Bayesian controller versus traditional controllers [1]

Rainer Deventer[1], Joachim Denzler[1], and Heinrich Niemann[1]
[1] University Erlangen-Nürnberg, Chair for Pattern recognition, 91058 Erlangen, Martensstr. 3
{deventer,denzler,niemann}@informatik.uni-erlangen.de

Abstract

Adaptive control is a field with a long tradition sine the early 1950's. Despite the fact that Bayesian networks offer attractive properties, proven in other domains like data mining, they are seldom used in adaptive control. This paper develops a new type of controller, based on Bayesian networks. It is shown that controllers, trained with impulse and sinus response, shows nearly the same performance as the analytically calculated Dead-Beat controller.

1 Introduction

When dealing with industrial control, it is impossible, or at least time consuming, to get a mathematical description of the plant. E.g. there is no possibility to describe a nonlinear system by its transfer function. Thus the engineer has to rely on his experience. Another problem is, that the plant might change. This might happen due to changing operating points and environment. This led in the early 1950's [1] to research in adaptive control.

Bayesian networks(BN) offer a lot of advantageous preconditions to contribute to this interesting field. For example, they are able to deal with hidden variables or missing information, which might lead to robust behavior when measurements fail. Hybrid Bayesian networks offer the possibility to deal with discrete and continuous variables at the same time. Also nonlinear processes can be modeled, as shown in [2].

As long as linear systems are concerned, the similarities between Kalman Filters and Dynamic Bayesian networks (DBN) can be used to deduce the structure of the model and for the incorporation of a-priori knowledge (see [3]). In principle both the state-space description and difference equations can be used as model for the Bayesian controller. Former experiments [3] has shown that the difference equation, used in this paper, leads to more stable training results with less examples. In this paper a new controller type, based on BNs, is evaluated using dynamic systems with different damping and order. The results are compared with PID controllers, widely used in industry. Their settings are figured out both empirically using the traditional method of Ziegler and Nichols [4, 5]. Additionally they are compared to Dead-Beat controller, where the controller's transfer function is calculated analytically using the plant's transfer function. The results show, that the Bayesian controller clearly outperforms the approach by Ziegler and Nichols and shows comparable results as the Dead-Beat controller.

The article is structured as follows. In section "Dynamic systems and control theory" the terminology of control theory is introduced together with different methods to figure out the settings of PID controllers and the measures to evaluate their performance. Afterwards in section "Bayesian networks" the main ideas of BNs are discussed, including

[1] This work was funded by the ,,Deutsche Forschungsgemeinschaft" (DFG) under grant number SFB 396, project-part C1; only the authors are responsible for the content of this paper

the structure of the Bayesian controller and the calculation of the manipulated variable in section “Training and calculation of control signals”. The test systems and the experiments are discussed in the section called “Experiments”. This article concludes with a short summary and an outlook.

2 Dynamic systems and control theory

Control is a frequently occurring problem in industry with a tradition of more than 200 years (confer [6]). Examples are the control of pressure, temperature and revolution. In this section a common description for linear dynamic systems is introduced. Afterwards the frequently used PID controller is discussed together with the performance measure used to compare the Bayesian controller with the PID and Dead-Beat controller.

2.1 Controlled systems

Linear dynamic systems can be described by its differential equation

$$\sum_{i=0}^{n} a_i^c \frac{d^i\, y(t)}{dt^i} = \sum_{j=0}^{m} b_j^c \frac{d^j\, u(t)}{dt^j} \tag{1}$$

where $u(t)$ denotes the input of the system, $y(t)$ the output. Only systems with $m <= n$ are physical realizable. In most of the cases it is more common to use the Laplace-transform

$$\mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)\exp(-st)dt \tag{2}$$

of a function, so that the convolution $*$ of two signals $f_1(t)$ and $f_2(t)$

$$\mathcal{L}\{f_1(t) * f_2(t)\} = F_1(s) \cdot F_2(s) \tag{3}$$

is mapped to the multiplication of the two Laplace transformed $F_1(s)$ and $F_2(s)$. For equation(1) this leads to

$$Y(s)\sum_{i=0}^{n} a_i^c s^i = U(s)\sum_{j=0}^{m} b_j^c s^j \tag{4}$$

and the transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_0^c + b_1^c s + \cdots + b_m^c s^m}{a_0^c + a_1^c s + \cdots + a_n^c s^n} \ , \tag{5}$$

which is the quotient of two polynomials for simple single input single output systems.

For digital control it is more common to work with discrete time signals. The derivation of a signal $f(t)$ can be approximated by

$$\frac{df(t)}{dt} \approx \frac{f((k+1)\Delta T) - f(k\Delta T)}{\Delta T} \ , \tag{6}$$

where ΔT denotes the time difference between two time steps. Thus, the differential equation is approximated by a difference equation

$$y_t = -\sum_{i=1}^{n} a_i y_{t-i} + \sum_{i=0}^{n} b_i u_{t-i} \ . \tag{7}$$

To distinguish discrete time from continuous time systems indices are used, e.g u_t instead of $u(t)$. To avoid dealing with convolution the z-transformation

$$\mathcal{Z}\{f(k)\} = F(z) = \sum_{k=0}^{\infty} f(k) z^{-k} \tag{8}$$

is used for the description of time discrete systems. Similar to the Laplace transformation this leads to a z-transformed transfer function

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_n z^{-n}}{a_0 + a_1 z^{-1} + \cdots a_n z^{-n}} z^{-d} \quad , \tag{9}$$

where the term z^{-d} is responsible for the dead-time of the system. For systems without dead-time $d = 0$. Also the z-transform maps the convolution of two signals to the multiplication of the two z-transformed signals in the z-plane. For the development of the closed loop transfer function one should keep in mind that the digital controller generates only signals at special points in time. To ensure that this signal is kept constant between two points in time a zero order lag element H_0 is used. When looking for an equivalent description $H_0G(z)$ of a continuous time system G(s) in discrete time space, a series connection between a zero order lag element, the continuous system and a sampling element is used. Also the time discrete equivalent of linear systems is a quotient of two polynomials

$$H_0G(z) = \frac{B(z)}{A(z)} z^{-d} \quad , \tag{10}$$

denoted by $B(z)$ and $A(z)$, multiplied with z^{-d} for systems with dead-time. The closed loop transfer function $G_w(z)$ of a dynamic system $H_0G(z)$ and a controller $D(z)$ can now be calculated by

$$G_w(z) = \frac{D(z)H_0G(z)}{1 + D(z)H_0G(z)} \quad . \tag{11}$$

This formula will later be solved for D(z), to calculate the controller when the plant's transfer function G(z) and the desired reference reaction $G_w(z)$ are given.

2.2 PID Controller

The idea of a PID controller is to calculate the manipulated variable $u(t)$ according to the deviation $e(t) = w(t) - q(t)$ of the reference input $w(t)$ from the observed output signal $q(t)$. The input $u(t)$ is composed of three parts, being proportional to the error (P), to the integral (I) of the the error, and its derivation (D). The signal of an ideal PID controller results to

$$u(t) = K_\mathrm{C} e(t) + \frac{K_\mathrm{C}}{T_\mathrm{I}} \int_0^t e(\tau) d\tau + K_\mathrm{C} T_\mathrm{D} \frac{de(t)}{dt} \quad , \tag{12}$$

where K_C denotes the feedback gain. T_I and T_D are the integral and the derivative time respectively. The next two sections “Controller settings according to Ziegler and Nichols” and “Dead-Beat Controller” deal with two different possibilities to figure out the settings of a controller. The early approach of Ziegler and Nichols determines the parameters empirically. The Dead-Beat controller is suitable for discrete control systems, but it is supposed that the transfer function of the system is known.

Table 1: Settings of a PID controller according to Ziegler and Nichols

Type of controller	K_c	T_I	T_D
P	$0.5K_\mathrm{crit}$		
PI	$0.45K_\mathrm{crit}$	$0.85T_\mathrm{crit}$	
PID	$0.6K_\mathrm{crit}$	$0.5T_\mathrm{crit}$	$0.12T_\mathrm{crit}$

2.2.1 Controller settings according to Ziegler and Nichols

There are several possibilities for determining the settings of a controller. Many industrial processes show a transfer function with pure overdamped behavior, regularly described by

$$G(s) = \frac{K_\mathrm{P}}{1+Ts}\exp(-T_\mathrm{d}s) \tag{13}$$

as a PT_1 plant with gain K_P, time constant T, and dead time T_d. The rule most frequently used (cf. [6]) goes back to Ziegler and Nichols [4]. In one of his two proposed methods the controller is firstly used as a pure proportional controller, whose gain is increased until the control loop starts oscillating. The feedback gain K_crit and the duration of one period are measured. Afterwards the parameters are set according to table 1. The second controller to be compared with the Bayesian controller is the Dead-Beat controller introduced in the next section.

2.3 Dead-Beat Controller

The idea of a Dead-Beat controller is based on the considerations of Ragazzini [5]. Ragazzini assumes, that the transfer function of the closed loop $G_w(z)$ is equal to the desired command response $F_\mathrm{w}(z)$. From equation (11) it follows that

$$F_\mathrm{w}(z) = G_w(z) = \frac{D(z)H_0G(z)}{1+D(z)H_0G(z)} \ , \tag{14}$$

so that the controller $D(z)$ results in

$$D(z) = \frac{1}{H_0G(z)}\frac{F_\mathrm{w}(z)}{1-F_\mathrm{w}(z)} \ . \tag{15}$$

For example, it is possible to assume as desired command response, that the output should converge exponentially to the new desired value. In a Dead-Beat controller it is supposed, that all poles of the transfer function are in the origin of the z plane, so that

$$F_\mathrm{w}(z) = z^{-d}\frac{k_0z^q + k_1z^{q-1} + \cdots + k_{q-1}z + kq}{z^q} = z^{-d}(k_0 + k_1z^{-1} + \cdots + k_qz^{-q}) \tag{16}$$

which leads to $e(t) = 0$ after a number of time steps. Selecting

$$k_0 + k_1z^{-1} + \cdots + k_qz^{-q} = \frac{b_0 + b_1z^{-1} + \cdots + b_nz^{-n}}{b_0 + b_1 + \cdots + b_n} = \frac{B(z)}{B(1)} \tag{17}$$

leads to a stable input signal [5]. The dead beat controller

$$D(z) = \frac{A(z)}{B(1) - z^{-d} B(z)} \tag{18}$$

can be calculated by the known nominator and denominator polynomial of the dynamic system. It is guaranteed that after a step of the desired value, the new value is reached in a finite number of time steps. In the following section the Dead-Beat controller and the PID controller, whose settings are calculated according to Ziegler and Nichols, are compared to the Bayesian controller for four different test systems. The test systems have different dampings and orders, so that they show different step responses.

2.4 Performance measures

To compare the performance of the Bayesian controller with the PID and the Dead-Beat controller a measure is needed. A metric frequently used in control theory is the integral of the squared error. For discrete time systems this is replaced by the squared error sum

$$Q_{\mathrm{d}} = \sum_{t=0}^{t_{\mathrm{conv}}} \Delta T (e_t - e_\infty)^2 \tag{19}$$

where e_t denotes the difference between the observed output and the desired value w. Summation is done until convergence took place, i.e. $e_t = e_\infty$. A second performance criterion is the steady state error e_∞. Additionally the overshoot, the difference between the maximal output and the desired value w, is measured for all controllers.

To get an impression how fast the effect of a disturbance is eliminated also the time $t_{\mathrm{s}}(z, p\%)$ until the error is smaller than $p\%$ of the desired value is measured. The parameter z indicates whether the measurements are done for the reference ($z = 0$) or the disturbance reaction ($z = 1$) of the closed loop.

3 Bayesian networks

In the last section the mathematical description of dynamic systems, for example by difference equation, is explained. This section will expound, how this description is mapped to a DBN and used as a controller in the end. Before the structure of such a network is deduced some basics about BNs are discussed.

A Bayesian network represents the distribution

$$p(X_1, X_2, \cdots, X_l) = p(X_1) \prod_{i=2}^{l} p(X_i | X_1, \cdots, X_{i-1}) \tag{20}$$

of l random variables X_i, where X_i might represent the input U_t of a dynamic process. Usually not all random variables in $\{X_1, \cdots, X_{i-1}\}$ have an influence of X_i, thus (20) is rewritten to

$$p(X_1, X_2, \cdots, X_l) = p(X_1) \cdot \prod_{i=2}^{l} p(X_i | \boldsymbol{Pa}(X_i)) \ , \tag{21}$$

where the random vector $\boldsymbol{Pa}(X_i)$ contains the parents of X_i, usually the nodes with an influence on X_i. For reasons of clarity the relationship between random variables and their parents is displayed in a directed acyclic graph with the random variables as nodes and directed edges from the parents $\boldsymbol{Pa}(X_i)$ to the node X_i. For example, in figure 1 Y_{t+1} has the parents Y_t, Y_{t-1}, U_t and U_{t-1}. In principle there is no restriction for the distribution p, but to guarantee an easy evaluation of the BN, all continuous nodes in a BN are normally distributed

$$p(x|\boldsymbol{pa}(x)) = \mathcal{N}(\mu_X + \boldsymbol{w}_\mathrm{X}\boldsymbol{pa}(X)^\mathrm{T}, \sigma) \tag{22}$$

with a mean $\mu_X + \boldsymbol{w}_\mathrm{X}\boldsymbol{pa}(x)^\mathrm{T}$ and a deviation σ. The transpose of a vector is denoted by T, i.e. the weight is multiplied with the transposed instantiations of the parent nodes $\boldsymbol{pa}(x)^\mathrm{T}$. The parameters x and $\boldsymbol{pa}(x)$ are the instantiations of X and $\boldsymbol{Pa}(X)$ respectively. The weight vector $\boldsymbol{w}$ characterizes the influence of the parents $\boldsymbol{Pa}(X)$ on X. When a node has no parents or all of them are instantiated to zero the mean is equal to μ_X. During the evaluation there are several tasks to be executed. The main tasks are recalculation of the distribution when new observations are made or the calculation of the marginal distribution of a set of nodes $\boldsymbol{X}$. Given a full distribution $p(\boldsymbol{X})$ with $\boldsymbol{X} = \{X_1, \cdots, X_l\}$ an arbitrary distribution $p(\boldsymbol{X}\backslash\boldsymbol{C})$ with $\boldsymbol{C} \subset \boldsymbol{X}$ can be calculated by integration over all variables in $\boldsymbol{C}$:

$$p(\boldsymbol{X}\backslash\boldsymbol{C}) = \int_{\boldsymbol{C}} p(\boldsymbol{X})d\boldsymbol{C}. \tag{23}$$

A more detailed description of the algorithms used for BNs are given in [7, 8] or [9].

3.1 Dynamic Bayesian networks

In the BNs discussed so far there is a single node for each random variable. Thus it is not possible to represent the temporal course of a random variable. The mathematical description of a dynamic system, e.g. equation (7), shows that the value of a variable has to be represented at different time steps. For such cases DBNs are developed which are able to monitor a set of variables at arbitrary points in time.

The main idea of DBNs is to model each point in time by a static Bayesian network and to add temporal links from one time-slice to the next. Examples for such temporal links are found in figure 1, e.g. the edges from Y_{t-1} to Y_t and Y_{t+1}. Usually all the time-slices have identical parameter settings, but this is not a necessary precondition. Also the nodes in a DBN are normally distributed. A comparison between difference equation (7) and the definition of the normal distribution (22), used for each node, shows that it is possible to calculate the parameters of a node Y_t

$$p(y_t|y_{t-1}, \cdots, y_{t-n}, u_{t-1}, \cdots, u_{t-n}) = \mathcal{N}([a_1 \cdots a_n\, b_1 \cdots b_n][y_{t-1} \cdots y_{t-n}\, u_{t-1} \cdots u_{t-n}]^\mathrm{T}, \sigma)\ , \tag{24}$$

so that it predicts the next output of the dynamic system. Equation (24) presupposes a system with no dead time. The weight vector contains the coefficients of the difference equations being multiplied with the transpose of the in- and output signals. The evaluation of a DBN can be done in the same way as a static BN with equal parameters for all time slices respectively between the time slices.

As equation (24) shows, the distribution of node Y_t does not only depend on the time-slices for t and $t-1$, but also on former time slices. Such a model does not meet the

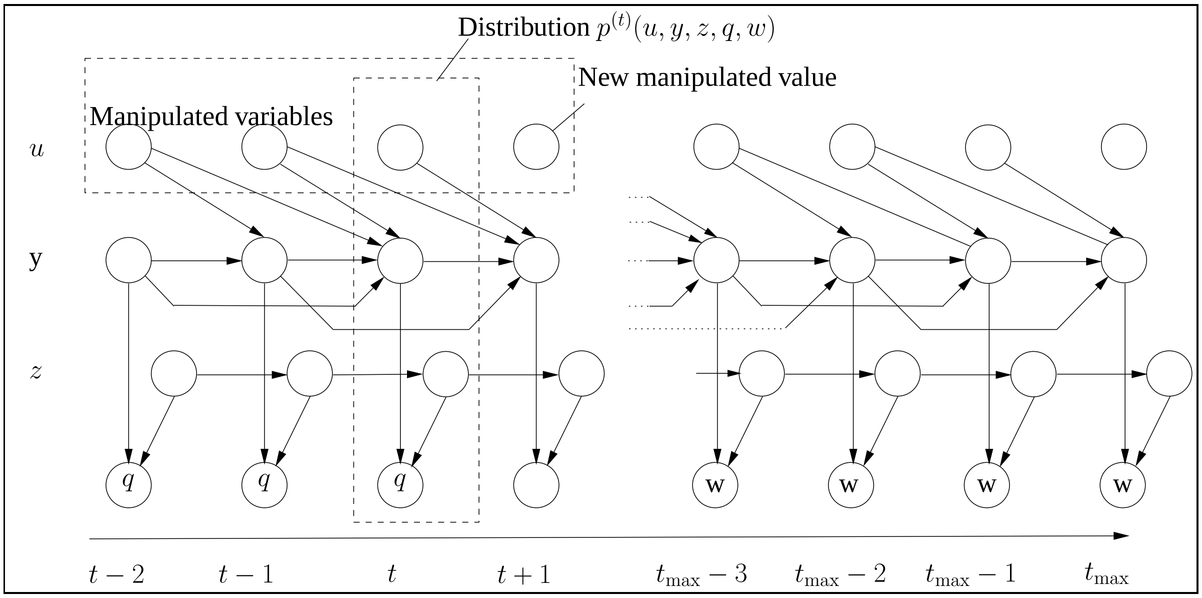

Figure 1: Principle structure of a BN used for control purposes

Markov assumption, that the future is independent of the past given the present. For this reason it is not possible to use standard toolboxes. For our experiments the BN-toolbox was expanded to model also Markov models of higher order. Of course a lot of details has to be omitted in this section. The reader is referred to [10] for a deeper introduction to DBNs.

3.2 Structure of the Bayesian Network

In the last section BNs and DBNs are introduced. This section will deepen this discussion and identify the structure of a Bayesian controller. The main part of such a controller can be inferred from equation (24). As y_t depends on $y_{t-1}, \cdots, y_{t-n}$ and $u_{t-1}, \cdots, u_{t-n}$ links are necessary from Y_{t-i} to Y_t $(1 \leq i \leq n)$ and from U_{t-i} to Y_t. The mean of the nodes Y_t is fixed to zero. Experience shows, that a low deviation is helpful for control purposes. This reflects the assumption that a proper model is learnt. The input nodes U_t have no parents, thus their only parameters are mean and deviation. The deviation is set to a maximum, so that there is no restriction for the instantiation of U_t. Thus the mean of U_t has only a negligible influence on u_t. It is set to zero.

A controller has to react also on the disturbance value z, so that the observed output q is not always equal to the modeled output y. For linear systems it is possible to suppose, that the disturbing value is added to the modeled output y_t, so that

$$q_t = y_t + z_t \quad . \tag{25}$$

This leads to the two additional nodes Q_t and Z_t for each slice. Both Y_t and Z_t are connected with weight one to Q_t. Both nodes get a zero mean. Additionally it is assumed, that the statistical behavior of Z_t changes only slightly in time, thus a link $Z_{t-1} \rightarrow Z_t$ with weight one is added. As an occurring error should result in changing u_t, the deviation of Z_t is selected smaller than the deviation of U_t. The deviation of Q_t is set to a minimal values, because an occurring deviation between y_t and q_t should be explained by changing the estimation of z_t.

The assumption, that there is no error during training, enables us to set $z_t = 0$ and $q_t = y_t$. Thus there are no hidden nodes left, which guarantees a stable training result.

4 Training and calculation of control signals

In the last section the structure of the DBN together with the mean of the nodes was concluded from the analytical description of dynamic systems. The used deviations of the nodes are selected, so that the BN is able to act as controller. The remaining parameters, i.e. the weights to Y_t are trained by the EM algorithm with 40 examples using the step, impulse, and sinus response of the system.

For the generation of the input signal a DBN with fixed length as depicted in figure 1 is used. The nodes $u_{t_0} \cdots u_t$ and $q_{t_0} \cdots q_t$ at the left hand side are used to enter formerly observed in- and output signals as evidence. For our experiments 10 nodes are used for the representation of the past. This part is also used to estimate the disturbing value as the difference between the modeled output y_t and the observed output q_t. To tell the system the desired value, w_t is entered also as evidence for the observed output $q_{t+2} \cdots q_{t_{\max}}$. For the experiments 15 nodes are used for the representation of the future. Now marginalization, as defined by equation (23), for the input nodes $u_{t+1} \cdots u_{t_{\max}}$ can be used to figure out input signals which leads to the desired output. When only one signal u_{t+1} is used this results in very high, sometimes even oscillating, manipulated variables (compare the results for BN1 in section "Experiments"). To damp down the input signal a weighted sum of $u_{t+1} \cdots u_{t+h}$ is used. The controller resulting from $h = 4$ is denoted as BN4. The response of the dynamic system to the input signal is measured (the dynamic system is simulated by Simulink) and entered together with the input signal as new evidence. As a DBN with fixed length is used all evidences are shifted to the left at the end of each cycle, the oldest values for the in- and output are removed. The results of our experiments, discussed in the section "Experiments" shows that the Bayesian controller described here shows comparable results to the analytically inferred Dead-Beat controller, defined in section "Dead-Beat Controller".

5 Experiments

The experiments are based on simulations of the dynamical systems, using Matlab. The usage of Matlab has different advantages, first it allows using the BN-toolbox [11], which is available free of charge. Second Simulink can be used for the simulation of the dynamic systems. As both toolboxes are based on Matlab, there is no problem with the integration.

As test systems three different systems of second order and one of third order, shortly introduced in table 5, are used. For all four systems the same scenario was applied. In the first phase the desired value $w(t)$ was set to zero. After convergence of the output the desired value was increased to 10. The overshoot, the squared error sum, the steady state error e_∞, and the time t_{s} until the new desired value is reached within an accuracy

Table 2: Description of test systems

No.	Transfer function G(s)	Description
1	$\frac{2}{0.01s^2+s+1}$	Damped system with gain two which has no tendency to overshoot
2	$\frac{2}{0.01s^2+0.1s+1}$	System with $D < 1$ which means that there is a tendency to overshoot
3	$\frac{10}{0.01s^2+0.05s+1}$	System with high gain and a large tendency to overshoot
4	$\frac{0.4s+2}{0.01s^3+0.5s^2+0.2s+1}$	System of third order, step response shows overshoot.

Table 3: Test results for system number 1 and 2

	Test System 1 $\Delta T = 0.05sec.$				Test System 2 $\Delta T = 0.05sec.$			
	ZN	DB	BN1	BN4	ZN	DB	BN1	BN4
$Q_\mathrm{d}(z=0)$	12.43	5.2	4.98	8.09	15.39	6.04	5.02	9.66
$e_\infty(z=0)$	0	0.01	0.03	0.06	0	0	0.01	0.02
Overshoot	8.82	0.01	0.06	0.06	4.03	0	0.36	0.54
$Q_\mathrm{d}(z=1)$	0.12	0.05	0.09	0.11	0.15	0.06	0.11	0.16
$e_\infty(z=1)$	0	0.01	0.02	0.03	0	0	0.03	0.04
$t_\mathrm{s}(z=0,1\%)$	1.1	0.1	0.10	0.50	3.35	0.1	0.35	0.45
$t_\mathrm{s}(z=0,3\%)$	0.85	0.1	0.10	0.35	2.5	0.1	0.20	0.40
$t_\mathrm{s}(z=1,1\%)$	0.55	0.1	0.15	0.55	1.5	0.1	0.15	0.30
$t_\mathrm{s}(z=1,3\%)$	0.3	0.05	0.15	0.20	0.5	0.1	0.15	0.25

of 1% or 3% is measured. The results are given in tables 3 and 4.

The squared error sum $Q_\mathrm{d}(z = 1)$, the steady state error $e_\infty(z = 0)$, the overshoot and the settling time in tables 3 and 4 gives an impression about the reference reaction. As the squared error sum includes a term for the steady state error it characterizes the oscillation until convergence is reached. Only together with the steady state error this term should be regarded as error measure.

Additionally to the reference reaction, the disturbance reaction is measured and characterized by the steady state error $e_\infty(z = 1)$, the squared error sum $Q_\mathrm{d}(z = 1)$ and the settling time $t_\mathrm{s}(z = 1, p\%)$ in presence of a disturbing value $z = 1$. For system 1, the settings of a PI controller, adjusted according to Ziegler and Nichols (ZN), shows worse performance both for the reference and the disturbance reaction of the control loop as the Dead-Beat (DB) and the Bayesian controller. For the Bayesian controller two different versions were tested. In the first one, in the column denoted with BN1, only one future signal u_{t+1} is used for the calculation of the new input signal. This leads to good results for the squared error sum, but to a great and oscillating manipulated variable. In the Bayesian controller BN4 the input signal is damped by calculating the used input as weighted sum of 4 input signals u_{t+1} to u_{t+4}. The signals obtained for these two controllers are displayed in figure 2 and 3. The comparison shows, that BN1 generates a larger input signal than BN4. This leads to a smaller squared error sum, as the error is reduced much faster in the beginning of the test. But in reality there are two disadvantages. First

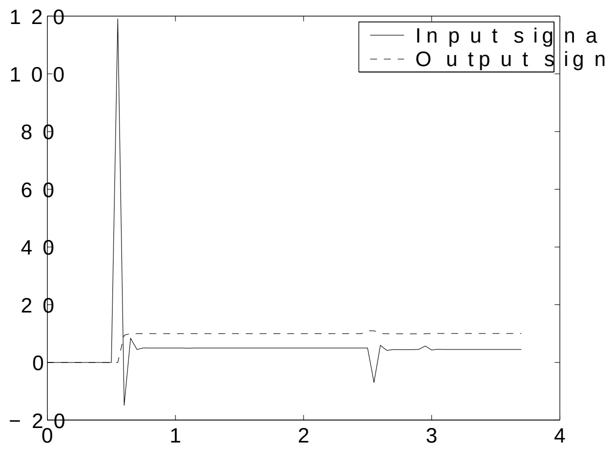

Figure 2: In- and output signals for BN1

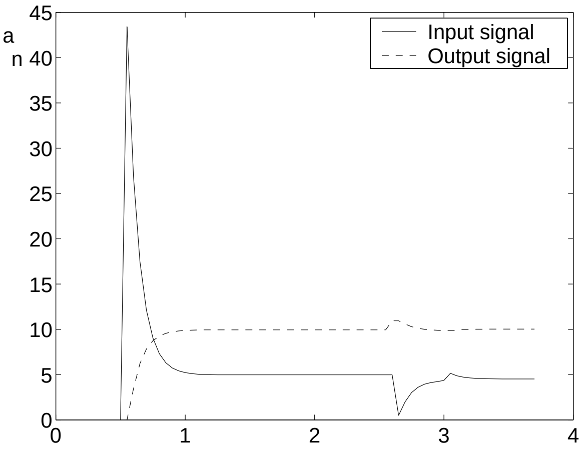

Figure 3: In- and output signals for BN4

the input signals of BN1 shows the tendency to oscillate, particularly for systems with dampings $D < 1$. The next question is whether the actuator is able to follow the sudden changes of the input signal. Also the Dead-Beat controller generates extremely large input signals. But as a consequence of the design the input signal is guaranteed to be stable after a finite number of time steps. But also for the Dead-Beat controller usually a larger sampling time would be selected, to come to smaller values for the manipulated variable.

When the disturbance reaction of the closed loop is regarded, the Dead-Beat controller is better than the Bayesian controller. But one should keep in mind that the Bayesian controller is calculated based on training signals, the Dead-Beat controller uses the knowledge of the dynamic system to calculate an optimal response.

The behavior of the four controllers is similar when the second test system is regarded. For the Ziegler-Nichols controller one should keep in mind, that systems 2-4 can not be described as a PT_1 element, due to the observed overshoot. They are too complex for this approach. Additionally the sampling time should be much smaller than the cycle duration, which is not met for systems 2 and 3. But measurements done with sampling time $\Delta T = 0.02sec.$, to meet this requirement, does not lead to better results in these cases. The squared error sum of the Bayesian controller BN1 is slightly better than for the Dead-Beat controller. A closer look at the signals (not displayed in this paper) of the Bayesian controller BN1 shows, that this is due to large values of the manipulated variable at the beginning, so that the error is diminished quickly. On the other side this results in a small overshoot. As a second consequence this leads to oscillations, before the output converges to the new desired value. The results for test system 3, depicted in table 4 agrees with the results obtained for the first two systems. For system four the sampling rate for the Dead-Beat controller and the Bayesian network is set to $\Delta T = 0.4sec$, the PI controller of Ziegler and Nichols does not work with this sampling rate, thus ΔT was decreased to 0.05 seconds. So the result for Q_d can not be compared in this case. Additionally the performance of the Bayesian controller decreases. An explanation might be the smaller proportion between the number of training examples and parameters.

Table 4: Test results for system number 3 and 4

	Test System 3 $\Delta T = 0.05 sec.$				Test System 4			
	ZN	DB	BN1	BN4	ZN	DB	BN1	BN4
$Q_{\mathrm{d}}(z=0)$	21.51	6.15	4.97	9.8914	13.87	40.20	49.88	66.45
$e_{\infty}(z=0)$	0.01	0	0.00	0.0025	0.00	0	0.05	0.05
Overshoot	1.67	0	0.00	1.1457	6.34	0.22	0.47	1.80
$Q_{\mathrm{d}}(z=1)$	0.22	0.06	0.12	0.1874	0.31	0.40	1.02	1.44
$e_{\infty}(z=1)$	0	0	0.03	0.0424	0	0	0.00	0.09
$t_{\mathrm{s}}(z=0, 1\%)$	6.05	0.1	0.05	0.6500	2.65	1.2	6.00	7.12
$t_{\mathrm{s}}(z=0, 3\%)$	4.4	0.1	0.05	0.4500	2.0	0.8	2.56	4.40
$t_{\mathrm{s}}(z=1, 1\%)$	2.5	0.1	0.55	0.6000	1.5	0.4	4.40	10.92
$t_{\mathrm{s}}(z=1, 3\%)$	1.05	0.1	0.15	0.2500	0.85	0.4	3.20	3.68

6 Conclusion

Bayesian networks show nearly the same performance as an analytically designed Dead-Beat controller. Despite promising results there is a lot of work to be done before Bayesian controller can be used in practice. In the future we will test our approach by controlling the forces of a hydroforming press. This step adds problems of real time performance and of controlling multiple input multiple output systems. A second branch of our research will concentrate on modeling non-linear systems with hybrid Bayesian networks. Later approaches for approximated evaluation of DBNs can be added to meet real time conditions also for more complex systems.

References

[1] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control.* Prentice Hall, London, 1991.

[2] R. Deventer, J. Denzler, and H. Niemann. Non-linear modeling of a production process by hybrid Bayesian Networks. In Werner Horn, editor, *ECAI 2000 (Berlin)*, pages 576–580. IOS Press, August 2000.

[3] Rainer Deventer, Joachim Denzler, and Heinrich Niemann. Bayesian control of dynamic systems. In Lakhmi C. Jain Ajith Abraham, editor, *Recent Advances in Intelligent Paradigms(to be published)*, chapter 3. Springer Verlag, 2003.

[4] J. G. Ziegler and N. B. Nichols. Optimum settings for automatic controllers. *Trans. ASME*, 64:759 – 768, 1942.

[5] Gerd Schulz. *Regelungstechnik, Mehrgrößenregelung – Digitale Regelungstechnik – Fuzzy Regelung.* Oldenburg Verlag, München, Wien, 2002.

[6] Heinz Unbehauen. *Regelungstechnik I.* Vieweg, 1997.

[7] S. L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, Vol. 87(420):1098 – 1108, December 1992.

[8] S. L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[9] F. V. Jensen. *An introduction to Bayesian networks*. UCL Press, 1996.

[10] U. Kjærulff. A computational schema for reasoning in dynamic probabilistic networks. In *Proceedings of the Eighth Conference of Uncertainty in Artificial Intelligence*, pages 121 – 129. Morgan Kaufmann Publishers, San Mateo, California, 1992.

[11] Kevin P. Murphy. The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, 33, 2001.