Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 327465, 10 pages
http://dx.doi.org/10.1155/2014/327465

Research Article

A Case Study on Formal Analysis of an Automated Guided Vehicle System

Jie Zhang,[1,2] Yuntao Peng,[1] William N. N. Hung,[3] Xiaojuan Li,[4] Jindong Tan,[2] and Zhiping Shi[4]

[1] *College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China*
[2] *University of Tennessee, Knoxville, TN, USA*
[3] *Synopsys Inc., Mountain View, CA, USA*
[4] *Beijing Engineering Research Center of High Reliable Embedded System, College of Information Engineering, Capital Normal University, Beijing, China*

Correspondence should be addressed to Jie Zhang; jzhang@mail.buct.edu.cn

Received 6 March 2014; Accepted 30 March 2014; Published 4 May 2014

Academic Editor: Xiaoyu Song

This paper considers a hybrid I/O automata model for an automated guided vehicle (AGV) system. A set of key properties of an AGV system are characterized for the correctness of the system. An abstract model is constructed from the hybrid automata model to simplify the proof of the constraints. The two models are equivalent in terms of bisimulation relation. We derive the constraints to ensure the correctness of the properties. We validate the system by analyzing the parameters of the constraints of the AGV system.

1. Introduction

Complex systems cannot be described by a pure discrete model or a continuous model [1–3]. Hybrid models have become increasingly popular in the last few decades as systems become increasingly complex. A hybrid system is a dynamic system with interacting continuous time triggered and discrete event triggered dynamics [1, 2, 4–6]. Many applications involve hybrid systems, such as embedded controllers [7], robotics [8, 9], mobile computing [10], and process control [11], in which high reliability is a requirement [4]. To model such a system, we need to describe and analyze it with the rigorous use of mathematics. An I/O automaton is used to model concurrent and distributed discrete event systems [12]. A hybrid input/output automaton (HIOA) [4] is a framework, which is developed by Lynch et al. and extended from hybrid automata for modeling complex hybrid systems. This is done by dividing the state variables of a HIOA into two sets, classified as internal variables and external variables, where the external variables include input variables and output variables. Discrete transitions and continuous trajectories can change the states of a system. An extremely important feature of the hybrid I/O automaton framework is that the hybrid system is divided into multiple modules. These modules are described so that the hybrid system can be modeled easily. The hybrid I/O automaton uses the external variables, input variables, and output variables to communicate among the automatons.

Automated guided vehicles (AGVs) are robots that move on the floor of a facility directed by a combination of software and sensor-based guidance systems. Earlier inventions on AGVs can be dated back to Barrett Electronics in 1953. One of the oldest publications on AGV can be found in [13]. In the past, AGVs were typically deployed to manufacturing facilities due to their efficiency, accuracy, and flexibility. Nowadays, AGVs are also used in warehouses, distribution centers and transshipment terminals, and so forth for repeated transportation tasks [14, 15]. The tracking path for the AGV can be designed as a circle, ellipse, sine wave, or other shapes such as arbitrary curves [16, 17]. The tracking trajectory is very important as many papers develop effective approaches to solve it, but our AGV is an example of applying HIOA modeling. Our modeling is inspired by [2]. But unlike [2] which uses a straight line orbit that can be approximated

to one-dimension, we investigate a two-dimensional problem where an automated guided vehicle moves along a circular painted orbit.

The first contribution of this paper is the formal modeling of an automated guided vehicle system using hybrid I/O automata. The second contribution of this paper is a set of important constraints which are characterized to ensure the correctness of the properties of the vehicle system. In order to simplify the model, we abstract a model from the hybrid automata of the AGV and establish a bisimulation relation between the two automata.

This paper is organized as follows. In Section 2, an automated guided vehicle system is introduced. In Section 3, the HIOA framework is introduced. In Section 4, we present a HIOA model of the AGV system and abstract a model from HIOA model. We prove that the two models have a bisimulation relation. In Section 5, we extract the key properties and deduce the corresponding constraints to ensure the correctness of the properties. We analyze the parameters of the AGV system at the end of Section 6. Finally, we point out some directions for future work.

2. An Automated Guided Vehicle System

We introduce the structure and behavior of a vehicle. The vehicle consists of five components: the left wheels, the right wheels, chassis, sensor, and controller. Figure 1 shows a circular orbit tracking of our vehicle which is the focus of the remainder of this paper. The vehicle has two degrees of freedom. One is the velocity such that, at any time t, it can move forward with a speed of $v(t)$, with the restriction that $0 \leq v(t) \leq 10$ mph (miles per hour). The other degree of freedom is the circular movement of the vehicle such that at any time t the vehicle can rotate its body via the wheels with an angular speed of $-\pi \leq \omega(t) \leq \pi$ rad/s (radians per second). Ignoring the inertia of the vehicle, we assume that we can instantaneously change the velocity or angular speed. The sensor measures the displacement $e(t)$ between the center of the vehicle and the center of the track using an array of photodiodes. As the AGV passes over the track, the diode directly above the track generates more current than the other diodes. If the vehicle is close enough to the track, it will move forward. When the vehicle strays too far to the left, it will steer to the right; and when the vehicle strays too far to the right, it will steer to the left. The vehicle can be stopped at any time as long as it receives the control signal. If the vehicle is too far away from the track that it is difficult to follow the track, then it moves backward.

3. Hybrid I/O Automata Framework

In this section, we first introduce some basic notions about the model we use and then consider the definitions and theories of hybrid automata, hybrid I/O automata, and their operations [4]. More detailed discussion of the hybrid I/O automata can be found in [4].

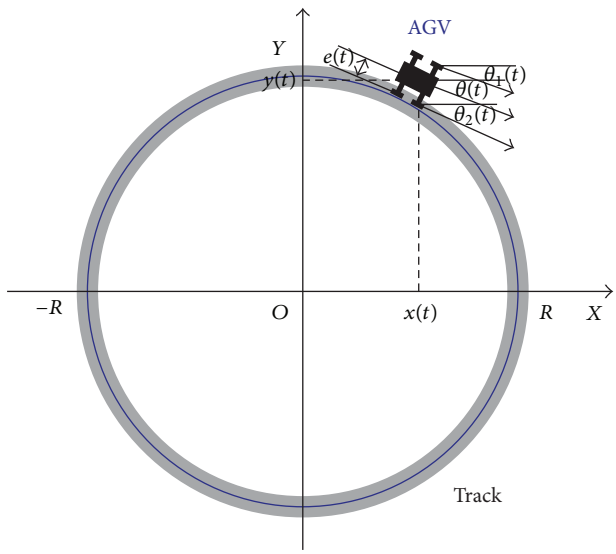

Figure 1: AGV tracking circular painted orbit.

3.1. Basic Notions. Hybrid behaviors, including discrete behavior, continuous behavior, and information flows into the system, are often described using static and dynamic variables, trajectories, and hybrid sequences. First, we introduce several basic notions involved in hybrid behavior. A location of the internal state of a system or a location of a connection between a component of a system and a component of another system can be represented as a variable, which may be static in type and denote a set of values of the variable, or dynamic in type and indicate a set of trajectories of the variable. A set of variables can be changed by discrete transitions, which are taken via discrete actions when they are enabled or by trajectories over a time interval. A hybrid sequence represents a series of changes that occur instantaneously along with the evolution of time and may be finite or infinite.

3.2. Hybrid Automata. As hybrid I/O automata are an extension of hybrid automata, we define the structure of hybrid automata first in order to describe the hybrid I/O automata. The definition of hybrid automata is given below, where $\triangleq$ denotes mathematical definition. For a more detailed description, see [4].

Definition 1. A hybrid automaton (HA) is an eight-tuple $H_M = (E_V, I_V, S, s_0, E_A, I_A, D_T, T)$, where

(i) E_V is a set of external variables,

(ii) I_V is a set of internal variables, and $V \triangleq E_V \cup I_V$ is the disjunction that represent all variables,

(iii) $S \subseteq \text{val}(I_V)$ is a set of states,

(iv) $s_0 \subseteq S$ is a nonempty set of initial states,

(v) E_A is a set of external actions,

(vi) I_A is a set of internal actions, and $A \triangleq E_A \cup I_A$ is the union of E_A and I_A,

(vii) $D_T \subseteq S \times A \times S$ is a set of discrete transitions,

(viii) T is a set of trajectories for V. For every $\tau \in T$ and $t \in \mathrm{dom}(\tau)$ (domain of τ), we have $\tau(t)\lceil I_V \in S$, where $\tau(t)\lceil I_v$ is the restriction of $\tau(t)$ to I_V; that is, the function g with $\mathrm{dom}(g) = \mathrm{dom}(\tau) \cap I_v$ such that $g(t) = \tau(t)$. We require the following axioms:

(A1) for all $\tau \in T$ for all $\tau' \leq \tau\ \tau' \in T$;

(A2) for all $\tau \in T$ for all $t \in \mathrm{dom}(\tau)\ \tau \triangleright t \in T$ where $\tau \triangleright t$ denotes $((\tau\lceil[t, +\infty)) - t)$;

(A3) suppose $\tau_0, \tau_1, \tau_2, \ldots, \tau_i, \ldots$ is a sequence of trajectories in T; if τ_i is closed and $\tau_i \cdot l\mathrm{state} = \tau_{i+1} \cdot f\mathrm{state}$, where $i \in \mathbb{N}$, τ_i is not the last trajectory of the hybrid sequence, then $\tau_0 \frown \tau_1 \frown \tau_2 \cdots \in T$, where $\tau \frown \tau' \triangleq \tau \cup (\tau'\lceil(0, \infty) + \tau \cdot l\mathrm{time})$.

The execution fragment of a hybrid automaton is a hybrid sequence $\alpha = \tau_0 a_1 \tau_1 a_2, \ldots, \tau_i a_{i+1}, \ldots$, where $\tau_i \in T$, where i is a nonnegative integer and T is defined in Definition 1; and if τ_i is not the last trajectory, then $\tau_i \cdot l\mathrm{state} \xrightarrow{a_{i+1}} \tau_{i+1} \cdot f\mathrm{state}$, where $l$state represents the last state and $f$state denotes the first state. Any input trajectory of the composition can be accepted by the composition, and we say that the components of the composition are strongly compatible HIOAs. Trace is the external behavior of a hybrid I/O automaton. Concatenation represents two hybrid sequences linked together. Let and be hybrid sequences and closed, with the concatenation being denoted by $\alpha \frown \alpha' \triangleq \mathrm{init}(\alpha)(\mathrm{last}(\alpha) \frown \mathrm{head}(\alpha'))\mathrm{tail}(\alpha')$.

3.3. Hybrid I/O Automata. We described the hybrid automata above. Here, we present the behavior and structure of a HIOA. A HIOA is used to model a complex hybrid system. The discrete state of the controller can be modeled by control modes, represented as internal variables. Each mode observes an invariant condition. The internal variables can be changed in two ways: in a discrete transition or in a continuous trajectory. External variables, including input variables and output variables, are used to exchange information between two automatons. Here is the definition of a hybrid input/output automaton. For a more detailed description, see [4].

Definition 2. A hybrid I/O automaton (HIOA) is a five-tuple $A_M = (H_M, U, Y, I, O)$, where

(i) $H_M = (E_V, I_V, S, s_0, E_A, I_A, D_T, T)$ is a hybrid automaton,

(ii) $U \subseteq E_V$ is a set of input variables,

(iii) $Y \triangleq E_V \setminus U$ is a set of output variables,

(iv) I is a set of input actions,

(v) O is a set of output actions,

(vi) the following axioms are satisfied:

(A1) for all $x \in S$ for all $\alpha \in I\ \exists x' \in S$ such that $x \xrightarrow{a} x'$,

(A2) let trajs(U) denote the set of all trajectories for U, for all $x \in S$ for all $v \in \mathrm{trajs}(U)\ \exists \tau \in T$ such that $\tau \cdot f\mathrm{state} = x, \tau \downarrow U \leq v$, and either

(a) $\tau \downarrow U = v$, or

(b) τ is closed and some $l \in L$ is enabled in $\tau \cdot l$state,

where $g = \tau \downarrow U$ represents $\mathrm{dom}(g) = \mathrm{dom}(\tau)$ such that, for all $c \in \mathrm{dom}(g)$ has $g(c) = \tau(c)\lceil U$.

We further define

(i) $Z \triangleq I_V \cup Y$ is a set of variables that are locally controlled, and

(ii) $L \triangleq I_A \cup O$ is a set of actions that are locally controlled.

Typically, it is difficult to model a complex system in one shot. HIOA can decompose a hybrid system into multiple components, model the modules as HIOAs, respectively, and then compose them in the end. We introduce a very important operation to compose two HIOAs, denoted as symbol ‖. For the proof of Theorem 3 and Lemma 4, see [4].

Theorem 3. *$A_{M1} \parallel A_{M2}$ is a hybrid I/O automaton when A_{M1} and A_{M2} are strongly compatible hybrid I/O automata and $U_1 \cap Y_2 = \emptyset$.*

Another important operation is hiding external variables in HIOA. Suppose $E_V \subseteq E_{V_A}$, $B_M = \mathrm{VarHide}(E_V, A_M)$, $E_{V_B} = E_{V_A} - E_V$, and $T_{B_M} = T_{A_M} \Downarrow (V_{A_M} - E_V)$.

Lemma 4. *If A_M is a HIOA and $E_V \subseteq E_{V_{A_M}}$, then* VarHide (E_V, A_M) *is a HIOA.*

Definition 5 (simulation relations). For all states x_{A_M} and x_{B_M} of A_M and B_M, given two comparable HIOAs, from A_M to B_M there exists a simulation relation $R_S \subseteq S_{A_M} \times S_{B_M}$ (denoted as $A_M R_S B_M$) when the following three conditions are met:

(i) knowing that $x_{A_M} \in s_{0_{A_M}}$ and suppose there exists a state $x_{B_M} \in s_{0_{B_M}}$ such that $x_{A_M} R_S x_{B_M}$, where $s_{0_{A_M}}$ is the set of initial states of A_M and $s_{0_{B_M}}$ is the set of initial states of B_M;

(ii) suppose $x_{A_M} R_S x_{B_M}$ and an execution fragment of A_M; execution fragment $\alpha = \tau_0 a_1 \tau_1 a_2, \ldots, \tau_i a_{i+1}, \ldots$, meets $\alpha \cdot f\mathrm{state} = x_{A_M}$; there exists a closed execution fragment β in B_M that meets $\beta \cdot f\mathrm{state} = x_{B_M}$, $\mathrm{trace}(\beta) = \mathrm{trace}(\alpha)$, and $\alpha \cdot f\mathrm{state} R \beta \cdot f\mathrm{state}$;

(iii) suppose $x_{A_M} R_S x_{B_M}$ and an execution fragment of $A_M \alpha = \tau_0$ has $\alpha \cdot f\mathrm{state} = x_{B_M}$; there exists a closed execution fragment β in B_M that meets $\beta \cdot f\mathrm{state} = x_{B_M}$, $\mathrm{trace}(\beta) = \mathrm{trace}(\alpha)$, and $\alpha \cdot l\mathrm{state} R \beta \cdot l\mathrm{state}$.

Corollary 6. *Given two comparable HAs A_M and B_M, and a simulation from A_M to B_M denoted as $A_M R_s B_M$, then $traces_{A_M} \subseteq traces_{B_M}$.*

The proof refers to [4]. According to [18, 19], we define a bisimulation as follows.

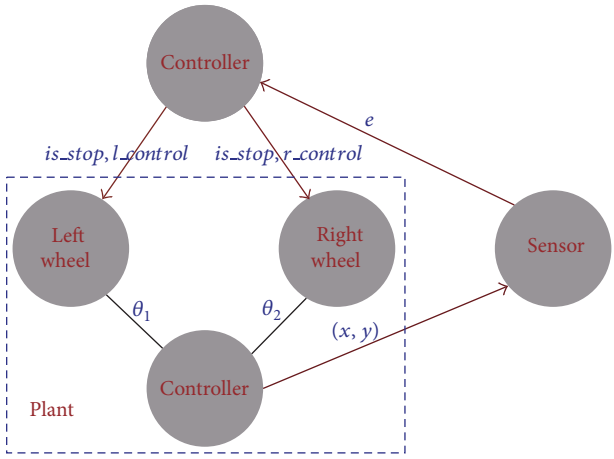

Figure 2: Network of hybrid automata for an AGV.

Definition 7 (bisimulation). Given two comparable HIOAs A_M and B_M, for all pairs (p, q) among all reachable states of A_M and B_M, p in A_M, and q in B_M. If all reachable states p^* in A_M have $p \xrightarrow{a} p^*$, this implies the existence of a state q^* in B_M such that $p \xrightarrow{a} p^*$. At the same time, all reachable states q^* in B_M have $p \xrightarrow{a} p^*$, implying that there exists a state q^* in A_M such that $p \xrightarrow{a} p^*$. Under these circumstances, we say that A_M and B_M have a bisimulation relation.

4. Modeling the AGV System

We model the AGV system using HIOA. Inspired by [2], the AGV system is modeled as a network of hybrid automata as shown in Figure 2. The model consists of five parts: chassis, left wheels, right wheels, sensor, and controller, respectively. The five components communicate via shared variables. In Figure 2, variables θ_1 and θ_2 are the angles of the left and right wheels relative to the x-axis positive direction of globe coordinate, respectively. Variables x and y represent the chassis coordinates with respect to the global coordinate frame. Variable e is the distance $e(t)$ from which the center of the AGV deviates from the center of the track at time t. The variable is used to communicate between the sensor and the controller. The controller receives the variable e, sends control signals to the left wheel and the right wheel, and then changes the mode of the AGV.

XOY is the global coordinate frame. v is the forward velocity of the car. t_{sample} is the sampling time. ω is the angular speed of the vehicle. $e(t)$ is the displacement of the center of the vehicle from the track at time t. ε_1 is the threshold indicating that the AGV is close enough to the center of the track that the AGV can move straight ahead in a forward mode. ε_2 is the threshold indicating that there is too great a distance between the center of the AGV and the center of track, and that the vehicle must therefore be steered to the other side. ε_3 is the threshold denoting that the vehicle has strayed so far from the center of track that the vehicle is in an unsafe condition and must be moved back via switching to the back mode. α is the maximum angle of vehicle velocity direction to the tangential direction of the center point on the track, where $0 \le \alpha \le \pi/2$. θ is the angle of the vehicle velocity direction to the x-axis positive direction. η is the angle of the vehicle velocity direction to the tangential direction of the center point on the track, where $-\alpha \le \eta \le \alpha$. R is the radius of the track.

The AGV system is decomposed into five components and modeled as hybrid automata: chassis, LWheel, RWheel, sensor, and controller, respectively.

4.1. Component Chassis. The chassis secures the position of each component. The state is composed of three state variables: $\langle x, y, \theta \rangle$ where x is the x-coordinate of the center of the vehicle; y is the y-coordinate of the center of the vehicle; and θ is the angle of the vehicle velocity direction to the x-axis positive direction. We use differential algebraic equations (DAEs) to describe the dynamic of the chassis. Initially, we ensure that the vehicle moves forward, and the initial condition is

$$\eta \in [-\alpha, \alpha] \wedge |e(t)| \le \varepsilon_1. \tag{1}$$

From Figure 2, the chassis secures the wheels. Hence the left wheels, right wheels, and the chassis have the same angles. We obtain the following algebraic equation:

$$\theta_1 = \theta_2 = \theta. \tag{2}$$

4.2. Component LWheel. We model the behavior of left wheels as the hybrid automaton LWheel. The left wheel has external variables: x_1, which gives the x-coordinate of the left wheel; y_1, which gives the y-coordinate of the right wheel; and θ_1, which is the angle of the moving direction of the left wheel to the x-axis positive direction. The types of these variables are real. This hybrid automaton model has no actions or discrete transitions, just satisfied trajectories. It communicates with the controller via the Boolean variable l_control and is_stop. We obtain differential equations for the left wheel as follows:

$$\begin{aligned}
\frac{dx_1}{dt} &= \textbf{if } \text{stop } \textbf{then } 0 \\
&\quad \textbf{else if } l_\text{control } \textbf{then } v\cos\theta_1 \\
&\quad \textbf{else } -v\cos\theta_1 \\
\frac{dy_1}{dt} &= \textbf{if } \text{stop } \textbf{then } 0 \\
&\quad \textbf{else if } l_\text{control } \textbf{then } v\sin\theta_1 \\
&\quad \textbf{else } -v\sin\theta_1.
\end{aligned} \tag{3}$$

4.3. Component RWheel. Since the left wheels and right wheels are symmetrical, we omit the description of the right wheels.

4.4. Component Sensor. We model the behavior of the *sensor* as the hybrid automaton sensor, whose output at time t,

for all t $e(t) = f(x(t), y(t))$, gives the center position of the AGV relative to the center of the track, shown in Figure 1. The sensor communicates with the controller through the variable e, which equals $e(t)$. Since the hybrid automaton sensor has no internal variables, and there are neither actions nor discrete transitions, only the following algebraic equation is met for the trajectories of the sensor:

$$\text{sensor} = e(t) = f(x(t), y(t)). \tag{4}$$

4.5. Component Controller. The controller can be divided into two levels. The supervisory controller determines the structure of the mode transition and guards the enabled transitions. The low-level controller determines the time-based inputs to the system. We are modeling the behavior of the controller as a hybrid automaton *controller*, whose input is the sensor value and the output the control signals to the left and right wheels that determine the operation of the wheels. There is a clock built into the controller for measuring the time interval since the last sampling. We use the variable c to represent this clock. A clock can be modeled as a first-order differential equation, and the clock variable c is defined as follows:

$$c = k \cdot t, \tag{5}$$

where k is the rate of the clock, t is the variable of time, and $k = d[c(t)]/dt$. In our model, the value of k can be a constant 1.

The controller has a variable e, which gets its value from the sensor. There are two variables recording the value of the sensor: variable new_sample, used to record the latest sample value, and variable sample, used to record the last sample value. In order to ensure that the vehicle moves forward, the initial states should satisfy:

$$c = 0 \wedge \text{sample} \in [-\epsilon_1, \epsilon_1]. \tag{6}$$

We define a transition as occurring when a guard in an outgoing transition from the current state becomes enabled. This control logic is captured in the mode transitions. The outputs are the pure signals is_stop, forward, and backward. There are three Boolean variables recording the outputs, is_stop, l_control, and r_control, respectively. We use an asterisk * to represent the next sample value. When the internal action clock transitions is taken, each state transition that is enabled will be taken:

$$c = t_{\text{sample}} \rightsquigarrow \begin{cases} c^* = 0 \\ \text{sample}^* = \text{new\_sample}, \end{cases} \tag{7}$$

where $\rightsquigarrow$ denotes the event trigger.

For trajectories we require that

$$\text{sample}(t_1) = \text{sample}(t_2), \tag{8}$$

for all time t_1, t_2 between clock transitions; that is, $t_1, t_2 \in [m \cdot t_{\text{sample}}, (m+1) \cdot t_{\text{sample}})$ for all $m \in \mathbb{N}$.

The new_sample will record the new value from the sensor:

$$\text{new\_sample} = e(t). \tag{9}$$

For every state, the following equations must hold:

$$\begin{aligned} \frac{dc}{dt} &= 1 \\ l\text{\_control} &= \textbf{if } \text{sample} \in [-\epsilon_3, \epsilon_3] \textbf{ then } \text{true} \\ &\quad \textbf{else } \text{false} \\ r\text{\_control} &= \textbf{if } \text{sample} \in [-\epsilon_3, \epsilon_3] \textbf{ then } \text{true} \\ &\quad \textbf{else } \text{false} \\ \text{is\_stop} &= \textbf{if } \text{stop} \textbf{ then } \text{true} \\ &\quad \textbf{else } \text{false}. \end{aligned} \tag{10}$$

The control logic determines the change in the state of the controller. Our AGV is running on the circular track. Since the circle is symmetrical, it suffices for us to just consider the situation of the first quadrant. The refinement of the mode gives the dynamic behavior of the output as a function of the input. We know that the displacement $e(t)$ is the function of x and y and that the control logic guards the transitions whether enabled or not. They are as follows:

$$\begin{aligned} \text{forward} &\Longrightarrow \neg\text{stop} \wedge -\epsilon_1 \le e(t) \le \epsilon_1 \\ &\Longrightarrow \begin{cases} x' = v\cos\theta, \\ y' = v\sin\theta, \\ \theta' = 0 \end{cases} \\ \text{right} &\Longrightarrow \neg\text{stop} \wedge \epsilon_3 \ge e(t) > \epsilon_2 \\ &\Longrightarrow \begin{cases} x' = v\cos\theta, \\ y' = v\sin\theta, \\ \theta' = -\omega \end{cases} \\ \text{left} &\Longrightarrow \neg\text{stop} \wedge -\epsilon_3 \le e(t) < -\epsilon_2 \\ &\Longrightarrow \begin{cases} x' = v\cos\theta, \\ y' = v\sin\theta, \\ \theta' = \omega \end{cases} \\ \text{back} &\Longrightarrow \neg\text{stop} \wedge e(t) < -\epsilon_3 \vee e(t) > \epsilon_3 \\ &\Longrightarrow \begin{cases} x' = -v\cos\theta, \\ y' = -v\sin\theta, \\ \theta' = 0 \end{cases} \\ \text{stop} &\Longrightarrow \begin{cases} x' = 0, \\ y' = 0, \\ \theta' = 0 \end{cases} \\ \text{maintain} &\Longrightarrow \begin{cases} -\epsilon_2 \le e(t) < -\epsilon_1 \quad \text{or} \\ \epsilon_1 < e(t) \le \epsilon_2. \end{cases} \end{aligned} \tag{11}$$

4.6. Composition. Since the left wheels, the right wheels, and the chassis have no output, they cannot be regarded as hybrid I/O automata. Since θ_1 and θ_2 are the internal variables of

wheels, we are modeling the three components as the hybrid I/O automaton Plant by hiding these variables. In our model, Plant, is_stop, l_control, and r_control are inputs, and x and y are outputs:

$$\begin{aligned}&\text{Plant}\\&\quad=\text{VarHide}\left(\{\theta_1,\theta_2\},(\text{Chassis}\,\|\text{LWheel}\|\,\text{RWheel})\right).\end{aligned}\tag{12}$$

Likewise, the sensor can be regarded as a hybrid I/O automaton, for which the inputs are x and y, and the output is e. The controller can also be viewed as a hybrid I/O automaton for which the input is e, and the outputs are is_stop, l_control, and r_control. According to Theorem 3 and Lemma 4, all of the components of the system are HIOAs and the composition also an HIOA. We have obtained a complete hybrid I/O automaton of the AGV system by hiding the external variables:

$$\begin{aligned}H_M=\text{VarHide}\left(\{x,y,e,\text{is\_stop},l\_\text{control},r\_\text{control}\},\right.\\\left.(\text{Plant}\,\|\text{Sensor}\|\,\text{Controller})\right).\end{aligned}\tag{13}$$

4.7. Abstraction. We expect that the AGV is always moving forward and never moves backward and use (14) to describe this situation. We select the appropriate threshold and ensure that the vehicle moves in the way we expect by specifying parameter constraints for all reachable states of the hybrid I/O automata H_M:

$$\left|\text{sample}\right|\le\epsilon_3.\tag{14}$$

In addition, we hope that the forward mode occurs infinitely often:

$$\text{GF}\left(c=0\wedge\left|\text{sample}\right|\le\epsilon_1\right).\tag{15}$$

In order to simplify the model, we abstract a model A_M from the previous model H_M. Then, we use model A_M instead of H_M. We find the constraints we need from model A_M to guarantee the correctness of the properties that we expect. Here, we simplify the model in several ways, as follows.

Based on (8), we know that the value of the variable sample remains unchanged during the interval after the current sampling and before the next sampling. Therefore we can easily prove that (14) is satisfied. We cannot consider the influence of the clock variable c. Furthermore, we assume that the vehicle is at the initial state at the time 0:

$$c=0\Longrightarrow\left|e\left(t\right)\right|\le\epsilon_3.\tag{16}$$

We find that the variables new_sample and sample are ruled out in our abstract model. Now, we use the refinements of the five modes of AGV to describe the dynamic behavior of an AGV. The formulas of the five modes are given as $\varphi_{\text{forward}},\varphi_{\text{right}},\varphi_{\text{left}},\varphi_{\text{back}}$, and φ_{stop}, respectively; φ_{step} is the disjunction of the five:

$$\begin{aligned}\varphi_{\text{step}}&\triangleq\varphi_{\text{right}}\vee\varphi_{\text{left}}\\&\quad\vee\varphi_{\text{forward}}\vee\varphi_{\text{back}}\vee\varphi_{\text{stop}}\\\varphi_{\text{forward}}&\triangleq\begin{cases}\neg\text{stop},\\\left|e\left(t\right)\right|\le\epsilon_1,\\x^*=x+v\cos\theta t_{\text{sample}},\\y^*=y+v\sin\theta t_{\text{sample}},\\\theta^*=\theta\end{cases}\\\varphi_{\text{right}}&\triangleq\begin{cases}\neg\text{stop},\\e\left(t\right)>\epsilon_2,\\x^*=x+v\cos\theta t_{\text{sample}},\\y^*=y+v\sin\theta t_{\text{sample}},\\\theta^*=\theta-\omega t_{\text{sample}}\end{cases}\\\varphi_{\text{left}}&\triangleq\begin{cases}\neg\text{stop},\\e\left(t\right)<-\epsilon_2,\\x^*=x+v\cos\theta t_{\text{sample}},\\y^*=y+v\sin\theta t_{\text{sample}},\\\theta^*=\theta+\omega t_{\text{sample}}\end{cases}\\\varphi_{\text{back}}&\triangleq\begin{cases}\neg\text{stop},\\\left|e\left(t\right)\right|>\epsilon_3,\\x^*=x-v\cos\theta t_{\text{sample}},\\y^*=y-v\sin\theta t_{\text{sample}},\\\theta^*=\theta\end{cases}\\\varphi_{\text{stop}}&\triangleq\begin{cases}\text{stop},\\x^*=x,\\y^*=y,\\\theta^*=\theta.\end{cases}\end{aligned}\tag{17}$$

Now we get the abstract model of the AGV system. Since the abstract model A omits the time variable, it is simpler than the original model. We will derive and verify the properties using the abstract model. There are two kinds of typical errors in formal verification, one is true error, where errors exist in the physical system, but the result of formal verification is correct. The reason of first kind of error is because we abstract a model from our original model, and the details we omitted may lead to the errors of original model being omitted, so we get a passing proof. The other is false error which do not exist in the physical system but the result of formal verification is incorrect. The reason is that abstract model omitted the details of original model. The abstract model cannot express the original system due to lack of information from the original system, and then the result of formal verification is incorrect.

In order to ensure the two kinds of errors never occur, we prove the original model H_M and the abstract model A_M have a bisimulation equivalence relationship.

Lemma 8. *Let S_K be a set of all reachable states of H_M, for all $s \in S_R$; one has*

$$c = 0 \Longrightarrow \left[\left|\, sample \,\right| \leq \epsilon_2 \Longleftrightarrow |e(t)| \leq \epsilon_2\right] \tag{18}$$

Proof. Use (14) and (15) to prove Lemma 8. □

Theorem 9. *The two comparable HIOAs H_M and A_M have a bisimulation relation.*

Proof. Owing to the limitation of space, we do not provide a detailed proof of Theorem 9, but the key step will be given. H_M and A_M satisfy the following condition:

$$c_{H_M} = 0 \wedge \theta_{H_M} = \theta_{A_M} \wedge x_{H_M} = x_{A_M} \wedge y_{H_M} = y_{A_M}. \tag{19}$$

For all state pairs (p, q) among all reachable states of H_M and A_M, $p \in H_M$ and $q \in A_M$, if states of state pair (p, q) hold the weakest condition of labeled transition system respectively, we say the pair (p, q) is bisimulation equivalent. If each initial state of H_M bisimulates an initial state of A_M, and there exists an execution fragment from p to p^*, where p^* in H_M has $p \rightarrow p^*$, implying the existence of a transition according to the transition predicate φ_{step} of A_M from q to q^* in A_M, such that $q \Rightarrow q^*$. At the same time, there exists a transition according to the transition predicate φ_{step} of A_M from q to q^*, where q^* in A_M has $q \Rightarrow q^*$, implying the existence of an execution fragment from p to p^*, where p^* in H_M, such that $p \rightarrow p^*$. We can then use Definition 5, Corollary 6, and Definition 7 to prove Theorem 9. □

5. Correctness

5.1. The Desired Properties of A_M. For a system, we often hope that bad things will never happen, a situation called safety, that good things will eventually happen, and that they will happen infinitely often, a situation called fairness. We express the properties via invariants. For our system, we expect the displacement from the center of the AGV to the center of the track to never be larger than the threshold ε_3, and never be less than the threshold $-\varepsilon_3$. At the same time, we ensure that η lies in the interval $[-\alpha, \alpha]$.

Property 1. The vehicle always moves forward and never moves backward. It can be described as

$$\varphi_{\text{safety}} \triangleq \eta \in [-\alpha, \alpha] \wedge |e(t)| \leq \varepsilon_3. \tag{20}$$

Property 2. The vehicle moves forward infinitely often. It can be described using the temporal logic formula

$$\varphi_{\text{fairness}} \triangleq \text{GF}\, |e(t)| \leq \varepsilon_1. \tag{21}$$

Lemma 10. *If φ_{safety} is an invariant of A_M and formula $\varphi_{fairness}$ holds for A_M, then* (14) *is an invariant of H_M and* (15) *holds for H_M.*

Proof. Use Lemma 8, Theorem 9, and (8) to prove Lemma 10. □

5.2. Parameter Constraints of the AGV System. In this section, we will give several parameter constraints for our AGV system. They are indispensable to guaranteeing the correctness of the properties of safety (20) and fairness (21). We define them in (22).

Parameter Constraints. Consider the following

$$\begin{aligned}
\varphi_1 &\triangleq \left(v\cos\alpha t_{\text{sample}}\right)^2 + \left(R + \varepsilon_2 + v\sin\alpha t_{\text{sample}}\right)^2 \\
&\quad \leq (\varepsilon_3 + R) \\
\varphi_2 &\triangleq (R - \varepsilon_2)^2 + v^2 t_{\text{sample}}^2 - 2(R - \varepsilon_2)\, v t_{\text{sample}} \sin\alpha \\
&\quad \geq (R - \varepsilon_3)^2 \\
\varphi_3 &\triangleq \omega t_{\text{sample}} \leq \alpha \\
\varphi_4 &\triangleq (R - \varepsilon_3)^2 \\
&\quad \leq \left(v\cos\alpha t_{\text{sample}}\right)^2 + \left(R - \varepsilon_2 + v\sin\alpha t_{\text{sample}}\right)^2 \\
&\quad \leq (R + \varepsilon_1)^2.
\end{aligned} \tag{22}$$

Theorem 11. *If φ_1, φ_2, and φ_3 are met, then the φ_{safety} property is an invariant of A_M; that is,*

$$\varphi_{safety} \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \Longrightarrow \varphi^*_{safety}. \tag{23}$$

Proof. In the first step, we prove that $\varphi_{\text{safety}} \wedge \varphi_1 \Rightarrow 0 \leq e^*(t) \leq \varepsilon_3$ holds. Since the circle is symmetrical, we only need to consider the situation of the first quadrant. In order to guarantee that $|e(t)| \leq \varepsilon_3$ is met in all cases of outside the circle track, we consider the most extreme case of the outside of the circle. First of all, suppose that the vehicle moves on the outside of the circle shown in Figure 3. The vehicle is very close to point A at the time of the current sampling, and $e(t) \leq \varepsilon_2$. The vehicle then moves forward to B with $\eta = \alpha$ at the next sampling; the $e(t)$ reaches the largest displacement. We use φ_1 to illustrate that $0 < e(t) \leq \varepsilon_3$ holds for A_M. The derivations in (24) show that $e^*(t) \leq \varepsilon_3$.

Deriving from φ_1. Consider the following:

$$\begin{aligned}
&\left(v\cos\alpha t_{\text{sample}}\right)^2 + \left(R + \varepsilon_2 + v\sin\alpha t_{\text{sample}}\right)^2 \leq (\varepsilon_3 + R)^2 \\
&\sqrt{\left(v\cos\alpha t_{\text{sample}}\right)^2 + \left(R + \varepsilon_2 + v\sin\alpha t_{\text{sample}}\right)^2} \leq (\varepsilon_3 + R) \\
&\sqrt{\left(v\cos\alpha t_{\text{sample}}\right)^2 + \left(R + \varepsilon_2 + v\sin\alpha t_{\text{sample}}\right)^2} - R \leq \varepsilon_3 \\
&\sqrt{|BD|^2 + |OD|^2} - R \leq \varepsilon_3 \quad \text{(use the Pythagorean theorem)} \\
&\sqrt{|BD|^2} - R \leq \varepsilon_3 \\
&|OB| - R \leq \varepsilon_3 \\
&e^*(t) \leq \varepsilon_3.
\end{aligned} \tag{24}$$

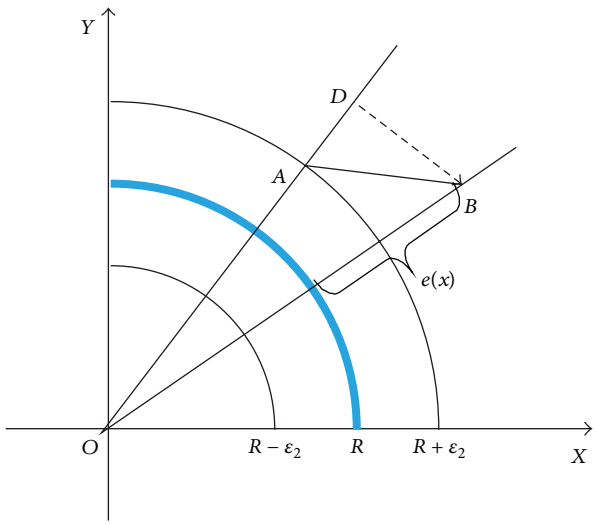

Figure 3: Illustration of the need for φ_1.

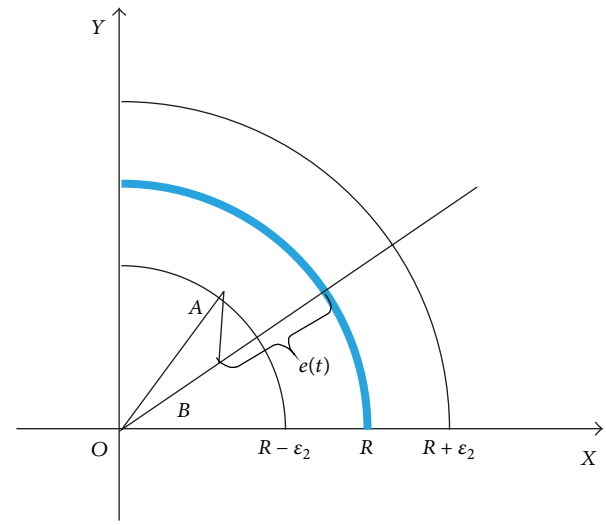

Figure 4: Illustration of the need for φ_2.

Since the vehicle moves on the outside of the circle, $e(t)^* > 0$. Therefore, $\varphi_{\text{safety}} \wedge \varphi_1 \Rightarrow 0 < e^*(t) \le \varepsilon_3$.

In the second step, we prove that $\varphi_{\text{safety}} \wedge \varphi_2 \Rightarrow -\varepsilon_3 \le e^*(t) \le 0$ holds. In order to guarantee that $|e(t)| \le \varepsilon_3$ in all cases inside the track of the circle, we consider the most extreme case inside. First of all, suppose that the vehicle moves on the inside of the circle shown in Figure 4. The vehicle is very close to point A at the time of the current sampling, with $e(t) > -\varepsilon_2$, and then moves forward to B with $\eta = -\alpha$ at the next sampling, where the vehicle reaches the farthest to the track. We use φ_2 to illustrate that $-\varphi_3 \le e(t) < 0$ holds for A_M. The derivations in (25) show that $e^*(t) \ge -\varepsilon_3$.

Deriving from φ_2. Consider the following:

$$\begin{aligned}
&(R-\varepsilon_2)^2 + v^2 t_{\text{sample}}^2 - 2(R-\varepsilon_2)\, v t_{\text{sample}} \sin\alpha \ge (R-\varepsilon_3)^2 \\
&\sqrt{(R-\varepsilon_2)^2 + v^2 t_{\text{sample}}^2 - 2(R-\varepsilon_2)\, v\, |\sin(-\alpha)|\, t_{\text{sample}}} \\
&\quad \ge R - \varepsilon_3 \\
&\sqrt{|OA|^2 + |AB|^2 - 2\,|OA\,\|AB\|\sin(-\alpha)|} \ge R - \varepsilon_3 \\
&\sqrt{|OA|^2 + |AB|^2 - 2\,|OA|\,|AB|\cos\left(\frac{\pi}{2} - \alpha\right)} \ge R - \varepsilon_3 \\
&\qquad\qquad \text{(use the Law of cosines)} \\
&|OB| \ge R - \varepsilon_3 \\
&OB - R \ge -\varepsilon_3 \\
&e^*(t) \ge -\varepsilon_3.
\end{aligned} \tag{25}$$

Since the vehicle moves on the inside of the circle, $e^*(t) < 0$. Therefore, $\varphi_{\text{safety}} \wedge \varphi_2 \Rightarrow -\varepsilon_3 \le e^*(t) \le 0$.

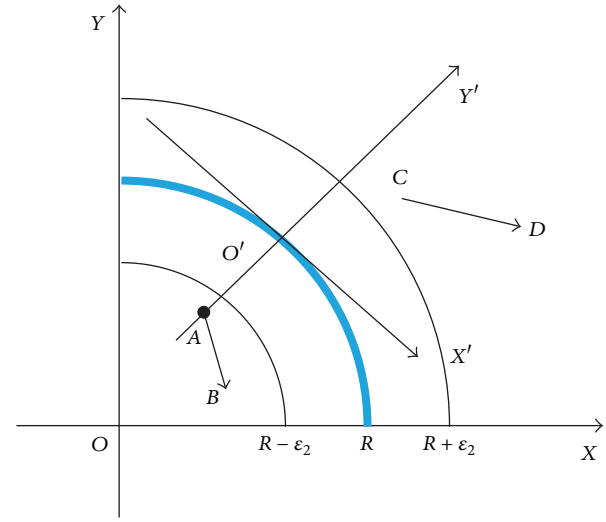

Figure 5: Illustration of the need for φ_3.

In the third step, we prove that $\varphi_{\text{safety}} \wedge \varphi_3 \Rightarrow \eta^* \in [-\alpha, \alpha]$ holds. Constraint $\omega t_{\text{sample}} \le \alpha$ is required to guarantee that η is always in the interval $[-\alpha, \alpha]$. First of all, we consider the scenario shown in Figure 5. We build a coordinate frame $X'O'Y'$ shown in Figure 5. If the vehicle reaches point A in the current sampling, the vehicle will steer to the left. The angle $\eta < 0$ (the angle between $\overrightarrow{AB}$ and $\overrightarrow{O'X'}$), relative to the coordinate frame $X'O'Y'$, is $\eta^* = \eta + \omega t_{\text{sample}} < \omega t_{\text{sample}}$ at the next sampling. If $\omega t_{\text{sample}} \le \alpha$, then $\eta^* = \eta + \omega t_{\text{sample}} < \alpha$. If the vehicle reaches point C in the current sampling, the vehicle will steer to the left. The angle $\eta > 0$ (the angle between $\overrightarrow{CD}$ and $\overrightarrow{O'X'}$), relative to the coordinate frame $X'O'Y'$, is $\eta^* = \eta - \omega t_{\text{sample}} > -\omega t_{\text{sample}}$ at the next sampling. If $\omega t_{\text{sample}} \le \alpha$, then $-\omega t_{\text{sample}} \ge -\alpha$, and we will get $\eta^* = \eta - \omega t_{\text{sample}} > -\alpha$. We have proved that $\eta^* \in [-\alpha, \alpha]$.

Therefore, $\varphi_{\text{safety}} \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \Rightarrow \varphi^*_{\text{safety}}$ is proved. □

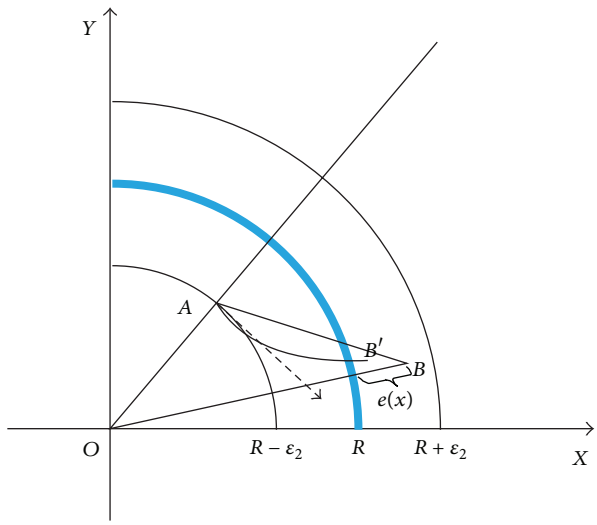

FIGURE 6: Illustration of the need for φ_4.

Theorem 12. *If φ_4 holds for A_M, then the property of $\varphi_{fairness}$ is an invariance of A_M; that is, $\varphi_{fairness} \wedge \varphi_4 \Rightarrow \varphi^*_{fairness}$.*

Proof. In order to ensure that the vehicle moves forward infinitely often, we avoid the situation of always steering to left after steering to right, and steering to right after steering to left. We consider the scenario shown in Figure 6. The center of the vehicle is very close to point A in the current sampling, and $e(t) < \varepsilon_2$, with the velocity direction approximate parallel to the direction of the tangent of point A, shown as a dashed line. The vehicle steers to the left, moves along the arc $\overset{\frown}{AB}$, and we look at $\overset{\frown}{AB}$ as a straight line AB. Suppose that $\eta = \alpha$, $e(t) < \varepsilon_1$ at the next sampling and that the vehicle switches to the forward mode. From φ_4, we can derive the following:

$$\begin{aligned} R - \varepsilon_1 \le |OB| &\le R + \varepsilon_1 \left(R > \varepsilon_3 > \varepsilon_2 > \varepsilon_1\right) \\ -\varepsilon_1 \le |OB| - R &\le -\varepsilon_1 \\ -\varepsilon_1 \le e^*(t) &\le -\varepsilon_1 \\ \left|e^*(t)\right| &\le -\varepsilon_1. \end{aligned} \tag{26}$$

□

6. Analysis of Constraints

In this section, we analyze the parameters of our AGV system. We rewrite the constraints as shown in (27).

Rewrite Constraints. Consider the following:

$$\begin{aligned} \varphi_1 &\triangleq v^2 t^2_{\text{sample}} + \varepsilon_2^2 + 2\varepsilon_2 v \sin\alpha t_{\text{sample}} \\ &\quad \le \varepsilon_3^2 + 2R\left(\varepsilon_3 - \varepsilon_2 - v\sin\alpha t_{\text{sample}}\right) \\ \varphi_2 &\triangleq \left(R - \varepsilon_2 - v t_{\text{sample}}\right)^2 + 2\left(R - \varepsilon_2\right) v t_{\text{sample}} \left(1 - \sin\alpha\right) \\ &\quad \ge \left(R - \varepsilon_3\right)^2 \\ \varphi_3 &\triangleq \omega t_{\text{sample}} \le \alpha. \end{aligned} \tag{27}$$

We assume that the value range of v is from $v_{\min}$ to $v_{\max}$, t_{sample} is from $t_{\min}$ to $t_{\max}$, α is from $\alpha_{\min}$ to $\alpha_{\max}$, ε_1 is from $\varepsilon_{1_{\min}}$ to $\varepsilon_{1_{\max}}$, ε_2 is from $\varepsilon_{2_{\min}}$ to $\varepsilon_{2_{\max}}$, ε_3 is from $\varepsilon_{3_{\min}}$ to $\varepsilon_{3_{\max}}$, R is from $R_{\min}$ to $R_{\max}$, and ω is from $\omega_{\min}$ to $\omega_{\max}$. The inequalities shown in (28) need to be met to ensure that the parameter constraints hold.

Inequalities Needed for the Parameter Constraints. Consider the following:

$$\begin{aligned} &v^2_{\max} t^2_{\max} + \varepsilon^2_{2_{\max}} + 2\varepsilon_{2_{\max}} v_{\max} \sin\alpha_{\max} t_{\max} \\ &\quad \le \varepsilon^2_{3_{\min}} + 2R_{\min}\left(\varepsilon_{3_{\min}} - \varepsilon_{2_{\max}} - v_{\max}\sin\alpha_{\max} t_{\max}\right) \\ &\left(R_{\max} - \varepsilon_{3_{\min}}\right)^2 \le \left(R_{\min} - \varepsilon_{2_{\max}} - v_{\max} t_{\max}\right)^2 \\ &\quad + 2\left(R_{\min} - \varepsilon_{2_{\max}}\right) v_{\min} t_{\min}\left(1 - \sin\alpha_{\max}\right) \\ &\omega_{\max} t_{\max} \le \alpha_{\min} \\ &\left(R_{\max} - \varepsilon_{1_{\min}}\right)^2 \le v^2_{\min} t^2_{\text{sample}} + \left(R_{\min} - \varepsilon_{2_{\max}}\right)^2 \\ &\quad + 2\left(R_{\min} - \varepsilon_{2_{\max}}\right) v_{\min} \sin\alpha_{\min} t_{\min} \\ &\left(R_{\min} + \varepsilon_{1_{\min}}\right)^2 \ge v^2_{\max} t^2_{\max} + \left(R_{\max} - \varepsilon_{2_{\min}}\right)^2 \\ &\quad + 2\left(R_{\max} - \varepsilon_{2_{\min}}\right) v_{\max} \sin\alpha_{\max} t_{\max}. \end{aligned} \tag{28}$$

It is obvious that the parameters $\varepsilon_{1_{\max}}, \varepsilon_{3_{\max}}$, and $\omega_{\min}$ do not appear in the constraint inequalities. Therefore, we increase ε_1 and ε_3 from the minimum and decrease ω from the maximum. We do not know the exact values of such parameters as $v, t_{\text{sample}}, \omega$, and can measure their values only by operating the vehicle. Errors cannot be avoided when we obtain these parameters. We can write the predicate logic formula asserting safety φ_{safety} as follows:

$$\begin{aligned} &\forall\alpha \in \left[\alpha_{\min}, \alpha_{\max}\right], \\ \forall\varepsilon_1 \ge \varepsilon_{1_{\min}}, \quad &\forall\varepsilon_2 \in \left[\varepsilon_{2_{\min}}, \varepsilon_{2_{\max}}\right], \quad \forall\varepsilon_3 \ge \varepsilon_{3_{\min}} \\ &\forall R \in \left[R_{\min}, R_{\max}\right]: \\ &\varphi_{\text{safety}} \wedge \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \Longrightarrow \varphi^*_{\text{safety}}. \end{aligned} \tag{29}$$

Parameters v and ω can be viewed as the internal variables of the vehicle.

7. Conclusion

In this paper, we have modeled an AGV system using a hybrid I/O system and investigated a two-dimensional problem where the vehicle moves in a circular orbit. We derived and proved the constraints of the parameters of the AGV system so that the vehicle always move forward closely following the circular track and never moves backward. We have also analyzed the constraints of the parameters and the range of the parameters. Future research can extend this formulation

from circular track to arbitrary complex curves, consider slopes or hilly terrains, and reason about multiple vehicle systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research for this study was supported by the NSFC (61373034, 61170304) and the International S&T Cooperation Program of China (2011DFG13000) (KZ201210028036).

References

[1] M. S. Branicky, "Introduction to hybrid systems," in *Handbook of Networked and Embedded Control Systems*, D. Hristu-Varsakelis and W. S. Levine, Eds., Control Engineering, pp. 91–116, Birkhäuser Boston, Boston, Mass, USA, 2005.

[2] A. Fehnker, F. Vaandrager, and M. Zhang, "Modeling and verifying a Lego car using hybrid I/O automata," in *Proceedings of the 3rd International Conference on Quality Software (QSIC '03)*, pp. 280–289, IEEE Computer Society, 2003.

[3] L. Balbis, A. W. Ordys, M. J. Grimble, and Y. Pang, "Tutorial introduction to the modelling and control of hybrid systems," *International Journal of Modelling, Identification and Control*, vol. 2, no. 4, pp. 259–272, 2007.

[4] N. Lynch, R. Segala, and F. Vaandrager, "Hybrid I/O automata," *Information and Computation*, vol. 185, no. 1, pp. 105–157, 2003.

[5] T. A. Henzinger, "Theory of hybrid automata," in *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS '96)*, pp. 278–292, IEEE Computer Society, July 1996.

[6] J. Lygeros, G. Pappas, and S. Sastry, "An introduction to hybrid systems modeling, analysis and control," in *Proceedings of the 1st Nonlinear Control Network Pedagogical School*, pp. 307–329, Athens, Greece, 1999.

[7] A. Balluchi, L. Benvenuti, M. D. D. I. Benedetto, S. Member, C. Pinello, and A. Luigi, "Automotive engine control and hybrid systems: challenges and opportunities," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 888–912, 2000.

[8] R. Alur, R. Grosu, Y. Hur, V. Kumar, and I. Lee, "Modular specification of hybrid systems in charon," in *Hybrid Systems: Computation and Control*, N. Lynch and B. Krogh, Eds., vol. 1790 of *Lecture Notes in Computer Science*, pp. 6–19, Springer, Berlin, Germany, 2000.

[9] M. Song, T. Tarn, and N. Xi, "Integration of task scheduling, action planning, and control in robotic manufacturing systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1097–1107, 2000.

[10] M. Katara, "Hybrid models for mobile computing," in *Coordination Languages and Models*, A. Porto and G. C. Roman, Eds., vol. 1906 of *Lecture Notes in Computer Science*, pp. 216–231, Springer, Berlin, Germany, 2000.

[11] B. Lennartson, M. Tittus, B. Egardt, and S. Pettersson, "Hybrid systems in process control," *IEEE Control Systems Magazine*, vol. 16, no. 5, pp. 45–56, 1996.

[12] N. A. Lynch and M. R. Tuttle, "An introduction to input/output automata," *CWI Quarterly*, vol. 2, no. 3, pp. 219–246, 1989.

[13] T. Muller, *Automated Guided Vehicles*, IFS, Kempston, UK, 1983.

[14] T. Le-Anh and M. B. M. de Koster, "A review of design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 171, no. 1, pp. 1–23, 2006.

[15] I. F. A. Vis, "Survey of research in the design and control of automated guided vehicle systems," *European Journal of Operational Research*, vol. 170, no. 3, pp. 677–709, 2006.

[16] W. Kang, N. Xi, and J. Tan, "Analysis and design of non-time based motion controller for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '99)*, pp. 2964–2969, May 1999.

[17] J. Tan, N. Xi, and W. Kang, "Non-time based tracking controller for mobile robots," in *Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 919–924, May 1999.

[18] R. Milner, *Communication and Concurrency*, Prentice-Hall, 1989.

[19] D. Park, "Concurrency and automata on infinite sequences," in *Theoretical Computer Science*, P. Deussen, Ed., vol. 104 of *Lecture Notes in Computer Science*, pp. 167–183, Springer, Berlin, Germany, 1981.