Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 739253, 7 pages
http://dx.doi.org/10.1155/2013/739253

Research Article

Automata-Based Analysis of Stage Suspended Boom Systems

Anping He,[1,2] Jinzhao Wu,[1] Shihan Yang,[1] Yongquan Zhou,[1] and Juan Wang[3]

[1] *Guangxi Key Lab of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning 530006, China*
[2] *School of Information Science and Engineering, Lanzhou University, Lanzhou 730000, China*
[3] *School of Civil Engineering and Mechanics, Lanzhou University, Lanzhou 730000, China*

Correspondence should be addressed to Anping He; hapetis@gmail.com

Received 7 February 2013; Accepted 23 February 2013

Academic Editor: Xiaoyu Song

A stage suspended boom system is an automatic steeve system orchestrated by the PLC (programmable logic controller). Security and fault-recovering are two important properties. In this paper, we analyze and verify the boom system formally. We adopt the hybrid automaton to model the boom system. The forward reachability is used to verify the properties with the reachable states. We also present a case study to illustrate the feasibility of the proposed verification.

1. Introduction

The special effects in live performance make the audience astonishing through colorful background and a stage suspended boom system. Generally, the background indicates a stable scene, but the suspended boom system shows a dynamic stunt. The special effects for live performance are always implemented by suspending objects and/or human in midair from the boom system, such as "little girl flying a kite" in the Olympic Games in Beijing 2008.

The contemporary boom systems contain a fixed physical mechanism and a programmable controller, such as the steeve system and PLC. The PLC samples the position of each steeve and restricts their movement periodically. Since human life is at stake, security is the top priority. The boom system is also mission critical due to its live performance nature-stunt. Generally, the stunt is against the security deeply; the audience wants exciting stunts, but actors need safe shows, which makes "Stunt Injuries and Fatalities Increasing" stated by McCann [1].

We focus on the formal aspect of analyzing and verifying stage suspended boom systems. The boom system exhibits a hybrid behavior; for example, the continuous behavior in a time period interacts with the discrete events. Therefore, it is natural to adopt hybrid automaton to model and verify this type of system.

Hybrid automaton is a formal model for precisely describing hybrid system in which computational processes interact with physical processes. Similar to other types of automaton, the hybrid contains states and transitions, but it also labels and groups relevant dense states as activities to express continuous behaviors, which are described by *state functions*. Then the behavior of the hybrid system is composed of discreteness of state transitions and continuity of state evolution.

The hybrid automaton was introduced in [2, 3] with analysis for some linear and nonlinear examples, and in [4], the authors focused on its verification aspect. There are many works on verification and analysis of hybrid automata now [4–7] studied the model checking of hybrid automata; [2, 3, 8] performed reachability analysis; [9, 10] studied probabilistic hybrid automata; [11] studied the hybrid automata with a domain-theoretic semantics. Moreover, there are many prior works on the case study of hybrid systems with automata. In [12], the safety properties of the automobile control system are studied. In [9], the hybrid automaton was used to analyze the circuit system. In [10], the authors focused on the sensor-driven hybrid automaton and gave a concrete example of goal network. In [2], an automated manufacturing system is studied. But to the best of our knowledge, no article studies the boom system formally in terms of hybrid automata.

In Section 2, we show the behaviors of the boom system in a formal way. In Section 3, we analyze a concrete case study for feasibility. We conclude in Section 4.

2. Modeling Stage Suspended Boom System

A stage suspended boom system is the system used to achieve a stunt. The contemporary boom system is composed of automatic electromechanical controllers and physical mechanisms, including PLCs, steeves, curtains, and electrical motors. We study this type of system in terms of interactions of steeves and PLCs.

2.1. Movement. A boom system performs a stunt by controllable steeves. The steeves are directly driven by electrical motors. We analyze the movement of the system by the movement of steeves. For example, we establish a 4-dimensional mathematical model describing the locus of each steeve, the first 3 dimensions express where the steeve is, and the last one specifies when it arrives there. Let x, y, and z be the first three dimensions and t the last one.

The whole movement of steeves is seen as a scene of a boom system. The controllable movement of each steeve is always adjusted and restricted manually or automatically by PLCs. In contemporary boom systems, manual control is only adopted to start a scene or stop it in an emergency. Once the automatic control is triggered, the PLCs manage the movement of steeves continuously unless an emergency occurs.

A stunt in a stage provides audience the astonishing effects; at the same time it also provides high risks for an actor/actress. Generally, the inertia and rotation are two main risks while steeves moving. In order to prevent these phenomena, the steeves in a real boom system move slowly and smoothly to reduce inertia; meanwhile, several steel wires connect a steeve and its driver for a consistent movement.

For each steeve, it is a mechanical device that behaves under the laws of physics, dealing with quantities of displacement, velocity, and acceleration. The steeve reports its current condition through sensors, adjusts its movement by the reference of actuators, and thus shows a controllable (piecewise) continuous time-variant property. We analyze its behavior by those physics laws.

We express the movement of a boom system with the velocity of steeves. To keep the movement slowly and smoothly, the acceleration is really low and close to 0; then in most cases, the velocity is a constant. In addition, the motors in the stage suspended system are commonly constant speed electric ones or the variable-frequency direct-current ones, which makes the control easy and effective. So this type of control belongs to linear ones.

Let us show the movement of a steeve. Each steeve of a stage is driven by the electro-motors. Each motor drives a steeve to move forward or backward, left or right, or up or down in terms of a control signal. So the movement of a steeve is a combination of drives of motors. We use the vector and matrix to express the following analysis formally. Let us consider xyz-coordinates of a 3-dimension stage; $\vec{u}_i$ be a velocity of the steeve indexed by i, a_i a control, $f_x(x_i)$, $f_y(y_i)$, and $f_z(z_i)$ time differential, for example, velocities; a_{x_i}, a_{y_i}, and a_{z_i} controls; then

$$a_i = \begin{pmatrix} a_{x_i} & 0 & 0 \\ 0 & a_{y_i} & 0 \\ 0 & 0 & a_{z_i} \end{pmatrix} \tag{1}$$

and $\vec{u}_i = (f_x(x_i), f_y(y_i), f_z(z_i)) \times a_i$. So the movement of the whole boom system, $\vec{u}$, is $\vec{u} = (\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_n)$. Let a be a control matrix, for example,

$$a = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ & \cdots & & \\ 0 & 0 & \cdots & a_n \end{pmatrix}. \tag{2}$$

The movement of boom system could be reexpressed as the following:

$$\vec{u} = \left(f_x(x_1), f_y(y_1), f_z(z_1), f_x(x_2), f_y(y_2), f_z(z_2), \ldots, f_x(x_n), f_y(y_n), f_z(z_n)\right) \times a. \tag{3}$$

Equation (3) shows that the movement of a boom system is depending on the velocity of each motor, for example, velocities in every direction and a control matrix.

We call the movement of boom system under a concrete control matrix an *activity*, whose number is finite because of the finite number of signals.

2.2. Scene. The steeves and the PLCs communicate and cooperate to implement a live performance, for example, a scene.

A scene shows the configuration of activities of a boom system in terms of controls from PLCs. Each control is a matrix of concrete control signals, for example, a concrete control value for its movement in a special direction. Let val_i be the valuation of a control matrix to a control value matrix. A scene formally defines a sequence of valuations, $\mathrm{val}_1(a), \mathrm{val}_2(a), \ldots, \mathrm{val}_k(a), \ldots$, that $\mathrm{val}_i(a) \neq \mathrm{val}_{i+1}(a)$. We call $(\mathrm{val}_i(a), \mathrm{val}_{i+1}(a))$ *scene related* controls.

PLCs implement a scene by configuring activities, which construct a hybrid system essentially. For example, the continuous behavior is determined by the steeves, while the discrete one by relation among their movement. The implementation of the scene is complex owing to the nondeterministics of the movement. Steeves driven by motors may not always move functionally, so the PLC owns a mechanism of exceptions.

There are two types of nondeterministics: timeout and inconsistency. The timeout indicates that the movement has to finish in a max time duration or the system suspends. The inconsistency involves two phenomena: the inconsistency of the steeve movement in different directions (the motors of a steeve do not cooperate well) or the inconsistency among the moving steeves (the steeves do not cooperate well). All of them have to be treated safely. Let us adjust our scene analysis in terms of these nondeterministics.

Let next be a function of getting a next control value matrix with current value. Then for a current control value matrix $\mathrm{val}_i(a)$, the next may involve three types of matrix: one for the scene requirement, one for the timeout fault, and one(s) for the inconsistent movement.

The inconsistency involves source control value matrices, destination matrix, and inconsistent matrix. Let

$$\mathrm{val}_i(a) = \begin{pmatrix} v_{i_1} & 0 & \cdots & 0 \\ 0 & v_{i_2} & \cdots & 0 \\ & \cdots & & \\ 0 & 0 & \cdots & v_{i_n} \end{pmatrix},$$

$$\mathrm{val}_j(a) = \begin{pmatrix} v_{j_1} & 0 & \cdots & 0 \\ 0 & v_{j_2} & \cdots & 0 \\ & \cdots & & \\ 0 & 0 & \cdots & v_{j_n} \end{pmatrix}, \quad (4)$$

$$\mathrm{val}_k(a) = \begin{pmatrix} v_{k_1} & 0 & \cdots & 0 \\ 0 & v_{k_2} & \cdots & 0 \\ & \cdots & & \\ 0 & 0 & \cdots & v_{k_n} \end{pmatrix}$$

be different controls; for example, not all control signals of motors are the same, $\exists l \in \{1, \ldots, n\}$ that $v_{i_l} \neq v_{j_l} \neq v_{k_l}$. We say $(\mathrm{val}_i(a), \mathrm{val}_k(a), \mathrm{val}_j(a))$ is *processably inconsistent* whenever $v_{i_l} = v_{j_l}$. Then if processable inconsistence holds, $v_{k_l} = v_{i_l}$ and then direction of the movement does not change.

Now we can show next control valuation matrix by the scene relation and the processable inconsistency relation. Let $\mathrm{next}(\mathrm{val}(a)) = \mathrm{val}'(a)$ with the following.

(i) $(\mathrm{val}(a), \mathrm{val}'(a))$ is scene related;

(ii) there exists a valuation matrix, $\mathrm{val}_i(a)$, that $(\mathrm{val}(a), \mathrm{val}'(a), \mathrm{val}_i(a))$ is processably inconsistent, and $(\mathrm{val}(a), \mathrm{val}_i(a))$ is scene related;

(iii) there exist two activities, $\mathrm{val}_i(a)$ and $\mathrm{val}_j(a)$ $(i \neq j)$, that $(\mathrm{val}_i(a), \mathrm{val}_j(a))$ is scene related and $(\mathrm{val}(a), \mathrm{val}'(a), \mathrm{val}_j(a))$ and $(\mathrm{val}_i(a), \mathrm{val}(a), \mathrm{val}_j(a))$ are processably inconsistent; or

(iv) $\mathrm{val}'(a)$ is a *suspend* control; for example, all control values are 0.

2.3. Hybrid Automaton Model. There are several types of definitions for hybrid automata [2–4], all of which construct a position (control model) graph with the events of jump transitions. A hybrid automaton is a sextuple of positions, real-valued variables, event labels, transitions, activities, and invariants. We study the formal model by (forward) reachability analysis; for example, let t, l, and x be a time elapsing, location and variable, and the activity is denoted by $\psi_l[v]$. We can verify a hybrid system by the *forward analysis* of reachability analysis. We compute "time can progress" of max time duration that elapsed in the position l by $tc_{p_l}[v](t)$, if for all $t_1 \in [0, t]$ that $\psi_l[v](t_1) \in \mathrm{Int}(l)$. Then we compute *forward time closure* of the valuation set of l-position and a special valuation set by $v_0 \in \langle P \rangle_\ell^{\nearrow}$, which means $\exists v \in V, t \in R^{\geq 0}$ if and only if $v \in P \wedge cp_\ell[v](t) \wedge v_0 = \psi_\ell[v](t)$. And then we compute *postcondition* of a set of valuations generated through transitions by $v_1 \in \mathrm{post}_e[P]$, which means $\exists v \in V$ if and only if $v \in P \wedge (v, v_1) \in \mu$. So finally we compute the set of reachable states by the fixpoint of the following equation:

$$X_\ell = \left\langle I_\ell \bigcup_{e=(\ell_0,\ell)\in Edg} \mathrm{post}_e[X_{\ell_0}] \right\rangle_\ell^{\nearrow} \quad (5)$$

with $I = \bigcup_{\ell \in \mathrm{Loc}} (\ell, I_\ell)$ being a set of initial states.

The (piecewise) continuous movement (see Section 2.1) builds the part of (piecewise) continuous behavior of a boom system. In contrast, periodically and discretely, the PLC implements a scene of live performance, by monitoring the continuous behavior (by sensors), generating decisions (by logic reasoning and data processing programs), and then writes the compatible controls into the actuators, instantaneously. The control may change the system movement by adding the boom system discrete behaviors (see Section 2.2). In short, the interacted (piecewise) continuity and discreteness of this system present a hybrid behavior, indeed.

It is very convenient to translate the analysis in previous subsections into a hybrid automaton. The variables include all movement variables, signal variables, and other special ones; the locations, as well as activities contained in the location, are corresponding to the activities of system movement directly; the edges between the activities are described by the next function; the invariant of each location involves the max time allowed for every activity, which is concluded from scene. We will show the hybrid automaton in the following case study.

3. A Case Study

For this specific boom system, each steeve is driven by two constant speed electric motors located in vertical and horizontal directions. A moving steeve could be stopped at any position if a stop button is pushed for some security reasons; moreover, after the operator pushes the start button, each steeve moves automatically under the control of the PLCs. The PLCs are used as intelligent controllers. The electric motor rotates in a constant speed to keep the steeve moving placidly. The transducers fixed on motors will send 256 pulses per motor rotation cycle, and the steeve will move 16cm. The PLCs memorize and calculate the number of pulses to control the electric motor. Moreover, the direction of motor rotation can be adjusted to move down (or left) or to up (or right) by the PLC signals.

We study an interesting scene, for example, a stunt that a actor/actress riding a bicycle to climb "hill", the bicycle is hung on a steeve in a xy coordination. Initially, the bicycle locates at the $(0, 0)$ position, then the bicycle, as well as the actor, begins to climbing a hill, for example, moves slowly to position (L, H), and then $(0, 2 \times H)$ of the top of the hill. Then the bicycle moves from one peak to another horizontally. And then moves down the hill and arrives at foot $(L, 0)$. Finally, moves to the initial position $(0, 0)$ and begins to another cycle.

We consider the following problems.

(i) Does the state suspended boom system perform safely?

(ii) Does the state suspended boom system perform correctly?

3.1. Hybrid Automata. We have two button-related variables: start and stop for start and stop stunt manually. The scene of this case study involves one steeve and one PLC and plays a 2D movement. So we get two movement related variables: x and y for x-axis and y-axis. The PLC samples the movement indirectly; it records and calculates the pulse number of each angular transducer of motors. Then two special variables are necessary, let xc be pulse counter for x-axis moving, and yc for the other. Moreover, we use another variable for local timer, for example, let d record the elapsed time of a position (or control model). Then the variable set is Var = {start, stop, x, y, xc, yc, d}, and controlled variable set Con = {x, y, xc, yc, d}.

Let v_x and v_y be constants for velocities of the steeve in x and y directions separately. Only one steeve be used in this scene; the movement is expressed by velocities and rates of pulse; so by the way introduced in Section 2, we get $a = \left(\begin{smallmatrix} a_x & 0 \\ 0 & a_y \end{smallmatrix}\right)$ and $\vec{u} = (v_x, v_y) \times a$.

According to the specification, the control a_x (a_y) makes a steeve moving left or right (up or down); so we set its value as $\{-1, 0, 1\}$ for moving left, stopping moving, and right (similar to a_y). Then we get the nine activities in Table 1 with (3).

The scene defines a sequence of controls: $\mathrm{val}_{\mathrm{ur}}(a)$, $\mathrm{val}_{\mathrm{ul}}(a)$, $\mathrm{val}_{\mathrm{r}}(a)$, $\mathrm{val}_{\mathrm{dl}}(a)$, $\mathrm{val}_{\mathrm{dr}}(a)$, and $\mathrm{val}_{\mathrm{l}}(a)$. We extend this control sequence by consistency and timeout analysis; let us take a scene relation $(\mathrm{val}_{\mathrm{ur}}(a), \mathrm{val}_{\mathrm{ul}}(a))$ as an example,

$$\begin{aligned} \mathrm{next}\left(\mathrm{val}_{\mathrm{ur}}(a)\right) &= \mathrm{next}\left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right) \\ &= \left\{\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\right\}. \end{aligned} \tag{6}$$

So when the steeve is moving up-right, it may suspend for timeout, or its horizontal/vertical movement finishes but the vertical/horizontal not (see the shadow in Figure 1). In this figure, the x-position movement finishes in t_1 time point; the y-direction displacement of any (red) line in the shadow is necessary for the consistency. After calculating all the next control valuation matrixes, we get the connected graph in Figure 2 without formulas.

PLC cannot add more movement of boom system but only can organize some activities to form a scene; so as we talked in Section 2.3, we can build the hybrid automaton according to the patterns of movement, for example,

$$\mathrm{Loc} = \{\mathrm{ur}, \mathrm{ul}, \mathrm{dl}, \mathrm{dr}, \mathrm{up}, \mathrm{down}, \mathrm{right}, \mathrm{left}, \mathrm{susp}\}. \tag{7}$$

We write ur as up-right for short; others are similar. The *Act* of a location is the same as the corresponding activity previously.

The PLC reads transducer sensors to adjust movement of a steeve to implement the stunt, the rate of pulse of a sensor is the same as the velocity of a steeve, and it is convenient to show the invariant and edges by pulse counter xc and yc. Let us define hypothesis to make the following expression simple. Let N_L and N_H be two pulse constants with $N_L = (256/16) \times L = 16 \times L$ and $N_H = 16 \times H$; let int returns the integer portion of a number and mod be a function of getting the remainder

Table 1: The activities.

	a	u
Up-right	$\mathrm{val}_{\mathrm{ur}}(a) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$u_{\mathrm{ur}} = (v_x, v_y)$
Up-left	$\mathrm{val}_{\mathrm{ul}}(a) = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$	$u_{\mathrm{ur}} = (-v_x, v_y)$
Down-left	$\mathrm{val}_{\mathrm{ur}}(a) = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$	$u_{\mathrm{dl}} = (-v_x, -v_y)$
Down-right	$\mathrm{val}_{\mathrm{ur}}(a) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$u_{\mathrm{dr}} = (v_x, -v_y)$
Up	$\mathrm{val}_{\mathrm{ur}}(a) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$	$u_{\mathrm{u}} = (0, v_y)$
Down	$\mathrm{val}_{\mathrm{ur}}(a) = \begin{pmatrix} 0 & 0 \\ 0 & -1 \end{pmatrix}$	$u_{\mathrm{d}} = (0, -v_y)$
Right	$\mathrm{val}_{\mathrm{ur}}(a) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$	$u_{\mathrm{r}} = (v_x, 0)$
Left	$\mathrm{val}_{\mathrm{ur}}(a) = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}$	$u_{\mathrm{l}} = (-v_x, 0)$
Suspend	$\mathrm{val}_{\mathrm{ur}}(a) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	$u_{\mathrm{sp}} = (0, 0)$

produced by being divided into two integers. We define the following assertions:

$$\begin{aligned} \mathrm{stop} &:= \mathrm{start} = 0 \wedge \mathrm{stop} = 1, \\ \mathrm{auto} &:= \mathrm{start} = 1 \wedge \mathrm{stop} = 0, \\ h_{\mathrm{mv}} &:= \mathrm{mod}\left(\mathrm{int}\left(xc\right), N_L\right) \le N_L, \\ v_{\mathrm{mv}} &:= \mathrm{mod}\left(\mathrm{int}\left(yc\right), N_H\right) \le N_H, \\ h_{\mathrm{slp}} &:= \mathrm{mod}\left(\mathrm{int}\left(xc\right), N_L\right) = 0, \\ v_{\mathrm{slp}} &:= \mathrm{mod}\left(\mathrm{int}\left(yc\right), N_H\right) = 0, \\ D &:= d \le T_{\max}, \end{aligned} \tag{8}$$

where stop and auto are two assertions for the steeve suspending and moving automatically, h_{mv} and v_{mv} for moving horizontally and vertically, h_{slp} and v_{slp} for sleeping on horizontal or vertical directions, and the last one D for the max time duration of a movement. Then the *Inv* of each location will be

$$\begin{aligned} \mathit{Inv}\left(\mathrm{ur}\right) &:= \mathit{Inv}\left(\mathrm{ul}\right) := \mathit{Inv}\left(\mathrm{dl}\right) \\ &:= \mathit{Inv}\left(\mathrm{dr}\right) := h_{\mathrm{mv}} \wedge v_{\mathrm{mv}} \wedge D \wedge \mathrm{auto}, \\ \mathit{Inv}\left(\mathrm{up}\right) &:= \mathit{Inv}\left(\mathrm{down}\right) := h_{\mathrm{slp}} \wedge v_{\mathrm{mv}} \wedge D \wedge \mathrm{auto}, \\ \mathit{Inv}\left(\mathrm{right}\right) &:= \mathit{Inv}\left(\mathrm{left}\right) := h_{\mathrm{mv}} \wedge v_{\mathrm{slp}} \wedge D \wedge \mathrm{auto}, \\ \mathit{Inv}\left(\mathrm{susp}\right) &:= \mathrm{stop}. \end{aligned} \tag{9}$$

We get Edg from the next function, which defines the transition among control valuation matrixes. According to the definition of next($\mathrm{val}_{\mathrm{ur}}$), we know the next location of up-right ur will be up, susp, and right as seen in Table 1. Let us show an example of the transition from ur to up. In each

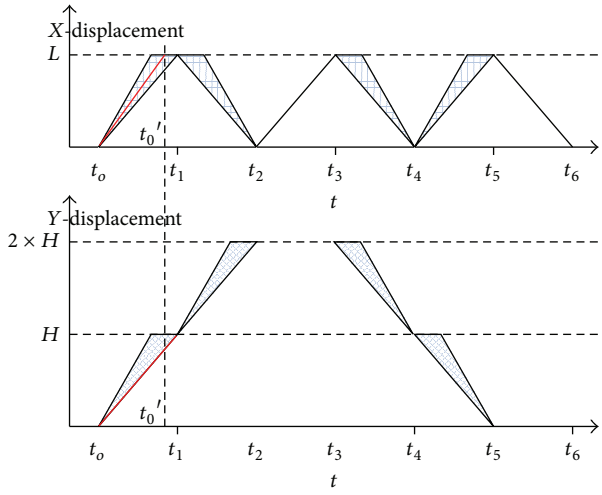

FIGURE 1: The fault self-recovered movement.

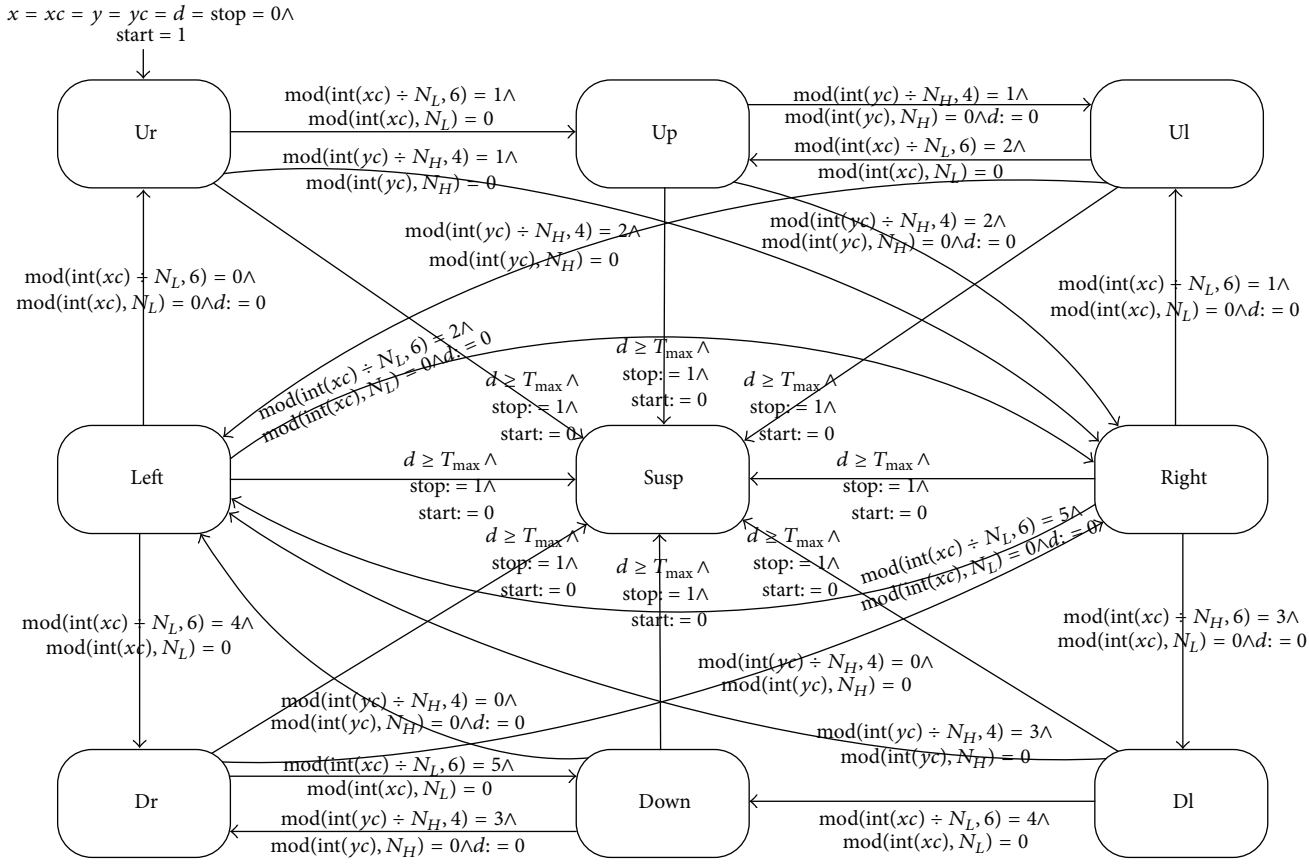

FIGURE 2: The hybrid automaton of riding a bicycle.

cycle, the moving distance on x-axis is $6 \times L$ that the direction of the displacement changes once increasing L and so is the movement along the y-axis. The ur to up transition happens to recover the moving error when the up-right movement of scene performs; for example, if the movement on x-axis finishes before the one on y-axis, the bicycle moves up-right, then up, and finally up-left; then the condition of the jump shows the finish of the first x-axis movement, for example, the $6k \times L + L$ distance. So we get the following formulae:

$$\mu := \text{mod}\left(\text{int}\left(xc\right) \div N_L, 6\right) = 1 \wedge \text{mode}\left(\text{int}\left(xc\right), N_L\right) = 0. \tag{10}$$

Then the transition expression is $e = (\mathrm{ur}, \alpha, \mu, \mathrm{up})$ in which α is an event label.

Step by step, we construct a hybrid automaton in Figure 2 (each position is indexed by an integer number for utility).

3.2. Reachability Analysis. We can prove some interesting properties by reachability analysis of the hybrid automaton in Figure 2. Let us use the forward analysis method in Section 2.3 to compute reachable state set from initial states.

The initial states illustrate that the steeve starts moving up-right from the coordination $(0, 0)$; meanwhile, the elapsed time begins to be recorded. Let pc be an integer over indexes of positions; we express the initial states as a formula ψ_I:

$$\begin{aligned}\psi_I &:= pc = 1 \wedge x = y = xc = yc = d = 0 \wedge \mathrm{start} \\ &= 1 \wedge \mathrm{stop} = 0.\end{aligned} \tag{11}$$

According to (5), the reachable states are characterized by the least fixpoint of the following nine equations:

$$\begin{aligned}
\psi_1 &= \left\langle \psi_{I_1} \vee \mathrm{post}_{(8,1)}[\psi_8] \right\rangle_1^{\nearrow}, \\
\psi_2 &= \left\langle \psi_{I_2} \vee \mathrm{post}_{(1,2)}[\psi_1] \vee \mathrm{post}_{(3,2)}[\psi_3] \right\rangle_2^{\nearrow}, \\
\psi_3 &= \left\langle \psi_{I_3} \vee \mathrm{post}_{(2,3)}[\psi_2] \vee \mathrm{post}_{(4,3)}[\psi_4] \right\rangle_3^{\nearrow}, \\
\psi_4 &= \left\langle \psi_{I_4} \vee \mathrm{post}_{(1,4)}[\psi_1] \vee \mathrm{post}_{(2,4)}[\psi_2] \right. \\
&\qquad \left. \vee \mathrm{post}_{(7,4)}[\psi_7] \vee \mathrm{post}_{(8,4)}[\psi_8] \right\rangle_4^{\nearrow}, \\
\psi_5 &= \left\langle \psi_{I_5} \vee \mathrm{post}_{(4,5)}[\psi_4] \right\rangle_5^{\nearrow}, \\
\psi_6 &= \left\langle \psi_{I_6} \vee \mathrm{post}_{(5,6)}[\psi_5] \vee \mathrm{post}_{(7,6)}[\psi_7] \right\rangle_6^{\nearrow}, \\
\psi_7 &= \left\langle \psi_{I_7} \vee \mathrm{post}_{(6,7)}[\psi_6] \vee \mathrm{post}_{(8,7)}[\psi_8] \right\rangle_7^{\nearrow}, \\
\psi_8 &= \left\langle \psi_{I_8} \vee \mathrm{post}_{(3,8)}[\psi_3] \vee \mathrm{post}_{(4,8)}[\psi_4] \right. \\
&\qquad \left. \vee \mathrm{post}_{(5,8)}[\psi_5] \vee \mathrm{post}_{(6,8)}[\psi_6] \right\rangle_8^{\nearrow}, \\
\psi_9 &= \left\langle \psi_{I_9} \vee \mathrm{post}_{(1,9)}[\psi_1] \vee \mathrm{post}_{(2,9)}[\psi_2] \right. \\
&\qquad \vee \mathrm{post}_{(3,9)}[\psi_3] \vee \mathrm{post}_{(4,9)}[\psi_4] \\
&\qquad \vee \mathrm{post}_{(5,9)}[\psi_5] \vee \mathrm{post}_{(6,9)}[\psi_6] \vee \mathrm{post}_{(7,9)}[\psi_7] \\
&\qquad \left. \vee \mathrm{post}_{(8,9)}[\psi_8] \right\rangle_9^{\nearrow}.
\end{aligned} \tag{12}$$

By the initial states, $\psi_{I_1} = \psi_I$ and $\psi_{I_2} = \cdots = \psi_{I_9} = \mathrm{false}$, we can calculate the fixpoint of (12) iteratively, for example, let $i = 1, 2, \ldots$ be the times iterated; we calculate the first equation by $\psi_{1,i} = \langle \psi_{1,i-1} \vee \mathrm{post}_{(8,1)}[\psi_{8,i-1}] \rangle_1^{\nearrow}$; then we get

$$\begin{aligned}\psi_1 = \big(&0 \le x \le L \wedge 0 \le y \le H \wedge 6 \times C_1 \le xc \le 6 \times C_1 \\ &+ N_L \wedge 4 \times C_2 \le yc \le 4 \times C_2 + N_H \wedge D \wedge \mathrm{auto}\big)\end{aligned} \tag{13}$$

with $k = 1, 2, 3, \ldots$ being the number of cycles of movement, $C_1 = (k-1) \times N_L$, $C_2 = (k-1) \times N_H$, and so are the others.

The safety properties can be studied in terms of the reachable states. We list properties as the lemmas bellow.

Lemma 1. *After a scene begins to perform, the movement of the boom system will be in a safe area, for example, a rectangle of* $(0, 0), (0, L), (L, 2 \times H)$, *and* $(0, 2 \times H)$:

$$t \ge 0 \Longrightarrow 0 \le x \le L \wedge 0 \le y \le 2 \times H. \tag{14}$$

Lemma 2. *In each activity, if duration of the activity is longer than* $T_{\max}$, *the movement will stop, for example.*

$$t > T_{\max} \Longrightarrow \psi_9. \tag{15}$$

These two security properties are direct from the reachable states.

Lemma 3. *The behavior of the system conforms to the specification of the scene.*

Proof. This lemma requires that the movement of steeves follows the scene specification; for example, implementation of the bicycle climbing the hill-zigzag traces appears once duration resetting transition ($d := 0$) is triggered. For example, if the steeve arrives at the position (L, H) (the PLC controller only knows this from $xc = 6 \times k \times N_L + N_L \wedge yc = 4 \times k \times N_H + N_H$ with k being the number of cycles of movement), a zigzag appears, for example,

$$\begin{aligned}
&xc = 6k \times N_L + N_L \wedge yc = 4k \times N_H + N_H \\
&\Longrightarrow \big(\big((6k \times N_L \le xc < 6k \times N_L + N_L) \\
&\qquad \wedge (4k \times N_H < yc < 4k \times N_H + N_H) \\
&\qquad \wedge (0 \le x < L \wedge 0 \le y < H)\big) \vee (x = L \wedge y = H)\big) \\
&\qquad \vee \big((6k \times N_L + N_L \le xc < 6k \times N_L + 2 \times N_L) \\
&\qquad\quad \wedge (4k \times N_H + N_H yc < 4k \times N_H + 2 \times N_H) \\
&\qquad\quad \wedge (0 \le x < L \wedge H < y \le 2 \times H)\big).
\end{aligned} \tag{16}$$

The formula $xc = 6 \times k \times N_L + N_L \wedge yc = 4 \times k \times N_H + N_H$ implies that ψ_1, ψ_2, ψ_3, ψ_4, and ψ_9 hold. Then we can check that the set of states characterized by the formulas after $\Rightarrow$ is a subset of states defined by $(\psi_1 \vee \psi_2 \vee \psi_3 \vee \psi_4 \vee \psi_9)$. The formal specifications and proofs of other zigzag trace, are similar. So we know that the hybrid system of the boom system holds the specification of the scene, and then lemma 3 is proved. □

Lemma 4. *The boom system recovers its fault movement by itself.*

Proof. Let us take the zigzag trace for example, if the time duration of the up-right movement is less than $T_{\max}$ and the x-direction movement finishes but y-direction movement

not, then x-direction movement stops and waits for the y-direction movement, for example,

$$\begin{aligned}\left(t \le T_{\max} \wedge xc = 6 \times k \times N_L + N_L \wedge yc < 4 \times k \times N_H\right. \\ \left. + N_H \wedge \text{start} = 1 \wedge \text{stop} = 0\right) \Longrightarrow \psi_2.\end{aligned} \tag{17}$$

The proof can be directly reasoned from the formulas of reachable states. Similarly, we can verify other fault-recovering requirements by their characterization of the reachable states. □

4. Conclusion

In this paper, we adopted hybrid automaton as the model of the boom system, and then used the forward method to analyze its reachability problem. Some important properties were verified in terms of the reachable states. An interesting case study of scene of bicycle climbing hill was shown to prove the feasibility of our study.

In future, we will adopt tools of hybrid automata to make the analysis and verification (possibly) automatically. After many case studies, we regard there could be a framework of modeling and verifying this type of system; so we will study more cases to find their characteristics, and then propose a practice framework (which may not only be solvable by hybrid automata).

Acknowledgments

This work is partly supported by Grants (HCIC201110) of Guangxi HCIC lab Open Fund, the Fundamental Research Funds for the Central Universities of Lanzhou University, no. 860772, and NSF of China no. 60973147, the Doctoral Fund of Ministry of Education of China under Grant no. 20090009110006 the NSF of Guangxi no. 2011GXNSFA018154, the Science and Technology Foundation of Guangxi no. 10169-1, and Guangxi Scientific Research Project No.201012MS274.

References

[1] M. McCann, "Stunt injuries and fatalities increasing, Tech. Rep.," http://www.uic.edu/sph/glakes/harts1/HARTS_library/stunts.txt.

[2] R. Alur, C. Courcoubetis, N. Halbwachs et al., "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, pp. 3–34, 1995.

[3] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Algorithmic analysis of nonlinear hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 540–554, 1998.

[4] T. A. Henzinger, "The theory of hybrid automata," Tech. Rep. UCB/ERL M96/28, EECS Department, University of California, Berkeley, Calif, USA, 1996.

[5] C. S. F. Balduzzi and A. Giua, "Modelling automated manufacturing systems with hybrid automata," in *Proceedings of the Workshop on Formal Methods and Manufacturing*, vol. 138, pp. 33–48, Zaragoza, Spain, 1999.

[6] R. Gentilini, K. Schneider, and B. Mishra, "Successive abstractions of hybrid automata for monotonic CTL model checking," in *Proceedings of the International Symposium on Logical Foundations of Computer Science (LFCS '07)*, pp. 224–240, June 2007.

[7] A. Podelski and S. Wagner, "Model checking of hybrid systems: from reachability towards stability," in *Hybrid Systems: Computation and Control*, vol. 3927 of *Lecture Notes in Computer Sciences*, pp. 507–521, Springer, Berlin, Germany, 2006.

[8] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?" *Journal of Computer and System Sciences*, vol. 57, no. 1, pp. 94–124, 1998.

[9] B. C. Williams, M. M. Henry, and M. M. Henry, *Model-Based Estimation of Proba- Bilistic Hybrid Automata*, 2002.

[10] J. M. B. Braman and R. M. Murray, "Probabilistic safety analysis of sensor-driven hybrid automata," in *Hybrid Systems: Computation and Control*, 2008.

[11] A. Edalat and D. Pattinson, "Denotational semantics of hybrid automata," in *Proceedings of the Foundations of software science and computation structures (FoSSaCS '06)*, vol. 3921, pp. 231–245, Springer, 2006.

[12] O. Msller and T. Stauner, "Modelling and verification using linear hybrid automata-a case study," *Mathematical Modelling of Systems*, vol. 1, no. 1, 111 pages, 1996.