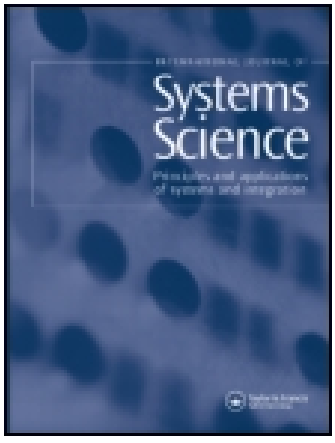

International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

Hybrid automata: an insight into the discrete abstraction of discontinuous systems

Eva M. Navarro-López [a] & Rebekah Carter [a]
[a] School of Computer Science , The University of Manchester , Oxford Road, Kilburn Building, Manchester, M13 9PL, UK
Published online: 19 Aug 2010.

International Journal of Systems Science
Vol. 42, No. 11, November 2011, 1883–1898

Taylor & Francis
Taylor & Francis Group

Hybrid automata: an insight into the discrete abstraction of discontinuous systems

Eva M. Navarro-López* and Rebekah Carter

School of Computer Science, The University of Manchester, Oxford Road, Kilburn Building, Manchester, M13 9PL, UK

(*Received 18 December 2009; final version received 27 April 2010*)

We develop a novel computational–dynamical framework for the modelling of a class of discontinuous dynamical systems (DDSs). In particular, what is referred to as the *DDS hybrid automaton* with inputs and outputs is proposed. This is a general hybrid automaton that provides a suitable mathematical model for DDSs with discontinuous state derivatives and sliding motions. The chief characteristic of this model is that, following the computational *divide-and-conquer* principle, a system with multiple discontinuous elements can be represented by the composition of several *DDS hybrid automata*. Although discontinuous, non-smooth or switched dynamical systems have been well-investigated within different frameworks, it is still a challenge to give satisfactory solutions for specifying the transitions between the different modes of operation of these systems. We propose a new way of solving this problem, which is especially effective for systems with multiple switching elements. An example is used to illustrate these ideas. Several simulations are presented. The simulations results are obtained with Stateflow® and Modelica®.

Keywords: hybrid systems; symbolic dynamics; discontinuous systems; hybrid automata; sliding motions; Stateflow®; Modelica®

1. Motivation

It might be as simple as jumping on a trampoline, or as sophisticated as the docking of vehicles in outer space; as trivial as playing billiards, or as crucial as maintaining the stability of an airplane during landing; as random as the flow of information in a communication network or as synchronised as a school of fish swimming in the sea. These are examples of systems that combine continuous and discrete, smooth and abrupt dynamics. In dynamical systems theory, this is called discontinuity or switching behaviour. The systems exhibiting it are called discontinuous, switched, non-smooth or piecewise-smooth systems. Their combined dynamics can be seen as a hybrid dynamical system.

The term 'hybrid system' has been used to label a wide variety of engineering problems, such as heterogeneous systems, multi-modal systems, multi-controller systems, logic-based switching control systems, discrete-event systems or variable structure systems, among others. In general, hybrid systems are dynamical systems consisting of continuous-type and discrete-event dynamics. They merge formal computational tools, dynamical systems theory and control engineering methodologies.

There are several modelling frameworks for hybrid dynamical systems. Each framework is oriented to specific types of problems and, indeed, reflects the background of the researchers behind it, whether their specialisation is computer science, control engineering or applied mathematics (Witsenhausen 1966; Tavernini 1987; Alur, Courcoubetis, Henzinger, and Ho 1993; Antsaklis, Stiver, and Lemmon 1993; Henzinger 1996; Branicky, Borkar, and Mitter 1998; Ye, Michel, and Hou 1998; van der Schaft and Schumacher 2000; Buss, Glocker, Hardt, von Stryk, and Schmidt 2002). In this article, the hybrid automaton-based framework is used (Alur et al. 1993; Henzinger 1996), which merges continuous dynamics and finite automaton theories. It is a graph-related representation which is closely connected to computer science discrete representations, and also to symbolic dynamics.

The contribution of this article is to reinterpret discontinuous dynamical systems (DDSs) with sliding-type behaviour as hybrid automata with inputs and outputs. The use of a computational model, such as a hybrid automaton, is an elegant way to specify the multiple transitions between the different modes of operation of a discontinuous system.

Three key analysis aspects make the modelling and simulation of DDSs more challenging. Firstly, DDSs entail a wide variety of complex behaviours which usually lead to faults or dynamics degrading performance. These complex behaviours usually have their

*Corresponding author. Email: eva.navarro@cs.man.ac.uk

ISSN 0020–7721 print/ISSN 1464–5319 online

http://dx.doi.org/10.1080/00207721.2010.495189
http://www.tandfonline.com

origin in what is known as a discontinuity-induced bifurcation (DIB; Awrejcewicz and Lamarque 2003; Kunze 2004; di Bernardo, Budd, Champneys, and Kowalczyk 2008). Secondly, the non-uniqueness or even non-existence of solutions when the trajectories of the DDS either cross or slide on a discontinuity surface. Finally, the simulation and numerical integration problems (mainly, the zero-crossing location and detection).

It is well-known that non-uniqueness or even non-existence of solutions may arise in discontinuous systems when the trajectories either cross or slide on a discontinuity surface. This has been extensively studied in systems with Coulomb friction. Even for simple systems, the non-uniqueness of solutions can appear if the system dynamics and all the transitions between the different modes of operation of the system are not appropriately specified (Lotstedt 1991). A complete overview of the problem can be found in Brogliato (1999).

The problem of uniquely defining the solution in a discontinuous system has been solved by means of different methods (Filippov 1988; Utkin 1992). However, there are still different challenges concerning the simulation and the numerical integration of these systems (Acary and Brogliato 2008). There are two main issues, the first of which is the problem of maintaining the trajectory on the discontinuity surface once it has entered the surface; this is called the *tracking error*. Several numerical solutions have solved this problem and are closely related to avoiding the chattering phenomenon (Zhao and Utkin 1996; Mosterman, Zhao, and Biswas 1999). The second major issue is the detection and location of points where the trajectory crosses the discontinuity surface; this is known as *zero-crossing detection* (Park and Barton 1996; Zhang, Yeddanapudi, and Mosterman 2008).

In the past decade, there has been an effort in proposing different semantics and computational-oriented frameworks for modelling systems exhibiting sliding-type behaviour. For example, *object-oriented models* (Elmqvist, Cellier, and Otter 1993; Mattsson 1996) or *hybrid dynamic models* (Mosterman and Biswas 2000) are used for different applications. In this article, two hybrid automaton models are proposed for this same purpose. Our approach differs from these computational models for DDSs. We propose a general framework for a class of discontinuous systems, while these works are focused on particular examples.

The two hybrid automata proposed model general DDSs with sliding-type behaviour and one discontinuity surface. The first hybrid model, called the *DDS hybrid automaton*, has three discrete locations. This model overcomes some problems encountered in the 3-discrete-state object-oriented model given in Mattsson (1996). The second hybrid model is the *extended DDS hybrid automaton* with five discrete locations. It is inspired by simulation-oriented models of discontinuous friction (Karnopp 1985; Leine, van Campen, de Kraker, and van den Steen 1998), and by the state-transition diagram of a friction model presented in Elmqvist et al. (1993). An important characteristic of the models proposed is that systems with multiple discontinuity surfaces can be obtained by the composition of several *DDS hybrid automata* (Navarro-López 2009c).

This article is inspired by the results of Navarro-López (2009c,d) where the *DDS hybrid automata* were originally presented. The basis of these hybrid automaton models is the hybrid model extracted from Johansson, Egerstedt, Lygeros, and Sastry (1999), Lygeros, Tomlin, and Sastry (1999) and Lygeros, Johansson, Simić, Zhang, and Sastry (2003). It is very similar to the *Hybrid State Model* (HSM) proposed in Buss et al. (2002). The main difference between the HSM and the hybrid model used here is that the HSM uses an equation-based representation, and the discontinuity surfaces are defined by means of switching functions instead of guard sets.

The specification of discontinuous systems given in this article leads to a simulation algorithm. The events or discrete transitions between the different modes of operation of the system are defined in order to clearly specify all the possible changes in the dynamics. As a consequence, the hybrid automata proposed can be translated to a program or to any other description language (e.g. Forbus 1984; Kuipers 1986; Brockett 1988; Woods 1991; Egerstedt and Brockett 2003).

In order to validate the models, a system with discontinuous friction and different sliding-mode-related dynamics is considered. It is the torsional model of a conventional vertical oilwell drillstring of 2 degrees of freedom (DOF), which has been widely studied, for instance, in Navarro-López and Cortés (2007), Navarro-López and Licéaga-Castro (2009) and Navarro-López (2009a,b) and references therein. In order to illustrate the use of the basic *DDS hybrid automata* for modelling DDSs with multiple discontinuity surfaces, a sliding-mode-based control is introduced in the example.

The simulation of the hybrid automata is carried out by means of the Simulink/Stateflow® toolbox of MATLAB® (The MathWorks 2008) and Modelica® (Modelica 2009). The translation between the Simulink/Stateflow or Modelica models and the hybrid automata is far from being trivial (Agrawal, Simon, and Karsai 2004; Alur, Kanade, Ramesh, and Shashidhar 2008); however, we do not have major

problems in our example. The simulation results given by the two packages are compared and Modelica® is concluded to be more appropriate in the numerical integration, especially when the number of discrete locations and transitions increase. The simulation of the hybrid automata presented here was extensively studied in Carter (2009).

Whilst a particular validation example is used, the proposed framework is general and applicable to a broader class of physical and engineering systems subject to different types of discontinuous interactions with their environment. The computational framework presented gives a fresh perspective for the dynamical analysis and control design of systems with discontinuities.

2. The DDS hybrid automaton

2.1. *The general hybrid automaton model*

The following general hybrid automaton is used. It is based on the hybrid model given in Johansson et al. (1999) and Lygeros et al. (1999, 2003).

Definition 2.1: A hybrid automaton with inputs and outputs is a collection

$$H = (Q, E, \mathcal{X}, \Sigma, \mathcal{U}, O, \mathcal{Y}, Dom, \mathcal{F}, Init, G, R, h, r),$$

where:

- $Q = \{q_1, q_2, \ldots, q_N\}$ is a finite set of discrete states or locations.
- $E \subseteq Q \times Q$ is a finite set of edges called transitions or events.
- $\mathcal{X} \subseteq \Re^n$ is the continuous state space.
- $\mathcal{U} \subseteq \Re^m$ and $\mathcal{Y} \subseteq \Re^m$ are the continuous input and output spaces, respectively.
- $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_M\}$ is a finite set of symbols labelling the edges and representing the discrete input events.
- $O = \{o_1, o_2, \ldots, o_K\}$ is a finite set of symbols representing the discrete output events.
- $Dom : Q \to 2^{\mathcal{X} \times \mathcal{U}}$ is the location domain. It is a mapping from the locations Q to the set of all subsets of $\mathcal{X} \times \mathcal{U}$, that is, Dom assigns a set of continuous states and inputs to each discrete state $q_i \in Q$, thus, $Dom(q_i) \subset \mathcal{X} \times \mathcal{U}$.
- $\mathcal{F} = \{\mathbf{f}_{q_i}(\mathbf{x}, \mathbf{u}) : q_i \in Q\}$ is the collection of vector fields describing the continuous dynamics such that $f_{q_i} : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$. Each $\mathbf{f}_{q_i}(\mathbf{x}, \cdot)$ is assumed to be Lipschitz continuous on the location domain for q_i in order to ensure that the solution within $Dom(q_i)$ exists and is unique.
- $Init \subseteq Q \times \mathcal{X}$ is a set of initial states.
- $G : E \to 2^{\mathcal{X}}$ is a mapping that defines a guard set. G assigns to each edge $e = (q_i, q_j) \in E$ a set of continuous states ($G(e) \subset \mathcal{X}$). Each guard set plays the role of an enabling condition in order to change the location.
- $R : E \times \mathcal{X} \times \mathcal{U} \to 2^{\mathcal{X}}$ is a reset map for the continuous states for each edge. It is assumed that $\forall e \in E$, $G(e) \neq \emptyset$ and $\forall \mathbf{x} \in G(e)$, $R(e, \mathbf{x}, \mathbf{u}) \neq \emptyset$. This is assumed so that the continuous dynamics cannot be destroyed, only changed.
- $h : Q \times \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$ is the continuous output mapping, there is one for each location.
- $r : Q \times \mathcal{X} \times \Sigma \times \mathcal{U} \to \mathcal{O}$ is the discrete output map, there is one for each location.

As long as the system is within location q_i, the continuous state $\mathbf{x}$ must satisfy $\mathbf{x} \in Dom(q_i)$. The transition from a discrete state q_i to another q_j is enabled when the continuous state $\mathbf{x}$ reaches the guard $G(q_i, q_j) \subset \mathcal{X}$ of some edge $(q_i, q_j) \in E$. Then, the discrete state changes to q_j and at the same time, $\mathbf{x}$ is reset to the value specified by $R(q_i, q_j, \mathbf{x}, \mathbf{u}) \subset \mathcal{X}$. H will be represented as a directed graph (Q, E) with vertices Q and edges E. For each vertex $q_i \in Q$, a set of initial conditions, a vector field and a domain are given. In addition, a guard, a label and a reset function are associated with each edge, $e \in E$. These semantics of H are defined below.

The state of H is $z = (q, \mathbf{x}) \in Q \times \mathcal{X}$. Consider a time domain $\Gamma = \Re^+$ and $T = \{t_0, t_1, \ldots, t_f\}$, with $t_k \in \Gamma$ for every $k \in N$, an ordered set of event time points. T contains the initial time t_0 and all times at which a transition from one discrete location to another occurs. t_f can be infinite. For all $t_k \in T$, the continuous states are written as $\mathbf{x}^k := \mathbf{x}(t_k)$, the locations as $q^k := q(t_k)$, the continuous inputs and outputs as $\mathbf{u}^k := \mathbf{u}(t_k)$ and $\mathbf{y}^k := \mathbf{y}(t_k)$, respectively. We define a continuous input and a continuous output sequence as $\phi_u := (u^0, u^1, \ldots, u^f)$ and $\phi_y := (y^0, y^1, \ldots, y^f)$, respectively. Moreover, a discrete input and a discrete output sequence is defined by $\phi_\sigma := (\sigma^0, \sigma^1, \ldots, \sigma^f)$ and $\phi_o := (o^0, o^1, \ldots, o^f)$, respectively, with $\sigma^k \in \Sigma$, and $o^k \in O$ $\forall k$.

Definition 2.2 (Stursberg 2006)**:** For an input sequence ϕ_u, a feasible execution (or run) of H is a sequence of hybrid states $\phi_z = (z(t_0), z(t_1), \ldots, z(t_f))$, with $z_k := z(t_k) = (q^k, \mathbf{x}^k)$ such that:

- **Initial condition:** $z_0 = (q^0, \mathbf{x}^0)$ with $q^0 = q(t_0) \in Q$ and $\mathbf{x}^0 \in Dom(q^0)$, $\mathbf{x}^0 \notin G((q^0, \cdot))$ for any $(q^0, \cdot) \in E$.
- **State evolution:** $z(t_{k+1})$, with $t_{k+1} := t_k + \tau$, $\tau \in \Gamma$, is obtained from $z(t_k)$ according to:
  - **The continuous evolution:** $\chi : [0, \tau] \to \mathcal{X}$, $\chi(0) = \mathbf{x}^k$, $\dot{\chi}(t) = f_{q^k}(\chi(t), \mathbf{u}^k)$ with unique solution for $t \in [0, \tau]$, and

$\chi(t) \in Dom(q^k)$. Moreover, $h: Q \times \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$, $y = h(q^k, \chi(t), \mathbf{u}^k)$.

- **Discrete transition:** $(q^k, q^{k+1}) \in E$, $\chi(\tau) \in G((q^k, q^{k+1}))$, and $\mathbf{x}^{k+1} = R((q^k, q^{k+1}), \chi(\tau), \mathbf{u}^k) \in Dom(q^{k+1})$.
- **Input and output selection:** $\mathbf{u}^{k+1} \in \mathcal{U}$, $\mathbf{y}^{k+1} \in \mathcal{Y}$, a discrete input $\sigma^{k+1} \in \Sigma$ and a discrete output $o^{k+1} \in O$ are associated with each $(q^k, q^{k+1}) \in E$.

2.2. *DSSs as DDS hybrid automata*

DDSs with discontinuous state derivatives and with one discontinuity (or switching) surface can be modelled as a hybrid automaton H. These systems may exhibit sliding-type behaviour on the discontinuity surface.

A state-dependent input control $\mathbf{u}(\mathbf{x})$ is used, consequently the vector fields can be written as $\mathbf{f}_{q_i}(\mathbf{x})$, and,

$$\dot{\mathbf{x}} = \begin{cases} \mathbf{f}^+(\mathbf{x}) & \text{if } \mathbf{x} \in S^+, \\ \mathbf{f}^-(\mathbf{x}) & \text{if } \mathbf{x} \in S^-, \end{cases} \tag{1}$$

where $\mathbf{x} \in \mathcal{X} \subseteq \Re^n$ is the state vector, $\mathbf{f}^+$ and $\mathbf{f}^-$ are continuous and smooth, and $S^+ = \{\mathbf{x} \in \mathcal{X} : s(\mathbf{x}) > 0\}$, $S^- = \{\mathbf{x} \in \mathcal{X} : s(\mathbf{x}) < 0\}$, with s a smooth scalar function with non-vanishing gradient. The discontinuity surface is $S^0 = \{\mathbf{x} \in \mathcal{X} : s(\mathbf{x}) = 0\}$, and $\mathcal{X} = S^+ \cup S^- \cup S^0$. On S^0, $\mathbf{f}^+(\mathbf{x})$ and $\mathbf{f}^-(\mathbf{x})$ do not agree.

The discontinuity surface is divided into two regions, the sliding set S_s^0, which is closed, and the crossing set S_c^0, which is open. Then $S^0 = S_s^0 \cup S_c^0$. S_s^0 is the set where a sliding motion can take place, and S_c^0 is the set of S^0 within which the trajectory crosses S^0 without sliding. The system dynamics on S^0 are $\dot{\mathbf{x}} = \mathbf{f}_s(\mathbf{x})$, where $\mathbf{f}_s$ is the equivalent dynamics (Filippov 1988; Utkin 1992).

In this article, Utkin's equivalent control method is used (Utkin 1992), which, as it is established in Zhao and Utkin (1996) and Mosterman et al. (1999), gives better chatter-free simulation results for some cases. Then,

$$\mathbf{f}_s(\mathbf{x}) = \frac{\mathbf{f}^+(\mathbf{x}) + \mathbf{f}^-(\mathbf{x})}{2} + u_{eq}(\mathbf{x}) \frac{\mathbf{f}^-(\mathbf{x}) - \mathbf{f}^+(\mathbf{x})}{2}, \tag{2}$$

with $u_{eq}(\mathbf{x}) \in [-1, 1]$ a scalar function playing the role of the equivalent control. From (2), and taking into account that $\mathbf{f}_s$ must be tangential to S^0, one yields to,

$$u_{eq}(\mathbf{x}) = -\frac{\langle \nabla s(\mathbf{x}), \mathbf{f}^+(\mathbf{x}) \rangle + \langle \nabla s(\mathbf{x}), \mathbf{f}^-(\mathbf{x}) \rangle}{\langle \nabla s(\mathbf{x}), \mathbf{f}^-(\mathbf{x}) \rangle - \langle \nabla s(\mathbf{x}), \mathbf{f}^+(\mathbf{x}) \rangle}, \tag{3}$$

where ∇s is the gradient of s and $\langle \cdot, \cdot \rangle$ denotes the scalar product of vectors. The sliding set has the following form,

$$S_s^0 := \{\mathbf{x} \in S^0 : \ -1 \le u_{eq}(\mathbf{x}) \le 1\}. \tag{4}$$

S_c^0 is the complement set of S_s^0 in S^0. It is assumed that there are no points on S_s^0 at which both f^+ and f^- are tangent to S^0. Furthermore, the sliding set is attractive for $\mathbf{x}$ such that $\langle \nabla s(\mathbf{x}), \mathbf{f}^-(\mathbf{x}) \rangle - \langle \nabla s(\mathbf{x}), \mathbf{f}^+(\mathbf{x}) \rangle > 0$ and repulsive for $\mathbf{x}$ such that $\langle \nabla s(\mathbf{x}), \mathbf{f}^-(\mathbf{x}) \rangle - \langle \nabla s(\mathbf{x}), \mathbf{f}^+(\mathbf{x}) \rangle < 0$ (Kuznetsov, Rinaldi, and Gragnani 2003).

Definition 2.3: A DDS hybrid automaton with three discrete states (H_{DDS_i}) describing the dynamics of system (1) is a particular case of H with,

- $Q = \{q_1, q_2, q_3\} = \{\text{slip}^+, \text{slip}^-, \text{stick}\}$, $\mathcal{X} \subseteq \Re^n$.
- $E = \{(q_1, q_2), (q_1, q_3), (q_2, q_1), (q_2, q_3), (q_3, q_1), (q_3, q_2)\}$.
- $\Sigma = \{a, b, c\}$. One edge label is assigned to each guard set.
- $\Sigma = O$. The discrete output events coincide with the discrete input events.
- $Dom(q_1) = S^+ \cup \{\mathbf{x} \in S^0 : u_{eq}(\mathbf{x}) > 1\}$, $Dom(q_2) = S^- \cup \{\mathbf{x} \in S^0 : u_{eq}(\mathbf{x}) < -1\}$, $Dom(q_3) = S_s^0$.
- $f_{q_1}(\mathbf{x}) = f^+(\mathbf{x})$, $f_{q_2}(\mathbf{x}) = f^-(\mathbf{x})$, $f_{q_3}(\mathbf{x}) = f_s(\mathbf{x})$.
- $Init = Q \times \mathcal{X} \backslash U_s$, with $U_s := \{\mathbf{x} \in S_s^0 : \langle \nabla s(\mathbf{x}), \mathbf{f}^-(\mathbf{x}) \rangle - \langle \nabla s(\mathbf{x}), \mathbf{f}^+(\mathbf{x}) \rangle < 0\}$. Then, the problem of non-uniqueness of solutions starting at unstable sliding sets is avoided. Indeed, in most cases, solutions starting away from unstable sliding surfaces do not usually reach them.
- $G(q_1, q_3) = G(q_2, q_3) = S_s^0$, $G(q_1, q_2) = G(q_3, q_2) = \{\mathbf{x} \in S^0 : u_{eq}(\mathbf{x}) < -1\}$, $G(q_2, q_1) = G(q_3, q_1) = \{\mathbf{x} \in S^0 : u_{eq}(\mathbf{x}) > 1\}$.
- $R(q_i, q_j, \mathbf{x}) = \{\mathbf{x}\}$ $\forall i, j$ with $j \neq 3$, $R(q_1, q_3, \mathbf{x}) = R(q_2, q_3, \mathbf{x}) = \{\mathbf{x} : s(\mathbf{x}) = 0\}$.
- $y = h(q_1, x) = h(q_2, x) = h(q_3, x)$ is the continuous output, which is the same for all the locations.

We point out that, in the domains $Dom(q_1)$ and $Dom(q_2)$, the following guard sets are included: $\{\mathbf{x} \in S^0 : u_{eq}(\mathbf{x}) > 1\}$, for $Dom(q_1)$, and $\{\mathbf{x} \in S^0 : u_{eq}(\mathbf{x}) < -1\}$, for $Dom(q_2)$. This has been done to ensure that, as soon as we enter the location, we are not going to go out of it. Under this consideration, the vector fields considered, f^+ or f^-, are still valid, because the guard set added to each domain is only valid at the instant we enter the location, being considered as a transition time. This is overcome in the hybrid model of five locations presented in the next section, because two transitions states from stick to motion are included.

A similar hybrid model to H_{DDS_1} was obtained in Sedghi, Srinivasan, and Longchamp (2002) and Sedghi (2003). The difference with H_{DDS_1} is that the hybrid framework used in Sedghi et al. (2002) and Sedghi (2003) is an equation-based representation.

3. Modifying the DDS hybrid automaton

A new hybrid automaton with five discrete locations is proposed in order to overcome some specification and numerical problems encountered in H_{DDS_1}. It will be called the *extended DDS hybrid automaton*, H_{DDS_2}.

This hybrid automaton considers intermediate transitions when leaving the discontinuity surface (stick-to-slip transition). H_{DDS_2} includes two intermediate states before motion (one for positive velocities, another for negative velocities), such as: $q_5 = \{trans^+\}$ and $q_4 = \{trans^-\}$. Indeed, what we are doing is splitting the *stick* state into three parts: first when the trajectories are within the sliding set (q_3), second and third when the trajectories are within the crossing set on the discontinuity surface (q_4 or q_5).

The necessity of these states for simulation purposes is well-known, see, e.g. Elmqvist et al. (1993) and Mattsson (1996). The *extended DDS hybrid automaton* H_{DDS_2} is also inspired in the state-transition diagram of a friction model presented in Elmqvist et al. (1993). The vector fields associated with q_4 ($f_s^-(\mathbf{x})$) and q_5 ($f_s^+(\mathbf{x})$) have to be different from the vector field within q_3, and they will vary depending on the application (see Section 4 for more details). In addition, for avoiding numerical problems with zero-crossing detection, a zero velocity band is used.

Furthermore, there is no direct switching between $slip^+$ and $slip^-$ or vice-versa. For example, if a transition $slip^+ \rightarrow slip^-$ should be carried out, if the system is in state $slip^+$ and s becomes zero, the system switches to *stick* before checking the conditions for switching to $slip^-$. Moreover, if the system is in location $trans^+$ and the velocity is reversed before starting to move, the system switches to state *stick* before going to $slip^-$. This is a way to avoid the non-determinism discussed in Mattsson (1996).

Definition 3.1: The extended DDS hybrid automaton with five discrete states (H_{DDS_2}) describing the dynamics of system (1) is a particular case of H with,

- $Q = \{q_1, q_2, q_3, q_4, q_5\} = \{slip^+, slip^-, stick, trans^-, trans^+\}$, $\mathcal{X} \subseteq \Re^n$.
- $E = \{(q_1, q_3), (q_2, q_3), (q_3, q_5), (q_3, q_4), (q_4, q_3), (q_5, q_3), (q_4, q_2), (q_5, q_1)\}$.
- $\Sigma = O = \{a, b, c, d\}$.
- $Dom(q_1) = \{\mathbf{x} \in \Re^n : s(\mathbf{x}) > \delta\}$, $Dom(q_2) = \{\mathbf{x} \in \Re^n : s(\mathbf{x}) < -\delta\}$, $Dom(q_3) = G_0^\delta = \{\mathbf{x} \in \Re^n : |s(\mathbf{x})| \leq \delta,\ |u_{\mathrm{eq}}(\mathbf{x})| \leq 1\}$, $Dom(q_4) = G_-^\delta = \{\mathbf{x} \in \Re^n : |s(\mathbf{x})| \leq \delta,\ u_{\mathrm{eq}}(\mathbf{x}) < -1\}$, $Dom(q_5) = G_+^\delta = \{\mathbf{x} \in \Re^n : |s(\mathbf{x})| \leq \delta,\ u_{\mathrm{eq}}(\mathbf{x}) > 1\}$, with $0 < \delta < 1$.
- $f_{q_1}(\mathbf{x}) = f^+(\mathbf{x})$, $f_{q_2}(\mathbf{x}) = f^-(\mathbf{x})$, $f_{q_3}(\mathbf{x}) = f_s(\mathbf{x})$, $f_{q_4}(\mathbf{x}) = f_s^-(\mathbf{x})$, $f_{q_5}(\mathbf{x}) = f_s^+(\mathbf{x})$.
- $Init = Q \times \mathcal{X} \backslash U_s$, with U_s as previously defined.
- $G(q_1, q_3) = G(q_2, q_3) = G(q_5, q_3) = G(q_4, q_3) = G_0^\delta$, $G(q_3, q_4) = G_-^\delta$, $G(q_3, q_5) = G_+^\delta$, $G(q_5, q_1) = G(q_4, q_2) = \{\mathbf{x} \in \Re^n : |s(\mathbf{x})| > \delta\}$.
- $R(q_i, q_j, \mathbf{x}) = \{\mathbf{x}\}\ \forall i, j$ with $j \neq 3$, $R(q_i, q_3, \mathbf{x}) = \{\mathbf{x} : s(\mathbf{x}) = 0\}\ \forall i \in \{1, 2, 4, 5\}$.
- $y = h(q_1, x) = h(q_2, x) = h(q_3, x) = h(q_4, x) = h(q_5, x)$.

It must be pointed out that, in hybrid automata (in general, in hybrid systems), in addition to ensure the existence and uniqueness of solutions for the dynamical systems within each location, we need to consider problems intrinsically associated with the hybrid nature of these systems, mainly: blocking behaviour, non-determinism in discrete transitions and Zeno-type behaviour – that is, an infinite number of discrete transitions in a finite amount of time. A similar phenomenon to Zeno behaviour appears in discontinuous/switched or variable structure systems (which are considered synonyms), and is known as chattering. Chattering can be considered as a class of Zeno behaviour and involves infinitely fast switching in the vicinity of a discontinuity surface (Zhao and Utkin 1996; Mosterman et al. 1999). It is eliminated by the definition of the system dynamics on the discontinuity surface – also referred to as regularisation. In our case, we propose the *DDS hybrid automata* in order to model DDSs with discontinuous state derivatives, and the switching is related to the discontinuity surfaces in the system and the sliding motions on them. Since we have uniquely defined the equivalent dynamics of the system on the discontinuity surfaces, we have avoided Zeno and chattering phenomena. Other regularisation methods have been proposed for other types of hybrid systems (Johansson et al. 1999).

In the next section, H_{DDS_1} and H_{DDS_2} are used to model a system with discontinuous friction. In this example, the discontinuity in the state derivatives comes from a sign-type function (within the friction model). In this type of discontinuous systems, in order to prevent the system trajectories exhibiting the chattering phenomenon, the sign function is substituted by a saturation-type function. In our case, since we have uniquely defined the trajectories of the system on the discontinuity surface, we have eliminated the problem of chattering or Zeno behaviour in the class of DDSs considered.

4. Example: the DDS hybrid automata for a system with friction

The example to validate the general hybrid models is a simplified 2-DOF-model of a vertical oilwell drillstring. It is a particular case of the *n*-DOF model proposed in Navarro-López and Cortés (2007). The change in the number of DOFs only implies the change in the dimension of the continuous state space within each location.

The drillstring torsional behaviour is described by a simple torsional pendulum driven by an electrical motor, and the bit-rock contact is described by a dry friction model. The drill pipes are represented by a linear spring with torsional stiffness k_t and a torsional damping c_t, which connect the inertias J_r and J_b.

The system state vector is $\mathbf{x} = (\dot{\varphi}_r, \varphi_r - \varphi_b, \dot{\varphi}_b)^T$, with φ_i, $\dot{\varphi}_i$ $(i \in \{r, b\})$ the angular displacements and angular velocities of the top-rotary system and the bit. At the top-drive system, a viscous damping torque is considered ($c_r x_1$). $T_m = u$ is the torque applied by a motor at the surface and is considered as a constant input. $T_b(x_3) = c_b x_3 + T_{f_b}(x_3)$ is the torque on the bit with $c_b x_3$ approximating the influence of the mud drilling on the bit behaviour. $T_{f_b}(x_3)$ is the friction modelling the bit-rock contact, and $T_{f_b}(x_3) = f_b(x_3)\text{sign}(x_3)$ with:

$$f_b(x_3) = W_{ob} R_b \left[\mu_{c_b} + (\mu_{s_b} - \mu_{c_b}) \exp^{-\frac{\gamma_b}{v_f}|x_3|} \right], \qquad (5)$$

with $W_{ob} > 0$ the weight on the bit (considered as a constant input), $R_b > 0$ the bit radius; μ_{s_b}, $\mu_{c_b} \in (0, 1)$ the static and Coulomb friction coefficients associated with J_b, $0 < \gamma_b < 1$ and $v_f > 0$. In addition, the Coulomb and static friction torque is T_{c_b} and T_{s_b}, respectively, with $T_{c_b} = W_{ob} R_b \mu_{c_b}$, $T_{s_b} = W_{ob} R_b \mu_{s_b}$. The sign function is considered as

$$\begin{aligned} &\text{sign}(x_3) = x_3/|x_3| \quad \text{if } x_3 \neq 0, \\ &\text{sign}(x_3) \in [-1, 1] \quad \text{if } x_3 = 0. \end{aligned} \qquad (6)$$

The drillstring dynamics is given by

$$\begin{aligned} \dot{x}_1 &= \frac{1}{J_r}[-(c_t + c_r)x_1 - k_t x_2 + c_t x_3 + u], \\ \dot{x}_2 &= x_1 - x_3, \\ \dot{x}_3 &= \frac{1}{J_b}\left[c_t x_1 + k_t x_2 - (c_t + c_b)x_3 - T_{f_b}(x_3)\right], \end{aligned} \qquad (7)$$

or in a compact form, $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) + \mathbf{T}_f(\mathbf{x}(t))$, where $\mathbf{A}$, $\mathbf{B}$ are constant matrices and $\mathbf{T_f}$ represents the torque on the bit. The inputs of the system are u and W_{ob}. The outputs are the angular velocities x_1 and x_3.

For (7) with (5), $s(\mathbf{x}) = s^b(\mathbf{x}) = x_3$, the switching surface is $S_0^b = \{\mathbf{x} \in \Re^3 : x_3 = 0\}$ and the sliding set is $S_s^b = \{\mathbf{x} \in S_0^b : |c_t x_1 + k_t x_2| \leq T_{s_b}\}$. Notice that T_{f_b} plays the role of the equivalent control (T_{fbeq}), and T_{fbeq} is the solution for T_{f_b} of equation $\dot{s}^b = 0$, that is, $u_{eq} = T_{fbeq} = c_t x_1 + k_t x_2 - (c_t + c_b)x_3$. Moreover, $-T_{s_b} \leq T_{fbeq} \leq T_{s_b}$. For more details, the reader is invited to read Navarro-López and Cortés (2007), Navarro-López and Licéaga-Castro (2009) and Navarro-López (2009a).

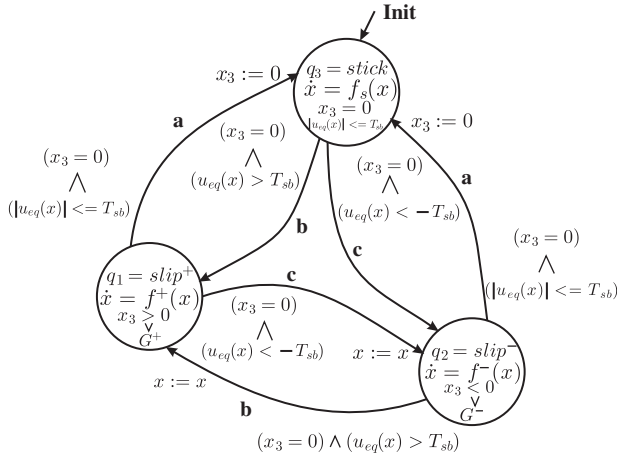

Figure 1. Directed graph associated with the *DDS hybrid automaton* H_{DDS_1} for the drillstring with $x_0 = (x_1, x_2, 0)^T$.

The *DDS hybrid automaton* H_{DDS_1} associated with system (7) and (5) has the following vector fields, guard and domain mappings:

$$\begin{aligned} &\mathbf{f}_{q1}(\mathbf{x}, W_{ob}, u) = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{T}_f(\mathbf{x})\big|_{T_{f_b} = T_{f_b}^+} = f^+(\mathbf{x}), \\ &\mathbf{f}_{q2}(\mathbf{x}, W_{ob}, u) = \mathbf{A}\mathbf{x} + \mathbf{B}u + \mathbf{T}_f(\mathbf{x})\big|_{T_{f_b} = T_{f_b}^-} = f^-(\mathbf{x}), \\ &\mathbf{f}_{q3}(\mathbf{x}, u) = \begin{pmatrix} \frac{1}{J_r}[-(c_t + c_r)x_1 - k_t x_2 + u] \\ x_1 \\ 0 \end{pmatrix} = f_s(\mathbf{x}), \\ &G(q_1, q_3) = G(q_2, q_3) = \{\mathbf{x} \in S_0^b : |u_{eq}(\mathbf{x})| \leq T_{s_b}\} = G^0 = S_s^b, \\ &G(q_1, q_2) = G(q_3, q_2) = \{\mathbf{x} \in S_0^b : u_{eq}(\mathbf{x}) < -T_{s_b}\} = G^-, \\ &G(q_2, q_1) = G(q_3, q_1) = \{\mathbf{x} \in S_0^b : u_{eq}(\mathbf{x}) > T_{s_b}\} = G^+, \\ &Dom(q_1) = \{\mathbf{x} \in \Re^3 : x_3 > 0\} \cup G^+, \\ &Dom(q_2) = \{\mathbf{x} \in \Re^3 : x_3 < 0\} \cup G^-, \\ &Dom(q_3) = S_s^b, \end{aligned} \qquad (8)$$

where $T_{f_b}^+$ and $T_{f_b}^-$ are $T_{f_b}(x_3)$ for $x_3 > 0$ and $x_3 < 0$, respectively. The directed graph associated with this *DDS hybrid automaton* is shown in Figure 1. The guards are close to the departure locations, and the reset functions are close to the arrival locations.

The 3-discrete-state hybrid automaton proposed here overcomes some non-determinism problems encountered in other hybrid formulations of

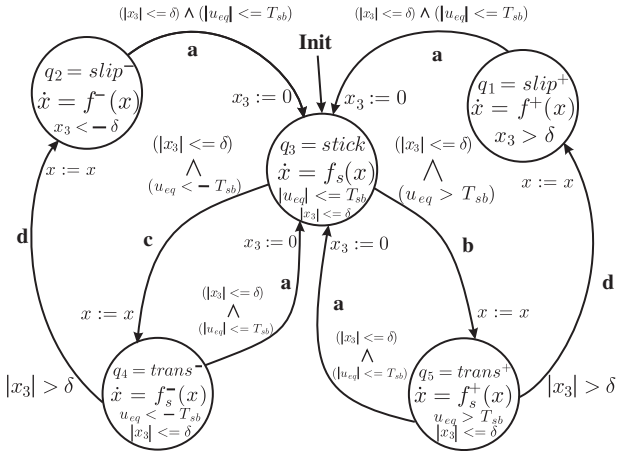

Figure 2. Directed graph associated with the *extended DDS hybrid automaton* H_{DDS_2} for the drillstring with $x_0 = (x_1, x_2, 0)^{\mathrm{T}}$.

discontinuous systems with sliding, for example, like the one presented in Mattsson (1996) where an object-oriented model with three locations is used.

Now, the *extended DDS hybrid automaton* of five discrete states is considered for the example. The discontinuous element is the friction, which is now considered as a combination of the switch model (Leine et al. 1998) and Karnopp's model, in which a zero velocity band is introduced (Karnopp 1985). Thus,

$$T_{\mathrm{f_b}}(\mathbf{x}) = \begin{cases} T_{\mathrm{e_b}}(\mathbf{x}) & \text{if } |x_3| \le \delta,\ |T_{\mathrm{e_b}}| \le T_{\mathrm{s_b}} \text{ (stick)}, \\ T_{\mathrm{s_b}} \mathrm{sign}(T_{\mathrm{e_b}}(\mathbf{x})) & \text{if } |x_3| \le \delta, |T_{\mathrm{e_b}}| > T_{\mathrm{s_b}} \\ & \text{(stick-to-slip transition)} \\ f_{\mathrm{b}}(x_3)\mathrm{sign}(x_3) & \text{if } |x_3| > \delta \text{ (sliding)}, \end{cases} \tag{9}$$

where $\delta > 0$, and $T_{\mathrm{e_b}}$ is the reaction torque, which coincides with the equivalent control.

The vector fields associated with $q_4(f_{\mathrm{s}}^-(\mathbf{x}))$ and $q_5\,(f_{\mathrm{s}}^+(\mathbf{x}))$ are obtained by considering in (7), $x_3 = 0$ and

$$T_{\mathrm{f_b}} = \max s(u_{\mathrm{eq}})\mathrm{sign}(u_{\mathrm{eq}}) = T_{\mathrm{s_b}}\mathrm{sign}(T_{\mathrm{e_b}}).$$

The discrete states, transitions, guards, location domains and reset functions are shown in the graphical representation of the hybrid dynamical system of Figure 2. Note that, when we enter the *stick* location, we will not necessarily have $x_3 = 0$, so we reset to this value on entry to *stick*.

The simulation results for these two hybrid models are given in Section 6.

5. Multiple discontinuity surfaces: composition of DDS hybrid automata

In order to model discontinuous systems with several discontinuity surfaces under the proposed framework, in the example considered in Section 4, another discontinuity surface is included, $s^{\mathrm{r}} = 0$, along which the system exhibits desired dynamics. For this purpose, a discontinuous control u is proposed so that the system trajectory reaches the surface $s^{\mathrm{r}} = 0$ and enters a sliding motion. Thus, the following controller is considered (Navarro-López and Licéaga-Castro 2009; Navarro-López 2009b):

$$\begin{aligned} s^{\mathrm{r}}(\mathbf{x}, t) &= (x_1 - \Omega) + \lambda \int_0^t [x_1(\tau) - \Omega]\mathrm{d}\tau \\ &\quad + \lambda \int_0^t [x_1(\tau) - x_3(\tau)]\mathrm{d}\tau, \quad \lambda > 0, \\ u &= c_{\mathrm{t}}(x_1 - x_3) + k_{\mathrm{t}}x_2 + c_{\mathrm{r}}x_1 - J_{\mathrm{r}}[\lambda(x_1 - \Omega) \\ &\quad + \lambda(x_1 - x_3) + \eta\,\mathrm{sign}(s^{\mathrm{r}})], \quad \eta > 0, \end{aligned} \tag{10}$$

where $\Omega > 0$ is the desired top-rotary velocity. It is ensured that $s^{\mathrm{r}}(\mathbf{x}, t)$ becomes zero in a finite time interval $t_{\mathrm{sr}} = \frac{|s^{\mathrm{r}}(\mathbf{x}, t_0)|}{\eta}$. Two new states x_4, x_5 are defined, such that, $\dot{x}_4 = x_1 - \Omega$ and $\dot{x}_5 = x_1 - x_3$. The following switching surface is defined: $S_0^{\mathrm{r}} := \{\mathbf{x} \in \Re^5 : s^{\mathrm{r}}(\mathbf{x}, t) = 0\}$. This surface has been designed in such a way to be attractive for all $\mathbf{x}$ and to be a sliding set for all $\mathbf{x} \in S_0^{\mathrm{r}}$. Control u is of switched type, with the form:

$$u = \begin{cases} u^+ & \text{if } s^{\mathrm{r}} > 0 \\ u^- & \text{if } s^{\mathrm{r}} < 0 \end{cases}, \tag{11}$$

obtaining u^+ and u^- by changing the sign of s^{r} in (10). From (10), the equivalent control associated with S_0^{r} is $u^- < u_{\mathrm{eq}}^{\mathrm{r}} < u^+$, with

$$\begin{aligned} u_{\mathrm{eq}}^{\mathrm{r}}(\mathbf{x}) &= c_{\mathrm{t}}(x_1 - x_3) + k_{\mathrm{t}}x_2 + c_{\mathrm{r}}x_1 \\ &\quad - J_{\mathrm{r}}[\lambda(x_1 - \Omega) + \lambda(x_1 - x_3)]. \end{aligned} \tag{12}$$

Consequently, the dynamics on S_0^{r} has the following form:

$$\begin{aligned} \dot{\mathbf{x}} &= f_{\mathrm{s}}^{\mathrm{r}}(\mathbf{x}, u)\big|_{u = u_{\mathrm{eq}}^{\mathrm{r}}} \\ &= \begin{pmatrix} -\lambda(x_1 - \Omega) - \lambda(x_1 - x_3) \\ x_1 - x_3 \\ \dfrac{1}{J_{\mathrm{b}}}\big[c_{\mathrm{t}}x_1 + k_{\mathrm{t}}x_2 - (c_{\mathrm{t}} + c_{\mathrm{b}})x_3 - T_{\mathrm{f_b}}(x_3)\big] \\ x_1 - \Omega \\ x_1 - x_3 \end{pmatrix}. \end{aligned}$$

In addition, control u has modified the dynamics on S_0^{b}, and

$$\dot{\mathbf{x}} = f_{\mathrm{s}}^{\mathrm{b}}(\mathbf{x}) = \begin{pmatrix} -2\lambda x_1 + \lambda\Omega - \eta\,\mathrm{sign}(s^{\mathrm{r}}) \\ x_1 \\ 0 \\ x_1 - \Omega \\ x_1 \end{pmatrix}.$$

In order to obtain the hybrid automaton associated with the closed-loop system (7)–(10), we will make the composition of several *DDS hybrid automata*. The basic hybrid automaton H_{DDS_1} is used and the result is a 9-location hybrid automaton described as follows. Similar results would be obtained by using H_{DDS_2}.

The switching surfaces S_0^{b} and S_0^{r} divide the state space in four regions where the system is smooth. For each of these regions, a location is defined:

$$q_1 = \{slip_{\mathrm{b}}^{+}, slip_{\mathrm{r}}^{+}\}, \quad q_2 = \{slip_{\mathrm{b}}^{+}, slip_{\mathrm{r}}^{-}\},$$
$$q_3 = \{slip_{\mathrm{b}}^{-}, slip_{\mathrm{r}}^{+}\}, \quad q_4 = \{slip_{\mathrm{b}}^{-}, slip_{\mathrm{r}}^{-}\}.$$

The domains and vector fields for each of these locations are:

$$\begin{aligned}
Dom(q_1) &= \{\mathbf{x} \in \Re^5 :\ s^{\mathrm{b}}(\mathbf{x}) > 0,\ s^{\mathrm{r}}(\mathbf{x}, t) > 0\} \cup \left(G^{+} \cap S_{+}^{\mathrm{r}}\right) \\
&= S_{+}^{\mathrm{r}} \cap \left(S_{+}^{\mathrm{b}} \cup G^{+}\right), \\
Dom(q_2) &= \{\mathbf{x} \in \Re^5 :\ s^{\mathrm{b}}(\mathbf{x}) > 0,\ s^{\mathrm{r}}(\mathbf{x}, t) < 0\} \cup \left(G^{+} \cap S_{-}^{\mathrm{r}}\right) \\
&= S_{-}^{\mathrm{r}} \cap \left(S_{+}^{\mathrm{b}} \cup G^{+}\right), \\
Dom(q_3) &= \{\mathbf{x} \in \Re^5 :\ s^{\mathrm{b}}(\mathbf{x}) < 0,\ s^{\mathrm{r}}(\mathbf{x}, t) > 0\} \cup \left(G^{-} \cap S_{+}^{\mathrm{r}}\right) \\
&= S_{+}^{\mathrm{r}} \cap \left(S_{-}^{\mathrm{b}} \cup G^{-}\right), \\
Dom(q_4) &= \{\mathbf{x} \in \Re^5 :\ s^{\mathrm{b}}(\mathbf{x}) < 0,\ s^{\mathrm{r}}(\mathbf{x}, t) < 0\} \cup \left(G^{-} \cap S_{-}^{\mathrm{r}}\right) \\
&= S_{-}^{\mathrm{r}} \cap \left(S_{-}^{\mathrm{b}} \cup G^{-}\right),
\end{aligned}$$

$$\mathbf{f}_{q_1} = \begin{pmatrix} \varphi_1(\mathbf{x}) - \eta \\ x_1 - x_3 \\ \varphi_2(\mathbf{x}) - \frac{T_{\mathrm{f_b}}^{+}(x_3)}{J_{\mathrm{b}}} \\ x_1 - \Omega \\ x_1 - x_3 \end{pmatrix}, \quad \mathbf{f}_{q_2} = \begin{pmatrix} \varphi_1(\mathbf{x}) + \eta \\ x_1 - x_3 \\ \varphi_2(\mathbf{x}) - \frac{T_{\mathrm{f_b}}^{+}(x_3)}{J_{\mathrm{b}}} \\ x_1 - \Omega \\ x_1 - x_3 \end{pmatrix},$$

$$\mathbf{f}_{q_3} = \begin{pmatrix} \varphi_1(\mathbf{x}) - \eta \\ x_1 - x_3 \\ \varphi_2(\mathbf{x}) - \frac{T_{\mathrm{f_b}}^{-}(x_3)}{J_{\mathrm{b}}} \\ x_1 - \Omega \\ x_1 - x_3 \end{pmatrix}, \quad \mathbf{f}_{q_4} = \begin{pmatrix} \varphi_1(\mathbf{x}) + \eta \\ x_1 - x_3 \\ \varphi_2(\mathbf{x}) - \frac{T_{\mathrm{f_b}}^{-}(x_3)}{J_{\mathrm{b}}} \\ x_1 - \Omega \\ x_1 - x_3 \end{pmatrix},$$

with G^0, G^{+} and G^{-} as defined in (8), but with $\mathbf{x} \in \Re^5$, and

$$S_{+}^{\mathrm{r}} = \{\mathbf{x} \in \Re^5 : s^{\mathrm{r}}(\mathbf{x}, t) > 0\}, \quad S_{-}^{\mathrm{r}} = \{\mathbf{x} \in \Re^5 : s^{\mathrm{r}}(\mathbf{x}, t) < 0\},$$
$$S_{+}^{\mathrm{b}} = \{\mathbf{x} \in \Re^5 : x_3 > 0\}, \quad S_{-}^{\mathrm{b}} = \{\mathbf{x} \in \Re^5 : x_3 < 0\}.$$

In addition, $\varphi_1(\mathbf{x}) = -\lambda(x_1 - \Omega) - \lambda(x_1 - x_3)$ and $\varphi_2(\mathbf{x}) = \frac{1}{J_{\mathrm{b}}}[c_{\mathrm{t}}x_1 + k_{\mathrm{t}}x_2 - (c_{\mathrm{t}} + c_{\mathrm{b}})x_3]$. As we mentioned previously for the 3-location hybrid automaton, we have included the guards defined by $G(q_j, q_i)$ in the domains $Dom(q_i)$.

The second group of locations corresponds to dynamics on the switching surfaces:

$$q_5 = \{slip_{\mathrm{b}}^{+}, stick_{\mathrm{r}}\}, \quad q_6 = \{stick_{\mathrm{b}}, stick_{\mathrm{r}}\},$$
$$q_7 = \{slip_{\mathrm{b}}^{-}, stick_{\mathrm{r}}\}, \quad q_8 = \{stick_{\mathrm{b}}, slip_{\mathrm{r}}^{+}\},$$
$$q_9 = \{stick_{\mathrm{b}}, slip_{\mathrm{r}}^{-}\}.$$

The domains and vector fields for each of these locations are:

$$\begin{aligned}
Dom(q_5) &= \{\mathbf{x} \in \Re^5 :\ s^{\mathrm{b}}(\mathbf{x}) > 0,\ s^{\mathrm{r}}(\mathbf{x}, t) = 0\} \cup \left(G^{+} \cap S_0^{\mathrm{r}}\right) \\
&= S_0^{\mathrm{r}} \cap \left(S_{+}^{\mathrm{b}} \cup G^{+}\right), \\
Dom(q_6) &= G^0 \cap S_0^{\mathrm{r}}, \\
Dom(q_7) &= \{\mathbf{x} \in \Re^5 :\ s^{\mathrm{b}}(\mathbf{x}) < 0,\ s^{\mathrm{r}}(\mathbf{x}, t) = 0\} \cup \left(G^{-} \cap S_0^{\mathrm{r}}\right) \\
&= S_0^{\mathrm{r}} \cap \left(S_{-}^{\mathrm{b}} \cup G^{-}\right), \\
Dom(q_8) &= G^0 \cap S_{+}^{\mathrm{r}}, \\
Dom(q_9) &= G^0 \cap S_{-}^{\mathrm{r}},
\end{aligned}$$

$$\mathbf{f}_{q_5} = \begin{pmatrix} \varphi_1(\mathbf{x}) \\ x_1 - x_3 \\ \varphi_2(\mathbf{x}) - \frac{T_{\mathrm{f_b}}^{+}(x_3)}{J_{\mathrm{b}}} \\ x_1 - \Omega \\ x_1 - x_3 \end{pmatrix}, \quad \mathbf{f}_{q_6} = \begin{pmatrix} -2\lambda x_1 + \lambda\Omega \\ x_1 \\ 0 \\ x_1 - \Omega \\ x_1 \end{pmatrix},$$

$$\mathbf{f}_{q_7} = \begin{pmatrix} \varphi_1(\mathbf{x}) \\ x_1 - x_3 \\ \varphi_2(\mathbf{x}) - \frac{T_{\mathrm{f_b}}^{-}(x_3)}{J_{\mathrm{b}}} \\ x_1 - \Omega \\ x_1 - x_3 \end{pmatrix}, \quad \mathbf{f}_{q_8} = \begin{pmatrix} -2\lambda x_1 + \lambda\Omega - \eta \\ x_1 \\ 0 \\ x_1 - \Omega \\ x_1 \end{pmatrix},$$

$$\mathbf{f}_{q_9} = \begin{pmatrix} -2\lambda x_1 + \lambda\Omega + \eta \\ x_1 \\ 0 \\ x_1 - \Omega \\ x_1 \end{pmatrix},$$

with $S_0^{\mathrm{r}} = \{\mathbf{x} \in \Re^5 : s^{\mathrm{r}}(\mathbf{x}, t) = 0\}$.

Some features of the system will be considered as constraints in order to reduce the number of feasible transitions between locations. The most important characteristic is that the surface S_0^{r} is attractive for all $\mathbf{x}$ and that all trajectories reach S_0^{r} in a finite time, and once the trajectory reaches S_0^{r}, it remains there. Because of these facts, once the system reaches locations q_5, q_6 and q_7, its future transitions will be restricted to these three locations. This can be considered as a desired *recurrent loop*. In addition, unfeasible

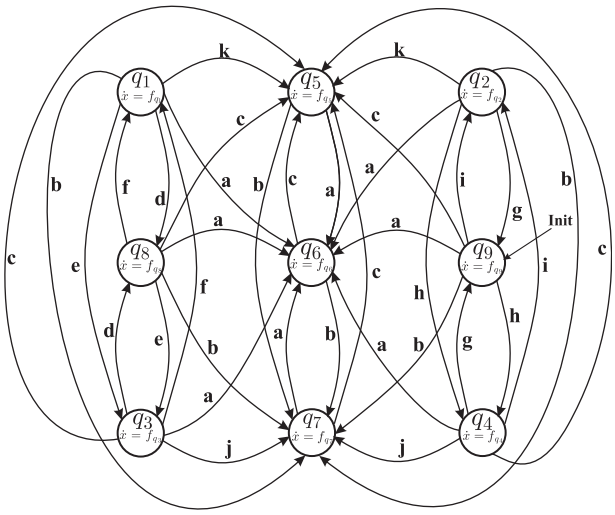

Figure 3. Graphical representation of the hybrid automaton for the closed-loop drillstring using the *DDS hybrid automaton* H_{DDS_1}.

transitions are also the transitions involving a cross through S_0^{r}. Consequently, there are 36 feasible edges or transitions:

$$\begin{aligned} E_{\mathrm{fea}} = \big\{&(q_1,q_8),(q_1,q_3),(q_3,q_1),(q_3,q_8),(q_8,q_1),(q_8,q_3),\\ &(q_3,q_5)(q_5,q_6),(q_5,q_7),(q_7,q_5),(q_7,q_6),(q_6,q_5),\\ &(q_6,q_7),(q_1,q_7)(q_2,q_9),(q_2,q_4),(q_4,q_2),(q_4,q_9),\\ &(q_9,q_2),(q_9,q_4),(q_9,q_5),(q_1,q_5),(q_8,q_6),(q_3,q_7),\\ &(q_2,q_5),(q_9,q_6),(q_4,q_7),(q_9,q_7),(q_1,q_6),(q_2,q_6),\\ &(q_3,q_6),(q_4,q_6),(q_8,q_5),(q_8,q_7),(q_2,q_7),(q_4,q_5)\big\}. \end{aligned}$$

In Figure 3, the graphical representation of the resulting hybrid control system is given. The 11 symbols associated with the edges represent the 11 types of guards in the hybrid automaton, that is:

$$\begin{aligned} &\mathbf{a} \Leftrightarrow G^0 \cap S_0^{\mathrm{r}}, \quad \mathbf{b} \Leftrightarrow G^- \cap S_0^{\mathrm{r}}, \quad \mathbf{c} \Leftrightarrow G^+ \cap S_0^{\mathrm{r}},\\ &\mathbf{d} \Leftrightarrow G^0 \cap S_+^{\mathrm{r}}, \quad \mathbf{e} \Leftrightarrow G^- \cap S_+^{\mathrm{r}}, \quad \mathbf{f} \Leftrightarrow G^+ \cap S_+^{\mathrm{r}},\\ &\mathbf{g} \Leftrightarrow G^0 \cap S_-^{\mathrm{r}}, \quad \mathbf{h} \Leftrightarrow G^- \cap S_-^{\mathrm{r}}, \quad \mathbf{i} \Leftrightarrow G^+ \cap S_-^{\mathrm{r}},\\ &\mathbf{j} \Leftrightarrow S_-^{\mathrm{b}} \cap S_0^{\mathrm{r}}, \quad \mathbf{k} \Leftrightarrow S_+^{\mathrm{b}} \cap S_0^{\mathrm{r}}. \end{aligned}$$

As is appreciated from Figure 3, the hybrid control system consists of three DDS hybrid automata represented by these three groups of locations: $\{q_1, q_8, q_3\}$, $\{q_5, q_6, q_7\}$, $\{q_2, q_9, q_4\}$.

6. Simulation of the hybrid automata: Modelica versus Stateflow

Stateflow® under Simulink® in MATLAB® and The MathWorks (2008) and Modelica® (Modelica 2009) under Dymola[1] are the software packages used to simulate the three *DDS hybrid automata* for the example. Firstly, a comparison of Stateflow and Modelica results is presented for the 3-location and 5-location hybrid automata. Secondly, for the 9-location hybrid automaton, having proved that Modelica is much superior, the simulations with Modelica are given.

6.1. *Comparing Stateflow and Modelica for the basic hybrid models*

The translation of the hybrid automata H_{DDS_1} and H_{DDS_2} into Stateflow charts is almost immediate. The Stateflow charts obtained look the same as the directed graphs associated with H_{DDS_1} and H_{DDS_2} given in Figures 1 and 2. Only, in the Stateflow chart for H_{DDS_1}, an additional boolean variable is introduced. Due to the difficulty in detecting the zero-crossing of the functions involved, $x_3 = 0$ is checked by means of the condition *novelocity* $== 1$. The boolean variable *novelocity* is obtained by passing x_3 through the *Threshold* Simulink block, and it is an external input to the Stateflow chart. This boolean variable is not necessary in H_{DDS_2}. The Simulink/Stateflow models for H_{DDS_1} and H_{DDS_2} are shown in Figures 4 and 5, respectively.

The simulations under Stateflow® and Modelica® of the two basic hybrid automata are compared with the simulation of the discontinuous system (7) with the friction model (9) in Figures 6–10. For the simulation of the discontinuous model, the integration function used is *ode45* of MATLAB®, and in order to have it as an accurate reference trajectory, the maximum step size and the tolerance are considered as 10^{-8}. The system parameters used for the simulations are:

$$\begin{aligned} &J_r = 2122\,\mathrm{kg\,m^2}, \quad J_b = 471.9698\,\mathrm{kg\,m^2},\\ &R_b = 0.155575\,\mathrm{m}, \quad k_t = 861.5336\,\mathrm{N\,m/rad},\\ &c_t = 172.3067\,\mathrm{N\,m\,s/rad}, \quad c_r = 425\,\mathrm{N\,m\,s/rad}, \quad (13)\\ &c_b = 50\,\mathrm{N\,m\,s/rad}, \quad \mu_{c_b} = 0.5,\\ &\mu_{s_b} = 0.8, \quad \delta = 10^{-6}, \quad \gamma_b = 0.9, \quad v_f = 1. \end{aligned}$$

From the figures, it can be seen how the hybrid automata reproduce the three main types of bit dynamical behaviour: positive velocity equilibrium, permanently stuck bit and stick–slip motion. Firstly, Figure 6 shows the positive velocity behaviour, which is such that the bit velocity converges to a positive equilibrium value, the same as the velocity of the top of the bit. The second behaviour (permanently stuck bit) occurs when the bit stops rotating after a period of time and never starts again, i.e. $x(t) \in S_0^{\mathrm{b}}\ \forall t > t_{\mathrm{s}}$, for some $t_{\mathrm{s}} > 0$ (Figure 7). Finally, Figure 8 shows stick–slip motion of the bit, which is where the bit velocity (x_3) oscillates between zero and positive velocity; the system enters and leaves repeatedly the sliding mode.

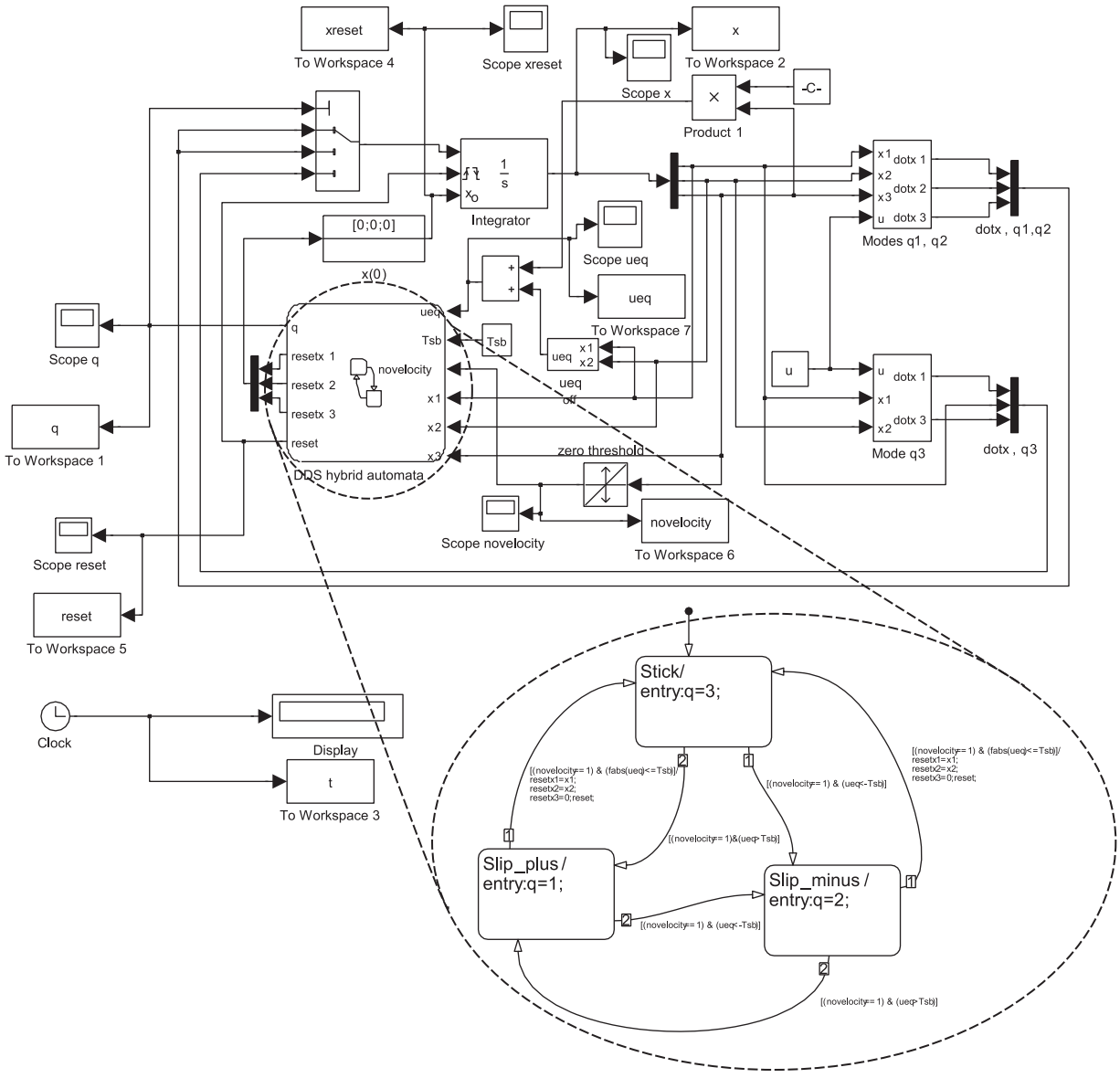

Figure 4. The Simulink/Stateflow model for H_{DDS_1}.

We should note that, in the evolution of q in H_{DDS_2}, it appears that we have an impulse when we change from $q=3$ to $q=1$ via $q=5$. When $q=3$ and u_{eq} becomes greater than $T_{\mathrm{s_b}}$, there is a change to $q=5$. However, this location corresponds to values of x_3 such that $|x_3| < \delta$, and since x_3 immediately starts to be increased when we enter this location, it very soon becomes large enough to leave this location. Hence, we have this impulsive-type q evolution.

6.2. *Choosing the step size and tolerance*

Several features are observed from the simulations, the first of which is the importance of the step size and the tolerance used in the numerical integration method. These two parameters have more impact on the Stateflow simulations than on the Modelica ones. For the Stateflow simulations, the function *ode45* of MATLAB® is used, which is a variable-step integration method. The maximum step size is changed in order to appreciate its effect on the system solutions obtained. In the case of Modelica, the only parameter of the integrator that can be changed within Dymola is the tolerance. Taking the tolerance to be too large can lead to incorrect long-term dynamical behaviour being seen, as we find is the case with either step size or tolerance being too large in the Stateflow simulations.

In Figures 6–8, the simulations obtained with Modelica and Stateflow are presented. The trajectories obtained in both cases are very similar. For the

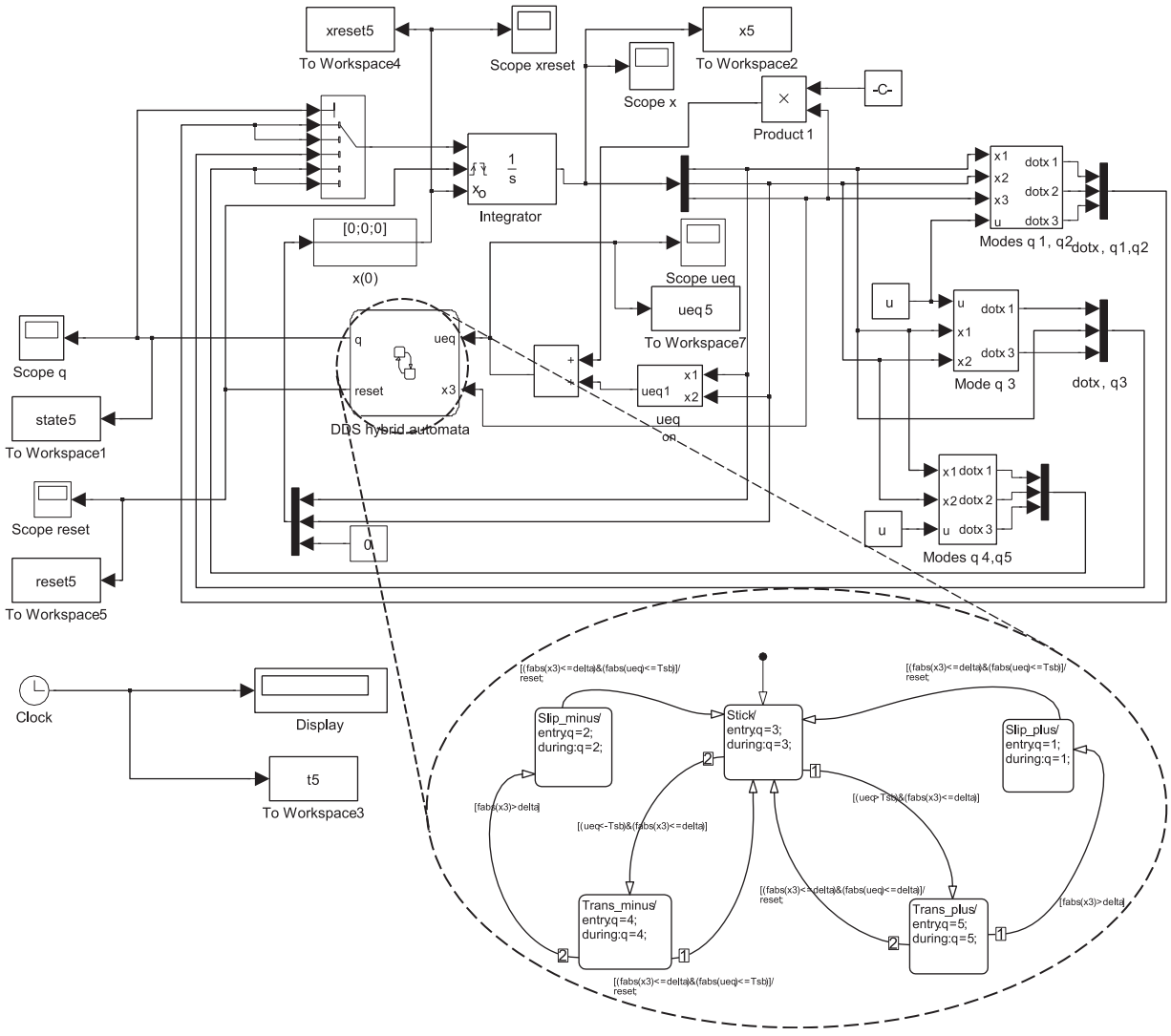

Figure 5. The Simulink/Stateflow model for H_{DDS_2}.

Stateflow simulations, the maximum step size is $h = 0.001$ s, the minimum step size is 10^{-5} s, and the tolerance is $\mathrm{tol} = 0.001$. With this h, the trajectories of the two hybrid automata coincide for the three dynamical behaviours. In Stateflow, when h is small enough, the impact of the tolerance in the solutions is minor. For Modelica simulations, a tolerance $\mathrm{tol} = 10^{-4}$ is used, except for the situation shown in Figure 6, in which $\mathrm{tol} = 10^{-8}$ must be used in order to reduce the difference between the trajectory obtained and the ones obtained with Stateflow. This tolerance is small enough to obtain accurate behavioural results, but higher tolerances do produce slightly different values for the maximum and minimum amplitudes after the last entry into the discontinuity surface.

In Stateflow, differences in the simulations appear when the maximum step size, h, is not small enough. In addition, the larger the number of entries into the discontinuity surface is, that is, the larger the number of transitions between discrete locations is, the larger the difference between the two hybrid trajectories given by H_{DDS_1} and H_{DDS_2} is, and the larger the difference between the hybrid trajectories and the trajectory of system (7)–(9) is. See Figures 9 and 10, where $h = 2$ s was used. This fact is especially visible in Figure 9(a), where a small tolerance of $\mathrm{tol} = 10^{-8}$ is used for the two hybrid automata. In this case, the 5-location hybrid automaton gives stick–slip oscillation, and although the 3-location automaton converges to the equilibrium, the convergence is delayed in time. This case displays such differing behaviour due to the combination of W_{ob} and u, since these parameters put the trajectory close to the bifurcation leading to stick–slip.

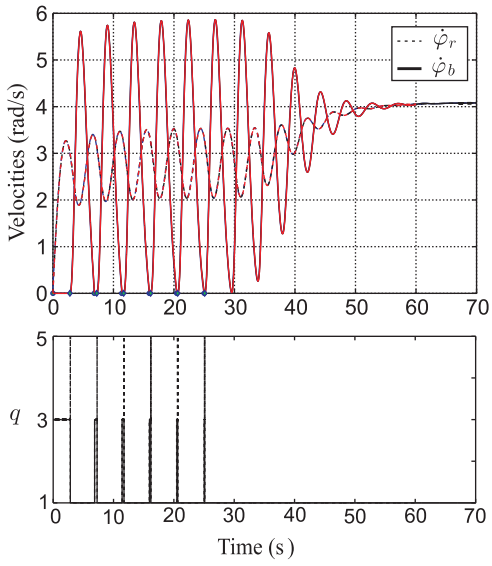

Figure 6. Trajectories obtained with Stateflow and Modelica. Convergence to the equilibrium entering several times the switching surface for the three systems: $u = 6\,\text{kN m}$, $W_{\text{ob}} = 51{,}408\,\text{N}$. For Stateflow, the maximum step size $h = 0.001\,\text{s}$, the minimum step size is $10^{-5}\,\text{s}$ and $\text{tol} = 0.001$. For Modelica, $\text{tol} = 10^{-8}$.

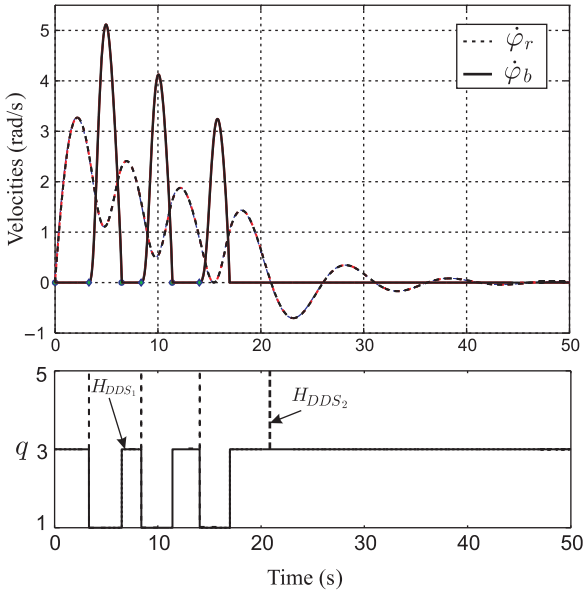

Figure 7. Trajectories obtained with Stateflow and Modelica. Permanent stuck bit for the three systems: $u = 6\,\text{kN m}$, $W_{\text{ob}} = 60\,\text{kN}$. For Stateflow, the maximum step size $h = 0.001\,\text{s}$, the minimum step size is $10^{-5}\,\text{s}$ and $\text{tol} = 0.001$. For Modelica, $\text{tol} = 10^{-4}$.

In the stuck situation of Figure 9(b), a tolerance $\text{tol} = 10^{-4}$ is used for the two hybrid automata. The difference between the discontinuous model simulation and the 5-location hybrid automaton simulation is greater than the difference between the discontinuous

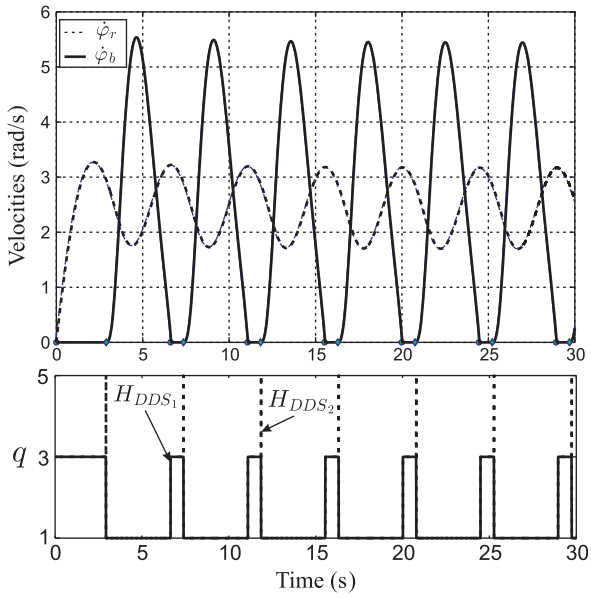

Figure 8. Trajectories obtained with Stateflow and Modelica. Stick-slip situation for the three systems: $u = 6\,\text{kN m}$, $W_{\text{ob}} = 53018\,\text{N}$. For Stateflow, the maximum step size $h = 0.001\,\text{s}$, the minimum step size is $10^{-5}\,\text{s}$ and $\text{tol} = 0.001$. For Modelica, $\text{tol} = 10^{-4}$.

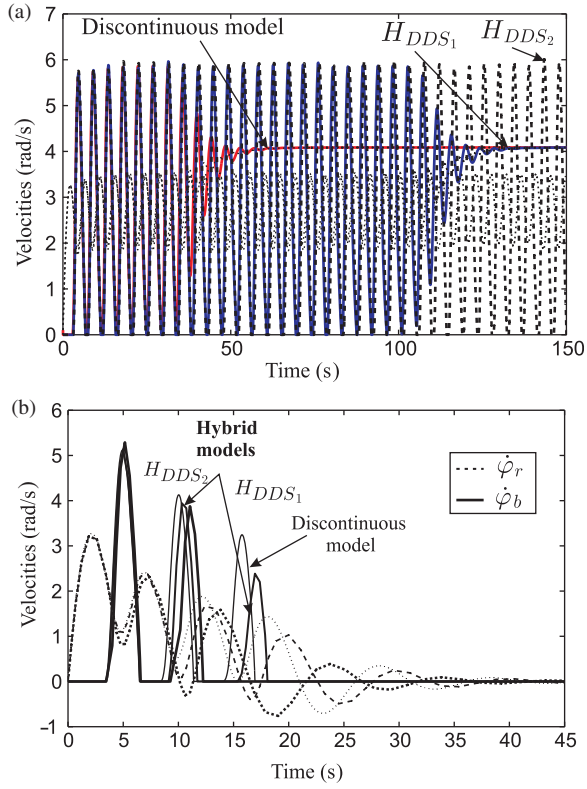

Figure 9. Comparison of Stateflow simulations. Use of a maximum step size $h = 2\,\text{s}$ in the numerical integration method: (a) Convergence to the equilibrium: $u = 6\,\text{kN m}$, $W_{\text{ob}} = 51{,}408\,\text{N}$, $\text{tol} = 10^{-8}$, (b) permanent stuck bit: $u = 6\,\text{kN m}$, $W_{\text{ob}} = 60\,\text{kN}$, $\text{tol} = 10^{-4}$.

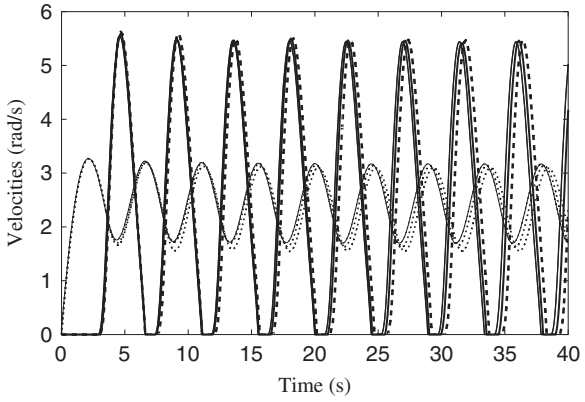

Figure 10. Comparison of Stateflow simulations with maximum step size $h = 2$ s, tol $= 10^{-8}$: Stick-slip situation with $u = 6$ kN m, $W_{ob} = 53{,}018$ N. — $\dot{\varphi}_b$ obtained with H_{DDS_1}, - - $\dot{\varphi}_b$ obtained with H_{DDS_2}, – $\dot{\varphi}_b$ of system (7)–(9), $\cdots$ $\dot{\varphi}_r$.

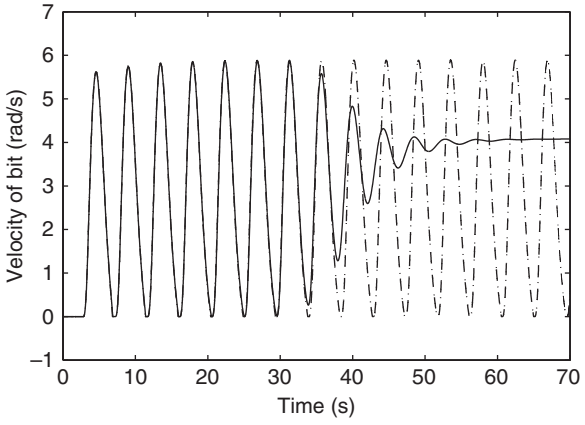

Figure 11. The effect of the tolerance in Modelica's LSODAR integration algorithm for $u = 6$ kN m, $W_{ob} = 51{,}408$ N. Plot shows the reference discontinuous system simulation (—) against the 3- and 5-location automata models (— · —), with tolerance 10^{-3}.

model simulation and the 3-location automaton simulation. Finally, in Figure 10, although a tolerance of tol $= 10^{-8}$ is used for the two hybrid automata, the lack of specific restriction on the step size means that there are visible differences between the different trajectories. This latter plot shows the importance of specifying sensible limits on the step size.

The plot in Figure 11 shows an example of the Modelica simulations for these hybrid automata when the tolerance is too large. The 3- and 5-location automata need a tolerance of 10^{-4} in order to achieve the correct long-term behaviour with $u = 6$ kN m and $W_{ob} = 51{,}408$ N. This plot shows the result if a tolerance of 10^{-3} is used instead. Note that, although the simulated long-term behaviour can change when near a bifurcation point, it is fairly resilient to changes under Modelica, provided enough accuracy is used. This is due to the Modelica integrator, LSODAR, which has a root finder, and so finds points where the location changes much better than MATLAB's *ode45*.

Using the Modelica language, rather than MATLAB's Stateflow, is much quicker for simulation of these hybrid automata systems. For example, on the same computer, for the highest accuracy needed in this article, the Stateflow simulation took roughly 100 s to compute a 100 s time span of result, whereas the comparable Modelica simulation took roughly 10 s to compute this 100 s time span of result. Added to this, most of the time taken by Modelica was used to compile the Modelica language code into C-code, which means that the time barely increases for any longer result interval. However, the majority of the time taken by Stateflow is used for the integration over the interval itself, so time for any longer interval increases proportionately. These observations show that simulations using the Modelica language are generally much more efficient than using Stateflow.

The overall picture is that Modelica can produce more accurate results in much shorter time, and so is usually better for simulations of hybrid-automaton systems, specially as these systems become larger. Given this conclusion, Modelica is used for the simulation of the automaton introduced in Section 5, which is a composition of three DDS hybrid automata.

6.3. *Simulation results with Modelica for the composition of several DDS hybrid automata*

The hybrid automaton introduced in Section 5 is now simulated, using the Modelica language through Dymola. The purpose of this hybrid automaton, as mentioned before, is to modify the input motor torque, u, so that the desired dynamics is seen in the drillstring system. However, the desired behaviour (convergence to equilibrium with positive velocity) is still only obtained under certain conditions which relate the weight on the bit (W_{ob}), the desired rotary velocity (Ω) and the parameter λ. The permanently stuck bit behaviour is eliminated by this hybrid automaton (Navarro-López and Licéaga-Castro 2009), so the only possibilities for long-term behaviour are convergence to equilibrium and stick–slip motion.

The first consideration to be made is of the two possible behaviour types that can be seen. The simulations of this hybrid automaton use parameters (13), along with $\lambda = 0.3$ and $\eta = 1$. Figure 12 depicts these behaviour patterns, with the left-hand plots showing the stick–slip behaviour, and the right-hand plots showing the convergence to equilibrium behaviour.

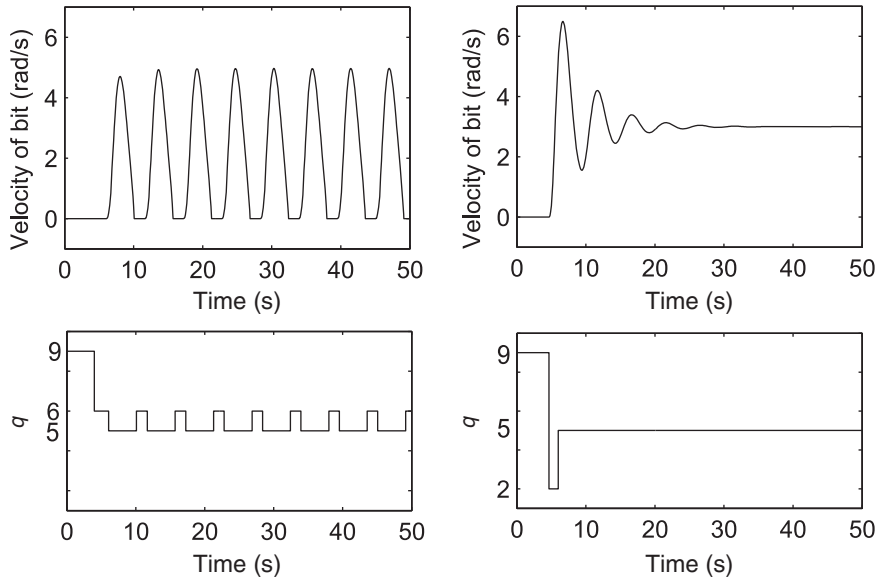

Figure 12. Bit velocity and location evolution for the two different long-term behaviour patterns. Both cases use $W_{ob} = 53018$ N. The stick–slip behaviour is depicted in the two plots on the left, and is obtained with $\Omega = 2$ rad/sec. The convergence to equilibrium behaviour is shown in the two plots on the right, and is obtained with $\Omega = 3$ rad/sec.

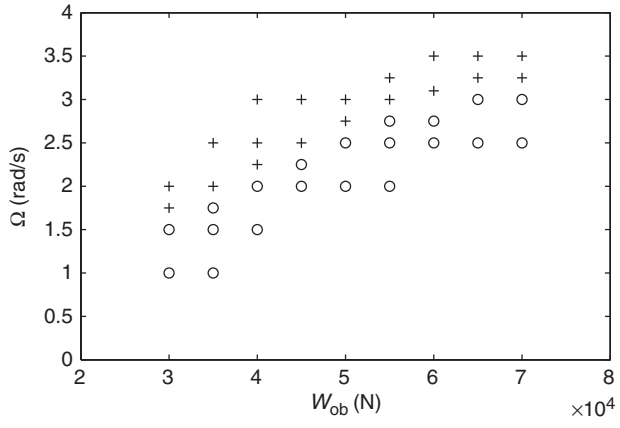

Figure 13. Long-term behaviour for the 9-location automaton obtained with Modelica simulations. ○: stick–slip behaviour; +: positive velocity.

Having noted the appearance of the time evolution, it is interesting to look at the overall trend of long-term behaviour for this hybrid automaton. To do this, the parameters W_{ob} (weight on the bit) and Ω (desired rotary velocity) have both been varied. The plot of the long-term behaviour pattern for each pair of parameters is given in Figure 13.

From this plot, it is clear that the region of convergence to positive velocity is any value of Ω above a curve dependent on W_{ob}. It is interesting to note that the equation for the boundary line can be calculated (Navarro-López and Licéaga-Castro 2009); in general, it is dependent on the value of λ, and the other typical parameters of the drillstring. However, the equation of the boundary does not change with η, since this only affects the speed of the controlled convergence to the long-term behaviour. These equations for the region of convergence to positive velocity mean that, provided the values of the parameters are known and provided the possible range of weight on the bit is known, the safe values for the desired velocity can be calculated. This calculation could also be made the other way round, so that possible values for desired velocity would specify a safe range for weight on the bit.

7. Conclusions

The *DDS hybrid automaton* and the *extended DDS hybrid automaton* are defined in this article in order to reinterpret a class of discontinuous systems with several switching surfaces, inputs and outputs within the hybrid-automaton framework. An example is used to illustrate the models proposed. It is a simplified torsional model of a drillstring including discontinuous friction and sliding-mode control. This article is a stepping stone of hybrid modelling of DDSs. Under the framework proposed, the control design, as well as the complex behaviours associated with discontinuous systems can be abstracted from a computational viewpoint.

Acknowledgements

The first author is grateful for the support of the Research Councils United Kingdom (RCUK) through the Fellowship EP/E500048/1. The authors also gratefully acknowledge the reviewers' valuable suggestions, which have improved the final paper version.

Notes on contributors

Eva M. Navarro-López has been a Lecturer/RCUK Academic Fellow at the School of Computer Science at the University of Manchester since September 2008. Having a MEng in Computer Science and Physics Systems Engineering, with a specialisation in Space Dynamics and Celestial Mechanics in the *Universidad de Alicante*, she completed a PhD in Control and Industrial Electronics at the *Universitat Politècnica de Catalunya* in Barcelona in 2002. From 1996 to 2000, she worked for the *Consejo Superior de Investigaciones Científicas* (CSIC) of Spain at the *Instituto de Automática Industrial* in Madrid and in the *Instituto de Robótica e Informática Industrial* in Barcelona. After her PhD, she worked at the *Instituto Mexicano del Petróleo* in México D.F., forming and leading a group on modelling, analysis and control of discontinuous dynamical systems and mechanical vibrations in oilwell drillstrings. In recognition of her contributions to research, since 2003, she has been awarded the membership of the Mexican National Research System (the national body for eminent scientists). From 2006 to 2008, she held a Ramón y Cajal Fellowship for outstanding researchers from the Spanish Government. Her current research topics are focused on modelling, analysis and control of hybrid dynamical systems and complex networks.

Rebekah Carter received her MMath in Mathematics from the University of Oxford in 2008, with first class honours. Since then she has completed an MSc with Distinction in Mathematics and Computational Science at the University of Manchester, and was awarded the NAG prize as the top student in the class. She is now continuing with her studies at Manchester, working towards her PhD degree in the area of hybrid dynamical systems. Her research interests include modelling, verification and control of hybrid systems, and the use of theoretical computer science in new areas.

Note

1. Dymola: Dynamic Modelling Laboratory. www.dynasim.se/index.htm.

References

Acary, V., and Brogliato, B. (2008), *Numerical Methods for Nonsmooth Dynamical Systems: Applications in Mechanics and Electronics* (Vol. 35), Lecture Notes in Applied and Computational Mechanics, Heidelberg: Springer-Verlag.

Agrawal, A., Simon, G., and Karsai, G. (2004), 'Semantic Translation of Simulink/Stateflow Models to Hybrid Automata using Graph Transformations', *Electronic Notes in Theoretical Computer Science*, 109, 43–56.

Alur, R., Courcoubetis, C., Henzinger, T.A., and Ho, P.H. (1993), *Hybrid Automata: an Algorithmic Approach to the Specification and Verification of Hybrid Systems* (Vol. 736), Lecture Notes in Computer Science, Berlin: Springer, pp. 209–229.

Alur, R., Kanade, A., Ramesh, S., and Shashidhar, K.C. (2008), 'Symbolic Analysis for Improving Simulation Coverage of Simulink/Stateflow Models', in *8th ACM and IEEE International Conference on Embedded software, EMSOFT 2008*, Atlanta, USA, 19–24 October 2008, pp. 19–24.

Antsaklis, P.J., Stiver, J.A., and Lemmon, M.D. (1993), *Hybrid System Modelling and Autonomous Control Systems* (Vol. 736), Lecture Notes in Computer Science, New York: Springer, pp. 366–392.

Awrejcewicz, J., and Lamarque, C. (2003), *Bifurcations and Chaos in Non-smooth Mechanical Systems* (Vol. 45), Singapore: World Scientific Series on Nonlinear Science, Series A.

Branicky, M.S., Borkar, V.S., and Mitter, S.K. (1998), 'A Unified Framework for Hybrid Control: Model and Optimal Control Theory', *IEEE Transactions on Automatic Control*, 43, 31–45.

Brockett, R.W. (1988), 'On the Computer Control of Movement', in *1988 IEEE Conference on Robotics and Automation*, New York, USA, April 1988, pp. 534–540.

Brogliato, B. (1999), *Nonsmooth Mechanics*, London: Springer-Verlag.

Buss, M., Glocker, M., Hardt, M., von Stryk, O., Bulirsch, R., and Schmidt, G. (2002), 'Nonlinear Hybrid Dynamical Systems: Modelling, Optimal Control, and Applications', in *Modelling, Analysis and Design of Hybrid Systems*, Vol. 279 of *NCIS*, eds. S. Engell, G. Frehse, and E. Schnieder, Berlin: Springer-Verlag, pp. 311–335.

Carter, R. (2009), 'Computational Model of a Rotary System with Discontinuous Elements', Msc Dissertation, The University of Manchester, Manchester, UK.

di Bernardo, M., Budd, C., Champneys, A., and Kowalczyk, P. (2008), *Piecewise-smooth Dynamical Systems. Theory and Applications*, London: Springer-Verlag.

Egerstedt, M., and Brockett, R.W. (2003), 'Feedback can Reduce the Specification Complexity of Motor Programs', *IEEE Transactions on Automatic Control*, 48, 213–223.

Elmqvist, H., Cellier, F.E., and Otter, M. (1993), 'Object-oriented Modelling of Hybrid Systems', *European Simulation Symposium*, The Netherlands: Delft, pp. 31–41.

Filippov, A.F. (1988), *Differential Equations with Discontinuous Right-hand Sides*, Dordrecht: Kluwer Academic Publishers.

Forbus, K.D. (1984), 'Qualitative Process Theory', *Artificial Intelligence*, 24, 85–168.

Henzinger, T.A. (1996), 'The Theory of Hybrid Automata', in *11th IEEE Symposium of Logic in Computer Science*, pp. 278–292.

Johansson, K.H., Egerstedt, M., Lygeros, J., and Sastry, S. (1999), 'On the Regularization of Zeno Hybrid Automata', *Systems and Control Letters*, 38, 141–150.

Karnopp, D. (1985), 'Computer Simulation of Stick-Slip Friction in Mechanical Dynamic Systems', *ASME Journal of Dynamic Systems, Measurement, and Control*, 107, 100–103.

Kuipers, B. (1986), 'Qualitative Simulation', *Artificial Intelligence*, 29, 289–338.

Kunze, M. (2004), *Non-smooth Dynamical Systems* (Vol. 1744), Lecture Notes in Mathematics, Berlin/ Heidelberg: Springer-Verlag.

Kuznetsov, Y.A., Rinaldi, S., and Gragnani, A. (2003), 'One-parameter Bifurcations in Planar Filippov Systems', *International Journal of Bifurcations and Chaos*, 13, 2157–2188.

Leine, R.I., van Campen, D.H., de Kraker, A., and van den Steen, L. (1998), 'Stick-Slip Vibrations Induced by Alternate Friction Models', *Nonlinear Dynamics*, 16, 41–54.

Lotstedt, P. (1991), 'Coulomb Friction in Two-dimensional Rigid Body Systems', *Zeitschrift für Angewandte Mathematik und Mechanik*, 64, 605–615.

Lygeros, J., Johansson, K.H., Simić, S.N., Zhang, J., and Sastry, S. (2003), 'Dynamical Properties of Hybrid Automata', *IEEE Transactions on Automatic Control*, 48, 2–17.

Lygeros, J., Tomlin, C., and Sastry, S. (1999), 'Controllers for Reachability Specifications for Hybrid Systems', *Automatica*, 35, 349–370.

Mattsson, S.E. (1996), 'On Object-Oriented Modelling of Relays and Sliding Mode Behaviour', in *13th Triennial IFAC World Congress*, San Francisco, USA, pp. 259–264.

Modelica (2009), 'Modelica® – A Unified Object-Oriented Language for Physical Systems Modeling'. www.modelica.org/documents/ModelicaSpec31.pdf

Mosterman, P.J., and Biswas, G. (2000), 'A Comprehensive Methodology for Building Hybrid Models of Physical Systems', *Artificial Intelligence*, 121, 171–209.

Mosterman, P.J., Zhao, F., and Biswas, G. (1999), 'Sliding Mode Model Semantics and Simulation for Hybrid Systems', in in *Hybrid Systems V*, Vol. 1567 of *LNCS*, eds. Antsaklis, et al., Berlin: Springer-Verlag, pp. 218–237.

Navarro-López, E.M. (2009a), 'An Alternative Characterization of Bit-sticking Phenomena in a Multi-Degree-of-Freedom Controlled Drillstring', *Nonlinear Analysis: Real World Applications*, 10, 3162–3174.

Navarro-López, E.M. (2009b), 'What Makes the Control of Discontinuous Dynamical Systems so Complex?', in *Mathematical Problems in Engineering, Aerospace and Sciences*, An International Series of Scientific Monographs and Text Books, ed. S. Sivasundaram, UK: Cambridge Scientific Publishers, pp. 63–83 (to appear).

Navarro-López, E.M. (2009c), 'Hybrid Modelling of a Discontinuous Dynamical System Including Switching Control', in *2nd IFAC Conference on Analysis and Control of Chaotic Systems Proceedings*, IFAC, London, UK, 22–24 June 2009c, pp. 1–6.

Navarro-López, E.M. (2009d), 'Hybrid-Automaton Models for Simulating Systems with Sliding Motion: Still a Challenge', in *3rd IFAC Conference on Analysis and Design of Hybrid Systems Proceedings*, IFAC, Zaragoza, Spain, 16–18 September 2009d, pp. 322–327.

Navarro-López, E.M., and Cortés, D. (2007), 'Avoiding Harmful Oscillations in a Drillstring Through Dynamical Analysis', *Journal of Sound and Vibration*, 307, 152–171.

Navarro-López, E.M., and Licéaga-Castro, E. (2009), 'Non-desired Transitions and Sliding-mode Control of a Multi-DOF Mechanical System with Stick-Slip Oscillations', *Chaos, Solitons and Fractals*, 41, 2035–2044.

Park, T., and Barton, P.I. (1996), 'State Event Location in Differential-algebraic Models', *ACM Transactions on Modeling and Computer Simulation*, 6, 137–165.

Sedghi, B. (2003), 'Control Design of Hybrid Systems via Dehybridization', Ph.D. Dissertation, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

Sedghi, B., Srinivasan, B., and Longchamp, R. (2002), 'Control of Hybrid Systems via Dehybridization', in *American Control Conference*, Anchorage, USA, 8–10 May 2002, pp. 692–697.

Stursberg, O. (2006), 'Supervisory Control of Hybrid Systems Based on Model Abstraction and Guided Search', *Nonlinear Analysis*, 65, 1168–1187.

Tavernini, L. (1987), 'Differential Automata and their Discrete Simulators', *Nonlinear Analysis, Theory, Methods and Applications*, 11, 665–683.

The MathWorks, I. (2008), 'Stateflow and Stateflow Coder User's Guide. For Complex Logic and State Diagram Modeling. www.mathworks.com/access/helpdesk_r13/help/pdf_doc/stateflow/sf ug.pdf

Utkin, V.I. (1992), *Sliding Modes in Control Optimization*, Berlin: Springer-Verlag.

van der Schaft, A.J., and Schumacher, J.M. (2000), *An Introduction to Hybrid Dynamical Systems*, London: Springer-Verlag.

Witsenhausen, H.S. (1966), 'A Class of Hybrid-state Continuous-time Dynamic System', *IEEE Transactions on Automatic Control*, AC-11, 161–167.

Woods, E.A. (1991), 'The Hybrid Phenomena Theory', in *12th International Joint conference of Artificial Intelligence*, Los Altos, CA: Morgan Kaufman (1991), pp. 71–76.

Ye, H., Michel, A.N., and Hou, L. (1998), 'Stability Theory for Hybrid Dynamical Systems', *IEEE Transactions on Automatic Control*, 43, 461–474.

Zhang, F., Yeddanapudi, M., and Mosterman, P.J. (2008), 'Zero-crossing Location and Detection Algorithms for Hybrid System Simulation', in *17th IFAC Triennial World Congress*, Seoul, Korea, 6–11 July 2008, pp. 7967–7972.

Zhao, F., and Utkin, V.I. (1996), 'Adaptive Simulation and Control Variable-structure Control Systems in Sliding Regimes', *Automatica*, 32, 1037–1042.