NUMERICAL SIMULATION OF NONSMOOTH SYSTEMS AND SWITCHING CONTROL WITH THE SICONOS/CONTROL TOOLBOX.

Vincent Acary, Constantin–Irinel Morărescu, Franck Pérignon , Bernard Brogliato
BIPOP Team Project
INRIA
France
firstname.lastname.@inrialpes.fr

Abstract. In this work we present the general concepts of the SICONOS platform (free opensource (GPL)). The reliability of this platform is pointed out by simulating the dynamics of a nonsmooth non-linear mechanical system. Some numerical results and their mechanical interpretations conclude the presentation.

Key words
Nonsmooth Dynamical Systems, Linear Complementarity Problem, Time–stepping scheme, Switching control.

1 Introduction

In order to emphasize the reliability of the SICONOS platform for the simulation of various types of dynamics we present a switching control scheme of a two-link manipulator. We consider the case of a manipulator with rigid joints but the algorithm can be extended to the flexible joint case as well.

2 Nonsmooth Lagrangian Mechanical Systems

2.1 Lagrangian mechanical systems

A generic n-dimensional Lagrangian dynamical system can be defined as:

$$M(q,z)\ddot{q} + N(\dot{q},q,z) + F_{\mathsf{int}}(\dot{q},q,t,z) = F_{\mathsf{ext}}(t,z) + p \tag{1}$$

where

- $q \in \mathbb{R}^n$ is the set of the generalized coordinates,
- $\dot{q} \in \mathbb{R}^n$ the velocity, *i.e.* the time derivative of the generalized coordinates.
- $\ddot{q} \in \mathbb{R}^n$ the acceleration, *i.e.* the second time derivative of the generalized coordinates.
- $p \in \mathbb{R}^n$ is an input due to some nonsmooth interaction with the environment.
- $M(q,z) \in \mathbb{R}^{n\times n}$ is the inertia term
- $N(\dot{q},q,z) \in \mathbb{R}^n$ is the nonlinear inertia matrix
- $F_{\mathsf{int}}(\dot{q},q,t,z) \in \mathbb{R}^n$ are the internal forces
- $F_{\mathsf{ext}}(t,z) \in \mathbb{R}^n$ are the external forces
- $z \in \mathbb{R}^{zSize}$ is a vector of arbitrary time–discrete algebraic variables.

In a more compact form, the system (1) can be stated as

$$M(q,z)\ddot{q} = F(\dot{q},q,t,z) + p \tag{2}$$

2.2 Nonsmooth interactions

General case The nonsmoothness is usually introduced through a nonsmooth generalized equation of the form

$$0 \in S(y,\lambda) + T(y,\lambda) \tag{3}$$

where $y \in \mathrm{I\!R}^m$ defines an output of the state and $\lambda \in \mathrm{I\!R}^m$ is a Lagrange multiplier. The inclusion (3), also called a generalized equation, defines the behavior of the Lagrange multiplier λ with respect to the output y. The function $S : \mathrm{I\!R}^{m\times m} \mapsto \mathrm{I\!R}^{m\times m}$ is assumed to be continuously differentiable and $T : \mathrm{I\!R}^{m\times m} \rightsquigarrow \mathrm{I\!R}^{m\times m}$ is a multivalued mapping with a closed graph. The unknowns (y,λ) are related to q and p through a set of kinematic output/input relations:

$$\begin{array}{l} y = g(t,q) \\ p = \nabla^T g(t,q)\lambda. \end{array} \tag{4}$$

Finally, if the evolution is nonsmooth, a reinitialization rule

$$v^+ = \mathcal{F}(v^-,q,t) \tag{5}$$

may be required. For multibody systems with unilateral constraints, this relation is usually termed as an impact law.

Unilateral constraints with Coulomb's friction Two well-known instances of (3) are:
a) the perfect unilateral constraints with $S(y,\lambda) = \partial\psi_{\mathrm{I\!R}_+}(\lambda)$ and $T(y,\lambda) = \lambda$, where ψ_K is the indicator function of the set K and the symbol ∂ denotes the subdifferential in the sense of the Convex Analysis (Rockafellar, 1970). The constraints $y \geqslant 0$ define[1]

[1] Throughout this work a vector is considered positive (negative) if all of its component are positive (negative). Thus $y \geqslant 0$ denotes a set of inequalities $y_i \geqslant 0,\ i = 1,\dots,m$

equivalently an admissible domain Φ in terms of generalized coordinates:

$$\Phi(t) = \{q \mid g(q,t) \geqslant 0\} \tag{6}$$

b) $y = [g_{\rm N}, v_{\rm T}]^T$ where $g_{\rm N}$ is the normal gap between two bodies and $v_{\rm T}$ is the tangential velocity at contact. We assume in the same way that λ is decomposed in normal and tangential component as $\lambda = [\lambda_{\rm N}, \lambda_{\rm T}]^T$. Choosing $S(y,\lambda) = \lambda$ and $T(y,\lambda) = [\partial\psi_{{\rm I\!R}_+}(g_{\rm N}), \mu\lambda_{\rm N}\partial\|v_{\rm T}\|]^T$, where μ is the coefficient of friction, we obtain the standard 3D Coulomb friction law.

In the sequel, we will only be interested in systems with perfect unilateral constraints and impacts.

3 Moreau's Time–stepping Scheme

3.1 Principle

The Moreau's Time stepping scheme (Moreau, 1988) for scleronomous holonomic perfect unilateral constraints is based on a formulation of the Newton impact law of coefficient e and the unilateral constraint in terms of velocity, together with an expression of the dynamical equation in terms of measures.

More precisely, the velocity v of the system is considered as a Right Continuous of Local Bounded Variations (RCLBV) function of the time and the acceleration as a differential measure dv associated with v. The absolutely continuous generalized coordinate q is derived from the velocity thanks to

$$q(t) = q(t_0) + \int_{t_0}^{t} v(\tau)\, d\tau \tag{7}$$

The dynamics (2) is reformulated as a measure differential equation,

$$M(q)\, dv - F(t,q,v)\, dt = \nabla g(q)\, dI \tag{8}$$

where dt is the Lebesgue measure. We omit the discrete state z to lighten the notation. The unilateral constraint $y = g(q) \geqslant 0$ is enforced by the multiplier measure dI. To complete this measure differential equation, Moreau proposed a compact formulation of the impact law as a measure inclusion,

$$dI \in \partial\psi_{T(y(t))}(U(t^+) + eU(t^-)) \tag{9}$$

where $T(q)$ is the tangent cone to ${\rm I\!R}^+$ at y, e is the coefficient of restitution and U is the velocity associated with the constraints y such that

$$U(t) = \nabla^T g(q)\, v(t) \tag{10}$$

Finally, we obtain a MDI, the so-called sweeping process,

$$\begin{aligned} &M(q(t))\, dv - F(t,q(t),v(t))\, dt \\ &\in -\nabla g(q)\partial\psi_{T(y(t))}(U(t^+) + eU(t^-)) \end{aligned} \tag{11}$$

The Moreau's Time stepping scheme performs the numerical time integration of the MDI (11) on an interval $]t_k, t_{k+1}]$ of length h. Using the notation $v_{k+1} \approx v(t_{k+1}^+); U_{k+1} \approx U(t_{k+1}^+); P_{k+1} \approx dI(]t_k, t_{k+1}])$, and $t_{k+\theta} = (1-\theta)q_{k+1} + \theta q_k, q_{k+\theta} = (1-\theta)q_{k+1} + \theta q_k, v_{k+\theta} = (1-\theta)v_{k+1} + \theta v_k$, the scheme may be written down as follows ($\theta \in [0,1]$):

$$\begin{cases} M(\tilde{q}_{k+1})(v_{k+1} - v_k) - hF(t_{k+\theta}, q_{k+\theta}, v_{k+\theta}) \\ = \nabla g(q_k) P_{k+1}, & \text{(12a)} \\ q_{k+1} = q_k + hv_{k+\theta}, & \text{(12b)} \\ U_{k+1} = \nabla^T g(q_k)\, v_{k+1} & \text{(12c)} \\ -P_{k+1} \in \partial\psi_{T_{{\rm I\!R}^+}(\tilde{y}_{k+1})}(U_{k+1} + eU_k), & \text{(12d)} \\ \tilde{y}_{k+1} = y_k + hU_k. & \text{(12e)} \end{cases}$$

The value $\tilde{q}_{k+1}$ is a prediction of the position which allows the computation of the explicit mass matrix. The value $\tilde{y}_{k+1}$ is a prediction of the constraint which allows the computation of the tangent cone $T_\Phi(\tilde{y}_{k+1})$.

The inclusion can be stated equivalently as a conditional complementarity problem for all $\alpha \in [1 \ldots m]$ as follows,

$$\begin{array}{l} \text{if} \quad \tilde{y}_{\alpha,k+1} \leqslant 0 \text{ then} \\ \quad 0 \leqslant U_{\alpha,k+1} + eU_{\alpha,k} \perp P_{\alpha,k+1} \geqslant 0, \\ \text{otherwise } \mu_{\alpha,k+1} = 0, \end{array} \tag{13}$$

The convergence of Moreau's time–stepping schemes has been shown under various assumptions in (Monteiro Marques, 1993; Mabrouk, 1998; Dzonou and Monteiro Marques, 2007) .

3.2 Comments

Moreau's time–stepping scheme is a *nonsmooth event capturing method*. In such a method, the time-integration is performed with a time step, which does not depend on the exact location of the nonsmooth events (impact, take-off, ...). The advantages of this class of methods are the convergence proofs and the efficiency even in the case of finite accumulation of impacts, and the fact that they are able to work without an accurate event detection. Finally, another practical interest of this method is that it does not require any kinematical reformulations.

The problem (12a)–(12e) is an instance of a Nonlinear Complementarity Problem (NCP) or a Linear Complementarity Problem (LCP) if F is linear. Numerous solvers have been designed in the Mathematical programming community to solve such a system; for further details, see (Facchinei and Pang, 2003) and for practical applications to Mechanics and Electronics see (Acary and Brogliato, 2008).

4 The SICONOS Platform and the SICONOS/CONTROL Toolbox

We briefly expose in this section some aspects of the SICONOS platform. For more details on how to use the platform and applications in various domains, we refer to (Acary and Pérignon, 2007) and the documentation at `http://siconos.gforge.inria.fr`.

4.1 Platform overview

The Siconos Platform is a scientific computing software dedicated to the modeling, simulation, control and analysis of (Non Smooth) Dynamical Systems (NSDS). Especially, the following classes of NSDS are addressed: Mechanical systems with contact, impact and friction, electrical circuits with ideal and piecewise linear components, Differential inclusions and Complementarity systems. The platform consists of the following components:

SICONOS/NUMERICS This stand-alone library contains a collection of low-level numerical routines in C and F77 to solve linear algebra problems and non-smooth problems(LCP, QP, NCP, 3D frictional contact). It is based on well–known netlib libraries such as BLAS/LAPACK, ATLAS, Templates. Numerical integration of ODE is also provided thanks to ODEPACK. (LSODE solver.)

SICONOS/KERNEL This component is the core of the platform. It offers a collection of C++ classes to model and to simulate NSDS. Some details will be given in the next Section. The Siconos Kernel can be informed by implementing external functions (for end-users) or by a plug-in/registration systems of new user-classes.

SICONOS/FRONT-END The Front-End is a “user-friendly” interface providing a more interactive way of using the platform through an API C++ with interactive environment Python scripting (Swig wrapper) and an API C for Scilab and Matlab interfaces.

4.2 Modeling principle

A NSDS can be seen as a set of dynamical systems that may interact in a nonsmooth way. The modeling approach in the Siconos platform consists in considering the NSDS as a graph with dynamical systems as nodes and nonsmooth interactions as branches. Thus, in order to describe each element of this graph in Siconos, one needs to define a `NonSmoothDynamicalSystem` object composed of a set of `DynamicalSystem` objects and a set of `Interaction` objects.

A `DynamicalSystem` object is no more than a set of equations to describe the behavior of a single dynamical system, with some specific operators and initial conditions.

An `Interaction` object describes the way one or more dynamical systems are linked or may interact. For instance, if you consider a set of rigid bodies, the `Interaction` objects define and describe what happens at the contacts. The `Interaction` object is characterized by some ”local” variables, y (also called output) and λ (input) and is composed of: a `NonSmoothLaw` object that describes the mapping between y and λ, and a `Relation` object that describes the equations between the local variables (y, λ) and the global ones as in (4).

As summarized in Figure 1, building a problem in Siconos relies on the proper identification and construction of some `DynamicalSystem`s and of all the potential interactions. A complete review of the dynamical systems and interactions available in Siconos can be found in (Acary and Pérignon, 2007).

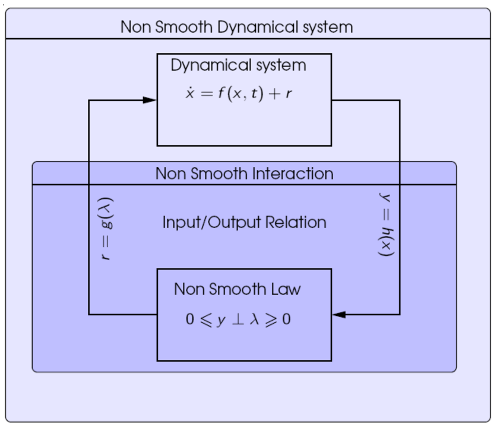

Figure 1. A simple `NonSmoothDynamicalSystem` with one `DynamicalSystem` object and one `Interaction`

4.3 Simulation principle

Once a NSDS has been fully designed and described thanks to the objects detailed above, it is necessary to build a `Simulation` object, namely to define the way the nonsmooth response of the NSDS will be computed.

First of all, let us introduce the `Event` object, which is characterized by a type and a time of occurrence. Each event has also a `process` method which defines a list of actions that are executed when this event occurs. These actions depend on the object type. For the objects related to nonsmooth time events, namely `NonSmoothEvent`, an action is performed only if an event-driven strategy is chosen. Finally, thanks to a registration mechanism, user–defined events can be added.

To build the `Simulation` object, we first define a discretisation, using a `TimeDiscretisation` object, to set the number of time steps and their respective size. Note that the initial and final time values are part of the `Model`. The time instants of this discretisation define `TimeDiscretisationEvent` objects used to initialize an `EventsManager` object, which contains the list of `Event` objects and their related methods. The `EventsManager` object belongs to the simulation and will lead the simulation process: the systems integration is always done between a ”current” and a ”next” event. Then, during the simulation, events of different types may be added or removed, for example when the user creates a `Sensor` or when an impact is detected.

4.4 Control principle in SICONOS/CONTROL

Two strategies are available to implement a control law in the Siconos platform:

Nonlinear continuous control with switches The control can be implemented in external functions F_{int} and F_{ext}. For an accurate simulation of the control law with Newton's method, the Jacobian of the control with respect to q and $\dot{q}$ must be provided (finite–difference approximation can be also used). For an explicit evaluation of the control law, the second strategy is preferable.

Sampled discrete control with delay Thanks to `Actuator` and `Sensor` objects, it is possible to schedule events of control type in the stack of the `EventsManager` object. The `Sensor` object is able to store any data of the model whenever an event is reached. The `Actuator` object is able to compute the control law with the stored values in the sensors. This strategy allows one to implement "real" sampled control laws with delay and switches independently of the time–step chosen for time–integration. It may be convenient for studying robustness of the control in sampled cases with delay. For the `SensorsEvent` and the `ActuatorEvent` related to control tools, an action is performed for both time–stepping and event–driven strategies at the times defined by the control law.

5 The Switching Nonlinear Control of a two–link Manipulator

In the sequel, let us introduce a model that allows us to test the Moreau's time-stepping algorithm of the SICONOS platform presented in the previous sections. Precisely, we consider a simple planar two-link manipulator whose end effector must track a desired circular trajectory that leaves the admissible domain. In order to accomplish its task the manipulator has to follow the constraint from the point where the circle leaves the admissible domain to the point where the circle re-enters in it.

5.1 Preliminaries

The time domain representation of the manipulator task can be described as (see (Brogliato *et al.*, 1997)):

$$\mathbb{R}^+ = \Omega_0 \cup I_0 \cup \Omega_1 \cup \Omega_2 \cup I_1 \cup \ldots \cup \Omega_{2k} \cup I_k \cup \Omega_{2k+1} \cup \ldots \quad (14)$$

where Ω_{2k} corresponds to free-motion phases, Ω_{2k+1} corresponds to constrained-motion phases and I_k represents the transient between free and constrained phases. It is worth to point out that during the phases I_k some impacts occur. The constraints defining the admissible domain are supposed frictionless and unilateral.

5.2 Controller design

In order to overcome some difficulties that can appear in the controller definition, the dynamical system (1) will be expressed in the generalized coordinates introduced in (McClamroch and Wang, 1988). The coordinates are $q \in \mathbb{R}^2$, with $q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$, such that the admissible domain $\Phi = \{q_1(t) \geqslant 0\}$ and then the set of complementary relations can be rewritten as $0 \leqslant \lambda \perp Dq \geqslant 0$ with $D = (1,0) \in \mathbb{R}^2$. The controller used here consists of different low-level control laws for each phase of the system. More precisely, the controller can be expressed as

$$T(q)U = \begin{cases} U_{nc} & \text{for } t \in \Omega_{2k} \\ U_t & \text{for } t \in I_k \\ U_c & \text{for } t \in \Omega_{2k+1} \end{cases} \quad (15)$$

where $T(q) = \begin{pmatrix} T_1(q) \\ T_2(q) \end{pmatrix} \in \mathbb{R}^{2\times 2}$.

Roughly speaking, we deal with a passivity based control law (see for instance (Brogliato *et al.*, 2007)) but some of the nonlinear terms are compensated during the constrained phases Ω_{2k+1}. We note also that the transition between constrained and free phases is monitored via a LCP (for further details see (Brogliato, 2003)). Without entering into details we precise that the switching controller is based on the fixed-parameter scheme presented in (Slotine and Li, 1988) and the closed-loop stability analysis can be found in (Bourgeot and Brogliato, 2005). Some of the events(impacts, detachment from the constraint) are state-dependent. Some others (switch between U_{nc} and U_t) are exogenous. The SICONOS/Control toolbox is able to simulate all these events, and to record them.

5.3 Dynamics Equation based on Lagrangian formulation

We consider the following notations (see figure 2): θ_i represents the joint angle of the joint i, m_i is the mass of link i, I_i denotes the moment of inertia of link i about the axis that passes through the center of mass and is parallel to the Z axis, l_i is the length of link i, and g denotes the gravitational acceleration.

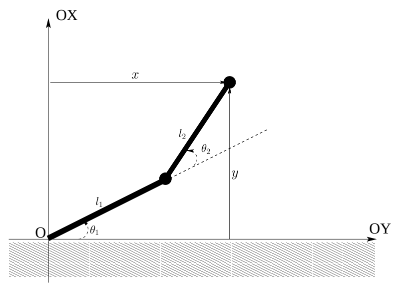

Figure 2. Two-link planar manipulator

Let us consider that the constraint is given by the ground (i.e. $y = 0$), thus the associated admissible domain is $\Phi = \{(x,y) \mid y \geqslant 0\}$. One introduces the generalized coordinates $q = \begin{bmatrix} y \\ x \end{bmatrix}$, $y \geqslant 0$ where (x,y) are the Cartesian coordinates of the end effector. The unconstrained desired trajectory of the manipulator's end effector is represented by a circle. However, we suppose that only a half of the circle is in the admissible domain. Concluding, the system has to track a half circle and then follow the ground from the point where the circle leaves the admissible domain to the point where the circle re-enters in it. Using the Lagrangian formulation we derive the dynamical equations of the system. Precisely, the inertia matrix is given by:

$$M_{11} = \frac{m_1 l_1^2}{4} + m_2 \left(l_1^2 + \frac{l_2^2}{4} l_1 l_2 \cos\theta_2 \right) + I_1 + I_2$$
$$M_{12} = M_{21} = \frac{m_2 l_2^2}{4} + \frac{m_2 l_1 l_2}{2} \cos\theta_2 + I_2$$
$$M_{22} = \frac{m_2 l_2^2}{4} + I_2$$

the nonlinear term containing Coriolis and centripetal forces is:

$$C_{11} = -m_2 l_1 l_2 \dot{\theta}_2 \sin\theta_2,\ C_{12} = -\frac{m_2 l_1 l_2}{2} \dot{\theta}_2 \sin\theta_2$$
$$C_{21} = \frac{m_2 l_1 l_2}{2} \dot{\theta}_1 \sin\theta_2, \quad C_{22} = 0$$

and the term containing conservative forces is:

$$G_1 = \frac{g}{2}[l_1(2m_1 + m_2)\cos\theta_1 + m_2 l_2 \cos(\theta_1 + \theta_2)]$$
$$G_2 = \frac{m_2 g l_2}{2} \cos(\theta_1 + \theta_2)$$

The generalized coordinates are obtained using the following transformation:

$$y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$
$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

5.4 Implementation details

The simulations were done using a nonlinear continuous control strategy. Precisely the term $N(\dot{q}, q, z)$ of equation (1) has been identified as $C(\dot{q}, q)\dot{q} + G(q)$ and the switching control has been introduced using the function F_{int}. Therefore, the Jacobian of N and F_{int} with respect to q and $\dot{q}$ have been explicitly computed and inserted in the algorithm. The SICONOS / Control toolbox also allows one to introduce a time-delay in the feedback loop, as we said in Section 4.4. The study of the robustness of the switching controller (15) with respect to sampling will be the object of a future work.

6 Numerical results

The stability analysis of the model and figures illustrating the behavior of the system during each phase of the motion (particularly during transition phases where the corresponding Lyapunov function is almost decreasing) can be found in (Morărescu and Brogliato, 2008). In the sequel, we discuss only some numerical aspects related to the time-stepping simulation strategy chosen in this work. The choice of a time-stepping algorithm was mainly dictated by the presence of accumulations of impacts which render the use of event-driven methods difficult[2]. The numerical values used for the dynamical model are $l_1 = l_2 = 0.5m$, $I_1 = I_2 = 1kg.m^2$, $m_1 = m_2 = 1kg$. It is noteworthy that the simulation results do not depend essentially on the chosen time-step for the scheme but, a smaller time-step allows to capture more precisely the behavior of the system.

We do not insist too much on the simulation results during the free-motion phases since the smoothness of the system is guaranteed on these phases and the behavior of the system is clear. The most interesting phases from the numerical point of view are the transition (accumulation of impacts) phases. It is worth to clarify that the number of impacts during the transition phases is not so important and the major issue is the finiteness of these phases. To be more clear we present in the next tables some numerical values. First, one can see that the lengths of the transition phase with respect to the time-step h do not vary significantly when the time-step decreases. Let us also denote by CPU the computing time necessary for the simulation (using an Intel(R) Core(TM)2 CPU 6300 1.86GHz) of one cycle (5 seconds).

h	$10^{-3}s$	$10^{-4}s$	$10^{-5}s$	$10^{-6}s$
$\lambda(I_1)$	0.945	0.9536	0.9525	0.9523
CPU	$1.5s$	$11.2s$	$111.3s$	$1072.2s$

The evolution of the number of impacts n_i with respect to the restitution coefficient e_N and the time-step h is quite different. As expected, n_i becomes larger when the restitution coefficient increases. Also, one can see that the accumulation of impacts can be captured with a higher precision when the time-step becomes smaller.

e_N \ h	$10^{-3}s$	$10^{-4}s$	$10^{-5}s$	$10^{-6}s$
0.2	$n_i = 3$	$n_i = 5$	$n_i = 6$	$n_i = 8$
0.5	$n_i = 6$	$n_i = 9$	$n_i = 12$	$n_i = 16$
0.7	$n_i = 9$	$n_i = 16$	$n_i = 23$	$n_i = 29$
0.9	$n_i = 23$	$n_i = 40$	$n_i = 64$	$n_i = 81$
0.95	$n_i = 32$	$n_i = 67$	$n_i = 108$	$n_i = 161$

However, a higher number of captured impacts does not change the global behavior of the system and the transition phase ends almost in the same moment when h varies, see $\lambda(I_1)$ in the first table.

In conclusion, reliable simulations with a reasonable CPU time can be performed with the Moreau's time-stepping scheme of the SICONOS platform, with a time-step $h = 10^{-4}s$.

Next, we present several simulations with different values of the controller parameters in order to see in an experimental way the influence of these parameters on the closed-loop dynamics. Some intuitive explanation will join the experimental results presented in the following. It is worth to precise that we keep for the next simulations a period of 5 seconds for each cycle and we fix the time-step at the value of 10^{-4} seconds.

In order to explain more easily the experimental results of this section we point out from the beginning that γ_1 is the coefficient of the velocity error and $\gamma_1\gamma_2$

[2] An event–driven algorithm is also available in SICONOS. Its use in case of accumulations needs some ad hoc numerical trick to pass through the accumulation (Abadie, 2000)

is the coefficient of the position error entering both in controller and Lyapunov function. In other words (see (Brogliato *et al.*, 2007)) the controller proposed in this work can be identified with a parallel interconnection (between the real position and desired position) of a spring with stiffness $\gamma_1\gamma_2$ and a damper with coefficient γ_1. First one considers a fixed value of $\gamma_1 = 25$ and one studies the behavior of the closed-loop dynamics with respect to γ_2 variation. The main influence in this case can be seen during the transition phases I_k. More precisely, diminishing the value of γ_2 the transition phases get larger. From the mechanical point of view decreasing γ_2 one decreases the stiffness of the spring entering the controller and therefore we get more jumps, higher first jump and longer transition phases.

The above discussion is also illustrated in the next table where we denote by H the height of the first jump.

γ_2	2	30	55
n_i	101	65	58
H	0.01091	0.00124	0.00056
$\lambda(I_1)$	4.088	0.313	0.148

Next one considers a fixed value of $\gamma_1\gamma_2 = 800$ and we repeat the simulation for different value of γ_1. From mechanical point of view, this means that we fix the spring stiffness entering the controller and we point out the effect of the damper gain on the system dynamics. Precisely, decreasing the value of γ_1 we detect an increasing tracking error, see figure 3. In other words, the real trajectory approaches more slowly the desired trajectory.

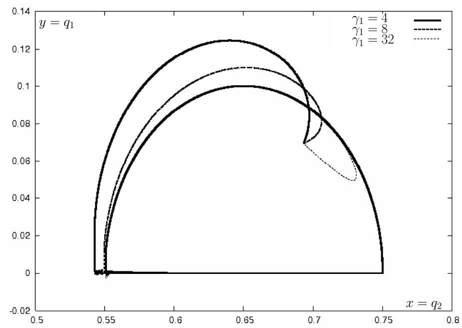

Figure 3. Smaller values of γ_1 lead to larger tracking errors.

7 Conclusion

In this paper we have presented the basic concepts of the SICONOS platform and its Control toolbox. Different control strategies (nonlinear continuous control, sampled discrete control) implemented in this platform are pointed out. In order to emphasize the qualitative performance of Moreau's time–stepping algorithm of the SICONOS platform we have presented and interpreted some numerical results concerning the tracking control of a two-link manipulator. The switching control used in the paper consists of different low-level control laws designed for each phase of the system. The flexible joint manipulators and the study of the robustness of the switching controller with respect to sampling will be considered in future works.

References

Abadie, M. (2000). *Dynamic Simulation of Rigid Bodies: Modeling of frictional contact*. B. Brogliato (Ed.) Lecture Notes in Physics LNP 551. Springer, pp. 1-60.

Acary, V. and B. Brogliato (2008). *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics*. Vol. 35 of *LNACM*. Springer Verlag.

Acary, V. and F. Pérignon (2007). An introduction to Siconos. Technical Report TR-0340. INRIA. http://hal.inria.fr/inria-00162911/en/.

Bourgeot, J.-M. and B. Brogliato (2005). Tracking control of complementary lagrangian systems. *International Journal of Bifurcation and Chaos* **15**(6), 1839–1866.

Brogliato, B. (2003). Some perspectives on the analysis and control of complementarity systems. *IEEE Trans. Autom. Contr.* **48**(6), 918–235.

Brogliato, B., R. Lozano, B. Maschke and O. Egeland (2007). *Dissipative Systems Analysis and Control. Theory and Applications*. Springer CCES, 2nd Ed.. London.

Brogliato, B., S.-I. Niculescu and P. Orhant (1997). On the control of finite-dimensional mechanical systems with unilateral constraints. *IEEE Trans. Autom. Contr.* **42**(2), 200–215.

Dzonou, R. and M. D. P. Monteiro Marques (2007). A sweeping process approach to inelastic contact problems. *Eur. J. Mech. A Solids* **26**, 474–490.

Facchinei, F. and J. S. Pang (2003). *Finite-dimensional Variational Inequalities and Complementarity Problems*. Vol. I & II of *Springer Series in Operations Research*. Springer Verlag NY. Inc.

Mabrouk, M. (1998). A unified variational for the dynamics of perfect unilateral constraints. *European Journal of Mechanics - A/Solids* **17**, 819–842.

McClamroch, N. and D. Wang (1988). Feedback stabilization and tracking of constrained robots. *IEEE Trans. Autom. Contr.* **33**(5), 419–426.

Monteiro Marques, M. D. P. (1993). *Differential Inclusions in NonSmooth Mechanical Problems: Shocks and Dry Friction*. Birkhauser, Verlag.

Moreau, J.J. (1988). *Nonsmooth mechanics and applications*. number 302 In: *CISM, Courses and lectures*. Moreau, J.J. and Panagiotopoulos, P.D. (Eds.) Springer Verlag. Wien- New York.

Morărescu, I.-C and B. Brogliato (2008). Tracking control of nonsmooth lagrangian systems with time constraints. In: *submitted to 6th Euromech Conference ENOC*. Saint Petersburg, Russia, June,30-July,4.

Rockafellar, R.T. (1970). *Convex Analysis*. Princeton University Press.

Slotine, J.J. and W. Li (1988). Adaptive manipulator control: A case study. *IEEE Trans. Autom. Contr.* **33**(5), 995–1003.