Parallelisation of quasi-static simulations of moving system: Application to the optimization of magnetic micro switch

Benoît Delinchant, Laurent Gerbaud, Frédéric Wurtz

Parallelization of quasi-static simulations of moving system: Application to the optimization of magnetic micro switch

B. DELINCHANT, L. GERBAUD, F. WURTZ

G2ELab, Grenbole Electrical Engineering Laboratory
BP 46 - 38402 Saint-Martin-d'Hères Cedex FRANCE
E-mail: Benoit.Delinchant@G2ELab.inpg.fr

***Abstract* — The aim of this paper is to improve simulation time of a moving system, especially for optimization needs. The solution provided here is applicable when displacement is known before simulations and very interesting if several computers are available for task parallelization. The proposed method is applied for the optimization of magnetic micro actuator switching time.**

I. INTRODUCTION

System design often needs optimization procedure to reach specifications and to improve the solution regarding competitors. For dynamic systems, a temporal simulation is required to extract values like average value, maximal value, final simulation value, etc. which are then constrained or optimized.

II. HOW TO REACH PARALLELIZATION

A. Time domain simulation

In the case of moving system, the dynamic equation (1) defines the differential equation for the position state to solve (2) (here in one dimension).

$$F_x = m.\Gamma_x \quad (1)$$

$$F_x = m.\frac{d^2}{dt^2}x \quad (2)$$

Where Fx is the static force along x axis, m the mass, Δx the acceleration, and x the position.

To solve this simple differential equation, 2 initial values are required, x_0 and v_0, the initial position and initial velocity. In our application domain, force computation depending on the position is the more consuming time than integration of the differential equation.

The issue of a simulation, like any iterative method, is the dependence between iterations. In this formulation, the dependence between time steps requires to wait computation of the next position, to compute next force value.

B. Space domain simulation

Considering force constant during a step, the symbolic integration of equation (2) is easy, indeed, with initial values (v_0, x_0) it gives the following result (3) for the first step, which can be written simply by (4).

$$x_1 = \frac{F_0}{2.m}.\Delta t^2 + v_0.\Delta t + x_0 \quad (3)$$

$$\frac{F_0}{2.m}.\Delta t^2 + v_0.\Delta t - \Delta x = 0 \quad (4)$$

From this expression, Δt can be extracted as defined by second order polynomial solutions of (4).

$$\Delta t = \frac{m}{F_0}.\left(-v_0 \pm \sqrt{{v_0}^2 + \frac{2F_0.\Delta x}{m}}\right) \quad (5)$$

From this expression, a space-domain simulation can be done (descritizing space instead of time). It gives time depending on position instead of position depending on time. It is the required to know positions of moving part, but most of time, moving parts are guided in the structure (allowing few degree of freedom like 1-axis rotation or 1-axis translation).

C. Parallelization

Algorithm:

1. Define N positions (x)
2. Computes N forces (F) on N computer in parallel
3. Initialization of velocity (v) and time (t) vectors
4. For i=1 to N : compute equation (6) and (7)

$$t_1 = \frac{m}{F_0}.\left(-v_0 + \sqrt{{v_0}^2 + \frac{2F_0.\Delta x}{m}}\right) + t_0 \quad (6)$$

$$v_1 = \frac{F_0}{m}(t_1 - t_0) + v_0 \quad (7)$$

In many applications, step '2' is the only step which consumes time, even with parallelizing. Then total simulation cost is divided by N.

III. APPLICATION TO THE OPTIMIZATION OF ELECTROMAGNTIC MICRO SWITCH

A. Electromagnetic switch design

The application can not be detailed in a 2 pages abstract. It can just be mentioned that design specifications are to constrain sizes and performances such as maximal choc on fixed positions and to minimize switching time. This last specification requires a simulation during an actuation from one position to the other. This MEMS is based on an electromagnetic actuation detailed in [1].

B. *Simulation code distribution*

From modelling equations, software component dedicated to optimization is built [2] to ensure diffusion, composition and reuse. Encapsulation of simulation code is very useful especially for our needs of parallelization with the distribution of simulation code to distant computers. A computer program acting as a service has been deployed on each computer of our network, waiting for a client call. A client is a program able to ask for available servers and able to send the software component to distant computers.

C. *Simulation code parallelization*

Each computer is able to compute force acting on the moving magnet depending on a predefined position. The client software (Fig 3.) is then acting as a supervisor; it defines N positions, and calls a computation for each of the N parallel computers. A listener mechanism is used to monitor then end of each computation and then to go to the next step of space-domain discretizing algorithm.

IV. RESULTS

A. *Parallelization results*

20 force computations have been done to compute force according to the position. The time average of force computation is about 5.7 seconds on a standard PC (Pentium 1.6 GHz – RAM 1Go). In the case of "time discretizing" method, 20 computations leads to a total time of 115 seconds. In the case of parallelization, it depends on the number of available computers. Parallelizing on two computers leads to about 68 seconds, which is not equal to 115/2=57.5 because network management has a cost. By extrapolation, if 20 computers are available for 20 computations, it will lead to a total computation time of about 7s. Indeed, as it can be seen on figure 1, when a computation is not distributed the time cost is equal to 5.7s, but parallelization add a time cost to reach approximately 7 seconds per force divided by computers number. It can be noticed that our experiment shows big value dispersions in the case of parallelizing on few computers.

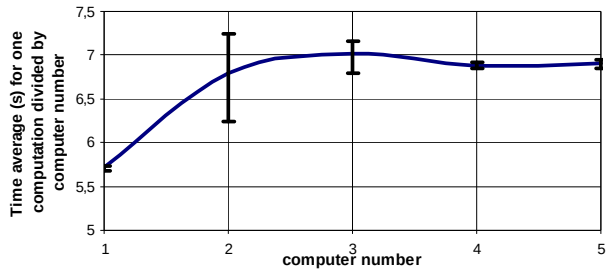

Fig. 1. Time average (with min/max) of a force computation depending on computer number parallelizing, divided by computer numbers

B. *Space integration results*

On figure 2, "time discretizing" and "space discretizing" curves are very close but not superposed. The difference comes from the hypothesis of the integration which considers a constant force between points.

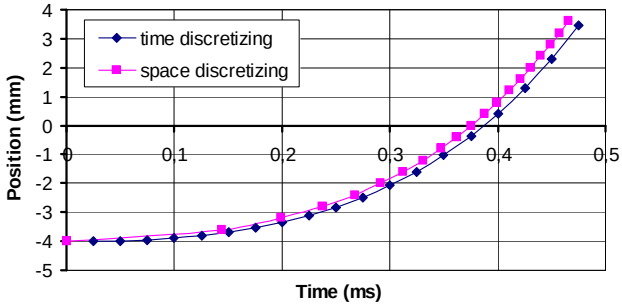

Fig. 2. space and time integration for time and space discretizing.

To improve the accuracy of "space discretizing" integration result, an interpolation procedure can be done on computed array of forces in order to decrease the integration step. To compare this new result, a fine simulation was done considered now as the reference results. In figure 3, it is shown on a zoom that last two curves are similar.

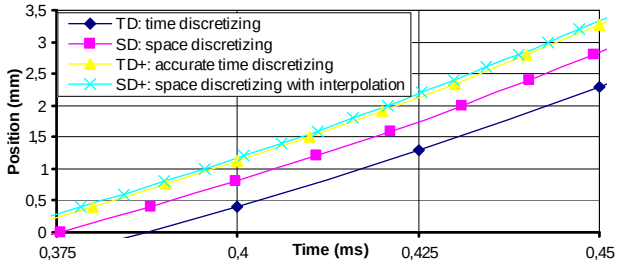

Fig. 3. Zoom on space and time integration for time and space discretizing adding a more accurate time and space discretizing.

TABLE 1: COMPUTATION TIME FOR 20 FORCE COMPUTATIONS

	Characteristics	**duration (s)**
TD :	Step=0.25ms	115
SD :	Step=0.4mm, 5 Computers	28
TD+ :	Step=0.10ms	226
SD+ :	Step=0.4mm, 5 Computers 1 interpolation point	28

V. CONCLUSIONS

The design of engineering systems, require optimization procedure which needs very fast models. This paper has shown that a drastic reduction of extracting simulation post-processing results is possible. It has been shown that a parallelization procedure can be done easily with a software component architecture. In out application, a ratio reduction of 8 was obtained only with 5 computers parallelizing.

We would like to acknowledge Dhia SEFSAFI for its work on software development during this study.

REFERENCES

[1] J. Stepanek, H. Rostaing, J. Delamare, O. Cugat, "Fast dynamic modeling of magnetic micro-actuator", Journal of Magnetism and Magnetic Materials, Volumes 272-276, May 2004, Pages 669-671.

[2] B. Delinchant, D. Duret, L. Estrabaut, L. Gerbaud, H. Nguyen Huu, B. DuPeloux, H.L. Rakotoarison, F. Verdiere, F. Wurtz, "An optimizer using the software component paradigm for the optimization of engineering systems", COMPEL vol.26, nb.2, 2007