A Hybrid Automaton for a Class of Multi-Contact Rigid-Body Systems with Friction and Impacts

DOI:
10.3182/20120606-3-NL-3011.00074
10.3182/20120606-3-NL-3011.00048

A Hybrid Automaton for a Class of Multi-Contact Rigid-Body Systems with Friction and Impacts ⋆

Michael D. O'Toole, Eva M. Navarro-López

School of Computer Science, The University of Manchester, Oxford Road, Manchester, M13 9PL, UK.
(emails: michael.otoole@cs.man.ac.uk, eva.navarro@cs.man.ac.uk)

Abstract: Hybrid automata is a powerful framework for representing the dynamics of rigid-body mechanical systems. Thus far however, work has mainly focused on examples with only single points of contact. Consider for instance, the ubiquitous bouncing-ball example. The case of multiple contacts is much more complex. An event at one contact can affect the other contacts even when there is no apparent transmission mechanism. In this paper, we present a general hybrid automaton for systems consisting of multiple rigid-bodies with convex differentiable surfaces. The hybrid automaton assumes that Newton's impact law (restitution) and Coulomb friction are acting at the points of contact. Multiple contacts are addressed by introducing computational nodes which can be considered as the well-known *mythical modes* in the discontinuous systems' literature. These nodes consist of non-dynamical discrete states and reset operations which find valid combinations of contact forces. In this manner, they will guide the executions of the hybrid automaton to the relevant hybrid discrete locations.

Keywords: Hybrid automata; Discontinuous systems; Computational models; Friction; Impacts; Multi-rigid bodies; Mechanics.

1. INTRODUCTION

Would it be useful to generate automatically a general-purpose transition system for the description of mechanical systems subject to multiple impacts with friction? The answer is yes, and the applications would be broad and wide-ranging. However, this is a difficult and ambitious task.

The results presented in this paper are the first step towards this task. We propose a modelling framework that facilitates the specification of the switching among different mechanical and contact modes, taking into account the wide variety of discontinuous changes present in a general system with multiple impacts and friction. The modelling framework that we will use is that of hybrid automata (Alur et al., 1993; Henzinger, 1996; Johansson et al., 1999; Lygeros et al., 1999, 2003). Hybrid automata is one of the existing frameworks to describe hybrid dynamical systems. Discontinuous systems with jumps, as systems with impacts and friction, can be considered a subclass of hybrid systems. Although hybrid automata is a very powerful computational-oriented modelling framework for specifying the behaviour of complex systems, in the case of systems with impacts, it has only been used for simple systems, typically represented by the benchmark of one bouncing ball (see Goebel et al. (2009) and references therein). We aim to go beyond the bouncing ball.

Mechanical systems can be described by piecewise-smooth models that switch between different dynamics as the configuration of the system changes, and feature jumps in the state-space due to impacts. This kind of model is well-suited to abstraction in the hybrid automata representation. See for example the DDS and extended DDS hybrid automaton used to model discontinuous dynamical systems (DDSs) with sliding motions in Navarro-López (2009) and Navarro-López and Carter (2010). Particularly, these works well-demonstrated the use of hybrid automata to describe the different complex behaviours present in a simplified model of a conventional vertical oilwell drillstring (Navarro-López and Cortés, 2007).

Modelling mechanical systems using hybrid automata becomes more difficult when there is the possibility of multiple contacts. Consider two separate objects sharing a set of contact points along their respective surfaces. Now, consider a third object which collides with one of the two original objects. We know that the collision, despite being separate, will affect the behaviour at the site of the other contacts between the first two objects. Physically, we know this to be caused by energy transfer occurring over extremely short time-scales. However, when we model mechanical systems, we often simplify the problem by assuming rigid bodies that neglect fast-acting internal physical processes. The transition of energy from one contact site to the next is assumed instantaneous, and the mechanism by which this process occurs is invisible to us. We must instead resort to non-analytical numerical procedures, such as linear and nonlinear complementarity or proximal point methods to obtain the contact forces (Pfeiffer, 2008; Acary and Brogliato, 2008; Studer, 2009; Simo and Laursen, 1992; Bayo and Ledesma, 1996; Alart and Curnier, 1991).

In this paper, we present a general hybrid automaton for a class of multi-rigid body mechanical systems. The class studied is any mechanical system that can be modelled as a set of rigid-bodies with strictly convex surfaces, and complete freedom in Cartesian space – that is, translations along and rotations about

⋆ This work has been supported by the Engineering and Physical Sciences Research Council (EPSRC) of the UK under the framework of the project *DYVERSE: A New Kind of Control for Hybrid Systems* (EP/I001689/1). The second author also acknowledges the support of the Research Councils UK under the grant EP/E50048/1.

each axis x,y,z of a Cartesian reference frame. Contact interactions are governed by Newton's impact law and Coulomb friction. An example of this kind of systems might include billiard balls.

We build on previous efforts to model mechanical systems as hybrid automata by supporting multiple contacts in the system that occur simultaneously. This is done by introducing to the hybrid automaton the new concept of the *computation node*. This is one of the key new elements that we add to the hybrid-automaton modelling framework. We propose a hybrid automaton with two types of discrete locations: dynamical discrete locations – the classical discrete locations in a hybrid automaton, that is, locations with an associated dynamical system – and non-dynamical discrete locations – the new type of discrete locations introduced in our modelling framework. The collection of this new type of discrete locations and the resets between them is called a computation node. The computation nodes are inspired by the well-known *mythical modes*, originally proposed in Nishida and Doshita (1987) and later expanded to hybrid systems by Mosterman and Biswas (1998, 2000).

Thus, computation nodes are a collection of non-dynamical discrete states, transitions, and reset functions which are used to find acceptable combinations of contact forces for the configuration of the mechanical system. While in the computation node, the executions of the hybrid automaton are assumed to operate over zero time as the discrete states are non-dynamical. The physical analogue of this is the fast-acting dynamics acting at a microscopic level which we assume to occur instantaneously in the rigid-body mechanics formalism. The inclusion of the computation nodes gives us the flexibility to encode more complex and even iterative algorithms into the hybrid-automaton framework, with the benefit that we can model more complex mechanical systems with multiple contacts.

The resulting computational-mechanical model is referred to as the *Multi-rigid-body hybrid automaton*.

2. PRELIMINARIES

We begin by formalising a rigid-body object or 'entity'. The word entity is preferred over object to avoid confusion with the standard computer science definition of object. Following this, the contact reference frame and its projection matrices and vectors are introduced. Position, velocity and acceleration in the contact reference frame are used in the description of the hybrid automaton. This notation is primarily drawn from (Pfeiffer, 2008) and will, as far as possible, be kept the same.

Definition 1. (Entity) An entity $\mathscr{E}_i$ is a representation of a rigid-body consisting of the collection,

$$\mathscr{E}_i = (q_i, m_i, J_i(q_i)),$$

where:

- $q_i = \left(\bar{q}_i^T, \alpha_i, \beta_i, \gamma_i\right)^T$ is a vector containing the position $\bar{q}_i \in \mathbb{R}^3$ of the centre of rotation of the entity in a global Cartesian coordinate frame, and the rotation of the entity about the x-axis (α_i), about the y-axis (β_i), and about the z-axis (γ_i) of the global frame.
- m_i is the total mass of the entity.
- $J_i : \mathbb{R}^6 \to \mathbb{R}$ is a strictly convex differentiable function, called the surface function, which defines the surface of the entity. Any point p on the surface of the entity satisfies the equality $J_i(q_i) = 0$.

The condition of strict convexity means that any pair of entities can only share one contact point. This simplifies the synthesis of the hybrid automaton in the next section. Note that this restriction does not preclude the multi-contact case. An entity can have multiple contact points provided that each of the contact points is within a different entity.

From now and so on, $|\cdot|$ denotes the absolute value for any real number, or the cardinality of a set, and $\|\cdot\|$ is the 2-norm on the Euclidean space $\mathbb{R}^n$.

Consider the case of two entities $\mathscr{E}_1, \mathscr{E}_2$ about to come into contact. Both entities have surfaces completely determined by $J_1(q_1) = 0$ and $J_2(q_2) = 0$, respectively. Denote p_1 as a point in the global reference frame that lies on the surface of $\mathscr{E}_1$, and p_2 similarly as a point on the surface of $\mathscr{E}_2$. The points are chosen so that when collision occurs, $p_1 = p_2$. We can define a gap function $g_{N,1}$ which returns the distance between the two points in a direction normal to the surface of $\mathscr{E}_1$,

$$g_{N,1} = \mathbf{n}(q_1) \cdot (p_2 - p_1), \tag{1}$$

where $\mathbf{n}(q_1) = \left(\frac{\partial J_1(q_1)}{\partial q_1}\right)^T / \left\| \left(\frac{\partial J_1(q_1)}{\partial q_1}\right)^T \right\|$ is a normal vector from the surface of $\mathscr{E}_1$ at the point p_1. We can put the relative velocity and acceleration between the surface points in terms of entity coordinates q_1, q_2,

$$\dot{g}_{N,1} = \mathbf{w}_{N,1}^T \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} + \dot{\mathbf{n}}^T(q_1)(p_2 - p_1), \tag{2}$$

$$\ddot{g}_{N,1} = \mathbf{w}_{N,1}^T \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} + w_{a^N}(q_1, q_2, \dot{q}_1, \dot{q}_2). \tag{3}$$

We can generalise the above result. Consider a multi-rigid-body system consisting of a set of M entities $\{\mathscr{E}_1, \mathscr{E}_2, ..., \mathscr{E}_M\}$ with $I = \{1, 2, ..., P\}$ points of contact between them occurring at any one time. Let $q = \left(q_1^T, q_2^T, ..., q_M^T\right)^T$ be a vector containing the individual coordinates of each entity. For the i^{th} contact we can define the following by using the procedure described above:

$$\dot{g}_{N,i} = \mathbf{w}_{N,i}^T(q)\dot{q}, \tag{4}$$

$$\ddot{g}_{N,i} = \mathbf{w}_{N,i}^T(q)\ddot{q} + w_{a^N,i}(q, \dot{q}). \tag{5}$$

The following terms are introduced:

$$\dot{g}_N = \left(\dot{g}_{N,1}, \dot{g}_{N,2}, .., \dot{g}_{N,P}\right)^T,$$

$$\mathbf{W}_N = \left(\mathbf{w}_{N,1}, \mathbf{w}_{N,2}, ..., \mathbf{w}_{N,P}\right),$$

$$\mathbf{w}_{a^N}(q, \dot{q}) = \left(w_{a^N,1}, w_{a^N,2}, ..., w_{a^N,P}\right)^T.$$

Then, for the whole system consisting of M entities:

$$\dot{g}_N = \mathbf{W}_N^T(q)\dot{q}, \tag{6}$$

$$\ddot{g}_N = \mathbf{W}_N^T(q)\ddot{q} + \mathbf{w}_{a^N}(q, \dot{q}). \tag{7}$$

The resulting equations (1), (6), (7) are useful for expressing the constraint that no two rigid-bodies can occupy the same space, i.e., penetration cannot occur. In the unilateral constraint formalism, we can state that: at the point of contact, the constraints $g_N \geq 0$, $\dot{g}_N \geq 0$, and $\ddot{g}_N \geq 0$ must be enforced. Furthermore, setting the constraint on the acceleration level allows the direct construction of ordinary differential equations (ODEs). This is necessary for building the hybrid automaton. In simulation, however, setting the constraint on the acceleration level invites drift caused by numerical errors, and some stabilisation constraint is necessary.

We now consider the constraints acting tangentially to the contact surface. Returning to the single contact in equation (1), define two vectors $\mathbf{t}_x(q), \mathbf{t}_y(q)$ orthogonal to one another and to the normal vector $\mathbf{n}(q)$. The combination of the three vectors form a coordinate reference frame at the site of contact. We refer to this frame as the contact frame. The position of point p_2 relative to p_1 projected onto the tangential plane is:

$$g_{T,1} = \begin{pmatrix} g_{x,1} \\ g_{y,1} \end{pmatrix} = \begin{pmatrix} \mathbf{t}_x(q_1) \\ \mathbf{t}_y(q_1) \end{pmatrix} \cdot (p_2 - p_1). \tag{8}$$

Following the same procedure as for the normal component, we can generalise for the i^{th} contact and obtain expressions for the relative tangential velocity and acceleration,

$$\dot{g}_{T,i} = \mathbf{w}_{T,i}^T(q)\dot{q}, \tag{9}$$
$$\ddot{g}_{T,i} = \mathbf{w}_{T,i}^T(q)\ddot{q} + w_{a^T,i}(q,\dot{q}). \tag{10}$$

For the whole system, we obtain:

$$\dot{g}_T = \mathbf{W}_T^T(q)\dot{q}, \tag{11}$$
$$\ddot{g}_T = \mathbf{W}_T^T(q)\ddot{q} + \mathbf{w}_{a^T}(q,\dot{q}), \tag{12}$$

where

$$\dot{g}_T = (\dot{g}_{T,1}, \dot{g}_{T,2}, .., \dot{g}_{T,P})^T,$$
$$\mathbf{W}_T = (\mathbf{w}_{T,1}, \mathbf{w}_{T,2}, ..., \mathbf{w}_{T,P}),$$
$$\mathbf{w}_{a^T}(q,\dot{q}) = \left(w_{a^T,1}, w_{a^T,2}, ..., w_{a^T,P}\right)^T.$$

There are two forces acting at the point of contact. The first is the reaction force, which prevents any two rigid-bodies from occupying the same space, i.e. penetration between entities cannot occur. The second force is friction, which resists motion along the surface of the point of contact.

The reaction force $\lambda_{N,i}$ acts along the surface normal of the point of contact, and is always sufficiently large to enforce the unilateral constraints $g_{N,i} = 0$, $\dot{g}_{N,i} = 0$, and $\ddot{g}_{N,i} = 0$ at the point of contact. We refer to these as the non-penetration constraints or Signorini's contact law. The reaction force can be expressed by the complementarity condition on the acceleration level as,

$$\begin{cases} \lambda_{N,i} \geq 0 \;\; if \;\; \ddot{g}_{N,i} = 0, \\ \lambda_{N,i} = 0 \;\; if \;\; \ddot{g}_{N,i} > 0. \end{cases}$$

Friction acts on the plane tangential to the point of contact $\lambda_{T,i} = (\lambda_{x,i}, \lambda_{y,i})^T$ where $\lambda_{x,i}$ is in the direction of $g_{x,i}$ and $\lambda_{y,i}$ is in the direction of $g_{y,i}$. Here we use the Coulomb friction given by the friction law in (Pang and Trinkle, 1996).

The friction force is bounded such that $\|\lambda_{T,i}\| < \mu_i(\dot{g}_{T,i})\lambda_{N,i}$ where $\mu(\dot{g}_{T,i})$ is the coefficient of friction and can either be a constant or a function of relative velocity. The contact can be in one of three modes; *stick*, *trans*, or *slip* mode. In stick mode, the friction force $\lambda_{T,i}$ is any value within the bound such that the relative acceleration along the surface of the contact point is constrained to zero, that is, the bilateral constraint $\|\ddot{g}_{T,i}\| = 0$ is enforced. In slip mode, the friction force is on the bound, acting in a direction opposing the velocity. The transitional-slip mode is the same as the slip mode but where the friction force acts in a direction opposing acceleration rather than velocity. Put succinctly, the friction force is determined by,

$$\begin{cases} \{\lambda_{T,i} : \|\lambda_{T,i}\| \leq \mu_i(\dot{g}_{T,i})\lambda_{N,i}\} \;\; if \;\; \begin{matrix} \|\ddot{g}_{T,i}\| = 0, \\ \|\dot{g}_{T,i}\| = 0, \end{matrix} \;\; (stick) \\ \lambda_{T,i} = -\dfrac{\ddot{g}_{T,i}}{\|\ddot{g}_{T,i}\|}\mu_i(\dot{g}_{T,i})\lambda_{N,i} \;\; if \;\; \begin{matrix} \|\ddot{g}_{T,i}\| > 0, \\ \|\dot{g}_{T,i}\| = 0, \end{matrix} \;\; (trans) \\ \lambda_{T,i} = -\dfrac{\dot{g}_{T,i}}{\|\dot{g}_{T,i}\|}\mu_i(\dot{g}_{T,i})\lambda_{N,i} \;\; if \;\; \|\dot{g}_{T,i}\| > 0. \; (slip) \end{cases}$$

Impact energy transitions can be modelled using simple Newton's restitution law:

$$\dot{g}_{N,i}^+ = -e_{N,i}\dot{g}_{N,i}^-, \;\; 0 \leq e_{N,i} \leq 1,$$
$$\dot{g}_{T,i}^+ = -e_{T,i}\dot{g}_{T,i}^-, \;\; \|e_{T,i}\| \leq 1,$$

where $\dot{g}_{N,i}^+, \dot{g}_{T,i}^+$ are the post-impact velocities and $\dot{g}_{N,i}^-, \dot{g}_{T,i}^-$ are the pre-impact velocities. The terms $e_{N,i}, e_{T,i}$ are the normal and tangential restitution coefficients respectively. These parameters represent the wave and damping effects that are otherwise ignored in a rigid-body analysis.

We can create complementarity conditions similar to the contact force laws. Define $\Lambda_{N,i}$ as an impulse along the surface normal of the i^{th} contact. Then, based on the non-penetration constraint, we have:

$$\begin{cases} \Lambda_{N,i} \geq 0 \;\; if \;\; \dot{g}_{N,i}^+ + e_{N,i}\dot{g}_{N,i}^- = 0, \\ \Lambda_{N,i} = 0 \;\; if \;\; \dot{g}_{N,i}^+ + e_{N,i}\dot{g}_{N,i}^- > 0, \end{cases}$$

and in the tangential plane,

$$\begin{cases} \{\Lambda_{T,i} : \|\Lambda_{T,i}\| \leq \mu_i\Lambda_{N,i}\} \;\; if \;\;\; \dot{g}_{T,i}^+ + e_{T,i}\dot{g}_{T,i}^- = 0, \\ \Lambda_{T,i} = -\dfrac{\dot{g}_{T,i}}{\|\dot{g}_{T,i}\|}\mu_i\Lambda_{N,i} \;\;\; if \;\; \left\|\dot{g}_{T,i}^+ + e_{T,i}\dot{g}_{T,i}^-\right\| > 0. \end{cases}$$

3. THE MULTI-RIGID-BODY HYBRID AUTOMATON

In this section, we propose what will be referred to as the *Multi-Rigid-Body* (MRB) hybrid automaton. It is based on the hybrid model given in Johansson et al. (1999); Lygeros et al. (1999) and the *Discontinuous Dynamical Systems* (DDS) hybrid automaton proposed in Navarro-López and Carter (2010). For the sake of simplifying the notation, we will not consider inputs and outputs for the basic hybrid MRB hybrid automaton.

3.1 The general MRB hybrid automaton

The MRB hybrid automaton has the following general definition. The details of its elements are explained in the subsequent sections.

Definition 2. (Multi-Rigid-Body hybrid automaton) Consider a system consisting of M entities with P *possible contacts*. The Multi-Rigid-Body (MRB) hybrid automaton is a collection,

$$H_{MRB} = (S, E, \mathscr{X}, Dom, \mathscr{F}, Init, G, R),$$

with:

- S a finite set of **discrete locations**. We will have two types of discrete locations: **dynamical discrete locations** (the classical discrete locations in a hybrid automaton, that is, locations with an associated dynamical system) and **non-dynamical discrete locations** – the new type of discrete locations introduced in our modelling framework – which are grouped into **computation nodes**. The set for dynamical discrete locations is denoted by S_d, and the set of non-dynamical discrete locations is denoted by S_{non}. $S = S_d \bigcup S_{non}$.

- $E \subseteq S \times S$, represents the **transitions** or **edges** in H_{MRB}. The set of edges is finite.
- $\mathscr{X} \subseteq \mathbb{R}^{12M+3P}$ is the **continuous state space**. The continuous state vector is a generalised coordinate vector $q \in \mathbb{R}^{6M}$ and a generalised velocity vector $\dot{q} \in \mathbb{R}^{6M}$ for all the M entities in the system, plus $\lambda_N = \{\lambda_{N,1}, \lambda_{N,2}, ..., \lambda_{N,P}\} \in \mathbb{R}^P$ and $\lambda_T = \left\{\lambda_{T,1}^T, \lambda_{T,2}^T, ..., \lambda_{T,P}^T\right\} \in \mathbb{R}^{2P}$ as the continuous vectors of contact forces, acting at each of the P possible contacts. That is, for $x \in \mathscr{X}$, $x = (q, \dot{q}, \lambda_N, \lambda_T)^T$.
- $Dom : S_d \to 2^{\mathscr{X}}$ are the **location domains**. Dom assigns a set of continuous states to each location of S_d, thus, for $s \in S_d$, $Dom(s) \subseteq \mathscr{X}$.
- $\mathscr{F}$ represents the **continuous dynamics**: $\mathscr{F}$ is a collection of dynamical systems that describe the dynamics in each dynamical discrete location.
- $Init \subseteq S_d \times \mathscr{X}$ is a **set of initial states**.
- G represents the **guard maps** for each transition. $G : E \to 2^{\mathscr{X}}$. G assigns to each edge $e \in E$ a set of continuous states ($G(e) \subset \mathscr{X}$) which enables transitions along that edge.
- R denotes the **reset maps**.

For space reasons, we are not going to define how the MRB hybrid automaton can evolve. That is, the definition of the hybrid time trajectory and the execution of the hybrid automaton on such a trajectory. These definitions are similar to the ones given for classical hybrid automata and can be found in Johansson et al. (1999). The consideration of computation nodes does not change the definition of the time trajectory and the execution of the hybrid automaton, since time does not evolve in the computation nodes. These nodes are considered as nodes that perform some necessary computations, and roughly speaking, they may be considered as part of the dynamical discrete locations of the MRB hybrid automaton.

3.2 Contact List and Contact-Combination Graph

To assist in the construction of the MRB hybrid automaton, two structures are defined: the *contact list* and the *contact-combination graph*.

Denote Γ_N as the set of all gap functions $g_{N,i}$ for every *possible contact* that could occur in the system. Let $\dot{\Gamma}_N = \{\dot{g}_{N,1}, \dot{g}_{N,2}, ...\}$ and $\ddot{\Gamma}_N = \{\ddot{g}_{N,1}, \ddot{g}_{N,2}, ...\}$ be the sets of successive derivatives of the gap functions with respect to time.

Assign to each gap function in the set Γ_N an index or label and let $\bar{I}$ be the collection of all these labels. Thus, $i \in \bar{I} \Leftrightarrow g_{N,i} \in \Gamma_N$. Note that the set $\bar{I}$ naturally labels the elements of Γ_N, $\dot{\Gamma}_N$, and $\ddot{\Gamma}_N$. That is, for some $i \in \bar{I}$, we have $g_{N,i} \in \Gamma_N$, $\dot{g}_{N,i} \in \dot{\Gamma}_N$ and $\ddot{g}_{N,i} \in \ddot{\Gamma}_N$.

The contact combination graph is a structure that contains information on the possible configurations the system of entities could take, and whether it is possible for the system to evolve from one configuration to another. Each vertex of the contact combination graph represents some unique combination of closed and open contacts which could possibly occur at some time in our system. Each edge represents a possible transition from one set of closed and open contacts to another. Denote $\Omega = (V, E_\Omega)$ as the contact combination graph, where V is a set of vertices and E_Ω a set of edges.

For each vertex $V_k \in V$, define $I_k \subseteq 2^{\bar{I}}$ as the set of indices of all closed contacts when the contact combination graph is in the k^{th} vertex,

$$I_k = \left\{ j \in \bar{I} : \; g_{N,j} \in \Gamma_N(q), \; g_{N,j} \leq 0 \right\}.$$

If it is possible for the configuration of the system to directly evolve from configuration I_k, to a configuration I_l then there is an edge between V_k and V_l, that is $(k,l) \in E_\Omega$. From here on, we will refer to any specific contact-combination by its index. That is, when the contact combination graph is in the k^{th} vertex, we say that the system is int the k^{th} contact-combination. Further, as a convention, we will reserve $k = 0$ for the contact-combination where no contacts occur ($I_k = \{\emptyset\}$). Finally, we denote $N_k = |I_k|$ as the number of contacts occurring in the k^{th} contact-combination.

3.3 Elements of the MRB hybrid automaton

Dynamical discrete locations and associated dynamics. For each vertex $V_k \in V$ there is a set of 3^{N_k} dynamical discrete locations:

$$S_k = \left\{ s_{k,1}, s_{k,2}, ..., s_{k,3^{N_k}} \right\}.$$

To each location $s_{k,i}$, we associate the following index sets:

- $I_{st,k,i}$ the indices of all contacts where friction is in *stick* mode:
$$I_{st,k,i} = \left\{ j \in I_k : \; \left\|\dot{g}_{T,j}\right\| = 0 \; \wedge \; \left\|\ddot{g}_{T,j}\right\| = 0 \right\}.$$
- $I_{tr,k,i}$ the indices of all contacts where friction is in *trans* mode:
$$I_{tr,k,i} = \left\{ j \in I_k : \; \left\|\dot{g}_{T,j}\right\| = 0 \; \wedge \; \left\|\ddot{g}_{T,j}\right\| > 0 \right\}.$$
- $I_{sl,k,i}$ the indices of all contacts where friction is in *slip* mode:
$$I_{sl,k,i} = \left\{ j \in I_k : \; \left\|\dot{g}_{T,j}\right\| > 0 \right\}.$$

The set $S_d = \bigcup_{V_k \in V} S_k$. We also note that $I_{st,k,i} \cup I_{tr,k,i} \cup I_{sl,k,i} = I_k$ and $I_{st,k,i} \cap I_{tr,k,i} \cap I_{sl,k,i} = \{\emptyset\}$.

For each dynamical discrete location $s_{k,i}$, we have a continuous dynamical system of the form:

$$\mathbf{M}(q)\ddot{q} = \mathbf{h}(q,\dot{q}) + \left(\mathbf{W}_{N,k}(q) + \mathbf{W}_{R,s_{k,i}}(q,\dot{q}) + \tilde{\mathbf{W}}_{R,s_{k,i}}(q,\dot{q})\right) f_N + \mathbf{W}_{T0,s_{k,i}}(q) f_T, \tag{13}$$

where $\mathbf{M}(q)$ is the mass matrix, $\mathbf{h}(q,\dot{q})$ is a force vector containing all conservative forces, input forces, and Coriolis terms, and:

$$\mathbf{W}_{N,k} = (..., \mathbf{W}_{N,j}, ...), \; f_N = (..., \phi_{N,j}, ...)^T, \; \forall j \in I_k,$$
$$\mathbf{W}_{st,k,i} = (..., \mathbf{W}_{T,j}, ...), \; f_T = \left(..., \phi_{T,j}^T, ...\right)^T, \; \forall j \in I_{st,k,i},$$
$$\mathbf{W}_{tr,k,i} = \left(..., -\mathbf{W}_{T,j} \frac{\ddot{g}_{T,j}^T}{\left\|\ddot{g}_{T,j}\right\|} \mu_j(\dot{g}_{T,j}), ... \right), \; \forall j \in I_{tr,k,i},$$
$$\mathbf{W}_{sl,k,i} = \left(..., -\mathbf{W}_{T,j} \frac{\dot{g}_{T,j}^T}{\left\|\dot{g}_{T,j}\right\|} \mu_j(\dot{g}_{T,j}), ... \right), \; \forall j \in I_{sl,k,i}.$$

The terms $\phi_{N,j} \in \mathbb{R}, \phi_{T,j} \in \mathbb{R}^2$ are obtained from the vectors $\phi_N = (\phi_{N,1}, \phi_{N,2}, ...)^T$ and $\phi_T = \left(\phi_{T,1}^T, \phi_{T,2}^T, ...\right)^T$ given by the following equation:

$$\begin{pmatrix} \phi_N \\ \phi_T \end{pmatrix} = \begin{pmatrix} \mathbf{W}_N^T \mathbf{M}^{-1} \mathbf{W}_N & \mathbf{W}_N^T \mathbf{M}^{-1} \mathbf{W}_T \\ \mathbf{W}_T^T \mathbf{M}^{-1} \mathbf{W}_N & \mathbf{W}_T^T \mathbf{M}^{-1} \mathbf{W}_T \end{pmatrix}^+ \left(\mathbf{M}^{-1}\mathbf{h} + \mathbf{w}_a\right). \tag{14}$$

The $^+$ indicates the pseudo-inverse. If there are no redundant constraints then the matrix on the right-hand side becomes non-singular and the pseudo-inverse can be replaced by a conventional matrix inversion.

Non-dynamical discrete locations and computation nodes. The MRB hybrid automaton includes a new type of discrete locations, referred to as non-dynamical discrete locations. The collection of these discrete locations, and the resets between them, is termed a computation node. The discrete locations in the computation nodes do not have any continuous dynamics. They are used for the computation of the contact forces for each contact combination. This type of locations are inspired by the well-known *mythical modes*, originally proposed in Nishida and Doshita (1987) and later expanded to hybrid systems by Mosterman and Biswas (1998, 2000). The computation nodes have to be understood in this sense, just a set of transition intermediate states associated to a discontinuous change. Further, one might consider the physical analogue of the computational nodes as being the fast-acting dynamics operating at the microscopic scale which we conventionally assume to act instantaneously in the framework of rigid-body mechanics. We detail the internal process of an example the computational nodes in Section 4.

For each vertex $V_k \in \{V_k \in V : N_k > 0\}$, there is a set of non-dynamical locations:

$$\mathscr{I}_k = \left\{\mathscr{I}_{k,en}, \mathscr{I}_{k,1}, ..., \mathscr{I}_{k,ex}\right\},$$
$$\mathscr{C}_k = \left\{\mathscr{C}_{k,en}, \mathscr{C}_{k,1}, ..., \mathscr{C}_{k,ex}\right\},$$

which are called the impact computation node and contact computation node respectively. Both computation nodes have similar form. The sets $\mathscr{I}_k, \mathscr{C}_k$ consist of an entrance discrete-location with subscript *en*, an exit discrete-location with subscript *ex*, and a set of discrete locations in between which are used for the computation of the contact forces. The number of these discrete locations varies in number depending on the algorithm used to calculate the contact forces. The edges going into the computation node are received by the entrance discrete-location, and all edges leaving the computation node leave from the exit discrete-location. The edges connecting the intermediate locations are explained in Section 4. We highlight that every different computation node will have a different number of non-dynamical discrete locations.

Edges or transitions. The set of edges of H_{MRB} consists of the union of the following:

- For all k such that $\{V_k \in V : N_k > 0\}$, there is an edge from the exit node of $\mathscr{I}_k$ to the entrance node of $\mathscr{C}_k$:
$$\left\{\left(\mathscr{I}_{k,ex}, \mathscr{C}_{k,en}\right)\right\}.$$
- For all k and j such that $(k, j) \in E_\Omega$ and $(N_k < N_j \wedge N_k > 0)$, there is an edge going from the exit node of the computation nodes $\mathscr{I}_k$ to the entrance node of $\mathscr{C}_j$:
$$\left\{\left(\mathscr{I}_{k,ex}, \mathscr{C}_{j,en}\right), ...\right\}.$$
- For all k and i such that $\left\{V_k \in V : N_k > 0 \wedge \left|I_{tr,k,i}\right| = 0\right\}$, there is an edge going from each of the discrete locations $s_{k,i} \in S_k$ to the entrance node of $\mathscr{C}_k$:
$$\left\{\left(s_{k,i}, \mathscr{C}_{k,en}\right), ...\right\}.$$
- For all k and i such that $\left\{V_k \in V : N_k > 0 \wedge \left(\left|I_{st,k,i}\right| > 0 \vee \left|I_{tr,k,i}\right| > 0\right)\right\}$, there is an edge going from the exit node of $\mathscr{C}_k$ to each of the discrete locations $s_{k,i} \in S_k$:
$$\left\{\left(\mathscr{C}_{k,ex}, s_{k,i}\right), ...\right\}.$$
- For all k, j and i such that $\left\{V_k \in V : N_k > 0 \wedge \left|I_{tr,k,i}\right| > 0 \wedge \left|I_{tr,k,j}\right| = 0 \wedge \left(I_{tr,k,i} \cup I_{sl,k,i} = I_{sl,k,j}\right)\right\}$, there is an edge going from the location $s_{k,i}$ to the location $s_{k,j}$:
$$\left\{\left(s_{k,i}, s_{k,j}\right), \ldots\right\}.$$
- For all k, j and i such that $(k, j) \in E_\Omega$ and $(N_k < N_j)$, there is an edge going from each of the discrete locations $s_{k,i} \in S_k$ to the entrance node of the computation nodes $\mathscr{I}_j$:
$$\left\{\left(s_{k,i}, \mathscr{I}_{j,en}\right), ...\right\}.$$
- For all k, j and i such that $(k, j) \in E_\Omega$ and $(N_k < N_j \wedge N_j > 0)$ there is an edge going from each of the discrete states in $s_{k,i} \in S_k$ to the entrance node of $\mathscr{C}_j$:
$$\left\{\left(s_{k,i}, \mathscr{C}_{j,en}\right), ...\right\}.$$
- If a discrete location S_0 such that $N_0 = 0$ exists, then for all $\{(k, 0) \in E_\Omega\}$ there is a pair of edges going to and from $\mathscr{I}_k$ and the free discrete state S_0:
$$\left\{\left(\mathscr{I}_{k,ex}, S_0\right), \left(S_0, \mathscr{I}_{k,en}\right)\right\}.$$
and from each dynamical location $s_{k,i} \in S_k$ to the free discrete state S_0:
$$\left\{(s_{k,i}, S_0), \ldots\right\}$$

Guards. The guards for each of the edges in H_{MRB} are as follows:

$$G(\mathscr{I}_{k,ex}, \mathscr{C}_{j,en}) = \left\{x \in \mathscr{X} : (\forall n \in I_j,\, \dot{g}_{N,n} = 0) \wedge (\forall n \in I_k \setminus I_j,\, \dot{g}_{N,n} > 0)\right\},$$
$$G(\mathscr{I}_{k,ex}, \mathscr{C}_{k,en}) = \{x \in \mathscr{X} : \forall n \in I_k,\, \dot{g}_{N,n} = 0\}.$$

Let us define $\Pi_k = \bigcup_{m:(k,m) \in E_\Omega} I_m$. For all $i \in \{1, 2, \ldots, N_k\}$:

$$G(s_{k,i}, \mathscr{C}_{k,en}) = \left\{x \in \mathscr{X} : \left(\left(\exists j \in I_{sl,k,i},\, \left\|\dot{g}_{T,j}\right\| = 0\right) \vee \left(\exists j \in I_{st,k,i},\, \mu_j \phi_{N,j} < \left\|\phi_{T,j}\right\|\right)\right) \wedge (\forall j \in I_k,\, \phi_{N,j} \geq 0) \wedge (\forall j \in \Pi_k \setminus I_k,\, g_{N,j} > 0 \vee \dot{g}_{N,j} > 0)\right\},$$
$$G(\mathscr{C}_{k,ex}, s_{k,i}) = \left\{x \in \mathscr{X} : \left(\forall j \in I_{st,k,i},\, \left\|\ddot{g}_{T,j}\right\| = 0\right) \wedge \left(\forall j \in I_{tr,k,i},\, \left\|\ddot{g}_{T,j}\right\| > 0\right) \wedge \left(\forall j \in I_{sl,k,i},\, \left\|\dot{g}_{T,j}\right\| > 0\right)\right\},$$
$$G(s_{k,i}, s_{k,j}) = \left\{x \in \mathscr{X} : \left(\forall n \in I_{tr,k,i},\, \|\dot{g}_{T,n}\| > 0\right) \wedge (\forall j \in I_k,\, \phi_{N,j} \geq 0) \wedge (\forall j \in \Pi_k \setminus I_k,\, g_{N,j} > 0 \vee \dot{g}_{N,j} > 0)\right\},$$
$$G(s_{k,i}, \mathscr{I}_{j,en}) = \left\{x \in \mathscr{X} : (\forall n \in I_j,\, g_{N,n} \leq 0 \wedge \dot{g}_{N,n} \leq 0) \wedge (\forall n \in \Pi_k \setminus I_j,\, g_{N,n} > 0 \vee \dot{g}_{N,n} > 0)\right\},$$
$$G(s_{k,i}, \mathscr{C}_{j,en}) = \left\{x \in \mathscr{X} : (\forall n \in I_j,\, \phi_{N,n} \geq 0) \wedge (\forall n \in I_k \setminus I_j,\, \phi_{N,n} < 0)\right\},$$

where $\phi_{N,j}, \phi_{T,j}$ are j^{th} rows of the vectors ϕ_N, ϕ_T given in equation (14). Finally, the following guards are specific only to S_0,

$$G(s_{k,i}, S_0) = \left\{x \in \mathscr{X} : \forall j \in I_k,\, \phi_{N,j} < 0\right\},$$
$$G(\mathscr{I}_{k,ex}, S_0) = \{x \in \mathscr{X} : \forall i \in I_k,\, \dot{g}_{N,i} > 0\}.$$

Location domains. The location domains *Dom* are only considered for dynamical discrete locations, $s_{k,i} \in S_d$:

$$Dom(s_{k,i}) = \Big\{ q \in \mathbb{R}^{6M} : \ (\forall j \in I_k, \ \phi_{N,j} \geq 0)$$
$$\wedge \ (\forall j \in \Pi_k \setminus I_k, \ g_{N,j} > 0 \ \vee \ \dot{g}_{N,j} > 0)$$
$$\wedge \ \left(\forall j \in I_{st,k,i}, \ \left\|\dot{g}_{T,j}\right\| = 0 \ \wedge \ \left\|\phi_{T,j}\right\| \leq \mu_j \phi_{N,j}\right)$$
$$\wedge \ \left(\forall j \in I_{tr,k,i} \ \left\|\dot{g}_{T,j}\right\| = 0 \ \wedge \ \left\|\ddot{g}_{T,j}\right\| > 0\right)$$
$$\wedge \ \left(\forall j \in I_{sl,k,i} \ \left\|\dot{g}_{T,j}\right\| > 0\right) \Big\} .$$

Reset maps. All the resets associated with transitions between dynamical discrete locations will not change the continuous state. The resets within the computation nodes are described in the next section. For the sake of clarity, we are taking some liberties with notation and do not employ the typical nomenclature of resets in hybrid automata.

4. THE COMPUTATION NODE

In this section, an example of an impact computation node $\mathscr{I}_k$ is described. It uses the augmented Lagrangian successive over relaxation proximal point method (SORPROX) as described in Studer (2009) to compute the contact forces.

Denote the following as the collections of contact forces or impulses, and contact parameters (coefficients of restitutions and friction) at the points of contact labelled by I_k:

$$f_N = (..., \lambda_{N,j}, ...)^T , \quad f_T = \left(..., \lambda_{T,j}^T, ...\right)^T ,$$
$$\varepsilon_N = (..., e_{N,j}, ...)^T , \quad \varepsilon_T = \left(..., e_{T,j}^T, ...\right)^T ,$$
$$\mu = (..., \mu_j, ...)^T ,$$

for all $j \in I_k$. Note that here we allow $\lambda_{N,j}, \lambda_{T,j}$ to be either forces (in contact computation nodes) or impulses (in impact computation nodes). The augmented Lagrangian functions for impacts using Newton's impact law are as follows:

$$\Phi_{N,i} = -\mathbf{A}_{i,i}^{-1} \sum_{\substack{j=1 \\ j \neq i}}^{N_k} (\mathbf{A}_{i,j} f_{N,j} + \mathbf{B}_{i,j} f_{T,j} - (1+\varepsilon_{N,i})\dot{g}_{N,i}) , \quad (15)$$

$$\Phi_{T,i} = -\mathbf{C}_{i,i}^{-1} \sum_{\substack{j=1 \\ j \neq i}}^{N_k} (\mathbf{B}_{i,j} f_{N,j} + \mathbf{C}_{i,j} f_{T,j} - (1+\varepsilon_{T,i})\dot{g}_{T,i}) , \quad (16)$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are obtained from,

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix} = \begin{pmatrix} \mathbf{W}_{N,k}^T \mathbf{M}^{-1} \mathbf{W}_{N,k} & \mathbf{W}_{N,k}^T \mathbf{M}^{-1} \mathbf{W}_{T,k} \\ \mathbf{W}_{T,k}^T \mathbf{M}^{-1} \mathbf{W}_{N,k} & \mathbf{W}_{T,k}^T \mathbf{M}^{-1} \mathbf{W}_{T,k} \end{pmatrix} ,$$

where as in the previous section,

$$\mathbf{W}_{N,k} = (..., W_{N,j}, ...) , \quad \mathbf{W}_{T,k} = (..., W_{T,j}, ...) , \quad \forall j \in I_k .$$

We start with an entrance discrete state $\mathscr{I}_{k,en}$. All edges going to the computation node go to this state. Equivalently, we will finish on an exit discrete state $\mathscr{I}_{k,ex}$. All edges leaving the computation node leave from this discrete state. The first edge $(\mathscr{I}_{k,en}, \mathscr{I}_{k,1})$ initialises the vectors f_N, f_T:

$$R(\mathscr{I}_{k,en}, \mathscr{I}_{k,1}, x) \Rightarrow f_{N,i} \to 0 \quad \forall i = \{1, 2, ..., N_k\} ,$$
$$R(\mathscr{I}_{k,en}, \mathscr{I}_{k,1}, x) \Rightarrow f_{T,i} \to \begin{pmatrix} 0 & 0 \end{pmatrix}^T , \quad \forall i = \{1, 2, ..., N_k\} .$$

The first step is to compute a value for the forces. For edges $(\mathscr{I}_{k,i}, \mathscr{I}_{k,i+1})$, with $i = 1, 2, .., N_k$, the guards and reset functions are as follows. Note that for each pair of intermediate locations within computation nodes, there are two edges $(\mathscr{I}_{k,i}, \mathscr{I}_{k,i+1})$, each with a different guard and reset:

$$G(\mathscr{I}_{k,i}, \mathscr{I}_{k,i+1}) = \{\Phi_{N,1} \geq 0\},$$
$$R(\mathscr{I}_{k,i}, \mathscr{I}_{k,i+1}, x) \Rightarrow f_{N,i} \to \Phi_{N,i},$$
$$G(\mathscr{I}_{k,i}, \mathscr{I}_{k,i+1}) = \{\Phi_{N,1} < 0\},$$
$$R(\mathscr{I}_{k,i}, \mathscr{I}_{k,i+1}, x) \Rightarrow f_{N,i} \to 0,$$

and for $(\mathscr{I}_{k,N_k+i}, \mathscr{I}_{k,N_k+i+1})$, with $i = 1, 2, .., N_k$:

$$G(\mathscr{I}_{k,N_k+i}, \mathscr{I}_{k,N_k+i+1}) = \{\|\Phi_{T,i}\| \leq \mu f_{N,i}\} ,$$
$$R(\mathscr{I}_{k,N_k+i}, \mathscr{I}_{k,N_k+i+1}, x) \Rightarrow f_{T,i} \to \Phi_{T,i},$$
$$G(\mathscr{I}_{k,N_k+i}, \mathscr{I}_{k,N_k+i+1}) = \{\|\Phi_{T,i}\| > \mu_i f_{N,i}\} ,$$
$$R(\mathscr{I}_{k,N_k+i}, \mathscr{I}_{k,N_k+i+1}, x) \Rightarrow f_{T,i} \to -f_{T,i} \frac{\mu_i f_{N,i}}{\|f_{T,i}\|} .$$

The second step is to check if the approximations for the contact forces have converged. If not, the transition back to state $\mathscr{I}_{k,1}$ is triggered and the first step is repeated. For edges $(\mathscr{I}_{k,2N_k+i}, \mathscr{I}_{k,2N_k+i+1})$, with $i = 1, 2, ..., N_k$:

$$G(\mathscr{I}_{k,2N_k+i}, \mathscr{I}_{k,2N_k+i+1}) = \{(|f_{N,i} - \Phi_{N,i}| < \delta) \vee$$
$$(|f_{N,i} - \Phi_{N,i}| \geq \delta \wedge f_{N,i} = 0)\},$$
$$G(\mathscr{I}_{k,2N_k+i}, \mathscr{I}_{k,1}) = \{(|f_{N,i} - \Phi_{N,i}| \geq \delta) \ \wedge \ (f_{N,i} \neq 0)\},$$

where δ is some small tolerance. For edges $(\mathscr{I}_{3N_k+i}, \mathscr{I}_{3N_k+i+1})$, with $i = 1, 2, ..., N_k$, we have:

$$G(\mathscr{I}_{k,3N_k+i}, \mathscr{I}_{k,3N_k+i+1}) = \{(\|f_{T,i} - \Phi_{T,i}\| < \delta) \vee$$
$$(\|\Phi_{T,i}\| > \mu_i f_{N,i} \ \wedge \ \|f_{T,i}\| \geq \mu_i f_{N,i})\} ,$$
$$G(\mathscr{I}_{k,3N_k+i}, \mathscr{I}_{k,1}) = \{(\|f_{T,i} - \Phi_{T,i}\| \geq \delta) \ \wedge$$
$$(\|\Phi_{T,i}\| \leq \mu_i f_{N,i} \ \vee \ \|f_{T,i}\| < \mu_i f_{N,i})\} .$$

After f_N, f_T have converged, the velocity $\dot{q}$ is reset. This occurs on the edge to the exit discrete state $(\mathscr{I}_{k,4N_k+1}, \mathscr{I}_{k,ex})$. The reset function is,

$$R(\mathscr{I}_{k,4N_k+1}, \mathscr{I}_{k,ex}, x) \Rightarrow \dot{q} \to \mathbf{M}^{-1} \left(\mathbf{W}_{N,k} f_N + \mathbf{W}_{T,k} f_T \right) + \dot{q}. \quad (17)$$

All edges leaving the computation node leave $\mathscr{I}_{k,ex}$. This is the last state of the computation node and is the last to be visited before the execution of the automaton returns to the dynamical discrete locations.

The contact computation node $\mathscr{C}_k$ has the same form as $\mathscr{I}_k$ but with equations (15) and (16) replaced by:

$$\Phi_{N,i} = -\mathbf{A}_{i,i}^{-1} \sum_{\substack{j=1 \\ j \neq i}}^{N_k} (\mathbf{A}_{i,j} f_{N,j} + \mathbf{B}_{i,j} f_{T,j} + \mathbf{c}_{N,i}) , \quad (18)$$

$$\Phi_{T,i} = -\mathbf{C}_{i,i}^{-1} \sum_{\substack{j=1 \\ j \neq i}}^{N_k} (\mathbf{B}_{i,j} f_{N,j} + \mathbf{C}_{i,j} f_{T,j} + \mathbf{c}_{T,i}) , \quad (19)$$

where $\mathbf{c}_N = \mathbf{W}_N^T \mathbf{M}^{-1} \mathbf{h} + \mathbf{w}_{aN}$, $\mathbf{c}_T = \mathbf{W}_T^T \mathbf{M}^{-1} \mathbf{h} + \mathbf{w}_{aT}$ and, now, the reset does not change the velocity $\dot{q}$.

5. EXAMPLE

Consider the three-ball example shown in Figure 1. The three balls are labelled a, b, and c. Each ball will be considered as an entity and the position and orientation of balls is given by the vector $q = \left(q_a^T, q_b^T, q_c^T\right)^T$ and has eighteen components, $q \in \mathbb{R}^{18}$. Each ball has a radius of 1 m, and mass, $m_a, m_b, m_c = 1$ kg. There are three possible contact sites *a-b*, *b-c*, and *a-c*.

This leads to eight possible contact combinations, as shown in Table 1, sixty-four dynamical discrete locations and fourteen computation nodes. The sixty-four discrete locations are obtained from: $|S_d| = 1 + \sum_{k=1}^{7} 3^{N_k} = 64$ (N_k are taken from Table 1). We have fourteen computation nodes because there are two nodes for every contact combination $k > 0$.

The set $\Gamma_N = \left\{g_{N,ab}(q), g_{N,bc}(q), g_{N,ac}(q)\right\}$ contains the normal distances between contact points along each of the possible contacts. At all points of contact, we assume a frictional coefficient $\mu = 0.5$ and a coefficient of restitution $\varepsilon_N = 1$ and $\varepsilon_T = -1$.

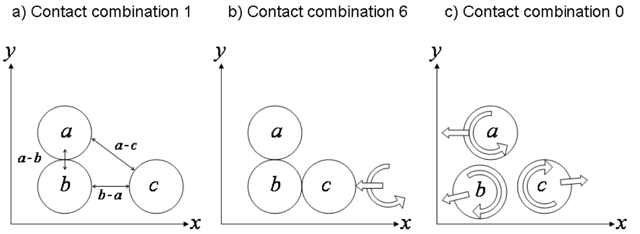

Fig. 1. The three ball example.

k	*a-b*	*a-c*	*b-c*	N_k
0	0	0	0	0
1	1	0	0	1
2	0	1	0	1
3	0	0	1	1
4	1	1	0	2
5	0	1	1	2
6	1	0	1	2
7	1	1	1	3

Table 1. Possible contact combinations in the 3-ball example.

Let us focus on the contact-combinations 1, 6, and 0. A physical interpretation of these contact combinations is shown in Figures 1.a), 1.b) and 1.c), respectively. For the contact combination 1, $g_{N,ab} = 0$, $g_{N,bc} > 0$ and $g_{N,ac} > 0$. There are $3^{N_1} = 3$ dynamical discrete locations $S_1 = \{s_{1,1}, s_{1,2}, s_{1,3}\}$, and two computation nodes, one for contact $\mathscr{C}_1$, and the second for impact $\mathscr{I}_1$. This section of the hybrid automaton is shown in figure 2. We designate $s_{1,1}$ as the location when the contact is in stick mode ($I_{st,1,1} = \{ab\}, I_{tr,1,1} = I_{sl,1,1} = \{\emptyset\}$), $s_{1,2}$ as the location when in trans mode, ($I_{tr,1,1} = \{ab\}, I_{st,1,1} = I_{sl,1,1} = \{\emptyset\}$), and $s_{1,2}$ as the slip mode ($I_{sl,1,1} = \{ab\}, I_{st,1,1} = I_{tr,1,1} = \{\emptyset\}$).

The dynamical discrete locations have the following associated dynamics:

$$s_{1,1} \quad \Rightarrow \quad \mathbf{M}\ddot{q} = \mathbf{h} + \mathbf{W}_{N,1}(q)\phi_N + \mathbf{W}_{st,1,1}(q)\phi_T,$$

$$s_{1,2} \quad \Rightarrow \quad \mathbf{M}\ddot{q} = \mathbf{h} + \left(\mathbf{W}_{N,1}(q) + \mathbf{W}_{tr,1,2}(q)\right)\phi_N,$$

$$s_{1,3} \quad \Rightarrow \quad \mathbf{M}\ddot{q} = \mathbf{h} + \left(\mathbf{W}_{N,1}(q) + \mathbf{W}_{sl,1,3}(q)\right)\phi_N,$$

where $\mathbf{h} = (0, -\delta f, 0, 0, \ldots)^T$, δf is some small force, $\mathbf{M}$ is a diagonal mass matrix containing the masses and moments of inertia of the balls, and

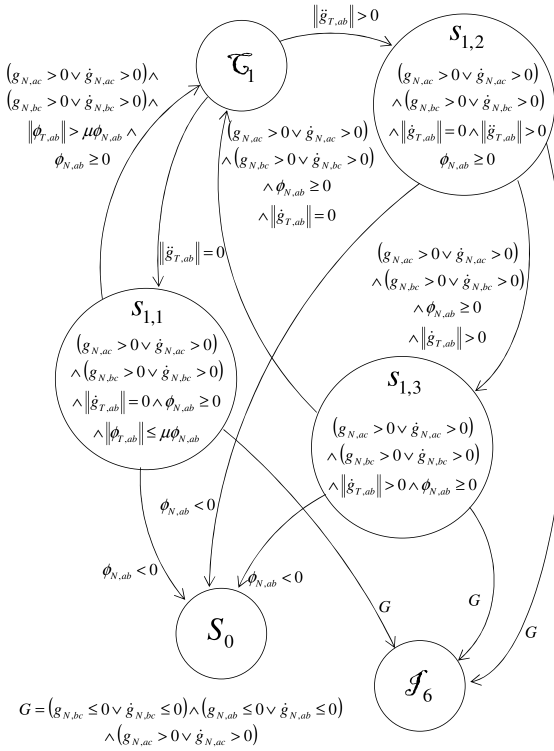

Fig. 2. The section of the hybrid automaton associated with contact combination 1. Notice that $\mathscr{I}_6$ and $\mathscr{C}_1$ are sets of non-dynamical locations forming the computations nodes.

$$\mathbf{W}_{N,1}(q) = \left(0\ 1\ 0\ 0\ 0\ 0\ 0\ -1\ 0\ 0\ 0\ 0\ \mathbf{0}_{1\times 6}\right)^T,$$

$$\mathbf{W}_{T,1}(q) = \begin{pmatrix} 0\ 0\ 1\ -1\ 0\ 0\ \ 0\ \ 0\ -1\ 1\ 0\ \ 0\ \ \mathbf{0}_{1\times 6} \\ 1\ 0\ 0\ \ 0\ \ 0\ 1\ -1\ 0\ \ 0\ \ 0\ 0\ -1\ \mathbf{0}_{1\times 6} \end{pmatrix}^T,$$

$$\mathbf{W}_{st,1,1}(q) = \mathbf{W}_{T,k}(q),$$

$$\mathbf{W}_{tr,1,2}(q) = -\mathbf{W}_{T,1}\frac{\ddot{g}_{T,ab}^T}{\left\|\ddot{g}_{T,ab}\right\|}\mu_1,$$

$$\mathbf{W}_{sl,1,3}(q) = -\mathbf{W}_{T,1}\frac{\dot{g}_{T,ab}^T}{\left\|\dot{g}_{T,ab}\right\|}\mu_1.$$

Note that $\mathbf{W}_{st,1,2}$, $\mathbf{W}_{st,1,3}$, $\mathbf{W}_{tr,1,1}$, $\mathbf{W}_{tr,1,3}$, $\mathbf{W}_{sl,1,1}$, and $\mathbf{W}_{sl,1,2}$ are empty because of our assignment of the index sets.

For the contact combination 6, $g_{N,ab} = g_{N,bc} = 0$ and $g_{N,ac} > 0$. There are $3^{N_6} = 9$ dynamical discrete locations $S_6 = \left\{s_{6,1}, s_{6,2}, \ldots, s_{6,9}\right\}$ and two computation nodes $\mathscr{C}_6$ and $\mathscr{I}_6$. Each dynamical discrete location pair is assigned some combination of stick, trans or slip occurring at the contacts *a-b* and *b-c*, i.e. $s_{6,1}$ (stick,stick), $s_{6,2}$ (stick,trans), $s_{6,3}$ (stick,slip), and so forth. For brevity, we will not state the vector fields and matrices for these states.

For the contact combination 0, $g_{N,ab}, g_{N,bc}, g_{N,ac} > 0$. That is, there are no contacts occurring. For this contact combination, there is only $2^{N_0} = 1$ dynamical discrete location, S_0, and there are no computation nodes. The dynamic equation for location S_0 is simply,

$$\mathbf{M}\ddot{q} = \mathbf{h}. \tag{20}$$

Assume that ball c is travelling along the x-axis towards ball b and is spinning anti-clockwise. Ball b and ball a are both stationary at the locations shown in Figure 1.a). The execution of the hybrid automaton is in location $s_{1,1}$, i.e., a-b contact is closed ($g_{N,ab} = 0$). The other contacts are open ($g_{N,bc} > 0, g_{N,ac} > 0$), and friction at contact site a-b is in stick mode.

At the onset of collision, the guard $G(s_{1,1}, \mathscr{I}_{6,en})$ is triggered. Consequently, a transition takes place from location $s_{1,1}$ to the entrance state $\mathscr{I}_{6,en}$ of the impact computation node. The computation node calculates approximations of the contact forces and iterates until an convergent combination is found. On the transition to the exit state $\mathscr{I}_{6,ex}$ of $\mathscr{I}_6$, the velocities $\dot{q}$ are reset.

At the exit node, the guard $G(\mathscr{I}_{6,ex}, S_0)$ is triggered, causing a transition to the free location S_0. The post-impact outcome is shown in Figure 1.c). The result is that all three balls move away from each other, opening all contacts between them ($g_{N,ab} > 0, g_{N,bc} > 0, g_{N,ac} > 0$).

6. CONCLUSIONS

In this paper, we have presented a novel computational model referred to as the multi-rigid-body hybrid automaton to fully describe the transitions and operation modes present in a general class of mechanical systems with impacts and friction. Significantly, our computational model can accommodate multiple contacts. This extends the already-existing results related to the modelling of systems with impacts by using hybrid automata. One of the chief characteristics of our model is the inclusion of computation nodes to calculate the contact forces. The computation node consists of a set of non-dynamical discrete locations, resets and guards, which can account for the energy transfer not explicitly considered within the rigid-body formalism. This hybrid automaton is well-suited for the formal verification of dynamical properties of realistic mechanical systems. Due to the complexity involved, verifying these systems manually is laborious, and could be eased by automatic computational techniques.

ACKNOWLEDGEMENTS

We are very grateful to Prof. Pieter J. Mosterman for pointing us out the interesting idea of mythical modes. This has been very helpful for building the MRB hybrid automaton. We also want to thank Prof. José Andrés Somolinos Sánchez from the *Universidad Politécnica de Madrid*, who has provided us with continuous feedback in the design of the MRB hybrid automaton. The MRB hybrid automaton is a sophisticated version of the general-purpose transition mechanical system that Prof. Somolinos and Eva Navarro started to sketch several years ago.

REFERENCES

V. Acary and B. Brogliato *Numerical Methods for Nonsmooth Dynamical Systems*. Springer-Verlag, Heidelberg, 2008.

P. Alart, and A. Curnier. A mixed formulation for fictional contact problems prone to Newton like solution methods. *Comp. Methods in App. Mech. and Eng.*, 92:353–375, 1991.

R. Alur, C. Courcoubetis, T.A. Henzinger, and P.H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. *Lecture Notes in Computer Science*, 736:209–229, 1993.

E. Bayo, and R. Laursen. Augmented Lagrangian and Mass-Orthogonal Projection Methods for Constrained Multibody Dynamics. *Nonlinear Dynamics*, 9:113–130, 1996.

B. Brogliato. *Nonsmooth Mechanics*. Springer, London, 2nd ed., 1999.

R. Goebel, R.G. Sanfelice, and A.R. Teel. Hybrid dynamical systems. Robust stability and control for systems that combine continuous-time and discrete-time dynamics. *IEEE Control Systems Magazine*, April:28–93, 2009.

T.A. Henzinger. The theory of hybrid automata. *Proc. 11th IEEE Symp. of Logic in Computer Science*, 1:278–292, 1996.

K.H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry. On the regularization of Zeno hybrid automata. *Systems & Control Letters*, 38:141–150, 1999.

J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3): 349–370, 1999.

J. Lygeros, K.H. Johansson, S.N. Simić, J. Zhang, and S. Sastry. Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48(1):2–17, 2003.

P.J. Mosterman, F. Zhao, and G. Biswas. Sliding mode model semantics and simulation for hybrid systems. *P. Antsaklis et al. (Eds); Hybrid Systems V, LNCS*, 1567:218–237, Springer-Verlag, 1999.

P.J. Mosterman, and G. Biswas. A theory of discontinuities in dynamic physical systems. *Journal of the Franklin Institute*, 335B(3):401–439, 1998.

P.J. Mosterman, and G. Biswas. A comprehensive methodology for building hybrid models of physical systems. *Artificial Intelligence*, 121:171–209, 2000.

P.J. Mosterman. Mode transition behavior in hybrid dynamic systems. *2003 IEEE Winter Simulation Conference*, 623–631, 2003.

E.M. Navarro-López, and D. Cortés. Avoiding harmful oscillations in a drillstring through dynamical analysis. *Journal of Sound and Vibration*, 307(1–2):152–171, 2007.

E.M. Navarro-López. Hybrid modelling of a discontinuous dynamical system including switching control. *2nd IFAC Conference on Analysis and Control of Chaotic Systems*, special session “Hybrid systems and grazing bifurcations”, (London, UK, June 22–24), , 2009.

E.M. Navarro-López, and R.Carter. Hybrid automata: an insight into the discrete abstraction of discontinuous systems. *International Journal of Systems Science*, August:1–16, 2010.

T. Nishida, and S. Doshita. Reasoning about discontinuous change. *6th National Conference on Artificial Intelligence (AAAI-87)*, (Seattle, Washington, USA), 643–648, 1987.

J.S. Pang and J.C. Trinkle. Complementarity formulations and existence of solutions of dynamic multi-rigid-body contact problems with Coulomb friction. *Math. Prog.*, 30:199–226, 1996.

F. Pfeiffer. *Mechanical System Dynamics*. Springer-Verlag, Heidelberg, 2008.

J.C. Simo, and T.A. Laursen. An Augmented Lagrangian Treatment of Contact Problems Involving Friction. *Computers and Structures*, 42(1):91–116, 1992.

C. Studer. *Numerics of Unilateral Contacts and Friction*. Springer-Verlag, Heidelberg, 2009.

F. Zhang, M. Yeddanapudi, and P.J. Mosterman. Zero-crossing location and detection algorithms for hybrid system simulation. *17th IFAC Triennial World Congress*, (Seoul, Korea, July 6–11)7967–7972, 2008.