A CONTROLLER FOR A MAGNETIC BEARING USING THE DYNAMIC PROGRAMMING METHOD OF BELLMAN

H. F. Steffani,[1] W. Hofmann,[1] B. Cebulski[1]

Abstract

Magnetic bearings with premagnetization can be described as a system either with the current as input which can take continuous values or with the voltage as input which can take only discrete values because of the used inverter. The controller for the latter has to be nonlinear (cf. Steffani and Hofmann (1997)). One method to design optimal controllers with boundaries is the DYNAMIC PROGRAMMING of BELLMAN. This paper describes how this method can be used to create a switching controller for a magnetic bearing. It is explained how the cost-function affects the stiffness. Experimental and simulation results are shown.

MODEL OF THE VOLTAGE CONTROLLED BEARING

The voltage controlled magnetic bearing with premagnetization showed in figure 1 is described by the differential matrix-equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} s \\ v \\ i_{st} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{k_s}{m} & 0 & \frac{k_i}{m} \\ 0 & -\frac{k_i}{L_{st}} & -\frac{R}{L_{st}} \end{pmatrix} \begin{pmatrix} s \\ v \\ i_{st} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{L_{st}} \end{pmatrix} u \quad , \tag{1}$$

which means that the system is of 3rd order. The variable s is the deviation of the shaft-position, v its time derivative and i_{st} the current. The k_i and k_s are the current and the position stiffness, which can be taken as constant as well as the inductance L_{st} and resistance R_{st} of the coil.

The system only allows the negative and the positive DC-link voltage and 0 as manipulated value u. This means that the conventional approaches in designing a continuous controller like pole-placement, LQR or the root-locus method cannot be

[1]TU Chemnitz, Electrical Machines and Drives, D-09111 Chemnitz, Germany, tel.: ++49 371/531-3377, fax.: ++49 371/531-3324, e-mail: hans.steffani@e-technik.tu-chemnitz.de.

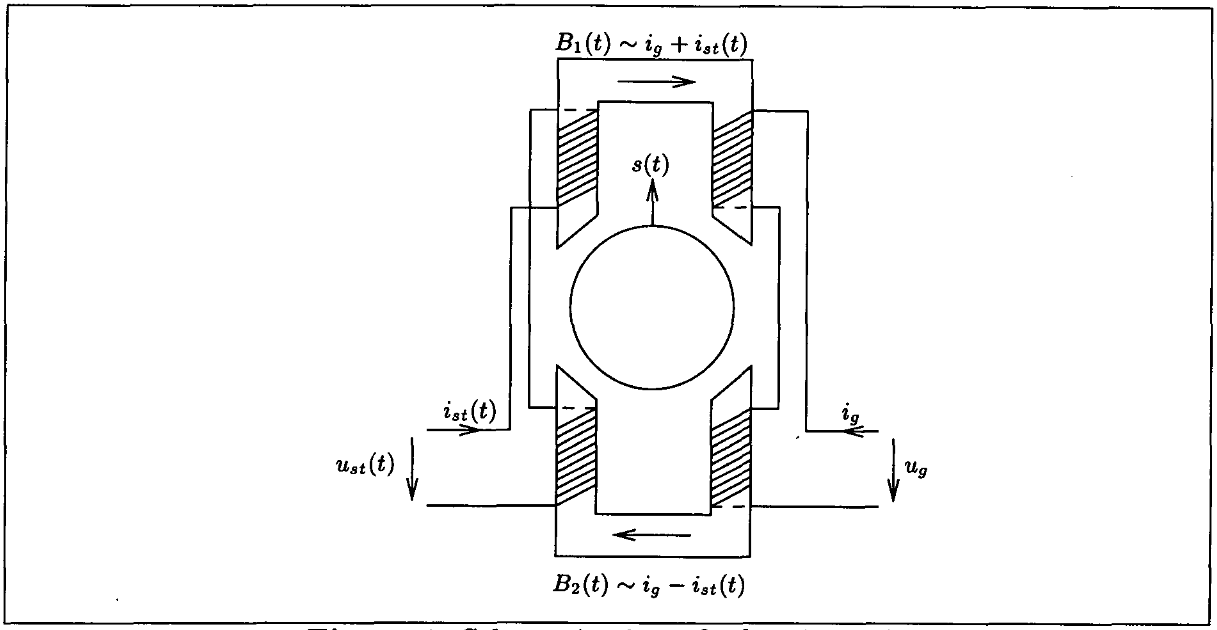

Figure 1: Schematic view of a bearing axis

applied. As the order is greater than 2 it is not possible to design a controller in the phase plane too.

However, the DYNAMIC PROGRAMMING seems to be an interesting method as it requires the quantification of the manipulated value which can be done easily.

DESIGN OF CONTROLLERS WITH THE METHOD OF BELLMAN

The main idea of the DYNAMIC PROGRAMMING is describing the control process as a number of consecutive decisions. Decision means that we have to chose a manipulated value out of a number of possible values at each control step. In our special system only 2 ($\pm u_{dc}$) or 3 ($\pm u_{dc}$, 0) values need to be tested. It is plausible and also can be proved (Föllinger, 1994, p. 254), that when taking an optimal trajectory every partial trajectory is optimal too. BELLMAN stated this as the

> **Principle of Optimality.** An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (Bellman, 1973, p. 87).

This principle leads, in contradiction to the approach using the MAXIMUM PRINCIPLE OF PONTRJAGIN (Boltjanski (1971)), immediately to the optimal feedback control law. The aim of optimal control is to minimize a cost functional like

$$I(\underline{x}_0) = \sum_k S(\underline{x}_k) \tag{2}$$

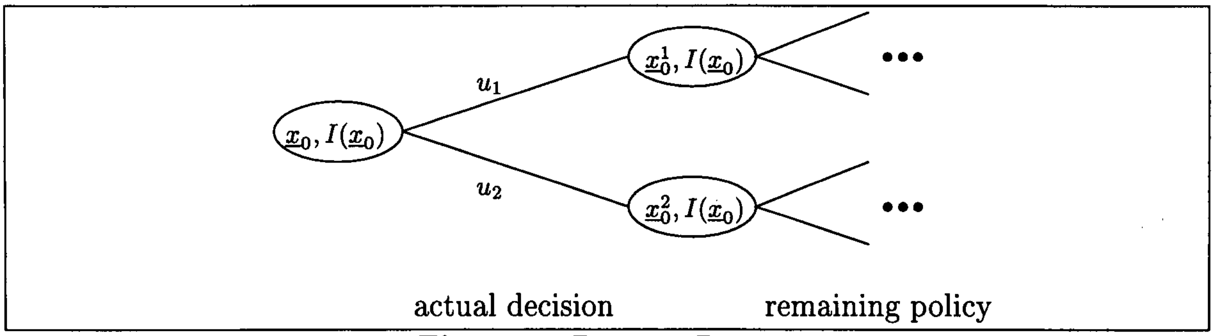

Figure 2: Decision Process

Initialization of the s-map

Repeat n times

    For all $\underline{x}_0$ do

        For all u_i do

            Calculate $\underline{x}^i(T, \underline{x}_0, u_i)$
            Calculate $I^{old}(\underline{x}^i)$

        Calculate $I^{new}(\underline{x}_0) = \min_i(S^{old} + I_i)$
        Store $I^{new}(\underline{x}_0)$ in S-map
        Store u belonging to I^{new} in u-map

Figure 3: Algorithm of the Controller-design with the Method DYNAMIC PROGRAMMING.

with $S(\underline{x}_k)$ being the cost of the state $\underline{x}_k$ on the trajectory from each starting point $\underline{x}_0$ to $\underline{0}$. In our case we use the quadratic form

$$S(\underline{x}_k) \quad = \quad \underline{x}_k^T \underline{\boldsymbol{Q}} \underline{x}_k \tag{3}$$

with a positive definite $\underline{\boldsymbol{Q}}$.

The DYNAMIC PROGRAMMING takes the control process as a multistage decision process. Given the principle of optimality, a policy leading to an optimal trajectory can be divided into two parts, the actual decision and the remaining policy (cf. figure 2). Since we have the choice between several values for the controller output, we have to take the one which leads to the state that owns the smallest cost for transferring to the goal state.

The question remains to calculate the cost for each state. This can be done using a recursive function

$$I(\underline{x}_0) = \min_i \left(S(\underline{x}_0) + I(\underline{x}_1^i)\right) \tag{4}$$

which can be derived from the principle of optimality (cf. (Bellman, 1967, sec. 3.7)).

The problem to solve is how to calculate (4) for each $\underline{x}_0$ of grid. An iterative approximation is chosen which is described in figure 3. Two maps[1] are built and updated several times. The one called I-map stores the cost of each state. It is initialized with the values taken from equation 3. An u-map is build to store the manipulated value which starts the optimal control-strategy. After repeating this several times the I-map is rebuilt taking the old one using equation (4).

The calculation of both maps are done within a subspace of the state-space. At the borders, the problem occurs, that at some points none of the valid values can move the state to a destination within the subspace. In this case, the cost $I(\underline{x}_1^i)$ cannot be calculated as a weighted sum of the costs of the three nearest neighbors. Instead, the cost of the nearest point within the state-space is taken.

What are the advantages of the DYNAMIC PROGRAMMING? The first is, that the result of the controller design-process is a look-up table, which is a fast but memory-consuming method for implementing a controller. However, the used DSP has enough memory so that this is not crucial. The second advantage is that boundaries of the state as well as of the manipulated value can be regarded. In our special case with only two or three possible values the method leads straight to an optimal bang-bang- or three-position controller. As a third advantage the DYNAMIC PROGRAMMING can be used for nonlinear plants as well as for linear plants.

The often mentioned disadvantage (cf. Föllinger (1994)) is the amount of computing power of the design process, which is and will be overcome by the increasing computing capacity. The algorithm can easily be distributed to several computers which run the algorithm in figure 3 for particular parts of the state-space. It has to be taken into account that the time-consuming part of the DYNAMIC PROGRAMMING is its off-line part which has only to be run once.

SCALING, WEIGHTING, RECURSION DEPTH AND PREDICTION TIME

Our software is coded in such a way, that it calculates the S- and u-maps for all states within the range $[-1, 1]$. To get a first solution, only a rough choice of the state-scale-factors is necessary. Also a simple rating of the three states with the same value leads to a stable simulation which shows us the way to a better choice of the range.

As mentioned above a quadratic form is used as cost-function. Principally it is possible to use other cost-functions. Because of our implementation, a time-variant cost-function is not possible.

The manipulated value is not weighted because its absolute value is always the same. It is common to set only the diagonal values of the weight matrix $\underline{\boldsymbol{Q}}$ to reduce the effort in finding sensible elements.

Instead of minimizing

$$I = \sum (s, v, i) \begin{pmatrix} q'_s & 0 & 0 \\ 0 & q'_v & 0 \\ 0 & 0 & q'_i \end{pmatrix} \begin{pmatrix} s \\ v \\ i \end{pmatrix} \tag{5}$$

[1]The term map was introduced applying the algorithm to 2nd order systems.

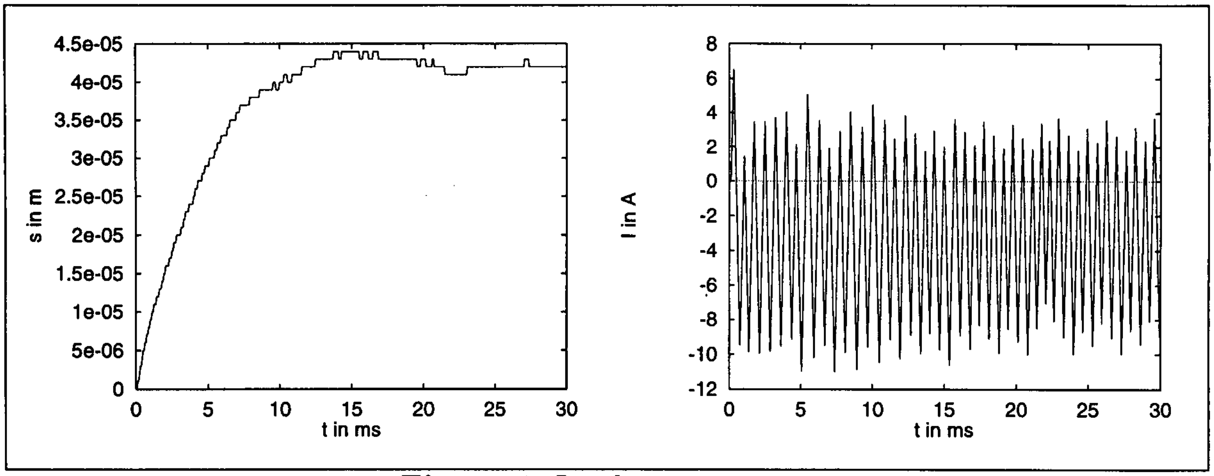

Figure 4: Load-step response

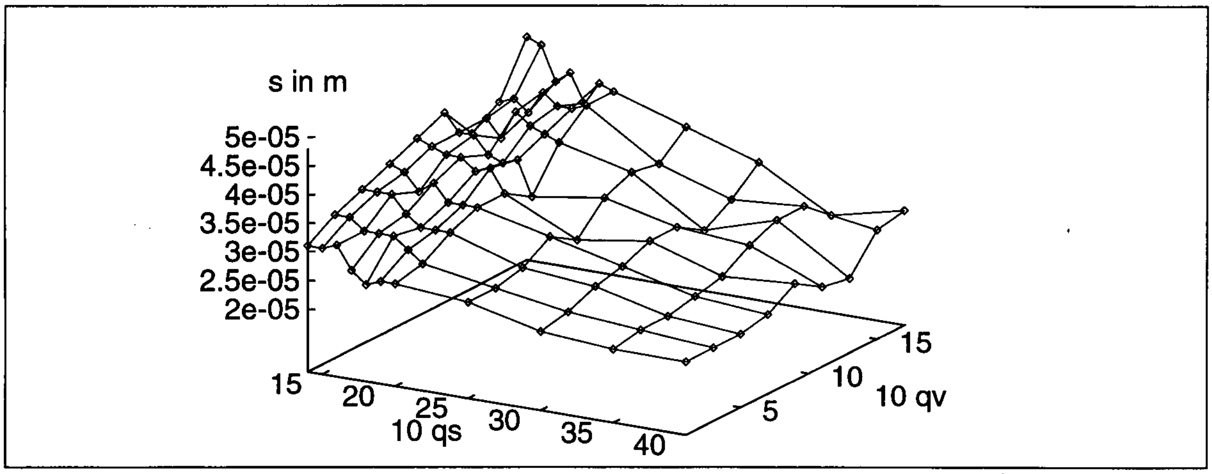

Figure 5: Control error of a magnetic bearing with 600 N load ($q_i = 1$)

we can minimize

$$I = \sum(s,v,i)\begin{pmatrix} \frac{q'_s}{q'_i} & 0 & 0 \\ 0 & \frac{q'_v}{q'_i} & 0 \\ 0 & 0 & \frac{q'_i}{q'_i} \end{pmatrix}\begin{pmatrix} s \\ v \\ i \end{pmatrix} = \sum(s,v,i)\begin{pmatrix} q_s & 0 & 0 \\ 0 & q_v & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} s \\ v \\ i \end{pmatrix} \quad (6)$$

which will be minimized by the same trajectories as the original cost function. The problem is still how to chose the parameters q_s and q_v.

For this paper the states have been scaled to $40\,\mu$m, $0.04\,{}^{\mathrm{m}}\!/\!_{\mathrm{s}}$ and 10 A. Each state is divided into 21 steps. This number should be odd to in order to turn each 0 to a discrete state. The simplest weights are $q_s = q_v = 1$ which means that all values are introduced with the same weight. The time to compute the lookup-table is about 136 s taking three possible manipulated values and 86 s using two on a DEC 3000.

One simulation result is shown in figure 4 using the same parameters as shown in table 1.

The simulated response to a load step is shown in figure 4. This simulation respects the dead-time of one step between sampling and output of the manipulated value. As well as the measured results, it leads to a fidgety current. The parameters of the design-process are the weights q_s and q_v and the prediction time T which is the time between two consecutive $\underline{x}$ and its consecutive state $\underline{x}_{k+1}$.

One crucial figure of a magnetic bearing is its stiffness. Figure 5 shows the position error assuming a constant load of 600 N. As the stiffness is calculated by $s_d = \frac{F}{s}$ large values signify small stiffness and vice versa.

It is obvious that the stiffness mainly depends on the weight q_s which weights the position error while q_v affects the damping. It is remarkable that a small weight for the velocity does not lead to an overshot but to a drop. This drop disappears if the weight of the velocity is increased.

All step responses do not overshoot which means that the system is heavily damped.

EXPERIMENTAL SETUP

A standard induction motor KR 250 S 2 with a nominal power of 75 kW and a nominal speed of 3000 min^{-1} with a 250 kg rotor is used as a test setup. The machine can be operated in field weakening to reach higher speed for experimental needs.

The levitation system consists of two radial bearings and one axial bearing. Each radial bearing consists of two perpendicular bearing axes. The bearing system can thus control the motion in three coordinates and the rotation in two coordinates. The remaining movement around the shaft axis is controlled by the induction motor.

The two opposite electromagnets are each fitted with a premagnetization coil and a second one for the manipulated current as it is shown in figure 1. The coils for the manipulated current are connected such, that a positive manipulated current reduces the magnetic flux in the one magnet and enhances it in the other one. This configuration is known as internal linearization.

An eddy current sensor is used as measurement system. Its time constant of about 8 μs can be neglected.

The amplifiers for the magnetic bearings are 50 kHz MOS-FET converters. They can either be used with an under-laying current control or the switches can directly be driven by the bearing control. In the first case the manipulated value of the air-gap controller is the current so it is called a current driven bearing while in the second case the manipulated value is the voltage so we call this a voltage driven bearing.

EXPERIMENTAL RESULTS

Figure 6 shows the progression of the axis position and the current using a controller designed with the DYNAMIC PROGRAMMING.

Table 1 shows the parameters which are used for the controller design.

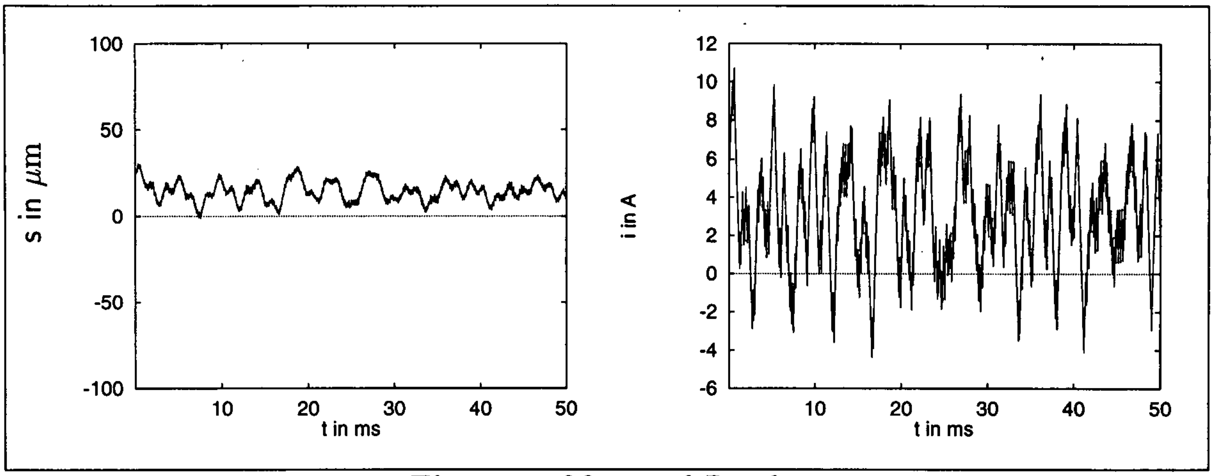

Figure 6: Measured Results

TABLE 1: Controller-design parameters

Number of quantization steps s, v, i_{st}	21
Range s	$\pm 40\mu\text{m}$
Range v	$\pm 0.039\frac{\text{m}}{\text{s}}$
Range i_{st}	$\pm 9.77A$
q_s	1.125
q_v	0.625
q_i	1.0
Prediction Time (DYNAMIC PROGRAMMING)	0.125 ms
Sample Time (controller)	0.066 ms

The shaft moves in an area of $\pm 15\mu$m around zero-position, However this result should be improved as other methods (Steffani and Hofmann (1997)) lead to movements within a range of $\pm 5\mu$m.

CONCLUSION

This paper shows how BELLMAN'S DYNAMIC PROGRAMMING can be used easily to design a bang-bang-controller for a voltage controlled magnetic bearing. Due to modern computers, the computing time becomes acceptable at least for a linear system with three states and a two-point manipulated value. The calculating effort can further be reduced by introducing a map to memorize the possible consecutive states for each state and manipulated value. However, this will heavily enlarge the memory used by the program. Further it is possible to distribute the algorithm to several computers. This is possible by dividing the state-space in several parts for which each step of calculating the s- and u-map can be done independently.

Theoretical investigations have to be done to find out how the depth, i. e. the number of iterations of calculating the maps has be chosen and how depth, discretization and range of the states are connected.

PARAMETERS

$$\begin{aligned} k_i &= 190\,{}^{\mathrm{N}}\!/\!_{\mathrm{A}} \\ k_s &= 6.421 \cdot 10^6\,{}^{\mathrm{N}}\!/\!_{\mathrm{m}} \\ L_{st} &= 3\mathrm{mH} \\ R &= 0.3\Omega \end{aligned}$$

ACKNOWLEDGMENT

We thank the DEUTSCHE FORSCHUNGSGEMEINSCHAFT which supports this project within its program SYSTEMINTEGRATION ELEKTRISCHER ANTRIEBE.

REFERENCES

Bellman, R. (1967). *Dynamische Programmierung und selbstanpassende Regelprozesse.* München, Wien: R. Oldenbourg Verlag.

Bellman, R. (1973). *Methods of Nonlinear Analysis*, Volume 2. New York, London: Academic Press.

Boltjanski, W. G. (1971). *Mathematische Methoden der optimalen Steuerung.* Leipzig: Akademische Verlagsgesellschaft Geest.

Föllinger, O. (1994). *Optimale Regelung und Steuerung* (3. ed.). München, Wien: Oldenbourg Verlag.

Steffani, H. F. and W. Hofmann (1997). A switching controller for a magnetic bearing. In M. Greene (Ed.), *Proceedings of the IECON 1997*, Piscataway, NJ, USA, pp. 85–90. IEEE: IEEE.