PAPER Special Section on Selected Papers from the 16th Workshop on Circuits and Systems in Karuizawa

Safety Verification of Material Handling Systems Driven by Programmable Logic Controller——Consideration of Physical Behavior of Plants——

Eiji KONAKA[†a)], *Nonmember*, **Tatsuya SUZUKI**[†], *Member*, *and* **Shigeru OKUMA**[†], *Nonmember*

SUMMARY The PLC (Programmable Logic Controller) has been widely used in the industrial world as a controller for manufacturing systems, as a process controller and so on. The conventional PLC has been designed and verified as a pure Discrete Event System (DES) by using an abstract model of a controlled plant. In verifying the PLC, however, it is also important to take into account the physical behavior (e.g. dynamics, shape of objects) of the controlled plant in order to guarantee such important factors as safety. This paper presents a new verification technique for the PLC-based control system, which takes into account these physical behaviors, based on a Hybrid Dynamical System (HDS) framework. The other key idea described in the paper is the introduction of the concept of signed distance which not only measures the distance between two objects but also checks whether two objects interfere with each other. The developed idea is applied to illustrative material handling problems, and its usefulness is demonstrated.

key words: *programmable logic controller, safety verification, hybrid dynamical system*

1. Introduction

The Programmable Logic Controller (PLC) has been widely used in the industrial world as a controller for manufacturing systems, for process control and so on. The PLC provides for a low-cost system since it uses binary sensors and actuators, and only the logical relationships between sensors and actuators are implemented on the PLC. The Ladder Diagram (LD) and Sequential Function Chart (SFC) are widely used as the standard programming language.

Before implementing control logic on the PLC, it is necessary to verify whether the control requirements can be met. In the verification of the control logic, not only the programmed control logic but also a model of the controlled plant should be included. From this point of view, automata and/or temporal logic [1] have been used to model the behavior of the plant. In these modeling methods, the plant is modeled as a Discrete Event System (DES), and the closed loop behavior is also treated as a DES. In [5], a hybrid system is abstracted by means of a DES and control specifications are verified using the abstracted system. This method, however, requires a huge amount of computation, and it cannot be applied to practical problems. Although a certain type of specification (e.g., specification for the sequence of actions) can be verified in this framework, the verification is likely to be conservative since the information regarding the physical behavior of the plant is ignored. Thus, in order to develop a verification algorithm available to real industrial systems, it is important to take into account the physical behavior of the plant. When we look at the problem of verifying the safety of material handling systems, the physical behavior of the plant consists of the dynamics of the manipulator and configurations of components. Dynamics is necessary to predict the motion of the objects of a plant, and configurations are necessary to check the interference between objects.

In the field of robotics, the C-Space approach [6] is well known as one of the methods for checking for interference. Since this approach calculates safe regions based on geometrical information (not based on algebraic expressions), it thus requires a large amount of computation. Moreover, only geometrical information is considered, and the dynamic information regarding the controlled plant is not utilized. In order to resolve this problem, in this paper, quantitative measures to ensure safety, which can be easily calculated, is defined and dynamical information is utilized in order to reduce the amount of computation.

Based on these considerations, a safety verification algorithm for material handling manipulators is developed. In the proposed framework, first, the dynamics of the plant is modeled as a time-driven difference equation, and the closed loop behavior is regarded as a hybrid dynamical system [2] which consists of a time-driven system and the DES. Secondly, the shape of each link of handling manipulators are considered in the verification process since they are regarded and modeled as rigid bodies. This framework can capture the physical behavior of the plant and allows for a more sophisticated verification, and as a result, enables us to guarantee such important properties as safety.

In order to develop an engine for verifying safety, first, the degree of danger is defined based on the distance between the components of the handling manipulators and a dangerous region. In order to reduce the computation burden, all components and dangerous regions are outer approximated by means of convex polyhedra. (Generally, the distance between nonconvex sets is hard to calculate.) Secondly, the signed distance between polyhedra is defined as a quantitative measure of safety. A computationally effective safety checking algorithm is developed by exploiting the

Manuscript received June 26, 2003.
Manuscript revised September 26, 2003.
Final manuscript received November 20, 2003.

[†]The authors are with the Graduate School of Engineering, Nagoya University, Nagoya-shi, 464-8603 Japan.
a) E-mail: konaka@okuma.nuee.nagoya-u.ac.jp

property of the signed distance. The proposed algorithm can achieve a more sophisticated control verification that cannot be achieved by means of the conventional DES-based framework. The developed idea is applied to an illustrative example of the control of material handling manipulators, and its usefulness is verified.

2. Verification of PLC Based on DES Framework

2.1 PLC

The PLC is a controller in which inputs and outputs are specified by a binary signal. Only logical relationships between them are implemented on it (Fig. 1). There are several program languages for the implementation on the PLC. Among them, LD and SFC have been widely used so far. In the following, these languages are briefly reviewed with simple illustrative examples.

2.1.1 Ladder Diagram (LD)

LD represents the logical relationship between sensors and actuators. Figure 2 shows an example of LD.

In Fig. 2, Y_1, Y_2, Y_3 and U_1, U_2 represent the states of sensors and actuators, respectively. The control logic of this diagram is $U_1 = (Y_1 + Y_3) \cdot \bar{Y}_2$, $U_2 = (Y_2 + U_2) \cdot Y_3$. (Note that $\cdot$, $+$ and $\bar{}$ represent AND, OR, and NOT, respectively.) The right side of these equations is called the firing condition. Each output U_i turns ON if and only if the corresponding firing condition turns ON (true). In this case, for example, actuator U_2 is ON if and only if both Y_2 and Y_3 are ON. Once U_2 is turned ON, it maintains its state while Y_3 is ON. This kind of output is called self-maintenance. Self-maintenance acts as a kind of memory in the LD, and appears frequently in practical applications. Outputs hold their value while inputs do not change. If inputs change their value, an input event occurs. Outputs change in synchronization with the input event. (Similarly, the change of output is called the output event.)

Note that the LD describes only logical relationships between inputs (sensors) and outputs (actuators). The precedence relationships between the outputs are not explicitly described.

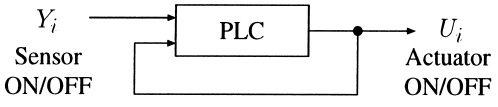


Fig. 1 Programmable logic controller (PLC).

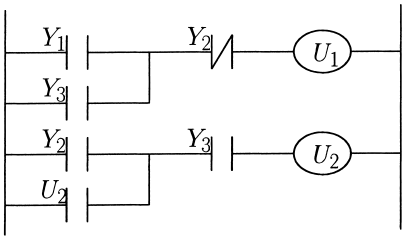


Fig. 2 Example of ladder diagram.

2.1.2 SFC

SFC is a programming language which originates from Petri Nets. Figure 3 shows an example of the SFC. The main components of the SFC are steps, actions, and transitions. Steps (depicted by rectangles) correspond to the states of the closed loop system. Actions (ellipses) represent the fired outputs at each step. Transitions (horizontal bars) represent the switch between the two steps which is triggered by the assigned input logic. At the start of the control process, all steps are OFF except the initial step (double rectangle). If the input condition associated with the transition turns ON, then the state of the SFC switches. For example, in Fig. 3, suppose that the state of the SFC is at Step 0. Then the state of the SFC switches to Step 1 after Tr 1 turns ON (i.e., the sensor Y_1 turns ON.), and the actuator U_1 turns ON while Step 1 is ON.

Moreover, SFC can express parallel path divergence and single path divergence.

Figure 4 shows an example of the parallel path divergence (depicted by double horizontal bar). If the state is at the Step 1 and Tr 1 turns ON, then Step1 turns OFF, and Step 2 and Step 3 turn ON simultaneously. Figure 5 shows an example of single path divergence (single horizontal bar). In single path divergence, only one sequence is executed depending on which transition turns ON. If the state is at Step 1 and Tr 1b turn ON, then Step 1 turns OFF and Step 3 turns

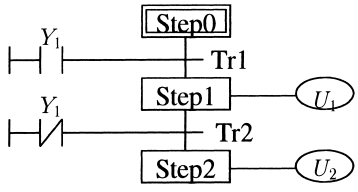


Fig. 3 Example of SFC.

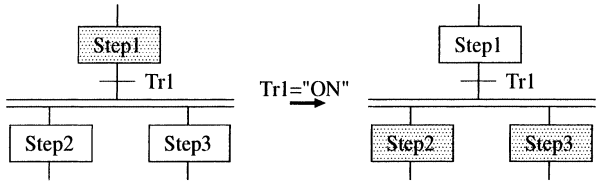


Fig. 4 Example of parallel path divergence.

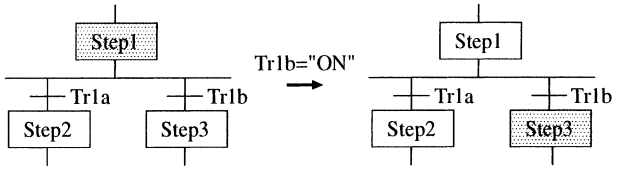


Fig. 5 Example of single path divergence.

ON.

Obviously, the SFC can explicitly describe the control sequence, different from the LD.

2.2 DES Model-Based Verification

In verifying the operation of the PLC, model of the plant is inevitable. The plant has been conventionally modeled as DES, and a closed loop system has been also as DES (Fig. 6). In this framework, only a "logical" specification (e.g., a sequence or precedence relationship) can be verified since the closed loop expression includes only information as pure DES. For example, in [1], a controlled plant was modeled by means of Temporal Logic [3], and the LD was then coupled with the plant model. Based on this closed loop structure, an algorithm which extracts the order of events has been proposed. As a result, the specifications of the precedence relationship between actions has been verified.

Although a DES-based plant model is consistent with PLC, information regarding the physical behavior of the plant is lost and only the symbolic aspects of the plant can be focused on. Thus, the physical behavior of the plant between successive output events is completely ignored in this model. Hence, two behaviors which generate the same output events in the same order may not be distinguished. For example, in Fig. 7, both trajectories (a) and (b) which generate same sensor output sequence (Y_2 is ON after Y_1 is ON) are regarded as the same behavior in the DES model.

In the field of computer science, verification has been usually considered based on the DES model. For example, in [5], a hybrid system was abstracted by means of a DES, and control specifications were verified using the abstracted model. In this method, however, a huge number of discrete states are generated in order to maintain the model's accuracy, and very large amount of computations are required to solve practical problems. In contrast to this method, the proposed algorithm needs no discrete abstraction and can be performed in a reasonable time.

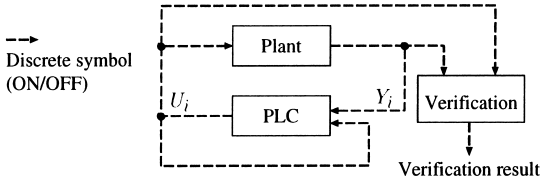


Fig. 6 DES based verification.

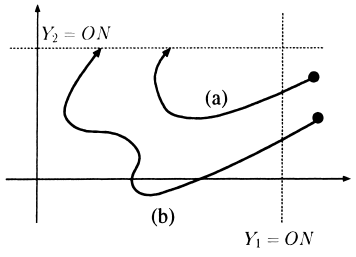


Fig. 7 Indistinguishable physical behaviors in DES model.

3. Safety Verification of PLC

3.1 Problem Formulation as Hybrid System

The mechanical system of an actual plant consists of rigid bodies, and the safety of the plant must be ensured by preventing these rigid bodies from interfering with one another. If the designer tries to verify a system's safety based on the DES model, the result tends to be conservative, and other important criteria such as completion time is sacrificed. In order to avoid this, we propose that the plant is modeled as a time-driven difference equation, with the closed loop behavior regarded as a so-called Hybrid Dynamical System (HDS) [2] as shown in Fig. 8. Also, a safety verification algorithm is developed based on the HDS model.

In our model, the controlled plant is modeled using the following difference equations:

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{A}(\boldsymbol{u}(k))\,\boldsymbol{x}(k) + \boldsymbol{B}(\boldsymbol{u}(k))\,\boldsymbol{u}(k) \\ \boldsymbol{y}(k) = \boldsymbol{f}(\boldsymbol{x}(k)) \end{cases} \tag{1}$$

State $\boldsymbol{x}$ represents a collection of physical variables of the plant (position, velocity, etc.) and $\boldsymbol{y}$ represents the output of the plant. Output is quantized into binary values (ON/OFF) through the sensors. PLC determines the binary control input (ON/OFF of the actuators) according to the programmed control logic. The binary control input is then transformed into the physical input for the plant $\boldsymbol{u}$ that evolves the state of the plant.

By regarding the system as HDS, the validity of the following items can be verified from the viewpoint of safety.

- Control logic
- Sensor position
- Physical parameters (e.g., control input, size of component, etc.)

Let the output of the plant be the position and orientation of the rigid body, and $S_{state}(\boldsymbol{y}(k)) \subset \Re^3$ be a region which is occupied by them. In verifying the safety of the system, it is necessary to investigate the behavior of $S_{state}(\boldsymbol{y}(k))$. Suppose that dangerous regions are denoted by $S_{danger}(k)$. Hence, the system is safe if and only if

$$\forall k \; S_{state}(\boldsymbol{y}(k)) \cap S_{danger}(k) = \emptyset. \tag{2}$$

Also, the distance between $S_{state}(\boldsymbol{y}(k))$ and $S_{danger}(k)$ can

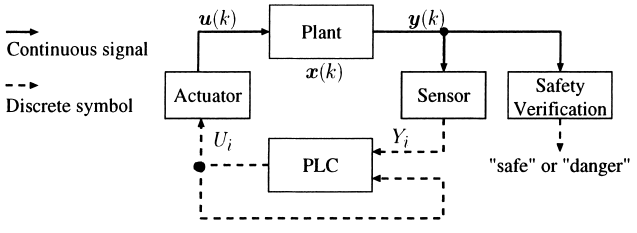


Fig. 8 HDS based verification.

be the quantitative measure for the safety. Since the distance between nonconvex sets is computationally cumbersome and hard to calculate, all objects and dangerous regions are outer approximated by means of convex polyhedra.

Definition 1: $P_{state}(\boldsymbol{y}(k))$ and $P_{danger}(k)$ are convex polyhedra, which are outer approximations of $S_{state}(\boldsymbol{y}(k))$ and $S_{danger}(k)$, respectively.

3.2 Quantitative Measure for Safety

In this section, we define a signed distance between convex polyhedra as the quantitative measure for safety. The signed distance is easy to calculate using linear programming, and its sign indicates the existence of an intersection.

3.2.1 Signed Distance between Convex Polyhedra

Suppose that two polyhedra $P^{(1)}$ and $P^{(2)}$ are given by

$$P^{(1)} = \{\boldsymbol{x} | \boldsymbol{m}^{(1)}\boldsymbol{x} \leqq \boldsymbol{p}^{(1)}\} \tag{3}$$

$$P^{(2)} = \{\boldsymbol{x} | \boldsymbol{m}^{(2)}\boldsymbol{x} \leqq \boldsymbol{p}^{(2)}\} \tag{4}$$

$$\boldsymbol{x} \in \Re^n,\ \boldsymbol{m}^{(i)} \in \Re^{s^{(i)} \times n},\ \boldsymbol{p}^{(i)} \in \Re^{s^{(i)}}.$$

Let us consider the following linear programming problem.

$$\begin{array}{ll} \text{Find} & \boldsymbol{x}_c, r \\ \text{which maximize} & r \\ \text{subject to} & \boldsymbol{m}_i\boldsymbol{x}_c + r\|\boldsymbol{m}_i\| \leqq \boldsymbol{p}_i \\ & \boldsymbol{m} = \begin{pmatrix} \boldsymbol{m}^{(1)} \\ \boldsymbol{m}^{(2)} \end{pmatrix}, \boldsymbol{p} = \begin{pmatrix} \boldsymbol{p}^{(1)} \\ \boldsymbol{p}^{(2)} \end{pmatrix} \\ & i = 1, \cdots, s^{(1)} + s^{(2)} \end{array} \tag{5}$$

($\boldsymbol{m}_i$ represents i-th row vector of matrix $\boldsymbol{m}$, and $\boldsymbol{p}_i$ represents i-th component of vector $\boldsymbol{p}$, respectively.)

Theorem 1: There exists the intersection of two convex polyhedra (3) and (4) if and only if r, which is the solution of (5), has positive value.

Proof The intersection of two convex polyhedra (3) and (4) is equivalent to a solution which satisfies inequalities that represent these polyhedra. (In the following, we do not distinguish a polyhedron from the inequalities corresponding to it.)

(only if) Let $\boldsymbol{x}_c$ be a solution which satisfies inequalities (3) and (4) simultaneously. Since $\boldsymbol{x}_c$ satisfies $\boldsymbol{m}_i\boldsymbol{x}_c \leqq \boldsymbol{p}_i$ ($i = 1, \cdots, s^{(1)} + s^{(2)}$) . There exists $r \geqq 0$ which satisfies

$$\boldsymbol{m}_i\boldsymbol{x}_c + r\|\boldsymbol{m}_i\| \leqq \boldsymbol{p}_i \quad i = 1, \cdots, s^{(1)} + s^{(2)}. \tag{6}$$

Problem (5) maximizes r, then the solution of (5) is positive.

(if) Prove the contraposition of sufficient part.

Since simultaneous inequalities (3) and (4) have no feasible solution, for all $\boldsymbol{x}_c$ there exists at least one i such that $\boldsymbol{m}_i\boldsymbol{x}_c > \boldsymbol{p}_i$. For such i,

$$\boldsymbol{m}_i\boldsymbol{x}_c + r\|\boldsymbol{m}_i\| \leqq \boldsymbol{p}_i \ \Rightarrow r \leqq 0. \tag{7}$$

In this case, the solution of (5) is negative. □

Definition 2: Signed distance between the point $\boldsymbol{x}$ and the hyperplane $\boldsymbol{m}_i\boldsymbol{x} = \boldsymbol{p}_i$ is defined by the following equation.

$$r = \begin{cases} \tilde{r} & \text{if} \quad \boldsymbol{m}_i\boldsymbol{x} \leqq p_i \\ -\tilde{r} & \text{if} \quad \boldsymbol{m}_i\boldsymbol{x} > p_i \end{cases} \tag{8}$$

In this definition, $\tilde{r}$ represents a Euclidean distance between a point and a hyperplane.

For one $\boldsymbol{x}_c \in \Re^n$ and all i, consider the point $\boldsymbol{x} = \boldsymbol{x}_c + r_i \frac{\boldsymbol{m}_i^T}{\|\boldsymbol{m}_i\|}$. If this point lies on the hyperplane $\boldsymbol{m}_i\boldsymbol{x} = \boldsymbol{p}_i$, then

$$\boldsymbol{m}_i\boldsymbol{x}_c + r_i\|\boldsymbol{m}_i\| = \boldsymbol{p}_i \quad i = 1, \cdots, s^{(1)} + s^{(2)} \tag{9}$$

holds. In this case, r_i corresponds to the signed distance between the point $\boldsymbol{x}_c$ and the hyperplane $\boldsymbol{m}_i\boldsymbol{x} = \boldsymbol{p}_i$ by definition. Here, let r_{min} be the minimum value of r_i over all i. Then the following inequality will be satisfied.

$$\boldsymbol{m}_i\boldsymbol{x}_c + r_{min}\|\boldsymbol{m}_i\| \leqq \boldsymbol{p}_i \quad i = 1, \cdots, s^{(1)} + s^{(2)} \tag{10}$$

Hence, if Eq. (5) has solution $\boldsymbol{x}_c$ and r, then r represents the minimum signed distance from $\boldsymbol{x}_c$ to $\boldsymbol{m}_i\boldsymbol{x} = \boldsymbol{p}_i$ over all i. In (5), since $\boldsymbol{x}_c$ is also a variable to be optimized, then Eq. (5) can be restated as the problem: "Find a point which maximizes the minimum signed distance to each hyperplane".

From these discussions, we can see that the solution r of (5) is associated with the signed distance to each hyperplane which constructs the polyhedra, regardless of whether or not the intersection exists. Therefore, we define the signed distance between two polyhedra as follows:

Definition 3: Given two convex polyhedra (3) and (4), the signed distance between these polyhedra is defined by the solution r of (5).

The signed distance as defined here has following features.

$$\begin{cases} r > 0 & \Leftrightarrow \text{ two polyhedra intersect} \\ r = 0 & \Leftrightarrow \text{ two polyhedra are tangent} \\ r < 0 & \Leftrightarrow \text{ two polyhedra do not intersect} \end{cases} \tag{11}$$

If $r > 0$, r is the radius of the largest ball that exists in the intersection, and $\boldsymbol{x}_c$ is the Chebyshev center of the intersection [4]. If $r < 0$, a ball whose radius is $|r|$ exists between two convex polyhedra (Fig. 9).

3.2.2 Degree of Danger of the System

Definition 4: The degree of danger in this system is defined by the signed distance $r(k)$ between $P_{state}(\boldsymbol{y}(k))$ and $P_{danger}(k)$.

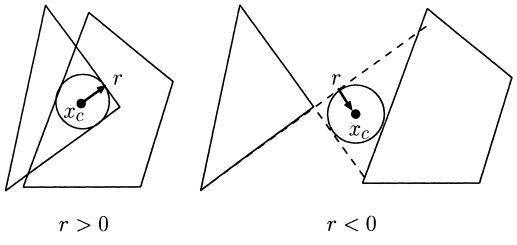


Fig. 9 Geometric meaning of signed distance.

The following theorem shows a criterion for the safety check (Fig. 10).

Theorem 2: The system is safe if $r(k) < 0$ for all k.

3.2.3 Effective Method for Computing Degree of Danger

If $r < 0$, a ball whose radius is $|r|$ exists between two convex polyhedra (Fig. 9). Then, the minimum Euclidean distance between two polyhedra is larger than $2|r|$. The computational burden can be reduced by exploiting this property as follows:

Definition 5: $\Delta r(k)$ denotes the maximum moving distance of the vertices of a polyhedron between k-th and $(k+1)$-th step. Then the estimate of the signed distance in the most dangerous case, which is denoted by $\hat{r}(k)$, is defined by the following equation.

$$2\hat{r}(k+m) = 2r(k) + \sum_{i=0}^{m-1} \Delta r(k+i) \tag{12}$$

The following lemma immediately follows.

Lemma 1: If $\hat{r}(k) < 0$, then $r(k) < 0$. Therefore, $\hat{r}(k) < 0$ implies that the system is safe.

Δr can be computed based on the plant model and its computational burden is much smaller than that of r. (r is computed by linear programming)

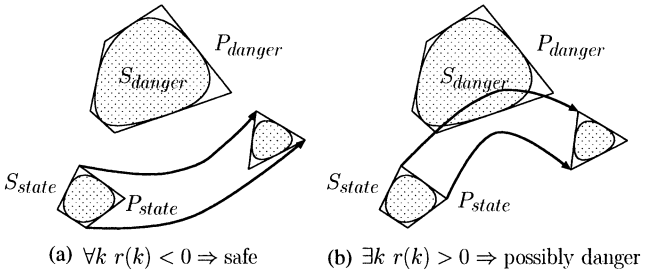
3.3 Safety Verification Algorithm

This section describes the safety verification algorithm based on the degree of danger. First of all, the following assumption is made.

Assumption 1: A model of the controlled plant (described by difference equation), the shape of its components, control law and P_{danger} are given. Also, the initial state of the plant $x(0)$ is known.

Safety verification algorithm

1. $k \leftarrow 0$
2. $P_{state}(\boldsymbol{x}(k))$, and degree of danger $r(k)$ are calculated using (5). Set $\hat{r}(k) \leftarrow r(k)$
3. The input $\boldsymbol{u}(k)$ is determined following the control law (implemented on the PLC), and $\boldsymbol{y}(k)$ and $\Delta r(k)$ are calculated by using the model.
4. If the control has been completed, then go to 5.
   If $2\hat{r}(k+1) = 2\hat{r}(k) + \Delta r(k) < 0$, then $k \leftarrow k+1$ and go to 3.
   If $2\hat{r}(k+1) = 2\hat{r}(k) + \Delta r(k) > 0$, then $k \leftarrow k+1$ and go to 2.
5. If $\hat{r}(k) < 0$ holds for any k, then the system is safe. Otherwise the system can be dangerous.



(a) $\forall k\ r(k) < 0 \Rightarrow$ safe (b) $\exists k\ r(k) > 0 \Rightarrow$ possibly danger

Fig. 10 Safety verification using degree of danger.

4. Example

4.1 Safety Verification of PLC-Driven Material Handling Manipulators

The usefulness of our idea is demonstrated through the application to the PLC-driven material handling manipulators (Fig. 11). The control requirements are stated as follows:

Control requirements

Manipulator 1 Carry a workpiece to work table 2 if it is on the work table 1.
Manipulator 2 Carry a workpiece to work table 3 if it is on the work table 2.
Whole system Carry a workpiece from work table 1 to the work table 3. In this case, the two manipulators must not collide.

Each manipulator is driven by the PLC which satisfies each control requirement. Figure 12 shows the SFC for manipulator 1. Two manipulators can be operated in parallel so as to make the operation faster. In order to verify the safety of the operation, i.e., that no collision should exist between manipulators 1 and 2, the proposed degree of danger is applied. Note that each link of the manipulator is outer approximated by use of the convex polyhedra.

The safety of the system is verified for various types of control law and physical parameter (velocity). Figure 13 shows three types of control law, and Table 1 shows the velocity of each manipulator. In Figs. 13, (a), (b), and (c) show the control in parallel operation, without parallel operation, and in partially parallel operation, respectively. Note that only control law (b) is verified as safe in the DES-based

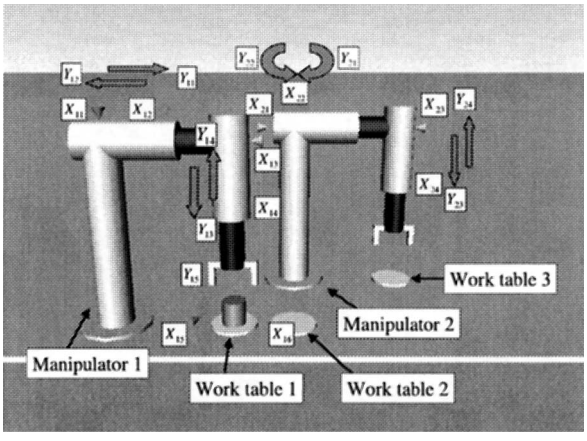


Fig. 11 PLC-driven material handling manipulators.

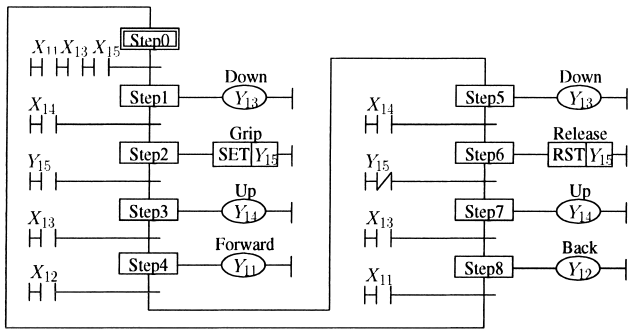


Fig. 12 SFC for manipulator 1.

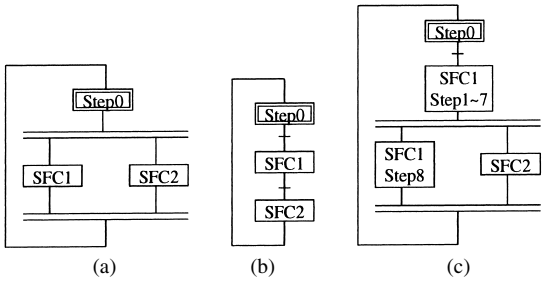


Fig. 13 SFCs for two manipulators.

Table 1 Parameters used in simulation.

manipulator	Velocity [s^{-1}] (left-right, up-down) case 1	case 2	case 3
1	(3, 3)	(2, 2)	(4, 4)
2	(0.9, 3)	(1.2, 4)	(0.6, 2)

Table 2 Verification result. (Safety/Time to complete the task)

	case 1 [s]	case 2 [s]	case 3 [s]
Control law (a)	D/12.75	D/12.75	S/14.85
Control law (b)	S/15.85	S/17.25	S/17.25
Control law (c)	S/14.15	D/14.75	S/15.95

S: Safe D: Danger

framework since two manipulators do not access work table 2 simultaneously. (Control laws (a) and (c) allow the possibility of the manipulators accessing work table 2 simultaneously.)

4.2 Verification Result

Table 2 shows the verification results, and Fig. 14 shows $r(k)$ and $\hat{r}(k)$ when using control law (c). From Table 2, it is clear that control law (b) is safe for all parameters. Control laws (a) and (c), however, have parallel operation, and their safety depend on physical parameters. In case 1, control law (c) provides the most effective control since it guarantees safety and it can minimize the total time necessary to complete the task. These results can be attained by considering the physical behavior of the plant, thus the usefulness of the proposed method has been demonstrated.

Figure 14 shows the profiles of $\hat{r}(k)$ and $r(k)$. We can see that $\hat{r}(k) \geqq r(k)$ holds and the sign of $\hat{r}(k)$ is equivalent to that of $r(k)$. The time point at which r was calculated is depicted by the circle in Fig. 14. At these times, r was used to check the safety of the system, otherwise $\hat{r}$ was used. Table 3 also shows the reduction of the computational burden by the introduction of $\hat{r}$. This result shows that the proposed introduction of $\hat{r}$ can reduce the computation time by 75%.

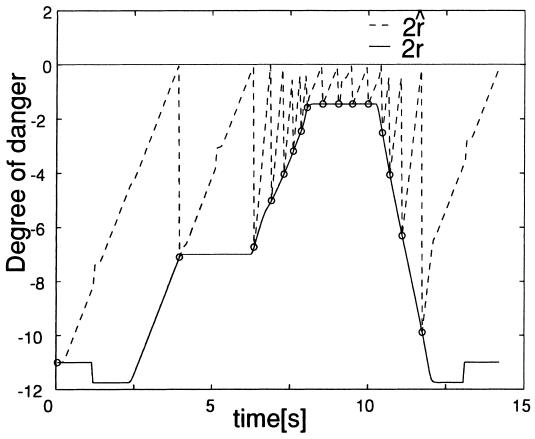


Fig. 14 Time profile of $\hat{r}$ and r. (case 1, control law (c))

Table 3 Computation time.

	Worst [s]	Average [s]
No reduction	11.3	10.4
Proposed method	3.03	2.65

5. Conclusions

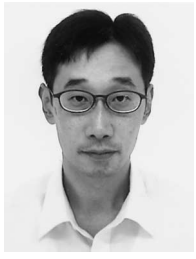
In this paper, the material handling system driven by PLC has been handled as a hybrid dynamical system, and the safety verification algorithm for such system has been proposed. Signed distance between two polyhedra has been defined and introduced as quantitative measure for the safety. Proposed safety verification algorithm has been applied to the material handling manipulators and its usefulness has been demonstrated. Note that proposed method has possibilities to be applied for the case where continuous valued controllers are used. Expanding the versatility of this method is our future issue.

References

[1] T. Zanma, T. Suzuki, and S. Okuma, "Design recovery for ladder diagram with information of controlled plant," Proc. 36th IEEE Conference on Decision and Control, pp.3580–3581, 1997.
[2] C. Tomlin and M. Greenstreet, eds., "Hybrid systems: Computation and control," Lecture Notes in Computer Science 2289, Springer-Verlag, Berlin, 2002.
[3] J.G. Thistle and W.M. Wonham, "Control problems in a temporal logic framework," Int. J. Control, vol.44, no.4, pp.943–976, 1986.
[4] E. Konaka, T. Suzuki, and S. Okuma, "Safety verification of programmable logic controller taking into account the physical dynamics—Application to material handling robots," Proc. SICE Annual Conference 2003, pp.1186–1191, 2003.
[5] A. Alur, T. Henzinger, G. Lafferriere, and G. Papps, "Discrete abstrac-

tion of hybrid systems," Proc. IEEE, vol.88, no.7, pp.971–984, 2000.
[6] T. Lozano-Pérez, "Automatic planning of manipulator transfer movements," IEEE Trans. Syst., Man. Cybern., vol.SMC-11, no.10, pp.681–698, 1981.

Eiji Konana was born in Aichi, Japan, in 1978. He received the B.E. and M.E. in Electrical Engineering from Nagoya University, Japan, in 2000 and 2002, respectively. Currently, he is a Ph.D. candidate in Information Electronics of Nagoya University. His research interests are in the areas of hybrid dynamical system and intelligent control systems. Mr. Konaka is a member of the SICE.



Tatsuya Suzuki was born in Aichi, Japan, in 1964. He received the B.E., M.E. and Ph.D. degrees in Electronic Mechanical Engineering from Nagoya University, Japan, in 1986, 1988 and 1991, respectively. From 1998 to 1999, he was a visiting researcher of the Mechanical Engineering Department of U.C. Berkeley. Currently, he is an Assistant Professor of the Department of Electronic Mechanical Engineering of Nagoya University. He won the paper award from IEEJ in 1995. His current research interests are in the areas of motion control systems, discrete event systems, hybrid dynamical systems and intelligent control systems. Dr. Suzuki is a member of the IEEJ, SICE, ISCIE, RSJ, JSME and IEEE.

Shigeru Okuma was born in Gifu, Japan, in 1948. He received the B.E., M.E., and Ph.D. degrees in Electronic Mechanical Engineering from Nagoya University, Japan, in 1970, 1972, and 1978, respectively. He also received the M.E. degree in Systems Engineering from Case Western Reserve University, Cleveland, OH, in 1974. Currently, he is a Professor in the Department of Information Electronics of Nagoya University. His research interests are power electronics, robotics, and evolutionary soft computing. Dr. Okuma was the recipient of the IEEE IECON'92 Best Paper Award, in addition to paper awards from the Japan Society for Precision Engineering and Institute of Electrical Engineers of Japan.