Siconos: A Software Platform for Modeling, Simulation, Analysis and Control of Nonsmooth Dynamical Systems

Vincent Acary, Franck Pérignon

Siconos: A Software Platform for Modeling, Simulation, Analysis and Control of Nonsmooth Dynamical Systems

Vincent Acary, Franck Perignon, INRIA Rhône-Alpes, France

vincent.acary@inrialpes.fr, franck.perignon@inrialpes.fr

In this paper, a brief overview of the so-called nonsmooth dynamical systems and their links with hybrid systems is first proposed. In particular, Lagrangian mechanical systems with contact and friction or electrical systems with ideal components (diodes, MOS transistors ...) belong to this framework. Secondly, the Siconos software, dedicated to simulation of such systems, is presented, starting from its architecture and the way NSDS are modeled within the platform. To conclude, three representative samples are shown in order to illustrate the Siconos platform abilities.

1 A short introduction to Non-Smooth Dynamical Systems (NSDS)

The Non Smooth Dynamical Systems (NSDS) are a special kind of dynamical systems characterized by the non-smoothness of their time-evolution and of their constitutive equations. The instants of discontinuity of the state or its derivatives can be viewed as events for which the system changes in structure, in other terms as a transition. In this way, a NSDS combines features of continuous dynamical systems with the characteristics of finite automata. Thus, as a mixture of time-continuous dynamics and discrete systems, the NSDS can be viewed as a subclass of hybrid systems.

Like the term "hybrid", the "nonsmooth" one is not a precise definition, in the sense that it is not restrictive enough. A better way to define a coherent class of dynamical systems is to consider the mathematical nature of their solutions depending on the chosen formulation. This introduction aims at giving a flavor of mathematical properties and resulting numerical consequences, which are shared by NSDS. The term "non-smooth" is partly inherited from the extensive use of a well-recognized mathematical theory: the non-smooth analysis [2]. Since they are specialization of hybrid dynamical systems, new mathematical results for NSDS can be derived and new efficient simulation tools can be designed. The role of the Siconos platform is then to take advantage of these properties in order to provide general modeling and simulation tools for NSDS.

1.1 The example of constrained dynamics

A standard example of NSDS is a dynamical system of the form:

$$\dot{x} = f(x,t), \quad x \in \mathbb{R}^n, \ t \in [0,T] \tag{1}$$

subjected to a set of constraints on its state (the inequality is to be understood component-wise):

$$y = h(x) = [h_\alpha(x), \alpha = 1...m]^T \geq 0 \tag{2}$$

The constraints (2) are usually enforced by an external input, say a multiplier $\lambda \in \mathbb{R}^m$ through an input function g such that

$$\begin{cases} g: \ \lambda \in \mathbb{R}^m \mapsto g(\lambda) \in \mathbb{R}^n \\ \dot{x} = f(x,t) + g(\lambda) \end{cases} \tag{3}$$

Finally, in order to complete the system, additional modeling information is needed. Particularly, two laws are of utmost importance:

a) A generalized equation [10] between the output w and the multiplier λ, denoted by the following inclusion:

$$0 \in F(y,\lambda) + Q(y,\lambda) \tag{4}$$

where $F : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is assumed to be continuously differentiable and $Q : \mathbb{R}^{m\times n} \sim\!\to \mathbb{R}^{m\times n}$ is a multi valued mapping with a closed graph. In simple cases, a complementarity condition is usually introduced for (4):

$$0 \leq y \perp \lambda \geq 0 \tag{5}$$

b) A reinitialization mapping or an impact law defining the state of the system after a non smooth event:

$$x(t^+) = \mathcal{F}(x(t^-),t) \tag{6}$$

A great number of dynamical systems with nonsmooth evolution can be casted into the previous formulation. To cite a few of them, we can mention differential inclusions, projected dynamical systems, piecewise affine systems and linear complementarity systems. For a comprehensive review of NSDS, we refer to the following monographs, [1] and [12].

Lagrangian dynamical systems

To shed more light on what NSDS can be, we can consider for instance the well known case of the Lagrangian mechanical systems with unilateral constraints, which model the dynamics of finite dimensional mechanical systems with contact. Let us consider a p-dimensional Lagrangian system in a configuration manifold $\mathcal{M}$, parameterized by a set of p generalized coordinates denoted by $q \in \mathcal{M}$. The matrix $M(q)$ is the mass and $F(q,\dot{q},t)$ the set of forces acting upon the system. Usually, the unilateral constraints are written in the coordinates as:

$$h_\alpha(q(t)) \geq 0, \quad \alpha = 1 \ldots \nu, \tag{7}$$

defining an admissible set for the system:

$$\Phi(t) = \{z(t) \in \mathcal{M}, \quad h_\alpha(z(t)) \geq 0, \alpha = 1 \ldots \nu\}. \tag{8}$$

With sufficient regularity in time, the Lagrange equations are:

$$M(q)\ddot{q} + F(q,\dot{q},t) = \sum_{\alpha=1}^{\nu} \nabla h_\alpha(q)\lambda_\alpha \tag{9}$$

where λ_α is the set of the Lagrange multipliers associated with the constraints $h_\alpha(q(t))$ trough a complimentary condition:

$$0 \leq h_\alpha(q) \perp \lambda_\alpha \geq 0 \tag{10}$$

In this case, the output function h is defined in terms of the coordinates q and the output function g is related to this function by:

$$g(\lambda) = \begin{bmatrix} 0 \\ \sum_{\alpha=1}^{\nu} \nabla h_\alpha(q)\lambda_\alpha \end{bmatrix} \tag{11}$$

On one side the Lagrangian dynamical system can be understood as an hybrid system with 2^ν modes, a mode being defined by the fact that each constraint α is active or not. In each mode, the solution is assumed to be smooth with discontinuities arising at transitions between two modes. On the other side, it can be considered as a single NSDS. Then its solution is defined as a global one, possibly non smooth, containing the instants of discontinuity. Usually, for Lagrangian systems, the coordinates are considered as absolutely continuous functions of time, the velocities as functions of bounded variations and accelerations as measures.

Linear complementary systems

In [5], the authors study the so-called Linear Complementarity Systems (LCS) defined by:

$$\begin{cases} \dot{x} = Ax + B\lambda \\ y = Cx + D\lambda \\ 0 \leq w \perp \lambda \geq 0 \end{cases} \tag{12}$$

In this type of systems, a linear output y of a linear dynamical system is defined as the complementary variable of an input λ. The linear complementarity condition can model for instance the behavior of an ideal diode. If the output y represents the current through the diode, this condition states that this current must be positive. The multiplier λ is then the opposite of the voltage across the diode. The graph of this relation is illustrated on Figure 1.

1.2 Comparison between the hybrid and the non-smooth approaches of NSDS

From the point of view of discrete systems, hybrid systems can be modeled as infinite-state transition systems [6]. For simple dynamics (constant or linear) and small systems (up to 100 degrees of freedom), the hybrid approach allows us to exploits the widespread analysis techniques for finite-state systems, such as the verification techniques to check some fundamental properties. For larger systems with fully nonlinear dynamics, it seems that these discrete techniques no longer apply.

In the case of NSDS, the mathematical properties of the solution and the associated numerical techniques allow the simulation and the analysis of large systems to be performed. We claim that the continuous approach is more efficient from the mathematical and the numerical point of view, rather than an event-driven or a discrete approach.

On the mathematical point of view, a NSDS can be considered as a unique time-continuous dynamical

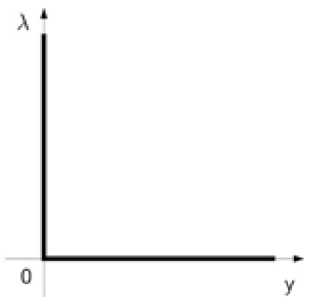

Figure 1. Complementarity condition as a constitutive law for an ideal diode

system which can encounter discontinuities and reinitializations. That leads to the definition of global solution of such systems in a class of appropriate functions. The case of the constrained Lagrangian dynamics leads for example to the formulation of the dynamics in terms of measure differential inclusions [11, 8, 7] valid on the continuous part of the evolution as well as at the events. Such types of solutions and formulations can comprised complex sequences of events, like concurrent events or accumulation (Zeno), together with mathematical results such as global existence and uniqueness.

On the numerical point of view, a non smooth approach of hybrid systems exploits the previous notion of solution defined everywhere in time. This fact leads to a design of powerful time–stepping schemes without explicit event handling procedure. For the case of the Lagrangian dynamics, the measure differential inclusion is evaluated on a fixed time interval and structural changes of the dynamics are taken into account in a weak sense. Contrary to event-driven schemes, such time-stepping are proved to be convergent even in presence of events accumulations.

Another advantage of the nonsmooth approach of NSDS is the algebraic formulation of certain class of state transitions at events. To shed more light on this aspect, the simple example of an ideal diode can be taken. This behavior can be modeled as a pure logical component thanks to an 'if' statement as in Modelica [4]. Indeed recognizing that the curve of the graph in Figure 1 can be parameterized by a parameter s, we can define the following Modelica script:

```
off = s < 0
  λ = if  off  then  -s  else 0
  y = if  off  then  0   else s
```

Same representations can be performed with ideal switches, piece-wise linear model of MOS transistors. The main difficulties to view systems with ideal components this way is that for each new boolean variable like off, two modes of the hybrid dynamical system are possible. If we introduce n-boolean variables, in the worst case, 2^n modes have to be checked. Therefore the problem complexity is exponential.

On the contrary, in the nonsmooth approach the discretized problem at each step can be reformulated as a Linear Complementarity Problem (LCP)[3] of the form:

$$\begin{cases} w = Mz + q \\ 0 \leq w \perp z \geq 0 \end{cases} \tag{13}$$

Under some usual assumptions on the matrix M, (positiveness, n-step property), and on the vector q, numerical algorithms can be used with polynomial complexity, avoiding an exhaustive enumerative verification of each modes at exponential time [9].

To conclude, on the mathematical point of view, the non smooth framework yields precise definitions of solutions together with uniqueness and existence results under appropriate assumptions. On the numerical point of view, the use of specific algorithms (time–stepping schemes, LCP solvers with polynomial complexity) leads to an efficient simulation environment.

2 NSDS in Siconos: problem formulation and software architecture

2.1 Overview of Siconos software

Siconos software is mainly dedicated to modeling and simulation of NSDS. It aims at providing a general and common tool for non smooth problems present in various scientific fields: applied Mathematics, Mechanics, Robotics, Electrical networks and so on. However, the platform is not supposed to re-implement the existing dedicated tools already used for the modeling of specific systems, but to possibly integrate them. For instance, strong collaborations exists with HuMAns[1] (humanoid motion modeling and control) or LMGC90[2] (multi-body contact mechanics) softwares.

Siconos software development is part of a European project involving different research teams (More details on `siconos.inrialpes.fr`), but is mainly developed in Bipop team at INRIA Rhônes-Alpes in Grenoble.

Siconos is a free software, under GPL GNU license, available on the Gforge web pages of the project, where one can also find documentation, support and all that sort of utilities: `http://siconos.gforge.-inria.fr/`.

Siconos is composed in three main parts: Numerics, Kernel and Front-End, as represent on figure 2.a below.

[1] http://bipop.inrialpes.fr/software/humans/

[2] http://www.lmgc.univ-montp2.fr/~dubois/LMGC90/

The Front-End provides interfaces with some specific command-languages such as Python or Scilab, this to supply more pleasant and easy-access tools for users, during pre and post treatment.

The Numerics part holds all low-level lgorithms, to compute basic well-identified problems (ordinary differential equations, LCP, QP solvers, Blas-Lapack linear algebra routines ...).

Finally, the Kernel is the core of the software, providing high level description of the studied system and numerical solving strategies. It is fully written in C++, and mainly composed of two rather distinct parts, modeling and simulation, that will be described more in details below. The whole dependencies among Kernel parts are fully depicted on figure 2.b. The Utils module contains tools, mainly to handle classical objects such as matrices or vectors. The Input-output module concerns objects for data management in XML format, thanks to the libxml2 library. Finally, a plug-in system is also available, essentially to allow user to provide his own computation methods for some specific functions (vector field of a dynamical system, mass ...), this without having to re-compile the whole platform. Moreover, the platform is designed in a way that allows user to add dedicated modules through object registration and factories mechanisms.

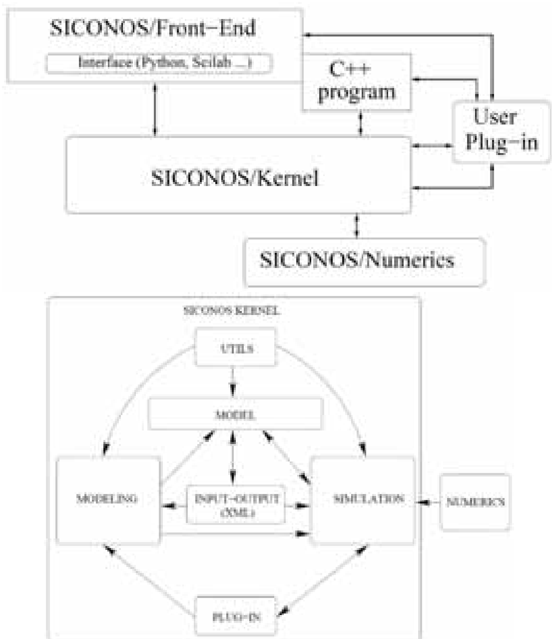

Figure 2: (a) General design of Siconos software, (b) Kernel component dependencies

2.2 NSDS modelling in Siconos software

As explained in Section 1, a non smooth dynamical system is a set of dynamical systems that interact altogether in a non smooth way. This is the role of Kernel modeling part to provide tools for these systems description.

Main objects are the pre-mentioned `DynamicalSystem`, `Relation` and `NonSmoothLaw`, both embedded in `Interaction`. To enlighten this point let us consider the example of a column of two spherical balls, in contact or not, falling down to the ground. Each ball constitutes a `DynamicalSystem`, including ordinary differential equations to described its dynamics, i.e. trajectories, initial conditions and other useful variables. Then, two interactions are possible: one between the balls and another between the ground and the lowest ball. A Siconos `Interaction` contains a list of dynamical systems (from one to any number, two in the present case), a relation that linked global and local variables and a nonsmooth law to describe the way systems interact. For the ball column case, the relation states that distance between two balls, and between lowest ball and ground, has to remain positive, and the law can be Newton impact one (see 2.2.c below for more details).

A reduced diagram of Siconos Kernel is presented on figure 3.a, which makes clearer various links between previous mentioned objects. In following paragraphs, various types of systems, relations and laws will be described more in details.

Dynamical systems

The most general case available in the platform is a first order system of the form:

$$\dot{x} = f(x, \dot{x}, t) + T(x)u(x, \dot{x}, t) + r \qquad (14)$$

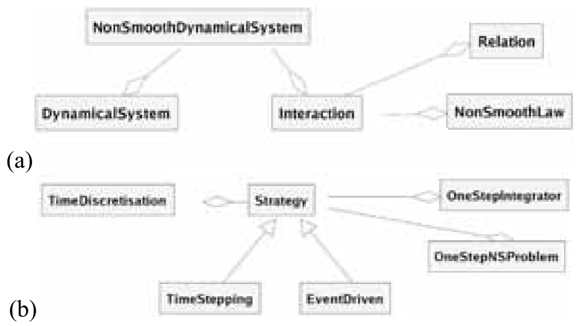

Figure 3: Simplified class diagram for kernel (a) modeling part, (b) simulation part.

where r is the non smooth part (typically contact forces for mechanical systems). (Note that to lighten equations, we will write x rather than $x(t)$). The terms Tu introduce the control variable into the system. All other dynamical systems in the software derived from the one above. They are:

- Linear Dynamical Systems:
$$\dot{x} = A(t)\,x + T\,u(t) + b(t) + r \tag{15}$$

- Lagrangian (second order) systems, usual in mechanical problems:
$$M(q)\,\ddot{q} + NNL(q,\dot{q}) = F_{\text{int}}(\dot{q},q,t) + F_{\text{ext}}(t) + r \tag{16}$$
where q denotes generalized coordinates, M the mass matrix, NNL the nonlinear inertia operator, F_{int} the internal, nonlinear forces and F_{ext} the external forces, depending only on time.

- Lagrangian Linear Time Invariant systems:
$$M\,\ddot{q} + C\,\dot{q} + K\,q = F_{\text{ext}}(t) + r \tag{17}$$
where C and K are respectively the classical viscosity and stiffness matrices.

The typical dimension of the state vector can range between a few degrees of freedom and more than several hundred thousand, for example for mechanical or electrical systems.

Relations

As explained above, and according to Section 1 notation, some relations between local, (y,λ), and global variables (x,r), have to be set to described interactions between systems. Their general form is:

$$y = h(x,t,\ldots)\,,\quad r = g(\lambda,t,\ldots) \tag{18}$$

Any other relations are derived from this one. Possible cases are:

- Linear time-invariant:
$$y = Cx + Fu + D\lambda + e,\quad r = B\lambda + a \tag{19}$$

- Lagrangian:
$$\dot{y} = H(q,t)\,\dot{q},\quad r = H^{t}(q,t)\lambda \tag{20}$$

- Lagrangian Linear:
$$y = H\dot{q} + b,\quad r = H^{t}\,\lambda \tag{21}$$

Note that, $H(q,t)$ can optionally depend on u or λ.

Nonsmooth laws

- Complementarity condition or unilateral contact:
$$0 \le y \perp \lambda \ge 0 \tag{22}$$

- Newton impact:
$$\text{if } y(t) = 0,\quad 0 \le \dot{y}(t^{+}) + e\,\dot{y}(t^{-}) \perp \lambda \ge 0 \tag{23}$$

- Relay:
$$\begin{cases} \dot{y} = 0 & |\lambda| \le 1 \\ \dot{y} \ne 0 & \lambda = \text{sgn}(y) \end{cases} \tag{24}$$

- Unilateral contact and Coulomb's Friction, with $y = [y_n, y_t]^T, \lambda = [\lambda_n, \lambda_t]^T$:
$$\text{if } y_n = 0, \begin{cases} 0 \le \dot{y}_n \perp \lambda_n \ge 0 \\ \dot{y}_t = 0, \quad \|\lambda_t\| \le \mu\lambda_n \\ \dot{y}_t \ne 0, \quad \lambda_t \le -\mu\lambda_n\,\text{sgn}(\dot{y}_t) \end{cases} \tag{25}$$

- Piece-wise linear relations with fill in graphs as depicted on the figure 4.

The complete NSDS

At the end, the complete NSDS object contains a list of dynamical systems and of interactions between them. An object `Topology` has been introduced, as a field of NSDS that "knows" all interactions and their links. That especially allows us to define the relative degrees between output and input of the different coupled systems.

2.3 Simulation of NSDS behaviour

The various possible strategies of simulation in Siconos are listed below:

- *Time-stepping* schemes.
- *Event-driven* schemes.
- Hybrid time integration methods between time-stepping and event-driven.
- Equilibrium analysis.
- Further ones dedicated to control and analysis (stability, bifurcations ...).

From a practical point of view, all those strategies are based on low level algorithms from Numerics, such as linear complementary problem, quadratic programming or non smooth Newton solvers.

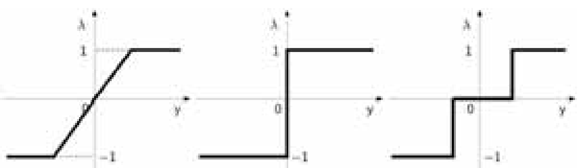

Figure 4: some multivalued piecewise linear laws: saturation, relay, relay with dead zone.

To conclude, general simulation strategy is summarized in the classes diagram of figure 3.b., with specific Siconos simulation objects, `OneStepIntegrator`, i.e. integrators for the dynamics without its non smooth part (Lsodar, Adams, Moreau ...) and `OneStepNSProblem`, which concerns the whole non smooth system.

3 Samples

In this part, specific examples of nonsmooth problems, solved with Siconos, are briefly presented: an electrical and two mechanical systems.

3.1 A first order system: electrical oscillator with 4 diodes bridge full-wave rectifier

Description of the physical problem

In this sample, a LC oscillator initialized with a given voltage across the capacitor and a null current through the inductor provides energy to a load resistance through a full-wave rectifier that consists in a 4 ideal diodes bridge (see fig. 5). Both waves of the oscillating voltage across the LC are provided to the resistor with current flowing always in the same direction. The energy is dissipated in the resistor, resulting in a damped oscillation.

Formalizing the linear complementary problem

A nonsmooth law describes the behavior of each diode of the bridge as a complementarity condition between current and reverse voltage (variables (y,λ)). Depending on the diode position in the bridge, y stands for the reverse voltage across the diode or for the diode current. There is only one dynamical system, which is linear, the whole oscillator that interacts with itself, through a linear time invariant relation between the state variable x and the non smooth law variables (y,λ).

Using Kirchhoff current and voltage laws and branches constitutive equations, we obtain the following linear dynamic system (see figure 5 for notations):

$$\begin{pmatrix}\dot{v}_L\\ \dot{i}_L\end{pmatrix}=\begin{pmatrix}0 & -1/C\\ 1/L & 0\end{pmatrix}\begin{pmatrix}v_L\\ i_L\end{pmatrix}+\begin{pmatrix}0 & 0\\ 0 & 0\\ -1/C & 0\\ 1/C & 0\end{pmatrix}^T\begin{pmatrix}-v_{DR1}\\ -v_{DF2}\\ i_{DF1}\\ i_{DR2}\end{pmatrix} \quad (26)$$

that fits in the frame of (15) with $x=(v_L \quad i_L)^T$ and $\lambda=(-v_{DR1} \quad -v_{DR2} \quad i_{DF1} \quad i_{DR2})^T$. Thus, linear relations are

$$r=B\lambda=\begin{pmatrix}0 & 0 & -1/C & 1/C\\ 0 & 0 & 0 & 0\end{pmatrix}(-v_{DR1} \quad -v_{DR2} \quad i_{DF1} \quad i_{DR2})^T$$

and

$$y=\begin{pmatrix}i_{DR1}\\ i_{DF2}\\ -v_{DF1}\\ -v_{DR2}\end{pmatrix}=\begin{pmatrix}0 & 0\\ 0 & 0\\ -1 & 0\\ 1 & 0\end{pmatrix}\begin{pmatrix}v_L\\ i_L\end{pmatrix}+\begin{pmatrix}1/R & 1/R & -1 & 0\\ 1/R & 1/R & 0 & -1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{pmatrix}\begin{pmatrix}-v_{DR1}\\ -v_{DF2}\\ i_{DF1}\\ i_{DR2}\end{pmatrix}$$

which corresponds to linear time-invariant relations of the form (19). Finally, complementarity conditions result from the ideal diode characteristic, and provide the required nonsmooth law:

$$0 \le -v_{DR1} \perp i_{DR1} \ge 0, \quad 0 \le -v_{DF2} \perp i_{DF2} \ge 0$$
$$0 \le i_{DF1} \perp -v_{DF2} \ge 0, \quad 0 \le i_{DR2} \perp -v_{DR2} \ge 0$$

Simulation and comparison with numerical results coming from SPICE models and algorithms

Siconos results are compared with those obtained using the SMASH simulator from Dolphin, with a smooth model of the diode as given by Shockley's law, a classical one step solver (Newton-Raphson) and the trapezoidal integrator.

Figure (6).a depicts the static $I(V)$ characteristic of two diodes with default SPICE parameters and two values for the emission coefficient N: 1.0 (standard

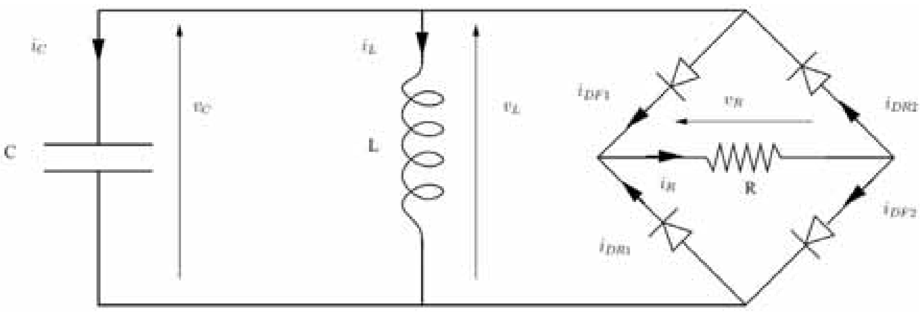

Figure 5: Electrical oscillator with 4 diodes bridge full-wave rectifier

diode) and 0.25 (stiff diode). The stiff diode is close to an ideal one with a threshold of 0.2 V.

Figure (6).b displays a comparison of the SMASH and SICONOS results with a trapezoidal integration ($\theta = 0.5$) and a fixed time step of 1 µs. A stiff diode model was used in SMASH simulations. One can notice that the results from both simulators are very close. The computational effort is much larger with Spice, which require a lot of Newton iterations to solve this very stiff and smooth problem, compared with a simple LCP solving.

3.2 A Lagrangian nonlinear system: simulation of a Robotic Arm

As a second example, we consider now the Mitsubishi PA-10 Robot, a seven-degree of freedom anthropomorphic robot, presented on figure 7. In Siconos, this robotic arm is modeled using a Lagrangian (nonlinear) dynamical system, that interacts with the ground. The arm falls down due to its own weight and bounces on the ground. Moreover, articular stops have been included to limit angular rotations. The degrees of freedom are the rotation angles, represented by the vector q. Neither external forces nor control terms are present. Finally, the robot dynamics

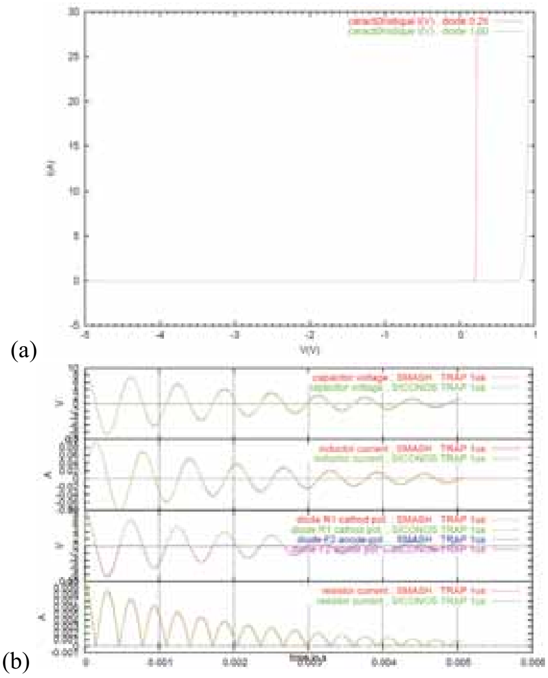

Figure 6: (a) Diodes characteristics from SPICE model with $N = 0.25$ and $N = 1$
(b) SMASH and SICONOS simulation results with trapezoidal integration, 1 µs time step.

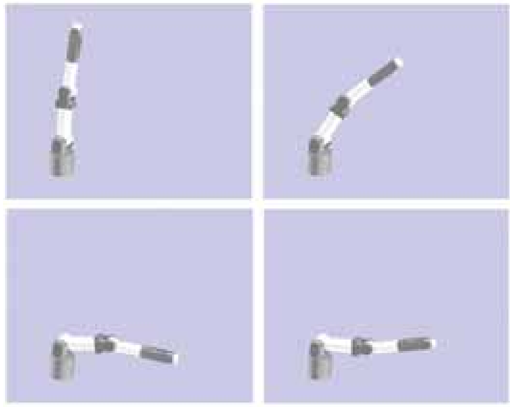

Figure 7: robotic arm fall-down: (a) initial position, (b) before contact with ground, (c) contact, (d) post-contact.

falls into the framework of Lagrangian dynamical systems as in (16). As a relation, we set that distance between arm and ground must remain positive, which means that contact can occurs but without penetration. The contact is supposed to be frictionless with a restitution coefficient denoted as e. This leads to nonlinear links between angular position (i.e. components of q), that can be written as:

$$\dot{y} = h(q)\dot{q}, \quad r = g(q), \quad \lambda = \nabla^t_q h(q)\lambda \qquad (27)$$

A Newton-Impact non smooth law complete this set of equations. with a restitution coefficient $e = 0.9$ for contact with the ground and $e = 0$ for angular stops.

The results of Siconos simulation, using Moreau time-stepping, Newton algorithm and a non smooth quadratic programming solver, are presented on figures 7 and 8. We denote A and B respectively the first (the one clamped on the ground) and second parts of the arm; θ_1 is the angle between A and the vertical position and θ_2 the angle between A and B. First, θ_1 and θ_2 increase, until the last one reaches its maximum angular position. Then θ_1, goes on increasing and θ_2 remains constant until the extremity of the arm touches the ground and bounces, here with a restitution coefficient e equal to 0.9. Finally A touches the ground and bounces also, together with B, until the complete arm lies on the ground. Note on 8.c and d that angular velocity cancels when angular stops are reached and change their signs when contact is established.

3.3 Beads column

We consider here a column of 1000 spherical beads, in contact or not, falling down to the ground. Dy-

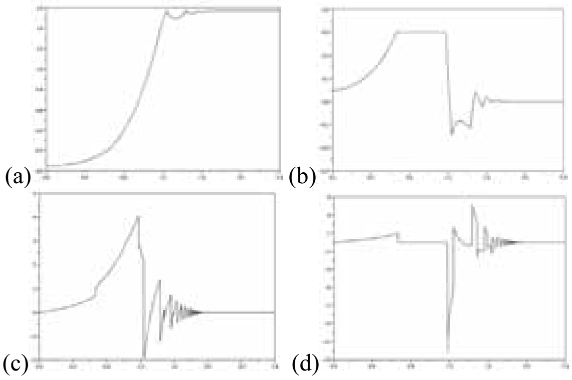

Figure 8: (a) $\theta_1(t)$, (b) $\theta_2(t)$, (c) $\dot{\theta}_1(t)$, (d) $\dot{\theta}_2(t)$

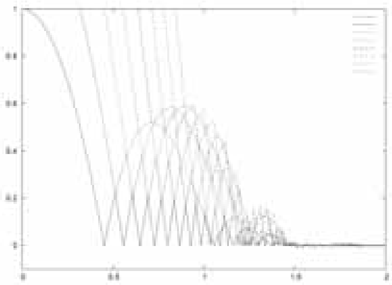

Figure 9: Vertical displacement of beads according to time

namical systems are Lagrangian ones and relations of type Lagrangian Linear, this with a Newton impact law. Figure 9 displays vertical displacements of the 8 lowest beads according to time. The interest of this example lies in the important number of degrees of freedom (i.e. size of vector q) and of relations (size of y and λ) equal to 1000.

Acknowledgements

This work is supported by the European Community through the Information Society Technologies thematic programme under the project SICONOS (IST-2001-37172).

References

[1] B. Brogliato. *Nonsmooth Mechanics: Models, Dynamics and Control*. Communications and Control Engineering. Springer-Verlag, second edition, 1999.

[2] F. H. Clarke. *Optimization and Nonsmooth analysis*. Wiley, New York, 1983.

[3] R. W. Cottle, J. Pang, R. E. Stone. *The linear complementarity problem*. Academic Press, Inc., Boston, MA, 1992.

[4] H. Elmqvist, S.E. Mattsson, M. Otter. *Object-oriented and hybrid modeling in Modelica*. Journal Européen des systèmes automatisés, 35(4):395 – 404, 2001.

[5] W. P. M. H. Heemels, J. M. Schumacher, S. Weiland. *Linear complementarity systems*. SIAM journal of Applied Mathematics, 60(4):1234–1269, 2000.

[6] T. A. Henzinger. *The theory of hybrid automata*. In M. K. Inan and R. P. Kurshan, editors, *Verification of Digital and Hybrid Systems*, volume 170 of NATO ASI Series F: Computer and Systems Sciences, pages 265-292. Springer Verlag, 2000.

[7] M. D. P. Monteiro Marques. *Differential Inclusions in Nonsmooth Mechanical Problems: Shocks and Dry Friction*. Birkhauser, Verlag, 1993.

[8] J. J. Moreau. *Unilateral contact and dry friction in finite freedom dynamics*. In J. J. Moreau and P. D. Panagiotopoulos, editors, *Nonsmooth mechanics and applications*, number 302 in CISM, Courses and Lectures, pages 1–82. Springer Verlag, 1988.

[9] J. S. Pang, R. Chandrasekaran. *Linear complementarity problems solvable by a polynomially bouded pivoting algorithms*. Math. Prog. Study, 25:13–27, 1985

[10] S. M. Robinson. *Generalized equations and their solutions. I. Basic theory*. Mathematical programming study, 10:128–141, 1979.

[11] M. Schatzman. *A class of nonlinear differential equations of second order in time*. Non linear Analysis, 2(3):355–373, 1978.

[12] van der Schaft, J. M. Schumacher. *An Introduction to Hybrid Dynamical Systems*. Springer Verlag London, 2000.

Corresponding author: Vincent Acary,
INRIA Rhône-Alpes – Bipop project
Zirst, 655 avenue de l'Europe
Montbonnot, 38334 Saint Ismier Cedex, France
vincent.acary@inrialpes.fr