Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Nonlinear One-Step Predictive Control of an Active Magnetic Bearing

Stéphane Bonnet * **Jérôme De Miras** * **Borislav Vidolov** *

* *U.M.R. C.N.R.S. 6599 "HeuDiaSyC", Université de Technologie de Compiègne, BP 20529 – 60205 COMPIÈGNE CEDEX, FRANCE*
e-mail: {bonnetst,demiras,bvidolov}@hds.utc.fr

Abstract: This paper deals with the stabilization of a five degree-of-freedom active magnetic bearing using a novel predictive nonlinear control approach. That approach is based on using a simplified nonlinear model of the bearing axes without considering couplings to synthesize a discrete-time controller. The controller is then able to stabilize the load with predetermined linear dynamics that form the control design parameters. Simulations and experimental results obtained from a laboratory bearing show the robustness of the proposed controller, both in terms of modeling errors and perturbation rejection.

1. INTRODUCTION

Non-contacting magnetic bearings are magnetic levitation devices that have been stirring up a lot of interest in recent years, both in academic and industrial communities. They are becoming important devices in applications ranging from axle bearings in high speed machinery such as turbines, pumps or compressors to flywheel inertial energy storage systems [CDF$^+$05]. Indeed, they are demonstrating numerous advantages when compared to traditional, mechanical bearings: as they are contactless devices, friction is low and predictable, they can work in a complete vacuum which usually tends to prevent proper lubrication of conventional bearings, and reduce wear-out and thus cost of ownership of expensive high-speed rotating machinery.

However, building stable, static magnetically positioned systems using only permanent ferromagnetic magnets, according to Earnshaw's theorem, is impossible. Even though passive solutions involving the use of diamagnetic materials are being investigated [MSB01], they are still relatively underdeveloped. Those limitations mean that most bearings are active magnetic bearings (AMBs), which are electromagnetic devices requiring active control systems in order to stabilize the load [SBT94].

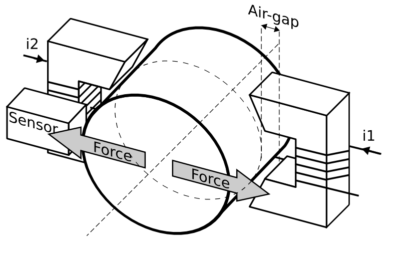

Fig. 1. An AMB axis.

The basic principle behind AMBs operation is simple, as shown by Fig. 1. Each control axis is fitted with two electromagnets and a position sensor that measures the displacement of the shaft. The electromagnets generate a force proportional to the square of the current intensity passing through them and inversely proportional to the air-gap between the shaft and the stator. By controlling those forces, it is possible to steer the position of the shaft along each control axis. However, that principle makes the model of an axis highly nonlinear, which in turn leads to many complexities in control synthesis. Furthermore, since AMBs are fast electromechanical systems, considering computational issues is essential when it comes to implementing real-time control strategies [GP05].

Hence, much work has been devoted to designing control laws for AMBs. In order to keep the computing effort needed by implementation reasonable, extensive studies have been carried out on linear modeling and control, such as state-space and transfer approaches, $\mathcal{H}_\infty$ control [FMM90], μ-synthesis [NY94], LQ control [GP05], or input-output linearization [CDC96]. In recent years however, thanks to the ever increasing computing power available, nonlinear controllers have gained a lot of popularity, as they are generally better than linear ones. The most popular technique used is feedback linearization combined with robust control techniques [CK05], but other nonlinear control techniques ranging from fuzzy control [VMDC96] or sliding mode control [CDC96] [CKS93] to discrete dynamic programming [SHC98] or continuous time model predictive control [HWH07] have been studied.

In this paper, we are focusing on the design of a heuristics based, nonlinear discrete-time predictive controller for a single axis. While predictive control usually implies to solve an on-line optimization problem for a given control horizon at each time step [GSDD05], the proposed controller only predicts the behavior of the plant one control step ahead. That knowledge is used to build a control input that makes, subject to certain limitations, the system follow linear dynamics given as design parameters, effectively building a linearizing control while not requiring too great a computing effort. As any predictive approach, the controller has to be robust towards both prediction errors and external perturbation forces. For the sake of clarity, we only study the stabilization of a single

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.3497

degree of freedom, using current applied to the coils as the constrained control input. As we consider couplings between different axes as unknown perturbations, that same controller can be applied to all five control axes. This simplification would not hold if the elimination of the unbalance (see eg. [DC98]) due to the offset between shaft rotational and gravity centers was the control problem, but this is an entirely different issue not addressed here. We first describe the proposed control approach. A model necessary to control synthesis and plant simulation is then derived. Finally, simulation and experimental results obtained on a five-degrees-of-freedom laboratory AMB show the robustness of the proposed approach.

2. PROPOSED CONTROL SCHEME

2.1 Overview

The basic idea behind the proposed discrete-time control strategy is to use a-priori knowledge of the plant dynamics to compute a control input that would steer its output towards a linear, predefined behavior given as a design parameter. This a-priori information is gathered from repeating off-line numerical integration of a numerical model of the system, varying the initial conditions in a properly chosen subspace of the model state-space, for each admissible, discrete, control input value. This yields a prediction map that, from a specific initial state and a given control input value, provides an approximation of the system state and outputs after one time-step.

The on-line algorithm purpose is to pick at each time-step the most appropriate control input to apply to the plant using that prediction map. Starting from the current state, which is supposed known, the system outputs and state for each admissible control value is predicted. Meanwhile, the output of a stable linear dynamical system of appropriate dimension is computed, using as initial conditions the current plant outputs. This is the objective for the next control step. The bounded control value that steers the plant outputs the closest to the objective is then computed through approximate distance minimization in the phase space, and applied to the plant. The main advantage of that algorithm if compared to usual predictive control is that no expensive on-line optimization step is needed. However, as it is limited to a horizon of one time-step, it does not intrinsically take inputs, outputs and state constraints into consideration. Respecting those kind of constraints depend on the choice of the reference system.

2.2 Prediction map

Consider the following time-invariant dynamical system representation:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = g(\mathbf{x}(t), \mathbf{u}(t)) \end{cases} \mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m, \mathbf{y} \in \mathbb{R}^p. \quad (1)$$

We can introduce the vector noted $\mathbf{v}$:

$$\mathbf{v}(t) = [\mathbf{x}(t)\ \mathbf{y}(t)]^T \in \mathbb{R}^{n+p}, \quad (2)$$

and the two permutation matrices $\mathbf{P_x}$ and $\mathbf{P_y}$ such that:

$$\mathbf{x}(t) = \mathbf{P_x} \cdot \mathbf{v}(t) \text{ and } \mathbf{y}(t) = \mathbf{P_y} \cdot \mathbf{v}(t). \quad (3)$$

What we want to obtain is a prediction map pr that, given an initial state condition $\mathbf{x}_k = \mathbf{x}(k \cdot \Delta t)$ and a control input $\mathbf{u}_k = \mathbf{u}(k \cdot \Delta t)$, predicts the a-priori state/output vector $\hat{\mathbf{v}}_{k+1}^- = \hat{\mathbf{v}}(k \cdot \Delta t + \Delta t)^-$, where Δt is the controller sample time, and k a positive integer:

$$\hat{\mathbf{v}}_{k+1}^- = pr(\mathbf{x}_k, \mathbf{u}_k). \quad (4)$$

In order to build that prediction map, a regular rectangular tessellation $\mathcal{T}$ on a subset $\mathcal{P}$ of $\mathbb{R}^n \times \mathbb{R}^m$ is constructed. $\mathcal{P} = [\mathbf{x}_{min}\ \mathbf{x}_{max}] \times [\mathbf{u}_{min}\ \mathbf{u}_{max}]$ arbitrarily limits the state and input intervals the predictor is built upon.

For each vertex of $\mathcal{T}$, that is every state/input combination in $\mathcal{T}$, the dynamical system given by Eq. (1) is numerically integrated using simulation tools such as Simulink on a single sample time-step. The solution $\mathbf{v}$ is stored in a table. The values of pr at unknown points can then be interpolated from that table, using multilinear interpolation. While multilinear interpolation leads to errors proportional to the square of the grid size, its main advantage is the low computing effort needed to get interpolated values, hence the choice of that approach instead of more precise but less efficient methods such as spline interpolation.

2.3 On-line control algorithm

The proposed control algorithm can be divided into five main steps:

- error estimation,
- objective computation,
- state/output prediction for all possible inputs,
- control input computation,
- next state/output prediction.

We suppose that at step k, an estimation $\hat{\mathbf{v}}_k$ of the plant state and outputs is available, eventually using a state estimator. We also suppose the prediction $\hat{\mathbf{v}}_k^-$ is available from the previous control step. That prediction is a representation of what the plant state and outputs should be if its initial state was $\mathbf{v}_{k-1}$ and the constant control input $\mathbf{u}_{k-1}$, computed at the start of the previous control step, was applied during all its duration. Of course, modeling errors in Eq. (1) and interpolation errors in Eq. (4) mean that the prediction is inherently unreliable. Therefore, it has to be corrected at each time-step. The prediction error estimation is given by:

$$\varepsilon_k = \varepsilon_{k-1} + \alpha(\hat{\mathbf{v}}_k^- - \hat{\mathbf{v}}_k), \quad (5)$$

where α is a damping coefficient in $]0, 1]$. The error estimate at step k is thus the cumulative prediction error made during all the previous steps. Indeed, by hypothesis, we consider the difference between consecutive prediction errors to be small if the displacement of the state/output is equally small. That means that when the system has reached the set point, the error estimate will converge towards a value mainly compensating for model errors at the set point, thus rejecting static errors. Indeed, a non-zero static error would mean that $\hat{\mathbf{v}}_k^-$ does not converge towards $\hat{\mathbf{v}}_k$, which can not be true if the system is stabilized at the set point.

The output objective $\mathbf{y}_{k+1}^* \in \mathbb{R}^m$ is computed by a stable reference linear system of the form

$$\dot{\mathbf{z}} = \mathbf{A} \cdot \mathbf{z},\ \mathbf{z} = \hat{\mathbf{y}}(t) - \mathbf{y}_c,\ \mathbf{A} \in \mathcal{M}^{p \times p}, \quad (6)$$

where $\mathbf{y}_c$ is the control set point. Discretizing Eq. (6), we obtain:

$$\mathbf{z}_{k+1} = e^{-\mathbf{A} \cdot \Delta t} \cdot \mathbf{z}_k \quad (7)$$

and finally

$$\mathbf{y}^*_{k+1} = e^{-\mathbf{A}\cdot\Delta t}(\mathbf{P_y}\cdot\hat{\mathbf{v}}_k - \mathbf{y}_c) + \mathbf{y}_c. \tag{8}$$

A variable set point can be plugged into y_c in order to achieve output trajectory tracking. However, the reference system dynamics act as a low pass filter, thus limiting the tracking capabilities of the control strategy.

The next step is to be able to select amongst all the possible inputs $\mathbf{u}_k \in \mathcal{U} = [\mathbf{u}_{min}, \mathbf{u}_{max}]$ which one will minimize the distance between $\mathbf{y}^*_{k+1}$ and $\hat{\mathbf{y}}^-_{k+1}$. In other words, the following minimization problem has to be solved:

$$\mathbf{u}_k = \arg\min_{\mathbf{u}_k\in\mathcal{U}} \left\|\hat{\mathbf{y}}^-_{k+1} - \mathbf{y}^*_{k+1}\right\|_2, \tag{9}$$

where

$$\hat{\mathbf{y}}^-_{k+1} = \mathbf{P_y}(pr(\mathbf{P_x}\cdot\hat{\mathbf{v}}_k, \mathbf{u}_k) - \varepsilon_k). \tag{10}$$

In order to limit the computational complexity of that step, a piecewise linear approximation of $\left\|\hat{\mathbf{y}}^-_{k+1} - \mathbf{y}^*_{k+1}\right\|_2$ is used instead. The search for the minimizing $\mathbf{u}_k$ value is carried out by iteratively computing the orthogonal projection of $\mathbf{y}^*_{k+1}$ on each linear part of that approximation. If that projection exists, then the minimizing value is computed by interpolation over that part. In order for that technique to attain an approximation of the minimum, the prediction map must be quasi-convex, limiting the kind of systems this control strategy can handle to those that are linearizable.

The last operation is to compute a prediction of the next state/outputs for the next error estimation step:

$$\hat{\mathbf{v}}^-_{k+1} = pr(\mathbf{P_x}\cdot\hat{\mathbf{v}}_k, \mathbf{u}_k) - \varepsilon_k, \tag{11}$$

which ends the on-line computations for time-step k.

This algorithm relies heavily on the prediction error not being too big, or varying too much in time. Hence, an implementation has to be careful not to over-correct the prediction errors. Usually, the state and outputs are not entirely known, which means an estimator has to be used, even if it is as simple as an Euler derivative estimator. Such estimators introduce delays that have to be taken care of in order for the error estimation step to be meaningful. In the same vein, the computed control input generally gets applied to the plant with a one sample time delay. This can be obtained by taking another prediction step before the objective computation and computing the control input to be used at step $k+1$ instead of step k. However, every added prediction step increases the error-induced noise, and even though it can be damped by adjusting the α parameter, that can lead to a degradation of the controller performance.

3. AMB MODELING

As all the axes are similar if we ignore couplings, we only need to model one axis in order to synthesize the controller. Hence, we are only considering the x axis. There are two possible control inputs that can be used for an axis: either the coils currents i_{xp} and i_{xm}, or the coils excitation voltages E_{xp} and E_{ym}. In our case, the currents are used by the experimental platform. As we will see, that makes the model of an axis a second-order system. However, the proposed controller can also be used with excitation voltages as control inputs, as simulations show in [BDMV05].

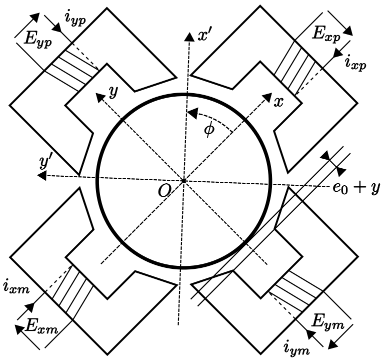

Fig. 2. $x - y$ control plane

The model needed for control synthesis does not have to be exact, as modeling errors are taken care of as external perturbation. However, if they are known, incorporating constant perturbations such as gravity into the synthesis model from the start is a good idea.

Considering only the x axis, as shown on Fig. 2, the shaft acceleration can be written as:

$$m\ddot{x} = F_1 - F_2 + F_p. \tag{12}$$

Were F_1, F_2 are the electromagnetic forces generated by the coils, and F_p a constant, additive perturbation force such as gravity. Neglecting magnetic saturation and hysteresis effects, we have:

$$F_1 = \frac{\lambda_1 {i_{xp}}^2}{2(e_0 - x)^2} \text{ and } F_2 = \frac{\lambda_2 {i_{xm}}^2}{2(e_0 + x)^2}, \tag{13}$$

where e_0 is the nominal air-gap of the axis. The λ_1 and λ_2 parameters depend on the geometry of the coils and the shaft. As each axis is composed of two symmetrical actuators, we can consider both of them equal to λ_x. If we combine Eq. (12) and (13), we obtain a model that is not linearizable when the shaft is at the origin and the currents equal zero [CDC96]. Traditionally, in order to obtain a linear model suitable to the application of linear control theory, a physical linearization is performed by introducing a bias current I_0 into the coils. That premagnetization current creates a constant magnetic flux in both actuators. The flux build-up time is thus eliminated and a nearly linear response on magnetic flux is obtained by varying the inputs currents around that value as the shaft makes small displacements along the axis (Fig. 3(a)).

However, using a bias current is quite energy-consuming since both coils are always active. It is thus more efficient to base the control on the nonlinear model where only one coil is active at any given time (Fig. 3(b)). In this mode of operation, currents i_{xp} and i_{xm} are mutually exclusive. They can be replaced by a single control input i_x as follows:

$$i_{xm} = \begin{cases} -i_x & \text{if } i_x < 0 \\ 0 & \text{otherwise} \end{cases} \text{ and } i_{xp} = \begin{cases} i_x & \text{if } i_x > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

Rewriting Eq. (13), we obtain:

$$F_1 = -F_2 = \frac{\lambda_x \operatorname{sign}(i_x){i_x}^2}{2(e_0 - \operatorname{sign}(i_x)x)^2}. \tag{15}$$

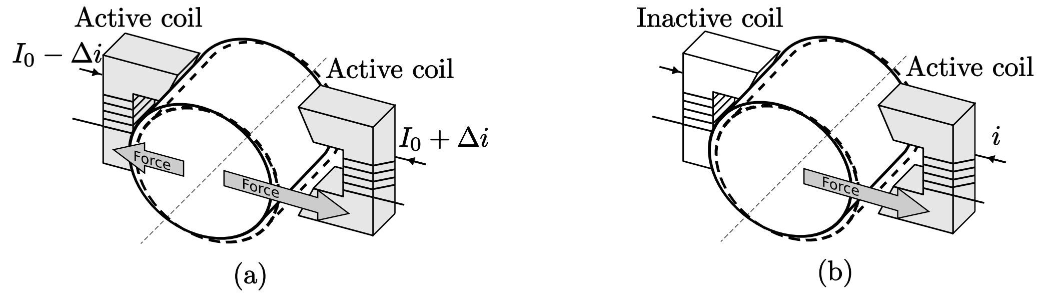

Fig. 3. Linear (a) vs. nonlinear (b) AMB operation

Eq. (14) implies that at any given time, either F_1 or F_2 vanishes. Finally the model used for control synthesis becomes:

$$m\ddot{x} = \frac{\lambda_x \, \mathrm{sign}(i_x) {i_x}^2}{2(e_0 - \mathrm{sign}(i_x)x)^2} + F_p. \tag{16}$$

4. EXPERIMENTAL SETUP

The device used for experiments is a laboratory AMB test bench (Fig. 4) from MECOS-TRAXLER AG, model miniVS. It is is built around a magnetic suspension unit made of a rotor and a stator, one industrial PC featuring a Pentium IV processor running The MathWorks xPC Target real-time executive linked to a desktop computer running MATLAB and Simulink through an Ethernet network, a power electronics control interface, a signal conditioning interface, and a power supply. The magnetic suspension device is composed by a rotor shaft driven by an electric motor, and five magnetic bearings controlling the shaft position along the $x_{1,2}$, $y_{1,2}$ and z axes. The shaft can rotate along the z axis. Those magnetic bearings are current-controlled by means of a high-gain controller built in the power electronics. Current references are set through a digital-to-analog converter interface card from National Instruments in the PC. They are fitted with magnetic transducers in charge of measuring the shaft position, and current-sensing devices. The sensor outputs are acquired by the industrial PC through the signal conditioning device and an analog-to-digital converter card also made by National Instruments. The magnetic transducers precision is limited to 1 μm by the acquisition chain. In order to simplify the development of the real-time controller implementation, we are using The MathWorks xPC Target rapid prototyping system. The controller is described using Simulink on the desktop PC through a model mixing built-in blocks when possible and custom C language function blocks for the prediction and control input computation parts of the controller. Indeed, optimizing these parts make sense as they form the bulk of the control algorithm. After code generation and compilation stages, the control application is loaded on the industrial PC and run, using a 700 μs cycle time. Table 1 summarizes the experimental test bench parameters.

A few implementation issues have to be taken into account. The main issue is the lack of shaft translational-speed sensing. Speed measurements have to be derived from the position sensors outputs through Euler derivation. However, a property of this class of derivative estimators is to produce an output delayed by half a time-step. That delay is difficult to handle, as the prediction map only contains full time-step evolution information. The approach used here is to use a two-time-step Euler derivative estimator, which delays its output for a full time-step. Using together the delayed speed estimation, a one-step-delayed position measurement, and knowledge of the currently applied control input, the controller computes the desired current control by doing two successive prediction steps. Those two prediction steps lead to an increase of error-induced noise, which is mitigated by setting the noise correction damping coefficient to $\alpha = .5$.

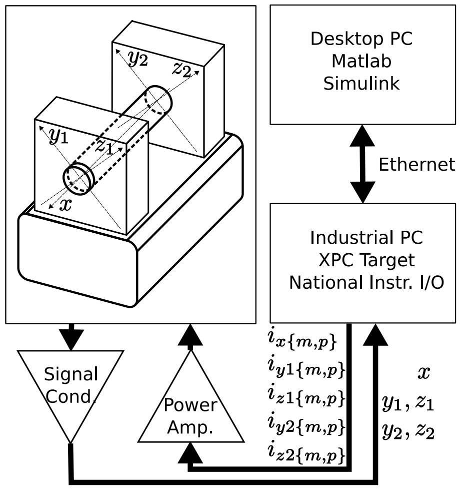

Fig. 4. Experimental setup

Table 1. AMB parameters

Parameter	Value	Description
m	3.097 kg	Shaft mass
e_0	0.410^{-3} m	Nominal air-gap
$\lambda_{\{x,y\}}$	1.210^{-6} mH · m	λ for control planes
λ_z	2.3210^{-6} mH · m	λ for the x axis
$\dot{\phi}_{max}$	30000 rpm	Max. shaft angular speed
I_{max}	6 A	Max. coil current
V_{max}	50 V	Max. coil voltage
F_p	$m \times \sqrt{2}/2 \times 9.81$ N	Gravity for axes y and z
F_p	0 N	Gravity for axis x

5. SIMULATION AND EXPERIMENTAL RESULTS

All the results are obtained by making the AMB x axis follow a 10 Hz square position reference trajectory with an amplitude of $2/3 \cdot e_0$ m. The algorithm used for all experiments and simulation is the same implementation of the control strategy presented in section 2. First, Fig. 5 shows simulation results obtained on the model derived in

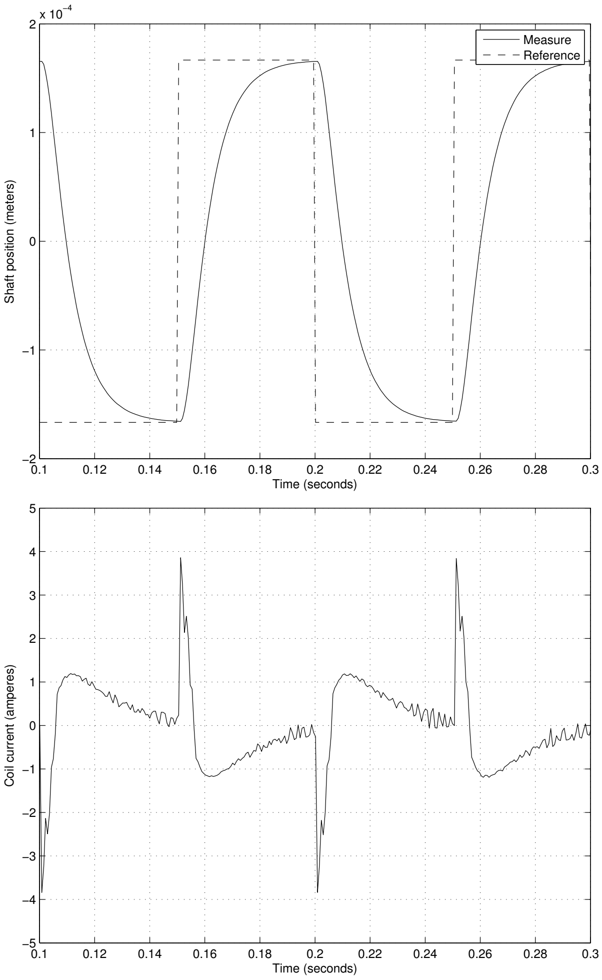

Fig. 5. Square position reference tracking - Simulation

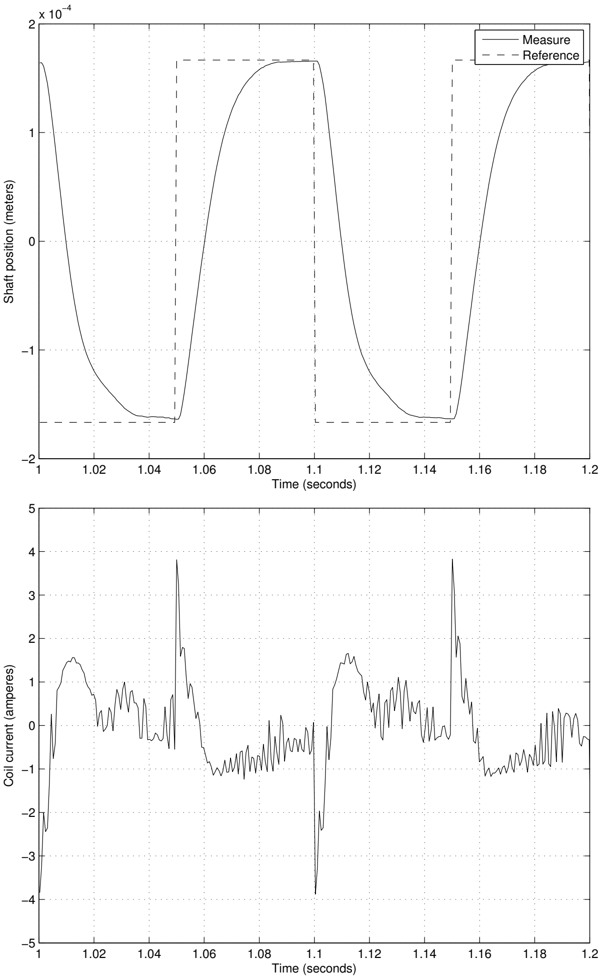

Fig. 6. Square position reference tracking - Experiment without gravity

section 3. The speed measurement $\dot{x}$ is obtained through an Euler derivative approximation from the position x. The current input i_x is computed one time-step ahead in order to account for the digital-to-analog conversion delay observed on the real experimental bench. As we can see, the shaft position follows the reference trajectory, but the square wave is filtered by the reference linear system used for the controller design. That linear system has a canonical second-order transfer function with a natural frequency $w_0 = 200$ Hz and a damping ratio $\zeta = 1.1$.

The noise observed on the current input is twofold. Firstly, the position measurement and current input are quantized, which adds noise to the state estimation and thus to the prediction. Secondly, since there is no gravity on the x axis, it is not controllable when the position and current are zero. That means that large control inputs will have small effects on shaft speed when it is at the origin. In order to keep the shaft stabilized, the control system has to produce large current excursions around zero that tend to lead to prediction-error overcompensation. That in turns creates some noise on the control input. In order to alleviate that behavior, some premagnetization could be used when x is close to the origin, making the shaft controllable.

Fig. 6 shows a real-life experiment in the same conditions as the simulation. As we can see, the simulation did predict the experiment quite well. However, the current input is noisier, which can be attributed to errors in the coil parameters used during the predictor construction, and especially the value of λ_x. Moreover, the current input is not exactly symmetrical, which is indicative of a discrepancy between the two coils of that axis. Note however that the observed noise is not significantly higher than what is obtained in [De 98] with a non-linear PID controller on the same setup.

Finally, the same experiment is conducted with the AMB making a 45° angle to its normal, horizontal position. The x axis is now subject to a perturbation due to gravity which has not been foreseen during control synthesis. Fig. 7 shows that the perturbation is rejected, the current being shifted away from zero as needed to compensate. The current noise is now significantly lower than that of Fig. 6, as the origin is now stabilizable with a non-null current, and is thus a controllable state.

6. CONCLUSION

A novel predictive nonlinear controller applied to the stabilization of an active magnetic bearing without premagnetization has been presented. The only work that has been needed on the experimental system is to derive a sufficiently precise simulation model in order to automatically synthesize the controller. Experimental results

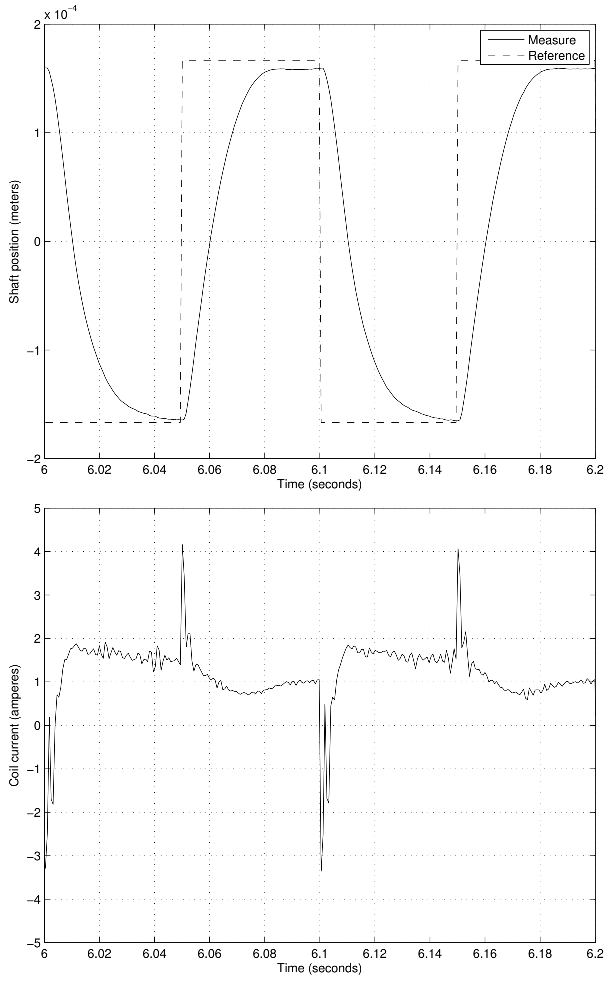

Fig. 7. Square position reference tracking - Experiment with gravity

show that this controller is able to make the device exhibit a predefined linear behavior while compensating for modeling errors and external perturbations, thus acting as an implicit linearizing controller. Its tracking capabilities have also been illustrated. That controller seems generic enough to be applied to other systems, while still maintaining desirable properties. However, being based on heuristics, its convergence properties have not been studied. In particular, the error correction step is essential to the algorithm and needs further refinement. Whether a well-chosen reference system could be used to constrain the state and/or outputs within specified bounds remains to be investigated.

REFERENCES

[BDMV05] S. Bonnet, J. De Miras, and B. Vidolov. Nonlinear implicit control of a magnetic bearing without premagnetization. In *Int. Conf. on Computational Intelligence, Control and Automation*, volume 2, pages 432–437, Vienna, Austria, 2005. IEEE Computer Society.

[CDC96] A. Charara, J. De Miras, and B. Caron. Nonlinear control of a magnetic levitation system without premagnetization. *IEEE Trans. Contr. Syst. Technol.*, 4(5):513–523, September 1996.

[CDF+05] A. Chiba, D. Dorrell, T. Fukao, O. Ichikawa, M. Oshima, and M. Takemoto. *Magnetic Bearings and Bearingless Drives.* Elsevier, New York, NY, USA, March 2005.

[CK05] Min Chen and C. R. Knospe. Feedback linearization of active magnetic bearings: current-mode implementation. *IEEE/ASME Trans. on Mechatronics*, 10(6):632–639, December 2005.

[CKS93] D. Cho, Y. Kato, and D. Spilman. Sliding mode and classical controllers in magnetic levitation systems. *IEEE Control Syst. Mag.*, 13(1):42–48, 1993.

[DC98] J. De Miras and A. Charara. Unbalance cancellation with rotating reference control for a horizontal shaft. In 6^{th} *Int. Symp. on Magnetic Bearings*, pages 673–682, MIT, Cambridge, MS, USA, 1998.

[De 98] J. De Miras. *Contribution à l'élimination du balourd pour une machine tournante à paliers magnétiques actifs par des techniques de commande non linéaires.* PhD thesis, Université de Technologie de Compiègne, France, 1998.

[FMM90] M. Fujita, F. Matsumara, and M.Shimizu. $\mathcal{H}_\infty$ robust control design for a magnetic suspension system. In 2^{nd} *Int. Symp. on Magnetic Bearings*, pages 349–356, Tokyo, Japan, 1990.

[GP05] W. Graga and A. Pilat. Comparison of linear control methods for an amb system. *Int. J. of Appl. Math. and Comp. Sci.*, 15(2):245–255, 2005.

[GSDD05] G. C. Goodwin, M. M. Seron, and J. A. De Doná. *Constrained Control and Estimation – An Optimisation Approach.* Computer and Control Engineering. Springer–Verlag, London, 2005.

[HWH07] J. Huang, L. Wang, and Y. Huang. Continuous time model predictive control for a magnetic bearing system. *Progress In Electromagnetics Research Symposium Online*, 3(2):202–208, 2007.

[MSB01] R. Moser, J. Sandtner, and H. Bleuler. Diamagnetic suspension system for small rotors. *Journal of Micromechatronics*, 1(2):131–137, January 2001.

[NY94] K. Nonami and H. Yamaguchi. μ-synthesis of a flexible rotor magnetic bearing system. In 4^{th} *Int. Symp. on Magnetic Bearings*, pages 73–78, Zürich, Switzerland, 1994.

[SBT94] G. Schweitzer, H. Bleuler, and A. Traxler. *Active Magnetic Bearings.* V.d.f. edition, Zürich, Switzerland, 1994.

[SHC98] H. F. Steffani, W. Hofmann, and B. Cebulski. A controller for a magnetic bearing using the dynamic programming of bellman. In 6^{th} *Int. Symp. on Magnetic Bearings*, Massachusets Institute of Technology, Cambridge, MA, USA, 1998.

[VMDC96] B. Vidolov, C. Melin, J. De Miras, and A. Charara. Two-rules-based fuzzy logic control and sliding mode control of an active magnetic bearing. In *FUZZ-IEEE'96*, volume 2, pages 1205–1209, New Orleans, LA, USA, 1996.