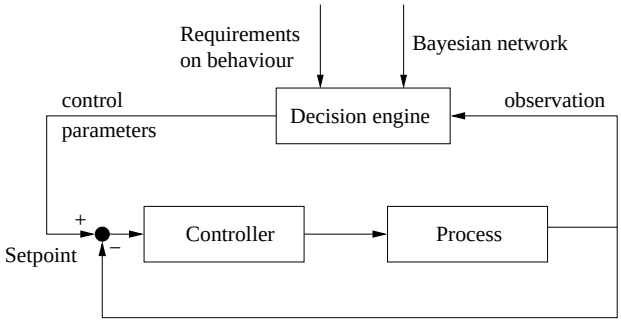

Fig. 1. Block diagram of an adaptive controller using a Bayesian network.

The problem of adaptability as defined above has two typical characteristics. First, decisions are typically required at a low frequency, i.e., it is not necessary and not even desirable to change the speed or energy usage many times per second. Second, there is a lot of uncertainty involved when making decisions, in particular about the environment, the state of the machine, but also about the exact dynamics of the system. Complex systems usually cannot be modelled accurately, whereas adaptability requires one to make system-wide, complex, decisions. In order to deal with this uncertainty, techniques where probability distributions are learned from available data seem therefore appropriate. In this paper, we explore the use of Bayesian networks [3] to tune setpoints of local controllers of the system. The block diagram in Fig. 1 offers an overview of this approach.

One advantage of Bayesian networks is that they contain a qualitative part, which can be constructed using expert knowledge, normally yielding an understandable, white-box model. Moreover, the quantitative parameters of a Bayesian network can be learned from data. Other AI learning techniques, such as neural networks, resist providing insight into why the machine changes its behaviour, as they are black-box models. Furthermore, rules—possibly fuzzy—are difficult to obtain and require extensive testing in order to check whether they handle all the relevant situations.

The purpose of the present paper is to convey some of our experience in building Bayesian-network based controllers in

the area of adaptive printing systems, which can be looked upon as special stochastic controllers. In our view, as systems get more and more complex, the embedded software will need to be equipped with such reasoning capabilities for making sound decisions. The paper is organised as follows. In the next section, we will introduce the necessary preliminaries with respect to Bayesian networks. In Section III, we will look at a specific case study where we would like to estimate the optimal setpoint under uncertainty. This example shows that some of the logic that might be needed in a rule-based system is implicitly encoded in the probability distribution. Another case is considered in Section IV; here the goal is to optimise the velocity of the engine. Both cases are compared to a traditional controller. In Section V the results obtained are compared to related approaches.

II. Preliminaries

A *Bayesian network* $B = (G, P)$ consists of a directed acyclic graph $G = (V, E)$, where V is a set of vertices and $E \subseteq V \times V$ is a set of directed arcs; with the probability distribution P is associated a set X of random variables that correspond one-to-one to the vertices of G, i.e., each vertex v corresponds exactly to one random variable X_v and vice versa. As the joint probability distribution P of the set of random variables X is factored in accordance to the structure of the graph G:

$$P(X) = \prod_{v \in V} P(X_v \mid X_{\pi(v)}),$$

where $\pi(v)$ is the set of parents of v, P can also be defined as a family of local conditional probability distributions $P(X_v \mid X_{\pi(v)})$, for each vertex $v \in V$. Bayesian networks can encode various probability distributions. Most often the variables are either all discrete or all continuous. Hybrid Bayesian networks, however, contain both discrete and continuous conditional probability distributions. A commonly used type of hybrid Bayesian network is the conditional linear Gaussian model (see [4], [5]). Efficient exact and approximate algorithms have been developed to infer probabilities from such networks (e.g., [6], [7], [8], and [9]). Also important in the context of embedded systems is the fact that real-time inference can be done using Bayesian networks, i.e., produce an approximate probability at any time (cf. [10] for a comprehensive overview).

A Bayesian network can be constructed with the help of one or more domain experts. However, building Bayesian networks using expert knowledge, although by now known to be feasible for some domains, can be very tedious and time consuming. Learning a Bayesian network from data is also possible, a task which can be separated into two subtasks: (1) structure learning, i.e., identifying the topology of the network, and (2) parameter learning, i.e., determining the associated joint probability distribution, P, for a given network topology. In this paper, we employ parameter learning. This is typically done by computing the maximum likelihood estimates of the parameters, i.e., the conditional probability distributions, associated to the networks structure given data [11].

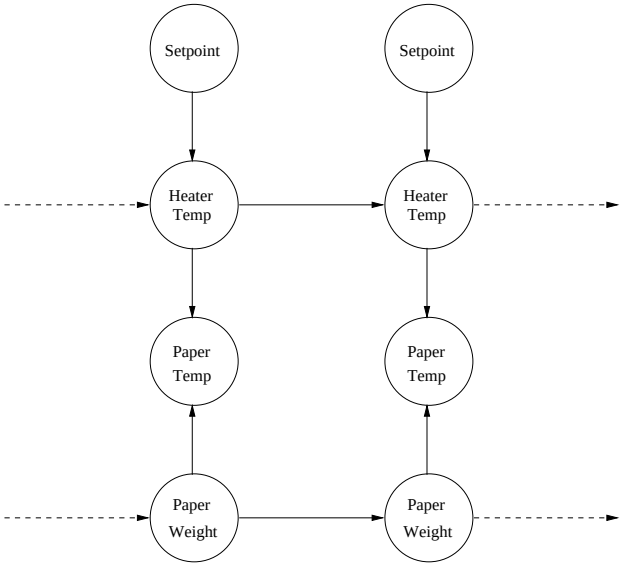

Fig. 2. Simplified Bayesian network of the print domain.

Temporal Bayesian networks are Bayesian network where the vertices of the graph are indexed with (discrete) time. All vertices with the same time index form a so-called *time slice*. Each time slice consists of a static Bayesian network and the time slices are linked to represent the relationships between states in time. If the structure and parameters of the static Bayesian network are the same at every time slice (with the exception of the first), one speaks of a *dynamic* Bayesian network, as such networks can be unrolled (cf. [12] for an overview).

III. Setpoint estimation

A. Description of the problem

For the type of printing system under consideration, various temperatures during the printing process play an important role. Low-level controllers make sure that the measurable temperatures are kept on setpoint. Due to design issues and considerations with respect to the cost price, it is not possible to place sensors at all places of interest; therefore, estimations have to be made.

In this section, we use a Bayesian network to estimate the appropriate setpoint for a heating component with the goal to influence the paper temperature when we can only measure the temperature of media (paper) that has passed this heating component. The temperature of the paper is influenced by uncertain aspects, such as the environmental temperature, the speed, the humidity of the paper, and the type of paper. In this case we focus on the latter aspect and assume that the other aspects are constant.

B. Experimental setup

The qualitative structure of the domain was elicited from the domain experts. For the purpose of this paper, we focus on certain relevant parts of the complete network dealing with the specific problem of determining the correct setpoint of the

heater. The structure of the domain consisting of two time slices is presented in Fig. 2.

The associated random variables for this network have been modelled as discrete variables by discretising the values to typical values that are used in the simulation. The setpoint variables have a domain size of 12; media temperature has a domain size of 16 and we consider three paper types: 80, 120, and 160 g/m^2 paper.

In order to acquire data and to test the system, a physical model of the system was created using Simulink [13]. The data that was generated was used to learn the conditional distributions of the model by calculating the parameters associated to the qualitative structure of the Bayesian network.

The adaptive control was implemented by a low-level PID controller (see e.g., [14]) that controls the temperature of the heater and a Bayesian network to manipulate the setpoint of this controller.

C. Case 1: Keep paper temperature on setpoint

First, we use a Bayesian network to choose the next setpoint such that the temperature of the paper will be at a setpoint T_{set} based on observations of the paper temperature and the setpoint at time t. Let $\text{OBS}_t = \{\text{PaperTemp}_t = T, \text{Setpoint}_t = \text{SP}\}$. We then calculate:

$$\text{SP}^* = \underset{\text{SP}' \in \text{Setpoint}}{\operatorname{argmax}} \; P(\text{PaperTemp}_{t+1} = T_{\text{set}} \mid \text{OBS}_t, \text{Setpoint}_{t+1} = \text{SP}')$$

and adapt the setpoint of the heater controller to SP^*. The Bayesian network is simplified, in particular by forgetting about the history, except for the immediate history when making decisions (first-order Markov assumption). Due to this simplification, sometimes the interpretation of the measurements can be misleading. There are several solutions to this problem. For example, we may extend the model to incorporate additional evidence of earlier states, or we may sample less, i.e., by waiting to the system returns to a steady situation. One simple heuristic that proved to be successful in this situation is avoid making decisions when the interpretation is highly uncertain, e.g., when:

$$P(\text{PaperWeight}_t = w \mid \text{OBS}_t) < k$$

for all paperweights w and where k is some tuning constant less than 1. Of course, such a controller can also be implemented well using a standard PID controller that controls directly based on the measurement of the paper temperature. The results of such a PID controller is compared to the Bayesian network approach (with $k = 0.9$) and is presented in Fig. 3. The PID controller seems smoother, which is most likely due to the fact that we have discrete value for the setpoint in the adaptive controlling setting. However, for the most part, the behaviour of the adaptive controller is similar to the PID control.

For this control task, Bayesian networks do not provide much benefit over PID controllers, and we can only hope not to do worse than these standard, well-understood, PID controllers. We are therefore aiming at more complex controllers, where traditional control theory starts to become more difficult. One example is discussed in the next section.

D. Case 2: Avoid faulty temperatures

As mentioned earlier, in order to get high quality prints, it is of importance to have a lower threshold for some temperatures. Fig. 3 shows that if we try to keep the paper at the T_{set} temperature, temperatures may drop below this value when the media changes. This could lead to a system fault. One solution is to put the setpoint at a higher temperature which provides a buffer for the media changes; however, if it is unnecessarily high, energy is lost and it may also cause problems at other parts of the printing process.

The advantage of Bayesian networks is that various probabilistic constraints can be put on the control signal. In this case, we are interested in the lowest temperature that ensures that we avoid dropping below T_{set}. Formally, to decide on the next setpoint, we calculate the minimal SP' such that

$$P(\text{PaperTemp}_{t+1} < T_{\text{set}} \mid \text{OBS}_t, \text{Setpoint}_{t+1} = \text{SP}') < \epsilon$$

i.e., the probability that the resulting temperature will be lower than T_{set} will be less than some threshold ϵ. The result can be found in Fig. 4 (with $\epsilon = 0.01$). What is interesting here is that the heater temperature is relatively high when the paper weight is lower. This is because the system anticipates on paper that might arrive with a high paper weight as this high paper weight causes a sudden large drop in temperature. This type of logic could be modelled by any system; however, it is interesting to see here that this is implicit in the probability distribution that has been learned from data.

IV. Dynamic speed adjustment

A. Description of the problem

The productivity of printers is limited to the amount of power available, in particular in environments which depend on weak mains. If there is insufficient power available, then temperature setpoints cannot be reached, which causes bad print quality. To overcome this problem, it is either possible to decide to always print at lower speeds or to adapt to the available power dynamically. In the section, we explore the latter option by a dynamic speed adjustment using a Bayesian network.

B. Modelling

The structure of the fragment of the model at each time slice is shown in Fig. 5. The requested power available is an observable variable that depends on low-level controllers that aim at maintaining the right setpoint for reaching a good print quality. The error variable models the deviation of the actual temperature from the ideal temperature, which can be established in a laboratory situation, but not during run-time. If this exceeds a certain threshold, then the print quality will be below a norm that has been determined by the printer manufacturer.

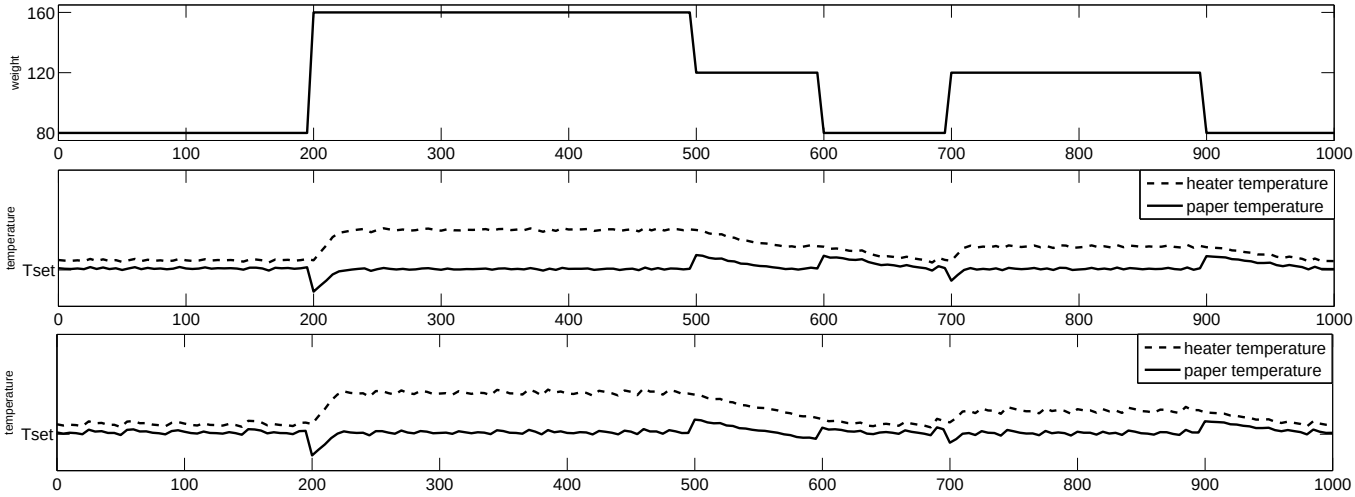

Fig. 3. Top figure: weight of paper that is being printed on, which changes dynamically. Middle figure: paper temperature controlled by a PID controller. Bottom figure: paper temperature controlled by an adaptive controller using a Bayesian network.

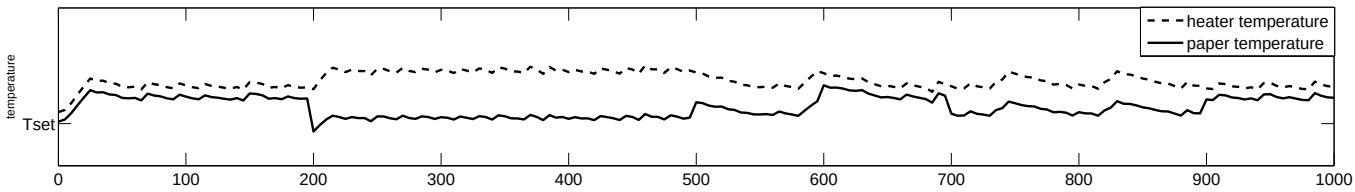

Fig. 4. Faults are avoided using an adaptive controller. Paper weight changes are the same as in Fig. 3.

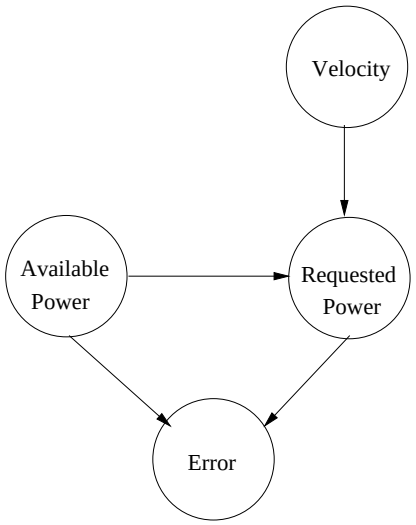

Fig. 5. Structure of the Bayesian network of each time slice.

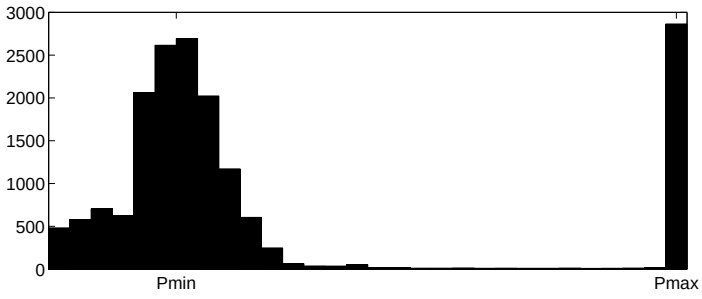

Fig. 6. Distribution of requested power.

Both velocity and available power influence the power that is or can be requested by the low-level controllers. Furthermore, the combination of the available power and the requested power is a good predictor of the error according to the domain experts.

For our experiments we again use two time slices with the interconnections between the available power – which models that the power supply on different time slices is not independent – and requested power, which models the state of the machine that influences the requested power.

In order to choose the family of distributions, we can consider to model the variables as Gaussian variables. This is reasonable as most variables are normally distributed, except for the available power (see Fig. 6). Fitting a Gaussian distribution to such a distribution will typically lead to insufficient performance. However, it can be interpreted as a mixture of two Gaussian distribution, one with mean P_{low}(Watt) and one with mean P_{high}(Watt) with a small variance. Such a distribution can be modelled using a hybrid network as follows. The network is augmented with an additional (binary) parent node S with values 'high' and 'low' for the requested power variable. For

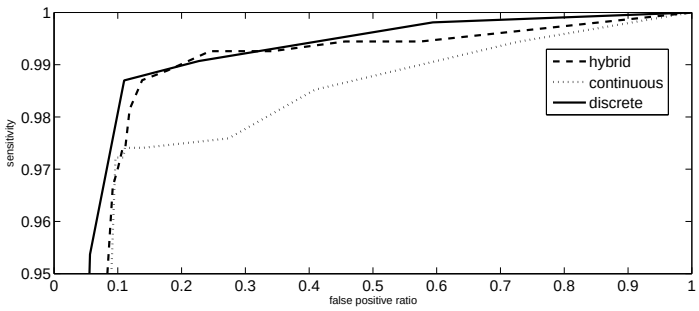

Fig. 7. ROC curves of the three Bayesian networks. The hybrid and discrete versions show the best classification performance.

both states of this node, a Gaussian model is associated to this variable. The marginal distribution of requested power is obtained by basic probability theory as

$$P(\mathrm{P_{req}}) = \sum_S P(\mathrm{P_{req}} \mid S) P(S).$$

C. Classification

One of the main reasoning tasks of the network is to estimate the error, i.e., the deviation from the ideal temperature, given a certain velocity and a certain observations. We could consider this a classification performance, i.e., the print quality is *bad* or *good*. This provides means to compare different models and see how well it performs at distinguishing between these two possibilities. A standard way to visualise and quantify this is by means of a Receiver Operating Characteristic (ROC) curve, which shows the relation between the false positive ratio and the true positive ratio (sensitivity). The area under the curve is a measure for its classification performance.

We have compared three models, i.e., a discrete model, a fully continuous model and a hybrid model for modelling the distribution of the requested power with two Gaussians. The classification performance is outlined in Fig. 7. As expected, the fully continuous model performs worse, whereas the hybrid and discrete show a similar trend. The advantage of the discrete version is that the probability distribution can easily be inspected and it has no underlying assumptions about the distribution, which makes it easier to use in practice. The hybrid version however allows for more efficient computation as we need a large number of discrete to describe the conditional distributions. For this reason, we have used the hybrid version in the in following experiments.

D. Simulation of the Bayesian controller

As the error information is not available during runtime, the marginal probability distribution of the error in the next time slice is computed using the information about the power available and power requested. This error is a Gaussian random variable with mean μ and standard deviation σ. Given a maximum error that we allow, denoted by $E_{\max}$, we pick the highest velocity v such that the marginal probability distribution of $P(\mathrm{Error}_{t+1})$ is such that $\mu + k\sigma < E_{\max}$, where k is a constant. Different values of k correspond to different points on the ROC curve as depicted in Fig. 7. For a Gaussian variable, more than 99.73% of the real value of the error will be within three standard deviations of the mean, so for example $k = 3$ would imply that $P(\mathrm{Error}_{t+1} < E_{\max}) > 99.87\%$. However, the sensitivity, i.e., the chance that the paper is warm enough, can be chosen arbitrarily, by also increasing the false positive ratio, i.e., by reducing the overall productivity.

In order to evaluate the approach, we compared the productivity of the resulting network with a rule-based approach that incorporates some heuristics for choosing the right velocity. The productivity is defined here simply as $\int_0^\tau v(t)dt$, where τ is the simulation time.

In order to smooth the signal that the network produces, we employ a a FIR (Finite Impulse Response) filter in which we average the decisions of the last 10 seconds. The resulting behaviour was simulated and is presented in Fig. 8 (with $k = 3$). Compared to the rule-based approach, we improve roughly 9% in productivity while keeping the error within an acceptable range. While it could certainly be the case that the rules could be improved and optimised, again, the point is that the logic underlying the controller does not have to be designed. What is required is a qualitative model, data, and a probabilistic criterion that can be inferred.

V. Discussion and Conclusions

So far, adaptive controllers based on explicit Bayesian networks have not been extensively investigated. The most closely related work is by Deventer [15], who investigates the use of dynamic Bayesian networks for controlling linear and nonlinear systems. The main difference with his work is that he estimates parameters from a Bayesian network using a given model of the system. In contrast, we aim at using models that were learned from data. Such data can come from measurements during design time or during run-time of the system.

Bayesian inference is well-known for trying to infer a hidden state in a dynamic model. Typical applications are filtering, i.e., trying to infer the current hidden state given the observations in the past and smoothing where past states are inferred. For example, the Kalman filter [16] is well-known in stochastic control theory (see e.g., [17] for an overview) and is a special case of a dynamic Bayesian networks, where the model is the linear Gaussian variant of a hidden Markov model, i.e., it describes a Markov process with noise parameters on the input and output variables. Non-linear variants, such as the extended Kalman filter or the unscented Kalman filter (see e.g., [18]) are approximate inference algorithms for non-linear Gaussian models by linearisation of the model. More recently, particle filters [19], also known as sequential Monte Carlo methods, have been proposed as an alternative, which relies on sampling to approximate the posterior distribution.

The difference with these filtering approaches is that for Bayesian networks there is an underlying domain model which is understandable. As Bayesian networks are general formalisms, they could also be used or re-used for diagnostic purposes, where it is typically required that a diagnosis

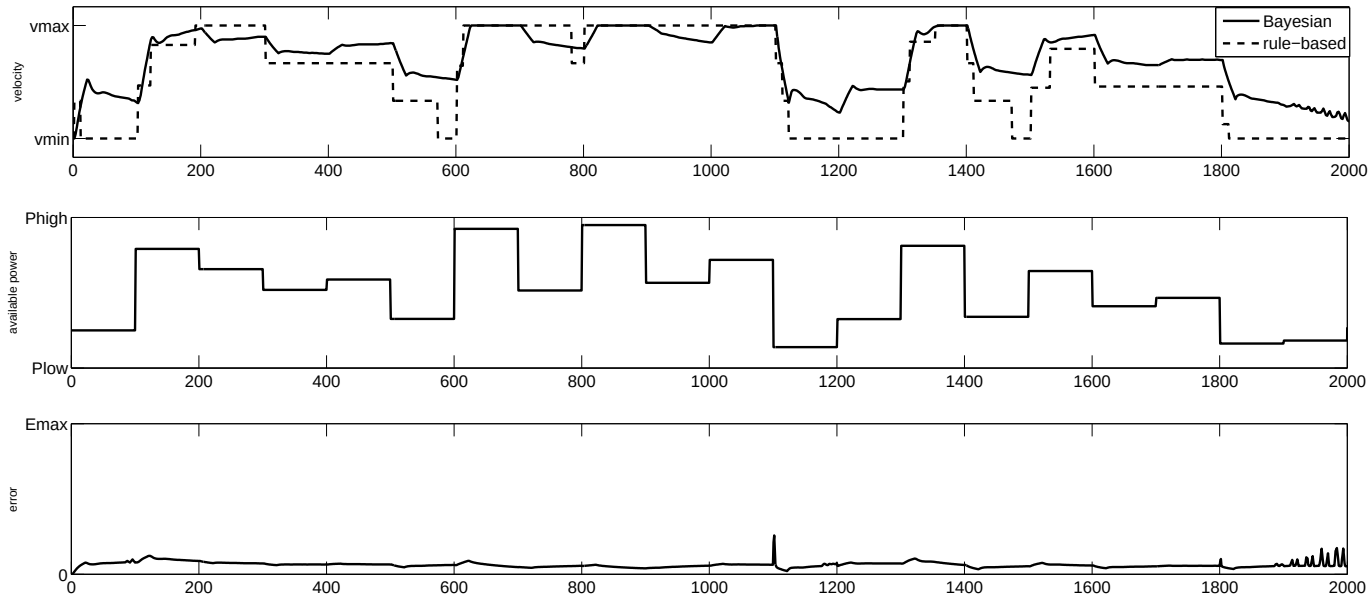

Fig. 8. In the centre figure, the available power is plotted, which is fluctuating. at the top, we compare the velocity of the engine which is controlled by a rule-based system and by a Bayesian network. Below, we present the error that the controller based on the Bayesian network yields, which is within the required limits.

can be represented in a human-understandable way so that proper action can be taken. Furthermore, it is well-known that the structure of the graphical part of a Bayesian network facilitates the assessment of probabilities, even to the extent that reliable probabilistic information can be obtained from experts (see [20] in the medical domain). One other advantage compared to black-box models is that the modelled probability distribution can be exploited for decision theory. This is particularly important if one wants to make real trade-offs such as between productivity and energy consumption. This is another direction that will be explored in the future.

With respect to the choice of the underlying distribution that is associated to a Bayesian network, several choices can be made. For exact inference, conditional linear Gaussian models are the standard way to deal with continuous variables in this area as exact inference can be used; however, they are restricted to modelling linear systems. Discrete distributions can then be used and have been successfully applied in, for example, medicine. Another option is to model the distribution on the basis of a mixture of truncated exponentials (MTE) [21], which allow one to model non-linear systems and, furthermore, allows for exact inference of required probabilities. This is a direction which we will explore in the near future. Of course, in particle filters, sampling methods can be employed for approximating the posterior from a wide range of non-linear distributions.

Bayesian networks have drawn attention in many different research areas, such as AI, mathematics and statistics. In this paper, we have explored the use of Bayesian networks for designing an adaptable printing systems. We have shown that the approach is feasible and can help to design an intelligent system. We believe that these techniques can have a wide application in the engineering sciences in particular for control and fault detection. The latter has been investigated before, e.g., [22], in which Bayesian networks were applied for both consistency-based as well as abductive diagnosis. Results of this paper provide evidence that explicit Bayesian networks can also be useful for the development of adaptive control systems.

ACKNOWLEDGEMENTS

This work has been carried out as part of the OCTOPUS project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Embedded Systems Institute program. We would like to thank the anonymous reviewers and the members of the OCTOPUS project for their helpful suggestions and feedback. We also thank Marcel van Gerven for making his Bayesian network toolbox available.

REFERENCES

[1] M. Chmarra, L. Arts, and T. Tomiyama, "Towards adaptable architecture," in *ASME 2008 International Design Engineering Technical Conferences*, 2008, dETC2008-49971.

[2] J. Farrell and M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*, ser. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Wiley-Interscience, 2006.

[3] J. Pearl, *Probabilistic Reasoning in Inteligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[4] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter, *Probabilistic Networks and Expert Systems*. Springer, 1999.
[5] S. Lauritzen, "Propagation of probabilities, means and variances in mixed graphical association models," *J. Am. Stat. Assoc.*, vol. 87, pp. 1098–1108, 1992.
[6] S. Lauritzen and D. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society*, vol. 50, pp. 157–224, 1988.
[7] G. Casella and C. Robert, *Monte Carlo Statistical Methods*. Springer-Verlag, 1999.
[8] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.
[9] U. Lerner and R. Parr, "Inference in hybrid networks: theoretical limits and practical algorithms," in *Uncertainty in Artificial Intelligence*, J. Breese and D. Koller, Eds., vol. 17. San Francisco, CA: Morgan Kaufmann, 2001, pp. 310–318.
[10] H. Guo and W. Hsu, "A survey of algorithms for real-time Bayesian network inference," in *AAAI/KDD/UAI02 Joint Workshop on Real-Time Decision Support and Diagnosis Systems*, A. Darwiche and N. Friedman, Eds., Edmonton, Canada, 2002.
[11] S. Lauritzen, "The EM algorithm for graphical association models with missing data," *Computational Statistics and Analysis*, vol. 19, pp. 191–201, 1995.
[12] K. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. dissertation, UC Berkeley, 2002.
[13] MATLAB, "The MathWorks Inc." 2008, version R2008A.
[14] K. Ogata, *Modern Control Engineering*, 4th ed. Prentice-Hall, Inc, 2002.
[15] R. Deventer, "Modeling and control of static and dynamic systems with Bayesian networks," Ph.D. dissertation, University Erlangen-Nürnberg, Chair for Pattern recognition, 2004.
[16] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
[17] K. Åström, *Introduction to Stochastic Control Theory*. Academic Press, 1970.
[18] P. Maybeck, *Stochastic models, estimation, and control*. Academic Press, 1979.
[19] J. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.
[20] P. Lucas, H. Boot, and B. Taal, "Computer-based decision-support in the management of primary gastric non-Hodgkin lymphoma," *Meth Inform Med*, vol. 37, pp. 206–219, 1998.
[21] S. Moral, R. Rumí, and A. Samarón, "Mixtures of truncated exponentials in hybrid Bayesian networks," in *Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, ser. LNAI, vol. 2143, 2001, pp. 156–167.
[22] I. Flesch, "On the use of independence relations in Bayesian networks," Ph.D. dissertation, University of Nijmegen, 2008.