Model Order Reduction of an axial magnetic bearing

Jérôme TOMEZYK[a], Sebastian RODRIGUEZ[b], Yves DUPUIS[a], Joaquim DA SILVA[a]

[a] *SKF Magnetic Mechatronics, 2 rue des champs, 27950 Saint-Marcel, France, jerome.tomezk@skf.com*
[b] *Laboratoire PIMM, Arts et Métiers institute of technology, CNRS, CNAM, HESAM Université, 151 Boulevard de l'Hôpital, 75013 Paris, France, sebastian.rodriguez_iturra@ensam.eu*

Abstract

We introduce a model order reduction (MOR) method applied to the simulation of an axial active magnetic bearing (AMB) which is based on Finite Element discretisation of non-linear magneto-dynamic Maxwell's equations coupled with circuit equations and a moving part (rotor that can only move in one axis). In order to speed up the computation time, we studied a MOR based on the Proper Orthogonal Decomposition (POD) to reduce the order of the problem coupled with the Discrete Empirical Interpolation Method (DEIM) to deal with the nonlinearity. We introduce a way to consider the rotor's displacement, which do not require remeshing, and is compatible with such a method. The technique is evaluated on a 2D axisymetrical axial magnetic bearing model. This technique allows a speed up about a minimun of 2 times faster than the full size model.

Keywords: Finite Element Method, Non linear materials, Eddy currents, Proper Orthogonal Decomposition, Discrete Empirical Interpolation Method

1. Introduction

Due to the mechanical constraints, axial active magnetic bearing (AMB) are usually composed of solid stator and rotor part. Under voltage excitation varying at high frequency, Eddy currents are generated within axial AMB which highly limits its bandwidth. These Eddy currents coupled with the use of non linear magnetic materials make it necessary to run Finite Element (FE) simulation. However the computational time required to solve such a physical process can become prohibitive due to the complexity of the problem.

Model Order Reduction (MOR) techniques can be used to accelerate the computation time. Among the different techniques one can mention the use of the Proper Orthogonal Decomposition (POD), where the reduced basis is constructed “offline" after a Sigular Value decomposition (SVD) of FE solutions, in order to be used later in an fast "online" resolution step. In addition, MOR techniques have also been used for low-rank approximation of nonlinear terms. In particular, there is the Discrete Empirical Interpolation Method (DEIM), which evaluates the nonlinearity only on a few points in order to interpolate the remaining points, which, when combined with the POD method applied to the solution, allows to drastically reduce the computational costs of nonlinear solvers.

Different works have proposed the use of these methods to accelerate the simulation of a three-phase transformer (Henneron & Clenet 2015) or a rotating electrical machine (Montier et al. 2016). For this reason, in the present work we propose to use a POD-DEIM approach to construct a reduced model of an axial AMB with axial rotor displacement, magnetic saturation and eddy currents.

The paper is structured as follows. Section 2. introduce the governing equations of the considered problem. Section 3. gives details on how to construct the reduced model. Following Section 4. shows numerical results and finally, Section 5. provides conclusions and perspectives.

2. Problem formulation

The mathematical formulation used in the modeling of the axial AMB corresponds to the Magneto-Dynamic Maxwell's equation. First, the equation is described in the continuous form.

2.1 Continuous formulation

In order to solve Maxwell's magnetodynamic equations, the vector potential formulation is considered. The vector potential $\mathbf{A}$ is defined as $\mathbf{B} = \nabla \times \mathbf{A}$, such that the strong formulation is given as follows

$$\nabla \times (\nu(B)\nabla \times A) + \sigma \frac{\partial A}{\partial t} - i_1 N_1 - i_2 N_2 = 0 \tag{1}$$

$$\frac{\partial \phi_j}{\partial t} + R_j i_j = u_j \text{ for } 1 \leq j \leq 2 \tag{2}$$

where N_j corresponds to the unit current density, i_j the current, u_j the voltage and R_j the resistor of the j^{th} coil, additionally, ν corresponds to the magnetic reluctivity and σ the electrical conductivity. The equations (2) are the circuit equations, to impose a voltage at the terminal of the stranded coils, then ϕ_j correspond to the magnetic flux. The force is computed with the Maxwell's tensor. Figure 1 shows one half of the magnetic bearing, with two coils that are connected in series. This strong formulation is then discretize using the FEM, which is detailed in the following section.

2.2 Semi-discrete formulation

We note $X \in \mathbb{R}^{N_{FE}}$ the finite element discretization of A, then the equations (1)-(2) becomes

$$M^{\nu}(X)X + M^{\sigma}\frac{dX}{dt} - \sum_{j=1}^{N_{coil}} i_j N_j = 0 \tag{3}$$

$$N_j^T \frac{dX}{dt} + R_j i_i = u_j \text{ for } 1 \leq j \leq N_{coil} \tag{4}$$

The exponent T corresponds to the transposition. In the following, the influence of rotor displacement is detailed.

2.3 Movement of the rotor

To take into account the rotor displacement, the mesh is deformed in the same spirit as in (Hasan et al. 2018). Thus, for a displacement in the y direction, all the rotor and air nodes are translated, except for the airgap, where an affine transformation is performed to keep a conform mesh (see figure 1). As a consequence, M^{σ} is independent of the displacement since σ is null in the air and the rotor is just translated, not deformed. Therefore, the stiffness matrix can be divided in two parts

$$M^{\nu}(X) = M^{\nu_{NL}}(X) + M_z^{\nu_0}$$

with $M^{\nu_{NL}}(X)$ a matrix independent from the position y and $M_y^{\nu_0}$ the stiffness matrix in the air. In contrast to the techniques that require a remesh for each time step, this way to consider the movement allows to have the number of degree of freedom N_{FE} constant and does not require any projection map during the resolution of the temporal problem. The fully discretized model can now be described.

2.3.1 Fully discretized model

A backward Euler is used for the time discretization with constant time step δt and Newton-Raphson procedure for the non linearity is considered. In this sense, the residual is defined as follows:

$$R\begin{pmatrix} X^n \\ i_1^n \\ i_2^n \end{pmatrix} = \begin{pmatrix} M^{\nu_{NL}}(X^n)X^n + M_y^{\nu_0}X^n + M^{\sigma}\frac{X^n - X^{n-1}}{\delta t} - i_1^n N_1 - i_2^n N_2 \\ N_1^T \frac{X^n - X^{n-1}}{\delta t} + R_1 i_1^n - u_1^n \\ N_2^T \frac{X^n - X^{n-1}}{\delta t} + R_2 i_2^n - u_2^n \end{pmatrix} \tag{5}$$

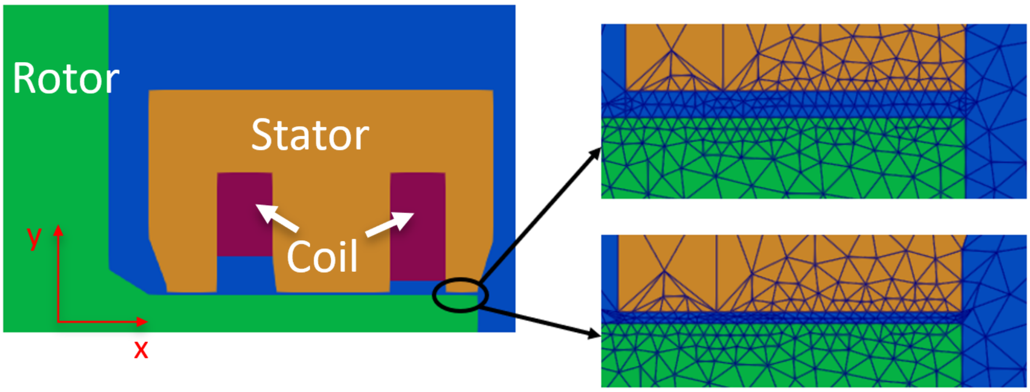

Figure 1: Half of axial magnetic bearing (in axisymmetric settings) with zoom on airgap in case of displacement.

Then, for each time step, one must find $\begin{pmatrix} X^n \\ i_1^n \\ i_2^n \end{pmatrix}$ such that $R\begin{pmatrix} X^n \\ i_1^n \\ i_2^n \end{pmatrix} = 0$. Then the Newton-Raphson procedure corresponds to iterate until convergence of the term

$$\tilde{X}_k = \tilde{X}_{k-1} - (J(\tilde{X}_{k-1}))^{-1} R(\tilde{X}_{k-1}) \tag{6}$$

with $\tilde{X}_{k-1} = \begin{pmatrix} X^{n-1} \\ i_1^{n-1} \\ i_2^{n-1} \end{pmatrix}$ and $J(\tilde{X}_{k-1}) = \dfrac{\partial R}{\partial \tilde{X}_{k-1}}$ the jacobian of R. We can write

$$J\begin{pmatrix} X^{n-1} \\ i_1^{n-1} \\ i_2^{n-1} \end{pmatrix} = \begin{pmatrix} M_y^{\nu_0} + (1/\delta t)M^{\sigma} & -N_1 & -N_2 \\ (1/\delta t)N_1^T & R_1 & 0 \\ (1/\delta t)N_2^T & 0 & R_2 \end{pmatrix} + \begin{pmatrix} J_{NL}(X^{n-1}) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{7}$$

with $J_{NL}(X) = \dfrac{\partial M^{\nu_{NL}}(X)X}{\partial X}$.

3. Reduced Model: POD and DEIM/G-POD

3.1 POD

The first step in reducing the model is to obtain a reduced basis that is complete enough to represent the solution of the problem properly. Thus, following the POD strategy, a matrix of snapshots is compute as follows:

$$M_{snap} = [X_1 \cdots X_N] \in \mathbb{R}^{N_{FE} \times N}, \tag{8}$$

which contains on each column the solution of the full order model at N time steps.
Then, the Singular Value Decomposition (SVD) is applied to this matrix to obtain the orthogonal matrices $U \in \mathbb{R}^{N_{FE} \times N_{FE}}$ and $V \in \mathbb{R}^{N \times N}$ and the rectangular diagonal matrix $\Sigma \in \mathbb{R}^{N_{FE} \times N}$ such that

$$M_{snap} = U \Sigma V^T \tag{9}$$

We can define the reduce basis $\Psi = [U_1, \cdots, U_{n_{POD}}]$. As the matrix Σ contains the singular values σ_i sorted in decreasing order, then we can choose n_{POD} such that $\dfrac{\sigma_{n_{POD}}}{\sigma_1}$ is smaller than a tolerance for example.

We can project the FE system onto this reduced basis, and then replace M^{ν}, M^{σ} and N_i by

$$M_R^{\nu_{NL}}(\Psi X_R) = \Psi^T M^{\nu_{NL}} \Psi,\ M_R^{\sigma} = \Psi^T M^{\sigma} \Psi,\ M_{y,R}^{\nu_0} = \Psi^t M_y^{\nu_0} \Psi \text{ and } N_{j,R} = \Psi^T N_j, \tag{10}$$

and then the new unknown of the reduced problem is $X_R \in \mathbb{R}^{n_{POD}}$. To compute $M_R^{\nu_{NL}}(\Psi X_R)$, we have to come back to the full order space, and build a full size matrix to then project it in reduced space, that is computational expensive.

3.2 DEIM/G-POD

The second step is then to reduce the computation time of the nonlinear term. To this end, we use DEIM, a method which have been proposed first in (Chaturantabut & Sorensen 2010). The basic idea is, in place of compute the full matrix $M^{\nu_{NL}}$ and then project it on the POD basis, we only compute a few number of rows of this matrix and then deduce the matrix via an appropriate linear operator. We start by taking a matrix of snapshot for the nonlinear term and take its SVD,

$$G_{snap} = [M^{\nu_{NL}}(X_1)X_1 \cdots M^{\nu_{NL}}(X_{N_{Newton}})X_{N_{Newton}}] = U_G \Sigma_G V_G^T, \tag{11}$$

where we save the nonlinear term for each time step and each step of the Newton-Raphson algorithm. Using the algorithm given in (Chaturantabut & Sorensen 2010), we get a reduce basis $\Phi \in \mathbb{R}^{N_{FE} \times n_{DEIM}}$ and a list $p \in \mathbb{R}^{n_{DEIM}}$ of the rows on which we will evaluate the nonlinear term. We can then build the row selection matrix $P \in \mathbb{R}^{N_{DEIM} \times N_{FE}}$

$$P = \begin{pmatrix} e_{p_1}^T \\ \cdots \\ e_{p_{n_{DEIM}}}^T \end{pmatrix} \in \mathbb{R}^{N_{FE} \times N_{DEIM}} \text{ with } e_i \in \mathbb{R}^{N_{FE}} \text{ such that } e_i(j) = \delta_{ij}.$$

Hence the approximation of the reduced non linear matrix is given by

$$M_R^{\nu_{NL}}(\Psi X_R) \simeq \Psi^T \Phi (P\Phi)^{\dagger} P M^{\nu_{NL}}(\Psi X_R)\Psi = M_{DEIM} M^{\nu_{NL}}(\Psi X_R)|_p \Psi \tag{12}$$

$$J_{NL,R} \simeq M_{DEIM} J_{NL}(\Psi X_R)|_p \Psi \tag{13}$$

where † corresponds to the pseudo-inverse, $M_{DEIM} \in \mathbb{R}^{n_{POD} \times n_{DEIM}}$ is a matrix which we have to compute once and $M^{\nu_{NL}}(\Psi X_R)|_p \in \mathbb{R}^{n_{DEIM} \times N_{FE}}$ (resp. $J_{NL}(\Psi X_R)|_p$) corresponds to the non linear matrix (resp. jacobian), but only for the row contains in p.

Remark: A good approach to increase the stability of the method is to use the Gappy-POD (G-POD) which only consists in evaluating the nonlinear term on more points than the basis contains vectors, ie $\Phi \in \mathbb{R}^{N_{FE} \times n_1}$ and $p \in \mathbb{R}^{n_2}$ with $n_2 > n_1$. It is for this generalization that there is a pseudo-inverse in the definition of M_{DEIM}.

3.3 Reduced model

Now, we have all the key ingredient to write the reduced problem. Like the full order model, we solve the POD-DEIM model using the Newton-Raphson algorithm to handle the nonlinear term, then for each time step, we have to solve

$$R_R \begin{pmatrix} X_R^n \\ i_1^n \\ i_2^n \end{pmatrix} = \begin{pmatrix} M_R^{\nu_{NL}}(\Psi X_R^n) X_R^n + M_y^{\nu_0} X_R^n + M^{\sigma} \dfrac{X_R^n - X_R^{n-1}}{\delta t} - i_1^n N_{1,R} - i_2^n N_{2,R} \\ N_{1,R}^T \dfrac{X_R^n - X_R^{n-1}}{\delta t} + R_1 i_1^n - u_1^n \\ N_{2,R}^T \dfrac{X_R^n - X_R^{n-1}}{\delta t} + R_2 i_2^n - u_2^n \end{pmatrix} = 0 \tag{14}$$

The algorithm 1 gives a description of the resolution scheme.

Algorithm 1 POD-DEIM

Input: Ψ, p, M_{DEIM}, t, z, U, X_0
Output: X_k, i_1^k and i_2^k for $1 \leq k \leq N$
  for $k = 1 : N$ **do**
    $\tilde{X} = \begin{pmatrix} X_{k-1} \\ i_1^{k-1} \\ i_2^{k-1} \end{pmatrix}$
    Compute $M_{y_k}^{\nu_0}$ and compute $M_{y_k,R}^{\nu_0} = \Psi^t M_{y_k}^{\nu_0} \Psi$. ▷ Take in account the displacement
    while Newton does not converge **do**
      Compute $M_R^{\nu_{NL}} = M_{DEIM} M^{\nu_{NL}}(\Psi \tilde{X}_{1:N_{FE}})|_p \Psi$ and $J_{NL,R} = M_{DEIM} J_{NL}(\Psi \tilde{X}_{1:N_{FE}})|_p \Psi$.
      $\delta X = -J_R^{-1} R_R(\tilde{X})$
      $\tilde{X} = \tilde{X} + \delta X$
    end while
    $X_k = \tilde{X}_{1:N_{FE}}$, $i_1^k = \tilde{X}_{N_{FE}+1}$, $i_2^k = \tilde{X}_{N_{FE}+2}$
  end for

4. Application

4.1 Fixed rotor

As a first test, rotor's movement is not considered. To get a sinusoidal flux, we use a simple P-controller with saturation to control the voltage ($Vdc = 70V$ and $f_s = 1kHz$) and the simulation starts from a steady state (i.e., $i_1 = i_2 = 15A$ for the initial timestep). We can see on the figure 2 the flux, current and force for the FE and the POD-DEIM solutions. Computation time is about 38m18s for the full order model while it is of 35m for the POD model and 7m15s for the POD-DEIM model.

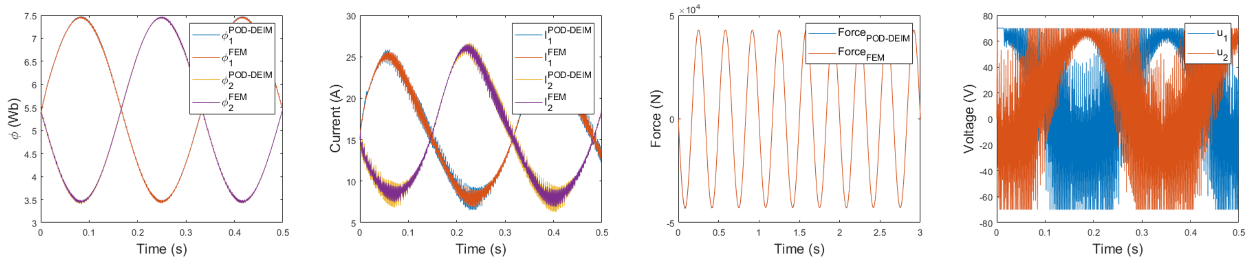

Figure 2: Flux, current and force ($n_{POD} = 15$ and $n_{DEIM} = 90$)

4.2 Constant voltage

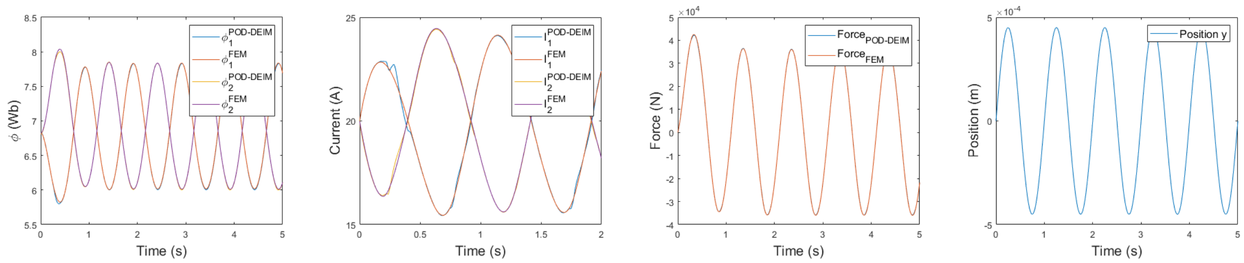

Figure 3: Flux, current and force ($n_{POD} = 10$ and $n_{DEIM} = 90$) for fixed voltage and sinusoidal displacement

In this case, the voltages stay at the same level during the whole simulation, such that $u_j = 20R_j$ $(j = 1,2)$ and the rotor displacement follows a sinusoidal curve. Figure 3 shows the results for this case. The computation time is here about 53m14s for the full order model, 45m47s for the reduced model with POD and 11m14s for that with POD-DEIM.

4.3 Concatenation of POD's basis

In this example, we combine both of the POD basis in order to obtain a new one that can handle a new scenario. For this test, we control the voltage to get a sinusoidal force ($Vdc = 70V$ and $f_s = 1kHz$). As the set point is the force, we have to add a mass-spring equation to address a physical movement (without taking into account the stiffness contact between stator and rotor parts). The DEIM parameters are computed using the data from the current test, which requires that n_{DEIM} is large to be able to reach the end of the simulation. The computation time is 54m15 for the full model, 47m56s for the POD and 27m28 for the POD-DEIM (figure 4).

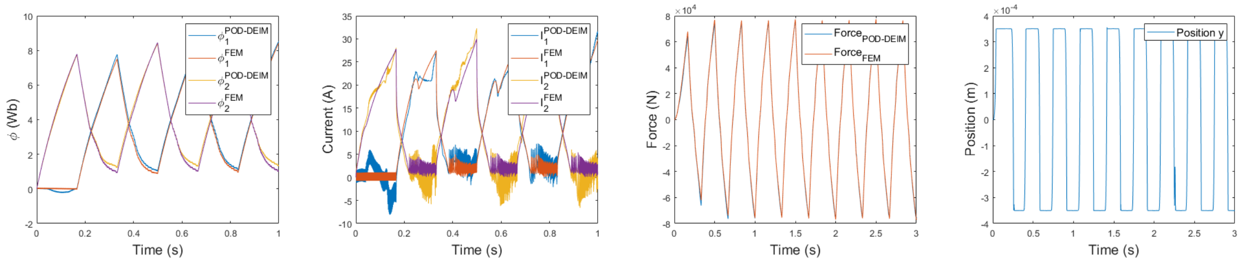

Figure 4: Flux, current, force and position ($n_{POD} = 25$, $n_{DEIM} = 1000$)

5. Conclusion and perspective

In this paper, we introduced a MOR technique based on the POD-DEIM applied to the simulation of an axial AMB, which takes into account Eddy current, magnetic saturation and rotor's displacement. We showed that this method allows to accelerate considerably the resolution time. However, some instabilities can appear when using the DEIM-GPOD, which makes the Newton-Raphson's algorithm fail to converge. These instabilities become more and more important when considering the rotor's displacement. In other hand, the POD approach alone is quite robust, but allow a speed up about only 10%.
In future work, we will try to circumvent this issue by updating the DEIM basis/linear operator during the simulation.

References

Chaturantabut, S. & Sorensen, D. C. (2010), 'Nonlinear model reduction via discrete empirical interpolation', *SIAM Journal on Scientific Computing* **32**(5), 2737–2764.

Hasan, M. R., Montier, L., Henneron, T. & Sabariego, R. V. (2018), 'Matrix interpolation-based reduced-order modeling of a levitation device with eddy current effects', *IEEE Transactions on Magnetics* **54**(6), 1–7.

Henneron, T. & Clenet, S. (2015), 'Model-order reduction of multiple-input non-linear systems based on pod and dei methods', *IEEE Transactions on Magnetics* **51**(3), 1–4.

Montier, L., Henneron, T., Clénet, S. & Goursaud, B. (2016), 'Transient simulation of an electrical rotating machine achieved through model order reduction', *Advanced Modeling and Simulation in Engineering Sciences* **3**(1), 1–17.