Automatic Differentiation for Optimization of Dynamical Systems

Petre Enciu, Laurent Gerbaud, Frédéric Wurtz

Automatic Differentiation Applied for Optimization of Dynamical Systems

Petre Enciu, Laurent Gerbaud, and Frederic Wurtz

Grenoble Electrical Engineering Lab. (G2ELab), UMR5269, CNRS/UJF/INPGENSE3, Domaine Universitaire, BP 46, F-38402 Saint Martin d'Hères Cedex, France

Simulation is ubiquitous in many scientific areas. Applied for dynamic systems usually by employing differential equations, it gives the time evolution of system states. In order to solve such problems, numerical integration algorithms are often required. Automatic differentiation (AD) is introduced as a powerful technique to compute derivatives of functions given in the form of computer programs in a high-level programming language such as FORTRAN, C, or C++. Such technique fits perfectly in combination with gradient-based optimization algorithms, provided that the derivatives are evaluated with no truncation or cancellation error. This paper intends to use AD employed for numerical integration schemes of dynamic systems simulating electromechanical actuators. Then, the resulting derivatives are used for sizing such devices by means of gradient-based constrained optimization.

***Index Terms*—Automatic differentiation (AD), dynamic systems, gradient constrained optimization.**

I. Introduction

SIZING by optimization is today of major interest since it provides a fast and reliable way to achieve, with low manufacturing costs, desired performances for suboptimal products usually by means of minimizing a cost function.

We are particularly interested by constrained gradient-based optimization using sequential quadratic programming (SQP) algorithms [1]. Such algorithms require error-free derivatives of the objective function and constraints. This may be at the origin of serious problems provided that often such functions may result from complex numerical algorithms. In this paper, we are particularly interested by those objective and constrained functions resulting from numerical integration of initial value problems (IVPs) of ordinary differential equations (ODEs) simulating the motion of an active body actuated by the electromagnetic force in the context of electromechanical actuators.

A good compromise in the optimization context is automatic differentiation (AD) that is a term applied for a technique able to compute derivatives of functions described by computer programs. This paper only uses AD for sizing dynamical actuators by means of gradient-based constrained optimization. In particular, AD will be applied using ADOL-C tool [2].

II. Optimization Problem

This paper considers the particular design optimization problem dealing with actuators rapidity. The paper defines the response time t_r as the instant when a specified state variable x_i reaches a prescribed threshold state value $\tilde{x}_f$. So the response time is implicitly defined, as stated in

$$x_i(t_r) = \tilde{x}_f. \tag{1}$$

The optimization problem is formulated by means of minimizing an objective function J by respecting additional inequality (g) or equality (h) constraints. The purpose is to find the value of design parameters set $P \in \mathbb{R}^p$ that satisfies such problem. Generally, such parameters determine the device geometry and shape.

The optimization problem is then formulated as

$$\begin{aligned} &\min J\left(x(t_r), P\right) \quad \text{together with} \\ &\quad g\left(x(t_r), P\right) \geq 0 \\ &\quad h\left(x(t_r), P\right) = 0. \end{aligned} \tag{2}$$

The difficulty in this paper is that the objective function J depends on the reached final states $x(t_r)$ solved from the ODE system modeling the device dynamics.

In this paper, two formulations of the state system are intentionally specified. The formulation in (3) represents an autonomous system, meaning that the time variable t does not appear in the differential equation, while the formulation in (4) refers to a nonautonomous system. The initial values x_0 are stated in (5)

$$\frac{dx}{dt} = \dot{x} = f(x, P) \tag{3}$$

$$\frac{dx}{dt} = \dot{x} = f(x, t, P) \tag{4}$$

$$x(0) = x_0, \qquad t = 0. \tag{5}$$

The gradient-based optimization algorithms applied for the optimization problem in (2) require the gradients of the objective function. The difficulty is to evaluate them like in (6) provided that the final states depend also on parameters

$$\nabla J = \left[\frac{\partial J}{\partial x}\right] \cdot \left[\frac{\partial x}{\partial P}\right] + \left[\frac{\partial J}{\partial P}\right]. \tag{6}$$

Another necessity is to compute the partial derivatives of the response time with respect to parameter set. So, the response time may be carried out in optimization as a constrained parameter in addition to (2)

$$t_r < t_{r\max}. \tag{7}$$

Indeed, our goal could be formulated by means of finding the fastest optimal device using gradient-based constrained optimization.

Manuscript received December 22, 2009; accepted February 19, 2010. Current version published July 21, 2010. Corresponding author: P. Enciu (e-mail: petre.enciu@g2elab.grenoble-inp.fr).

Digital Object Identifier 10.1109/TMAG.2010.2044770

III. Automatic Differentiation

Currently, several techniques may be considered to deal with functions differentiation. Among them, the symbolic differentiation can evaluate error-free derivatives. The impossibility to differentiate more sophisticated algorithms implemented in codes rather than mathematical explicit expressions represents the major drawback of this method. Another traditional approach is the finite-differences method. Its major drawback is the setting of the derivative step that may carry out to a bad approximation of the partial derivative values. Note that the efficiency of gradient-based algorithms decreases strongly if errors are introduced in the derivatives values.

AD is introduced as a powerful technique that computes error-free derivatives, up to machine precision, of functions described as computer programs in high-level languages such FORTRAN or C/C++. In [3], a rich list of tools implementing AD is provided. Therefore, an AD tool could be a library that instruments a user program in order to be differentiated. Such tools require minor modifications on the initial source and they are using the operator overloading capabilities of certain programming languages such as C++ and FORTRAN95. Note that AD is not limited only to first-order derivatives. Some AD tools may efficiently compute high-order derivatives, Taylor coefficients, or other useful differential calculations (see, for instance, ADOL-C [2]).

In a previous paper [4], AD was already applied to differentiate sizing models of electromagnetic devices dealing only with analytical explicit expressions. The difference in this paper is that AD is applied to differentiate ODE solvers.

IV. AD of ODE Numerical Integration Schemes

In order to evaluate the partial derivatives in (6), one may use an AD tool over a numerical scheme integrating the ODE system in (3) or (4). The paper is then subject to two numerical integration strategies.

A. AD of Runge–Kutta Solvers

In this section, we propose to apply AD over an adaptive time step Runge–Kutta (RK) scheme as in

$$x_{t_{i+1}}(P) = x_{t_i}(P) + h_i(P) \cdot \dot{\bar{x}} \tag{8}$$

where $\dot{\bar{x}}$ is a slope estimation, $x_{t_i} = x(t_i)$ are the states previously computed at t_i instant, and h_i is the adaptive integration step which depends on design parameters. Recent studies [5] were carried out to differentiate such schemes. The difference in [5] is that the response time is prescribed in advance at a fixed value. Our approach intends to make use of it as a constrained design variable, carried out further in optimization, so, its corresponding derivatives are to be evaluated as explained before.

Differentiating such integration schemes is rather simple with AD. The RK scheme may be considered as an input program to the AD tool. Fig. 2 shows the entire procedure.

The considered AD tool outputs the user desired derivatives. We are particularly interested by those partial derivatives of the dynamic system response $x(t_r)$ with respect to design parameters. These partial derivatives are then exploited in (6) in order to compute the objective function gradient.

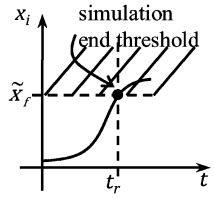

Fig. 1. Simulation end criterion.

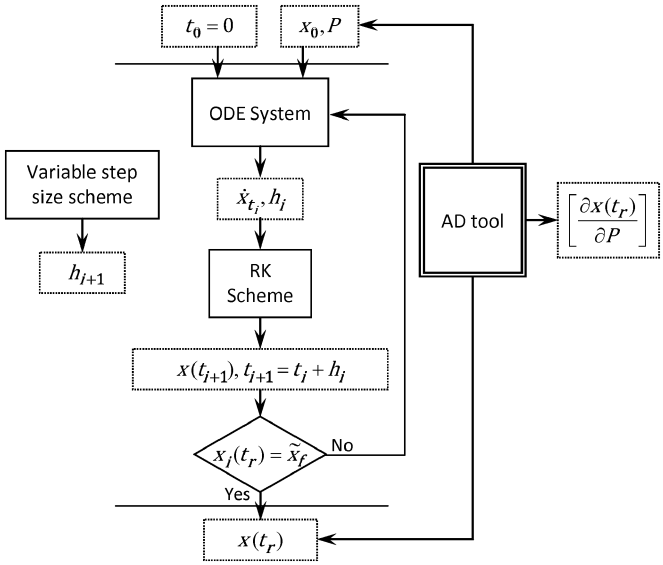

Fig. 2. AD of RK solvers.

Note that if a variable time step scheme is used, it is highly recommended to force the AD tool to avoid taking into account the partial derivatives of the time step, provided that h_{i+1} in Fig. 2 is also depending on parameters. Depending how the step size is computed, its partial derivatives may give different and wrong values for the total gradient of $x(t_r)$. For this aspect, [6] is a strong reference. However, we report that the time step differentiation deactivation is not representing a complicated task. It could be accomplished with minimal user knowledge about the used AD tool.

B. AD for Taylor Series Expansion Schemes

Here, truncated Taylor series (TS), as in [7], are applied to advance the solution of the ODE system in (3) or (4) over a time interval as in (9), supposing that f in (3) or (4) is sufficiently smooth

$$x_{t_{i+1}}(P) = x_{t_i}(P) + \frac{1}{1!}\frac{dx_{t_i}}{dt}h + \cdots + \frac{1}{n!}\frac{d^n x_{t_i}}{dt^n}h^n \tag{9}$$

where $(1/i!)(d^i x/dt^i)$ denotes the ith-order Taylor coefficient. In order to compute such coefficients, a high-order differentiation AD tool is required. For this reason, the paper considers ADOL-C tool. A better reason is that ADOL-C provides special drivers to deal directly with ODEs in its natural fashion.

Barrio [7] provides numerical solutions for adaptive time step schemes for ODE solvers using Taylor expansion. Interesting here is that AD is used to solve the dynamic system. The differentiation of such integration schemes is made by using special drivers implemented in ADOL-C.

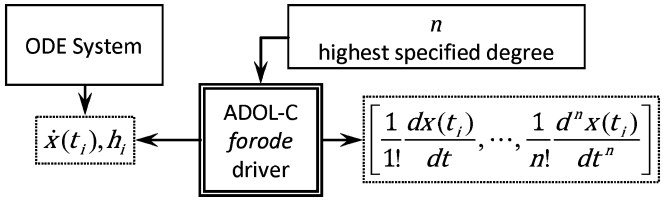

Fig. 3. ADOL-C tool applied for Taylor coefficients of ODE systems.

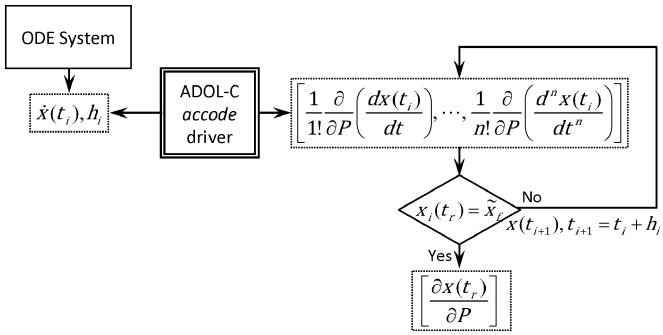

Fig. 4. ADOL-C tool applied for Taylor coefficient differentiation.

Fig. 3 shows the entire procedure applied in order to deal with the ODE system solving using TS expansion. Note that this procedure is ADOL-C specific. The Taylor coefficients are evaluated up to a user specified degree using the ADOL-C special driver, namely, *forode*. In its initial state, this driver deals only with autonomous system (3). However, in the nonautonomous case, like in (4), a special version of *forode* driver, implemented by the authors, is applied for Taylor coefficients evaluation.

Interesting here is that the input ODE system is described in its natural fashion as in (3) or (4) which is usually preferable for modeling dynamic systems. We are reporting no major difficulties in order to deal with high-order Taylor coefficients evaluation using ADOL-C tool.

The partial derivatives of the intermediate states $\partial x/\partial P$ have to be propagated starting from the initial values until the threshold is reached. In order to do this, the partial derivatives of the Taylor coefficients have to be evaluated. For this purpose, ADOL-C tool proposes a second driver, namely, *accode* helpful to evaluate cheaply these partial derivatives. The initial version of this driver is capable to compute the partial derivatives of the intermediate states (x_{t_i}) only with respect to initial values (x_0). However, the authors have implemented a special version capable to deal also with user defined design parameters.

The entire procedure that computes the desired derivatives as in the RK scheme case is shown in Fig. 4.

Obviously, as for RK-based integration schemes, it is also recommended to cancel the variable time step differentiation.

C. The Response Time

The response time is computed in this paper by solving the implicit relation

$$x_i(t_r) = \tilde{x}_f. \tag{10}$$

The major drawback to determine the response time is to find exactly the last step size necessary to satisfy this implicit formulation, provided that usually a variable time step scheme is preferred. The idea is to let the ODE solver naturally iterate until it slightly *exceeds* the imposed user specified value $\tilde{x}_f$. A regular implicit solver method may be then employed to determine with high accuracy the last step size. Note that in the paper we use the secant method, which converges in two-maximum three iterations, provided that the implicit equation (10) is quite linear in the region $[t_{r-1}, t_{r+1}]$.

The partial derivatives of the response time can be simply computed by considering the implicit function theorem discussed in [8]. Thanks to this theorem, the partial derivatives can be formulated as in

$$\nabla t_r = \left[\frac{\partial t_r}{\partial P}\right] = -\frac{1}{\dot{x}_i(t_r)}\left[\frac{\partial x_i(t_r)}{\partial P}\right]. \tag{11}$$

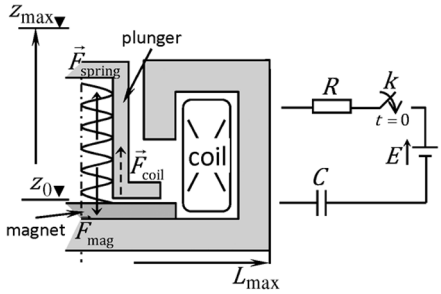

Fig. 5. Studied benchmark—electromechanical actuator.

V. Electromechanical Actuator Sizing

A. The Studied Benchmark

The benchmark in [9] (Fig. 5) of an electromechanical actuator modeling a circuit breaker is proposed for a sizing by gradient constrained optimization. In [9], Atienza *et al.* consider an optimization done by adjusting the design parameters by hand of the dynamic model in addition with a full SQP optimization of the static model. Obviously, this could be an error-prone and time-consuming approach. Our paper deals with SQP optimization of the dynamic part in a completely automatic fashion and with minimal user differentiation effort.

When the feeding circuit switch is turned off, the vacuum force produced by the magnet equilibrates the spring force. The simulation starts when the switch turns on. The electromagnetic force created by the coil cancels partially the magnet force. Consequently, the plunger will move, starting from initial position z_o toward the upper bound $z_{\max}$.

The dynamic system of the proposed device combines both equations of the electrical circuit feeding the coil and the movement equations. The states are the coil current (i) and its time derivative (di/dt), the plunger position (z), and its speed (s)

$$\begin{cases} i = \dfrac{di}{dt} \\ \ddot{l} = \dfrac{R}{L}\dfrac{di}{dt} - \dfrac{i}{L \cdot C} \\ \dot{z} = \begin{cases} 0, & \text{if } F \le 0 \\ s, & \text{otherwise} \end{cases} \\ \dot{s} = \begin{cases} 0, & \text{if } F < 0 \\ \dfrac{F}{m}, & \text{otherwise.} \end{cases} \end{cases} \tag{12}$$

F denotes the force acting on the mobile plunger, R, L, C are the coil parameters, and m is the total device mass. Note that

TABLE I
OPTIMIZATION SPECIFICATIONS

Variable	Constraint	Formula
Percussion energy at z_{max}	[0.12…10] [J]	$ms^2/2$
Response time t_r	[0…3.5] [s]	Implicity defined
Total force at z_{max}	15 [N]	
Shock resistance at z_0	[2000…10000] [m/s^{-2}]	$(F_{spring}$-$F_{mag})$/m
Total mass	minimize [kg]	

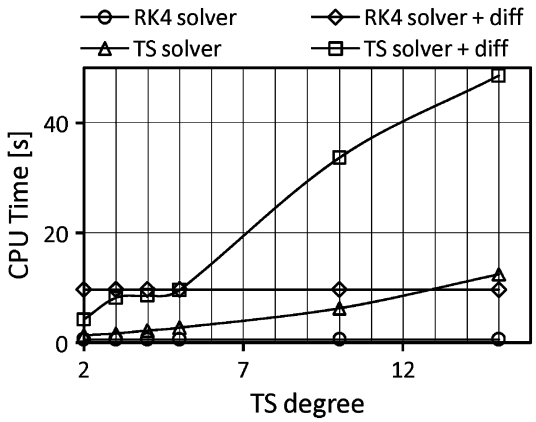

Fig. 6. CPU time performance of TS versus RK4.

the simulation is restricted only upward by imposing the conditions for the force. The residual force is evaluated in this paper by modeling the electromagnetic device by the equivalent reluctance circuit as in [9].

The response time is implicitly defined by the simulation stop criterion in (13), satisfied when the mobile plunger is bounded at $z_{\max}$

$$z(t_r) = \tilde{z}_f = z_{\max}. \tag{13}$$

B. Optimization Goal

A single objective optimization problem with three interval constraints and one imposed value rises from this particular case. These objectives are given in Table I.

The design parameters are represented by all geometrical and electrical parameters of the studied benchmark.

C. Integration Schemes Performances

In Fig. 6, the performances of TS expansion of various orders are compared to a regular RK scheme of fourth order, for solving the ODE system in (12). It results that the Taylor expansion is acceptable up to the fifth degree in terms of CPU time. We observed that its memory requirements are rather higher even at the second degree.

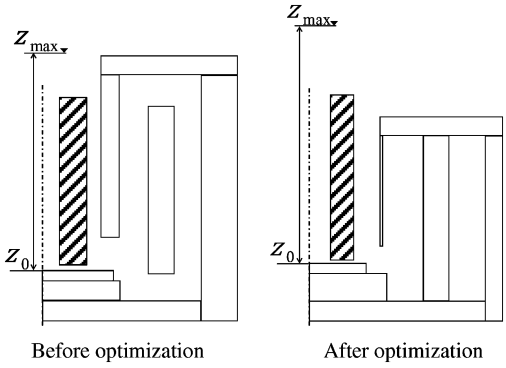

Fig. 7. Device configuration before and after optimization.

D. Optimization Results

The third-order TS expansion proved to be accurate enough for this particular problem. Its associated optimization results were identical with a fourth-order RK scheme. The gradient-based optimization made possible a reduction of 42% of total device mass and of 18% for the response time. The initial and final configuration of the studied device is given in Fig. 7.

VI. CONCLUSION

This paper presents a particular optimization problem on a benchmark dealing with state variables in ODEs. RK and Taylor expansion integration schemes are used to approximate these states. Both schemes are differentiated by employing AD in order to evaluate the highly accurate gradients needed by SQP algorithms. Identical results were obtained for TS starting from the third degree compared to RK of fourth order.

REFERENCES

[1] N. I. M. Gloud and D. P. Robinson, "A second derivative SQP method: Local convergence," Rutherford Appleton Lab., U.K., 2009, STFC.
[2] A. Griewank, D. Juedes, and J. Utke, "Algorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++," *ACM Trans. Math. Softw.*, vol. 22, no. 2, pp. 131–167, 1996.
[3] M. Bücker and P. Hovland, "Automatic differentiation," 2000 [Online]. Available: www.autodiff.org
[4] V. Fischer, L. Gerbaud, and F. Wurtz, "Using automatic code differentiation for optimization," *IEEE, Trans. Magn.*, vol. 41, no. 5, pp. 1812–1815, May 2005.
[5] A. Walther, "Automatic differentiation of explicit Runge-Kutta methods for optimal control," *Comput. Optim. Appl.*, vol. 36, pp. 83–107, 2006.
[6] P. Eberhard and C. Bischoff, "Automatic differentiation of numerical integration algorithms," *Math. Comput.*, vol. 68, no. 226, pp. 717–731, Apr. 1999.
[7] R. Barrio, "Performance of Taylor series method for ODEs/DAEs," *Appl. Math. Comput.*, no. 163, pp. 525–545, 2005.
[8] C. Coutel, F. Wurtz, and J. Bigeon, "A comparative study of two methods for constrained optimization with analytical models dealing with implicit parameters," *IEEE Trans. Magn.*, vol. 35, no. 3, pp. 1738–1741, May 1999.
[9] E. Atienza, M. Parrault, F. Wurtz, V. Mazauric, and J. Bigeon, "A methodology for the sizing and the optimization of an electromagnetic release," *IEEE Trans. Magn.*, vol. 36, no. 4, pp. 1659–1663, Jul. 2000.