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

Eva M. Navarro-López; Rebekah Carter

* School of Computer Science, The University of Manchester, Manchester, M13 9PL, UK

First published on: 19 August 2010
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
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, Yeddnapudi, 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 Ličag-a-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

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

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.
- \( X \subseteq \mathbb{R}^n \) is the continuous state space.
- \( U \subseteq \mathbb{R}^m \) and \( \mathcal{Y} \subseteq \mathbb{R}^n \) 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.
- \( \text{Dom}: Q \rightarrow 2^{X \times U} \) is the location domain. It is a mapping from the locations \( Q \) to the set of all subsets of \( X \times U \), that is, \( \text{Dom} \) assigns a set of continuous states and inputs to each discrete state \( q_i \in Q \), thus, \( \text{Dom}(q_i) \subseteq X \times U \).
- \( \mathcal{F} = \{f_q(x, u): q_i \in Q\} \) is the collection of vector fields describing the continuous dynamics such that \( f_q(x, u) : X \times U \rightarrow X \). Each \( f_q(x, u) \) is assumed to be Lipschitz continuous on the location domain for \( q_i \) in order to ensure that the solution within \( \text{Dom}(q_i) \) exists and is unique.
- \( \text{Init} \subseteq Q \times X \) is a set of initial states.
- \( G: E \rightarrow 2^X \) 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 \( (G(e) \subset X) \). Each guard set plays the role of an enabling condition in order to change the location.
- \( R: E \times X \times U \rightarrow 2^Y \) is a reset map for the continuous states for each edge. It is assumed that \( \forall e \in E, G(e) \neq \emptyset \) and \( \forall x \in G(e), R(e, x, u) \neq \emptyset \). This is assumed so that the continuous dynamics cannot be destroyed, only changed.
- \( h: \mathcal{Q} \times X \times U \rightarrow \mathcal{Y} \) is the continuous output mapping, there is one for each location.
- \( r: \mathcal{Q} \times X \times \Sigma \times U \rightarrow \mathcal{O} \) is the discrete output map, there is one for each location.

As long as the system is within location \( q_i \), the continuous state \( x \) must satisfy \( x \in \text{Dom}(q_i) \). The transition from a discrete state \( q_i \) to another \( q_j \) is enabled when the continuous state \( x \) reaches the guard \( G(q_i, q_j) \subset X \) of some edge \( (q_i, q_j) \in E \). Then, the discrete state changes to \( q_j \) and at the same time, \( x \) is reset to the value specified by \( R(q_i, q_j, x, u) \subset X \). \( H \) 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 \( z = (q, x) \in \mathcal{Q} \times X \). Consider a time domain \( \Gamma = \mathbb{R}^+ \) and \( T = \{t_0, t_1, \ldots, t_k\} \), with \( t_k \in \Gamma \) for every \( k \in \mathbb{N} \), 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 \( x^k := x(t_k) \), the locations as \( q^k := q(t_k) \), the continuous inputs and outputs as \( u^k := u(t_k) \) and \( y^k := y(t_k) \), respectively. We define a continuous input and a continuous output sequence as \( \phi_u := (u^0, u^1, \ldots, u^k) \) and \( \phi_y := (y^0, y^1, \ldots, y^k) \), respectively. Moreover, a discrete input and a discrete output sequence is defined by \( \phi_o := (o^0, o^1, \ldots, o^k) \) and \( \phi_o := (o^0, o^1, \ldots, o^k) \), respectively, with \( \sigma_k \in \Sigma, o_k \in O \forall k \).

**Definition 2.2** (Stursberg 2006): For an input sequence \( \phi_u \), a feasible execution (or run) of \( H \) is a sequence of hybrid states \( \phi_b = (z(t_0), z(t_1), \ldots, z(t_i)) \), with \( z_k := z(t_k) = (q^k, x^k) \) such that:

- **Initial condition:** \( z_0 = (q_0^0, x_0^0) \) with \( q_0^0 = q(t_0) \in Q \) and \( x_0^0 \in \text{Dom}(q_0^0), x_0^0 \notin G(q_0^0) \) for any \( (q_0^0, \cdot) \in E \).
- **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:** \( x:[0, \tau] \rightarrow X \), \( x(0) = x^k \), \( \dot{x}(t) = f_q(x(t), u^k(t)) \) with unique solution for \( t \in [0, \tau] \), and
with sliding. The system dynamics on $S^f$ account that $x$ is continuous and smooth, and $S^\text{dis}$ with non-vanishing gradient. The discontinuity surface is divided into two sets $S^f_0$ and $S^g_0$ such that $\nabla s(x), f^+(x) > 0$ and repulsive for $s(x)$ such that $\nabla s(x), f^-(x) - \nabla s(x), f^+(x) < 0$ (Kuznetsov, Rinaldi, and Gragnani 2003).

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

- $Q = \{q_1, q_2, q_3\} = \{\text{slip}^+, \text{slip}^-, \text{stick}\}, X \subseteq \mathbb{R}^n$.
- $E = \{(q_1, q_2), (q_1, q_3), (q_2, q_1), (q_2, q_3), (q_3, q_1)\}$.
- $\Sigma = \{a, b, c\}$. One edge label is assigned to each guard set.
- $\Sigma = O$. The discrete output events coincide with the discrete input events.

- $Dom(q_1) = S^+ \cup \{x \in S^0 : u_{eq}(x) > 1\}$.
- $Dom(q_2) = S^- \cup \{x \in S^0 : u_{eq}(x) < -1\}$.
- $Dom(q_3) = S^0_\text{S}$.

- $f_{q_1}(x) = f^+(x)$, $f_{q_2}(x) = f^-(x)$, $f_{q_3}(x) = f_0(x)$.

- $Init = Q \times X \cup u$, with $u := \{x \in S^0 : \nabla s(x), f^+(x) < 0\}$. 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.

- $G(q_1, q_2) = G(q_2, q_3) = S^0_\text{S}$, $G(q_1, q_3) = \{x \in S^0 : u_{eq}(x) < -1\}$, $G(q_2, q_1) = \{x \in S^0 : u_{eq}(x) > 1\}$.

- $R(q_i, q_j, x) = \{x_1, x_2, x_3\}$, $i, j \neq 3$, $R(q_i, q_j, x) = \{x_1, x_2, x_3\}$, $x_1, x_2, x_3 \in S^0$.

- $y = h(q_1, x) = h(q_2, x)$ 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: $\{x \in S^0 : u_{eq}(x) > 1\}$, for $Dom(q_1)$, and $\{x \in S^0 : u_{eq}(x) < -1\}$, 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 $H_{\text{DDS}}$ was obtained in Sedghi, Srinivasan, and Longchamp (2002) and Sedghi (2003). The difference with $H_{\text{DDS}}$ 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 $H_{\text{DDS}}$. It will be called the extended DDS hybrid automaton, $H_{\text{DDS}}$.

This hybrid automaton considers intermediate transitions when leaving the discontinuity surface (stick-to-slip transition). $H_{\text{DDS}}$ includes two intermediate states before motion (one for positive velocities, another for negative velocities), such as: $q_3 = \{\text{trans}^+\}$ and $q_4 = \{\text{trans}^-\}$. 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 $H_{\text{DDS}}$ is also inspired in the state-transition diagram of a friction model presented in Elmqvist et al. (1993). The vector fields associated with $q_4$ ($f_4^s(x)$) and $q_5$ ($f_5^s(x)$) 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 $\text{slip}^+$ and $\text{slip}^-$ or vice-versa. For example, if a transition $\text{slip}^+ \rightarrow \text{slip}^-$ should be carried out, if the system is in state $\text{slip}^+$ and $s$ becomes zero, the system switches to stick before checking the conditions for switching to $\text{slip}^-$. Moreover, if the system is in location $\text{trans}^+$ and the velocity is reversed before starting to move, the system switches to state stick before going to $\text{slip}^-$. 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 ($H_{\text{DDS}}$) describing the dynamics of system (1) is a particular case of $H$ with,

- $Q = \{q_1, q_2, q_3, q_4, q_5\} = \{\text{slip}^+, \text{slip}^-, \text{stick}, \text{trans}^+, \text{trans}^-\}$, $X \subseteq \mathbb{R}^n$.
- $E = \{(q_1, q_3), (q_2, q_3), (q_3, q_5), (q_4, q_4), (q_4, q_3), (q_5, q_3), (q_4, q_2), (q_5, q_1)\}$.
- $\Sigma = O = \{a, b, c, d\}$.
- $\text{Dom}(q_1) = \{x \in \mathbb{R}^n : s(x) > \delta\}$, $\text{Dom}(q_2) = \{x \in \mathbb{R}^n : s(x) < -\delta\}$, $\text{Dom}(q_3) = G_0^s = \{x \in \mathbb{R}^n : |s(x)| \leq \delta, |u_{eq}(x)| \leq 1\}$, $\text{Dom}(q_4) = G_0^p = \{x \in \mathbb{R}^n : |s(x)| \leq \delta, u_{eq}(x) < -1\}$, $\text{Dom}(q_5) = G_0^p = \{x \in \mathbb{R}^n : |s(x)| \leq \delta, u_{eq}(x) > 1\}$, with $0 < \delta < 1$.
- $f_4^s(x) = f^s(x)$, $f_5^s(x) = f^s(x)$, $f_4^s(x) = f^s(x)$. $f_4^s(x) = f^s(x)$.
- $\text{Init} = \mathcal{Q} \times \mathcal{X} \setminus U_s$, with $U_s$ as previously defined.
- $G(q_1, q_3) = G(q_2, q_4) = G(q_2, q_5) = G(q_4, q_1) = G(q_4, q_3) = G(q_4, q_2) = \{x \in \mathbb{R}^n : |s(x)| > \delta\}$.
- $R(q_1, q_1, x) = \{x \} \forall i, j$ with $j \neq 3$, $R(q_i, q_j, x) = \{x : s(x) = 0\} \forall i \in \{1, 2, 4, 5\}$.
- $y = h(q_1, x) = h(q_2, x) = h(q_3, x) = h(q_4, x) = h(q_5, x)$.

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 automaton 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, $H_{\text{DDS}}$ and $H_{\text{DDS}}$ 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 pendulum 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 \( J_B \) and \( J_B \).

The system state vector is \( x = (\dot{x}_1, \dot{x}_2, x_3)^T \), with \( \dot{x}_1, \dot{x}_2, \dot{x}_3 \) 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 \( (c_x x_3) \). \( T_m = u \) 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:

\[
  f_b(x_3) = W_{ob} R_B \left( \mu_{c_b} + (\mu_{s_b} - \mu_{c_b}) \exp \left( -\frac{2}{J_F} x_3 \right) \right),
\]

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

\[
\text{sign}(x_3) = x_3/|x_3| \quad \text{if} \quad x_3 \neq 0,
\]

\[
\text{sign}(x_3) \in [-1, 1] \quad \text{if} \quad x_3 = 0.
\]

The drillstring dynamics is given by

\[
\begin{align*}
  \dot{x}_1 &= \frac{1}{J_t} (-c_t + c_b) x_1 - k_t x_2 + c_b x_3 + u, \\
  \dot{x}_2 &= x_1 - x_3, \\
  \dot{x}_3 &= \frac{1}{J_b} \left( c_b x_1 + k_t x_2 - (c_b + c_t) x_3 - T_{f_b}(x_3) \right),
\end{align*}
\]

or in a compact form, \( \dot{s}(t) = Ax(t) + Bu(t) + T_f(x(t)) \), where \( A, B \) are constant matrices and \( T_f \) represents the torque on the bit. The inputs of the system are \( u \) and \( W_{ob} \). The outputs are the angular velocities \( \dot{x}_1 \) and \( \dot{x}_3 \).

For (7) with (5), \( s(x) = b^b(x) = x_3 \), the switching surface is \( S_0^b = \{ x \in \mathbb{R}^3 : x_3 = 0 \} \) and the sliding set is \( S_b^b = \{ x \in S_0^b : |c_t x_1 + k_t x_2| \leq T_{s_b} \} \). Notice that \( T_{s_b} \)

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

plays the role of the equivalent control \( T_{f_b} \), and \( T_{f_b} \) is the solution for \( T_b \) of equation \( \dot{s} = 0 \), that is, \( u_{eq} = T_{f_b} = c_b x_1 + k_t x_2 - (c_t + c_b) x_3 \). Moreover, \( -T_{s_b} \leq T_{f_b} \leq T_{s_b} \). For more details, the reader is invited to read Navarro-López and Cortés (2007), Navarro-López and Licéaga-Castro (2009) and Navarro-López (2009a).

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

\[
\begin{align*}
  f_{q_1}(x, W_{ob}, u) &= Ax + Bu + T_f(x)|_{T_b=T^+_b} = f^+(x), \\
  f_{q_2}(x, W_{ob}, u) &= Ax + Bu + T_f(x)|_{T_b=T^-_b} = f^-(x), \\
  f_{q_3}(x, u) &= \left( \begin{array}{c} \frac{1}{J_t} (-c_t + c_b) x_1 - k_t x_2 + u \\ x_1 \end{array} \right) = f_i(x),
\end{align*}
\]

\[ G(q_1, q_2) = G(q_2, q_3) = \{ x \in S_0^b : |u_{eq}(x)| \leq T_{s_b} \} = G^0 = S_0^b, \]

\[ G(q_1, q_2) = G(q_3, q_2) = \{ x \in S_0^b : u_{eq}(x) < -T_{s_b} \} = G^-, \]

\[ G(q_2, q_1) = G(q_1, q_1) = \{ x \in S_0^b : u_{eq}(x) > T_{s_b} \} = G^+, \]

\[ \text{Dom}(q_1) = \{ x \in \mathbb{R}^3 : x_3 > 0 \} \cup G^+, \]

\[ \text{Dom}(q_2) = \{ x \in \mathbb{R}^3 : x_3 < 0 \} \cup G^-, \]

\[ \text{Dom}(q_3) = S_b^b, \]

where \( T^+_b \) and \( T^-_b \) 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
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,

\[
T_b(x) = \begin{cases} 
T_{b_1}(x) & \text{if } |x_3| \leq \delta, |T_{b_1}| \leq T_{b_0} \text{ (stick)}, \\
T_{b_1} \text{sign}(T_{b_1}(x)) & \text{if } |x_3| \leq \delta, |T_{b_1}| > T_{b_0} \text{ (stick-to-slip transition)}
\end{cases}
\]

if \( |x_3| > \delta \) (sliding),

where \( \delta > 0 \), and \( T_{b_0} \) is the reaction torque, which coincides with the equivalent control.

The vector fields associated with \( q_d(f_1^{-}(x)) \) and \( q_d(f_1^{+}(x)) \) are obtained by considering in (7), \( x_3 = 0 \) and

\[
T_{b_1} = \max(u_{eq})\text{sign}(u_{eq}) = T_{b_1}\text{sign}(T_{b_1}).
\]

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, \( s' = 0 \), along which the system exhibits desired dynamics. For this purpose, a discontinuous control \( u \) is proposed so that the system trajectory reaches the surface \( s' = 0 \) and enters a sliding motion. Thus, the following controller is considered (Navarro-López and Liccaga-Castro 2009; Navarro-López 2009b):

\[
s'(x, t) = (x_1 - \Omega) + \lambda \int_0^t [x_1(\tau) - \Omega]d\tau + \lambda \int_0^t [x_1(\tau) - x_1(\tau)]d\tau, \quad \lambda > 0, \]

\[u = (x_1 + x_3) + k_1x_2 + c_1x_1 - J_s[x_1(\Omega) - (x_1 - x_3)] + \eta \text{sign}(s'), \quad \eta > 0, \quad (10)\]

where \( \Omega > 0 \) is the desired top-rotary velocity. It is ensured that \( s'(x, t) \) becomes zero in a finite time interval \( t_0 = \frac{|x_1(0) - \Omega|}{\eta} \). Two new states \( x_4, x_5 \) are defined, such that, \( x_4 = x_1 - \Omega \) and \( x_5 = x_1 - x_3 \). The following switching surface is defined: \( S_0' := \{ x \in \mathbb{R}^3 : s'(x, t) = 0 \} \). This surface has been designed in such a way to be attractive for all \( x \) and to be a sliding set for all \( x \in S_0' \). Control \( u \) is of switched type, with the form:

\[
u = \begin{cases}
u^+ & \text{if } s^+ > 0 \\
u^- & \text{if } s^- < 0
\end{cases}
\]

obtaining \( u^+ \) and \( u^- \) by changing the sign of \( s^+ \) in (10).

From (10), the equivalent control associated with \( S_0' \) is \( u^e_0 < u^e_0 < u^e_0 \), with

\[
u^e_0(x) = c_1(x_1 - x_3) + k_1x_2 + c_1x_1 - J_s[x_1(\Omega) + \lambda(x_1 - x_3)].
\]

Consequently, the dynamics on \( S_0' \) has the following form:

\[
x = f_1^{+}(x, u)|_w = u_0 = \begin{cases}
-(\lambda x_1 - \Omega) - \lambda(x_1 - x_3) & x_1 - x_3 \\
1 \left[c_1x_1 + k_1x_2 - (c_1 + c_0)x_3 - T_{b_1}(x_3)\right] & x_1 - x_3
\end{cases}
\]

In addition, control \( u \) has modified the dynamics on \( S_0' \), and

\[
x = f_1^{+}(x, u) = \begin{cases}
-2\lambda x_1 + \lambda \Omega - \eta \text{sign}(s') & x_1 \\
0 & x_1 - \Omega
\end{cases}
\]

\[
x = f_1^{+}(x, u) = \begin{cases}
x_1 & x_1 - \Omega \\
x_1 & x_1 - x_3
\end{cases}
\]
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 $H_{\text{DDS}}$, is used and the result is a 9-location hybrid automaton described as follows. Similar results would be obtained by using $H_{\text{DDS}}$.

The switching surfaces $S_0^+$ and $S_0^-$ divide the state space in four regions where the system is smooth. For each of these regions, a location is defined:

$q_1 = \{\text{slip}^+, \text{slip}^+_r\}, \quad q_2 = \{\text{slip}^+, \text{slip}^-_r\},
q_3 = \{\text{slip}^-, \text{slip}^+_r\}, \quad q_4 = \{\text{slip}^-, \text{slip}^-_r\}.$

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

\[
\begin{align*}
\text{Dom}(q_1) &= \{x \in \mathbb{R}^5 : \text{s}^b(x) > 0, s'(x, t) > 0\} \cup (G^+ \cap S_0^+), \\
&= S_0^+ \cap (S_0^b \cup G^+), \\
\text{Dom}(q_2) &= \{x \in \mathbb{R}^5 : \text{s}^b(x) > 0, s'(x, t) < 0\} \cup (G^+ \cap S_0^-), \\
&= S_0^- \cap (S_0^b \cup G^+), \\
\text{Dom}(q_3) &= \{x \in \mathbb{R}^5 : \text{s}^b(x) < 0, s'(x, t) > 0\} \cup (G^- \cap S_0^+), \\
&= S_0^+ \cap (S_0^b \cup G^-), \\
\text{Dom}(q_4) &= \{x \in \mathbb{R}^5 : \text{s}^b(x) < 0, s'(x, t) < 0\} \cup (G^- \cap S_0^-), \\
&= S_0^- \cap (S_0^b \cup G^-),
\end{align*}
\]

\[
\begin{align*}
f_{q_1} &= \begin{pmatrix}
\varphi_1(x) - \eta \\
x_1 - x_3 \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_2} &= \begin{pmatrix}
\varphi_2(x) - \frac{T_{\text{slip}}^b(x)}{2} \\
x_1 - \Omega \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_3} &= \begin{pmatrix}
\varphi_1(x) - \eta \\
x_1 - x_3 \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_4} &= \begin{pmatrix}
\varphi_2(x) - \frac{T_{\text{slip}}^b(x)}{2} \\
x_1 - \Omega \\
x_1 - x_3
\end{pmatrix},
\end{align*}
\]

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

\[
S_0^+ = \{x \in \mathbb{R}^5 : s'(x, t) > 0\}, \quad S_0^- = \{x \in \mathbb{R}^5 : s'(x, t) < 0\}, \\
S_0^b = \{x \in \mathbb{R}^5 : x_3 > 0\}, \quad S_0^0 = \{x \in \mathbb{R}^5 : x_3 < 0\}.
\]

In addition, $\varphi_1(x) = -\lambda(x_1 - \Omega) - \lambda(x_1 - x_3)$ and $\varphi_2(x) = \frac{1}{2} [c_1(x_1 + k_1x_2 - (c_1 + c_2)x_3)]$. 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 $\text{Dom}(q_i)$.

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

\[
q_5 = \{\text{slip}^+, \text{stick}_r\}, \quad q_6 = \{\text{stick}_b, \text{stick}_r\}, \\
q_7 = \{\text{slip}^+, \text{stick}_r\}, \quad q_8 = \{\text{stick}_b, \text{slip}^+_r\}, \\
q_9 = \{\text{stick}_b, \text{slip}^-_r\}.
\]

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

\[
\begin{align*}
\text{Dom}(q_5) &= \{x \in \mathbb{R}^5 : \text{s}^b(x) > 0, s'(x, t) = 0\} \cup (G^+ \cap S_0^+), \\
&= S_0^+ \cap (S_0^b \cup G^+), \\
\text{Dom}(q_6) &= G^0 \cap S_0^0, \\
\text{Dom}(q_7) &= \{x \in \mathbb{R}^5 : \text{s}^b(x) < 0, s'(x, t) = 0\} \cup (G^- \cap S_0^-), \\
&= S_0^- \cap (S_0^b \cup G^-), \\
\text{Dom}(q_8) &= G^0 \cap S_0^+, \\
\text{Dom}(q_9) &= G^0 \cap S_0^-;
\end{align*}
\]

\[
\begin{align*}
f_{q_5} &= \begin{pmatrix}
\varphi_1(x) \\
-x_1 + x_3 \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_6} &= \begin{pmatrix}
\varphi_2(x) - \frac{T_{\text{stick}}(x)}{2} \\
x_1 - \Omega \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_7} &= \begin{pmatrix}
\varphi_1(x) \\
-x_1 + x_3 \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_8} &= \begin{pmatrix}
\varphi_2(x) - \frac{T_{\text{stick}}(x)}{2} \\
x_1 - \Omega \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_9} &= \begin{pmatrix}
\varphi_1(x) + \eta \\
x_1 - x_3 \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_6} &= \begin{pmatrix}
\varphi_2(x) \\
-x_1 - \Omega \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_7} &= \begin{pmatrix}
\varphi_1(x) + \eta \\
x_1 - x_3 \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_8} &= \begin{pmatrix}
\varphi_2(x) \\
-x_1 - \Omega \\
x_1 - x_3
\end{pmatrix}, \\
f_{q_9} &= \begin{pmatrix}
-2\lambda x_1 + \Omega \\
x_1 \\
x_1
\end{pmatrix},
\end{align*}
\]

with $S_0^0 = \{x \in \mathbb{R}^5 : s'(x, t) = 0\}$.

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 $S_0^0$ is attractive for all $x$ and that all trajectories reach $S_0^0$ in a finite time, and once the trajectory reaches $S_0^0$, 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, unacceptable
transitions are also the transitions involving a cross through $S^3_{b}$. Consequently, there are 36 feasible edges or transitions:

$$E_{feas} = \{(q_1, q_6), (q_1, q_3), (q_3, q_1), (q_3, q_8), (q_8, q_1), (q_8, q_3), (q_3, q_6), (q_5, q_7), (q_7, q_5), (q_7, q_6), (q_6, q_7), (q_1, q_7)(q_2, q_9), (q_2, q_8), (q_4, q_2), (q_4, q_9), (q_9, q_2), (q_9, q_4), (q_9, q_6), (q_1, q_5), (q_5, q_6), (q_3, q_7), (q_2, q_6), (q_9, q_6), (q_4, q_7), (q_9, q_7), (q_1, q_6), (q_2, q_6), (q_3, q_6), (q_4, q_6), (q_8, q_5), (q_8, q_7), (q_2, q_7), (q_4, q_5)\}.$$ 

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:

$$a \leftrightarrow G^0 \cap S^0_a, \quad b \leftrightarrow G^- \cap S^0_a, \quad c \leftrightarrow G^+ \cap S^0_a,$$
$$d \leftrightarrow G^0 \cap S^0_1, \quad e \leftrightarrow G^- \cap S^0_1, \quad f \leftrightarrow G^+ \cap S^0_1,$$
$$g \leftrightarrow G^0 \cap S^0_2, \quad h \leftrightarrow G^- \cap S^0_2, \quad i \leftrightarrow G^+ \cap S^0_2,$$
$$j \leftrightarrow S^b \cap S^0_3, \quad k \leftrightarrow S^b \cap S_3.$$ 

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

### 6. Simulation of the hybrid automata: Modelica versus Stateflow

Stateflow® under Simulink® in MATLAB® and The MathWorks (2008) and Modelica® (Modelica 2009) under Dymola® 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 $H_{DDS}$ and $H_{DDSS}$ into Stateflow charts is almost immediate. The Stateflow charts obtained look the same as the directed graphs associated with $H_{DDS}$ and $H_{DDSS}$, given in Figures 1 and 2. Only, in the Stateflow chart for $H_{DDSS}$, 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 $\text{novelocity} = 1$. The boolean variable $\text{novelocity}$ 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 $H_{DDSS}$. The Simulink/Stateflow models for $H_{DDSS}$ and $H_{DDSS}$ 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:

$$J_r = 2122 \text{ kg m}^2, \quad J_b = 471.9698 \text{ kg m}^2,$$
$$R_0 = 0.155575 \text{ m}, \quad k_r = 861.5336 \text{ N m/rad},$$
$$c_t = 172.3067 \text{ N m/s/rad}, \quad c_r = 425 \text{ N m/s/rad},$$
$$c_b = 50 \text{ N m/s/rad}, \quad \mu_c = 0.5,$$
$$\mu_b = 0.8, \quad \delta = 10^{-6}, \quad \gamma_b = 0.9, \quad \nu = 1.$$ 

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. $x(t) \in S_b^0 \forall t > t_s$, for some $t_s > 0$ (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.
We should note that, in the evolution of $q$ in $H_{DDS}$, 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 $T_{sb}$, 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.
Stateflow simulations, the maximum step size is \(h = 0.001\) s, the minimum step size is \(10^{-5}\) s, and the tolerance is \(\text{tol} = 0.001\). 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 \(\text{tol} = 10^{-4}\) is used, except for the situation shown in Figure 6, in which \(\text{tol} = 10^{-8}\) 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 \(H_{\text{DDS}}\), and \(H_{\text{DDS}}\), 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 \(\text{tol} = 10^{-8}\) 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 \(W_{\text{ab}}\) and \(u\), since these parameters put the trajectory close to the bifurcation leading to stick–slip.
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

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

model simulation and the 3-location automaton simulation. Finally, in Figure 10, although a tolerance of $\text{tol} = 10^{-3}$ 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 $u = 6 \text{kN m}$ and $W_{ob} = 51,048 \text{N}$. 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 ($\Omega$) and the parameter $\lambda$. 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.
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 $\Omega$ (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 $\Omega$ 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 Liceaga-Castro 2009); in general, it is dependent on the value of $\lambda$, and the other typical parameters of the drillstring. However, the equation of the boundary does not change with $\eta$, 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 y 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

References


