Discrete Abstraction for a Class of Stochastic Hybrid Systems Based on Bounded Bisimulation

Koichi Kobayashi, Yasuhiro Fukui, and Kunihiko Hiraishi

Abstract—Stochastic hybrid systems can express complex dynamical systems such as biological systems and communication networks, but computation for analysis and control is frequently difficult. In this paper, for a class of stochastic hybrid systems, a discrete abstraction method in which a given system is transformed into a finite-state system is proposed based on the notion of bounded bisimulation. In the existing discrete abstraction method based on bisimulation, a computational procedure is not in general terminated. In the proposed method, only the behavior for the finite time interval is expressed as a finite-state system, and termination is guaranteed. The obtained discrete abstract model can be used for model predictive control in which the finite-time optimal control problem is solved at each time. Furthermore, as an application, analysis of genetic toggle switches is also discussed.

I. INTRODUCTION

In recent years, the framework on analysis and control of hybrid systems has been extended to stochastic hybrid systems (see, e.g., [1], [7], [16]). SHSs are well known as a model of biological systems [1] and communication networks [15], and development of analysis and control methods is one of the significant works from theoretical and practical viewpoints. Although a general class of SHSs has been proposed in [8], it is difficult to solve control/analysis problems, and special classes are frequently considered. One of the typical classes is to assume that continuous dynamics are deterministic. Also in this case, several applications such as failure-prone systems can be considered [4], [6], [20].

However, even if the system is limited to such a class, computation for analysis and control is frequently difficult. In recent years, much attention has been paid on discrete abstraction methods as one of the methods to simplify computation for analysis and control of complex dynamical systems (see, e.g., [2], [3], [9], [17], [22], [23]). In discrete abstraction methods, first, the state space in a given system is partitioned into finite regions, and the behavior of the original system is expressed by the transitions between regions. In this paper, we focus on the discrete abstraction method based on bisimulation [3]. By using discrete abstract models obtained by this method, the property such as reachability, safety, and liveness can be verified. The discrete abstraction method based on bisimulation has one weakness, that is, a computational procedure does not terminate in general [3]. Classes of hybrid systems such that a computational procedure terminates have been proposed in e.g., [3], [24]. However, the obtained classes are restrictive.

On the other hand, in the field of computer science, bounded model checking [5] has been proposed so far. The basic idea in this method is to search for a counterexample in executions whose length is bounded by a given integer. Also in model predictive control in which the finite-time optimal control problem is solved at each time, it is enough to consider the behavior for the finite time interval [10]. Furthermore, bisimulation for the finite time interval, i.e., bounded bisimulation has been proposed in e.g., [12], [19]. To the best of our knowledge, for stochastic hybrid systems, the notion of bounded bisimulation has not been defined.

In this paper, for a class of stochastic hybrid systems, the discrete abstraction method based on bounded bisimulation is proposed. First, the notion of bounded bisimulation is defined according to the notion of bisimulation. Next, a computational procedure for deriving a discrete abstract model is proposed based on bounded bisimulation. The obtained discrete abstract model can be illustrated as a directed graph. Finally, as an application, analysis of genetic toggle switches [13] is discussed. From this case study, we show the effectiveness of the proposed method.

Notation: Let \( I_n, 0_{m \times n} \) denote the \( n \times n \) identity matrix, the \( m \times n \) zero matrix, respectively. For a finite set \( A \), let \( |A| \) denote the number of elements in \( A \).

II. STOCHASTIC PIECEWISE LINEAR SYSTEMS

As a model of stochastic hybrid systems, we consider the following stochastic piecewise linear (SPWL) system:

\[
x(t + 1) = \begin{cases} 
A_{I(t),1}x(t) + B_{I(t),1}u(t) & \text{with the prob. } pr(I(t),1), \\
A_{I(t),2}x(t) + B_{I(t),2}u(t) & \text{with the prob. } pr(I(t),2), \\
\vdots \\
A_{I(t),q(I(t))}x(t) + B_{I(t),q(I(t))}u(t) & \text{with the prob. } pr(I(t),q(I(t))),
\end{cases}
\]

where \( x(t) \in \mathcal{X} \subseteq \mathbb{R}^n \) is the (continuous) state, \( u(t) \in \mathcal{U} \subseteq \mathbb{R}^m \) is the (continuous) control input. The discrete state (the mode) is expressed by \( I(t) \in \mathcal{M} := \{1, 2, \ldots, M\} \). For each \( I \), a convex polyhedron \( S_I \) is assigned, and we assume that for all \( I \neq J \in \mathcal{M} \), the relations \( \bigcup_{I \in \mathcal{M}} S_I = \mathcal{X} \) and \( S_I \cap S_J = \emptyset \) are satisfied.

Suppose that for the mode \( I \), \( q(I) \) linear systems are assigned, and one linear system is randomly selected at each time. In addition, the probability that the linear system
In the SPWL system (1), only the selection of continuous dynamics is probabilistic. Hence, we consider only a discrete probability distribution. However, several systems such as failure-prone systems can be modeled.

III. NOTION OF BOUNDED BISIMULATION

In this section, we define the notion of bounded bisimulation for the SPWL system (1). First, a discrete-time Markov decision process expressing the SPWL system (1) is introduced. Next, we define the notion of bisimulation and that of bounded bisimulation.

A. Discrete-time Markov Decision Processes

To define the notions of bisimulation and bounded bisimulation, we consider expressing the SPWL system (1) as a discrete-time Markov decision process.

First, \( s = (x, u) \in X \times U =: S \) is called an enlarged state. If for given \( s = (x, u) \in S \) and \( s' = (x', u') \in S \), there exists a control input \( u \) such that the system transits from \( x \) to \( x' \) with some probability \( p > 0 \), then it is said that the system transits from \( s \) to \( s' \) with the probability \( p \). Note that \( u' \in U \) does not influence the state transition. So \( u' \) is given as any value on \( U \). Such an enlarged state is also used in the existing method [22]. By \( p : S \times S \rightarrow [0, 1] \), define the transition probability that the enlarged state transits from \( s \) to \( s' \). Noting that for a given \( x \in X \) the mode \( I \) is uniquely determined, the transition probability \( p(s, s') \) can be derived by using \( pr(I, i) \) in (1). For given \( s, s' \), define

\[
V := \{ i \in \{1, 2, \ldots, q(I)\} \mid x' = A_{I,i}x + B_{I,i}u, \ x \in S_I \}.
\]

Then, we can obtain \( p(s, s') = \sum_{i \in V} pr(I, i) \). In addition, let \( s_j', j = 1, 2, \ldots, q(I) \) denote the candidates of the next enlarged state for a given \( s \in S \). The candidates \( s_j', j = 1, 2, \ldots, q(I) \) can be computed from the SPWL system (1).

Then the following relation \( \sum_{j=1}^{q(I)} p(s, s'_j) = 1 \) holds.

Next, a discrete-time Markov decision process is defined as follows.

**Definition 1:** A discrete-time Markov decision process \( \mathcal{H} \) is defined by

\[
\mathcal{H} = (S, p)
\]

where \( S \) is the set of enlarged states, and \( p : S \times S \rightarrow [0, 1] \) is the transition probability.

By the definition, we see that in the discrete-time Markov decision process (2), the set of enlarged states is in general given by an infinite set.

Finally, the quotient transition system of a given discrete-time Markov decision process is defined. By \( \sim \), denote an equivalence relation on \( S \). For the discrete-time Markov decision process \( \mathcal{H} \) in Definition 1, without loss of generality, we consider only a class of equivalence relations such that the transition probability between two equivalence classes is uniquely determined. This is because for a fixed \( s = (x, u) \), a discrete probability distribution in the next state is uniquely determined. We also remark that the quotient set \( S/\sim \) is in general given by an infinite set. In addition, for two equivalence classes \( C, C' \in S/\sim \), we define

\[
\tilde{V} := \{ i \in \{1, 2, \ldots, q(I)\} \mid \forall s = (x, u) \in C, \ \exists s' = (x', u') \in C', \\
x' = A_{I,i}x + B_{I,i}u, \ C \subseteq S_I \}.
\]

Then, for a given equivalence relation \( \sim \), the quotient transition system \( \mathcal{H}/\sim \) of the discrete-time Markov decision process \( \mathcal{H} \) is defined as follows.

**Definition 2:** For a given equivalence relation \( \sim \), the quotient transition system \( \mathcal{H}/\sim \) of \( \mathcal{H} \) in (2) is defined by

\[
\mathcal{H}/\sim = (S/\sim, p/\sim)
\]

where \( S/\sim \) is the quotient set of \( S \) by \( \sim \). The transition probability \( p/\sim : S/\sim \times S/\sim \rightarrow [0, 1] \) is given as

\[
p/\sim(C, C') = \sum_{i \in V} pr(I, i)
\]

where \( C, C' \in S/\sim \) are any equivalence classes.

B. Bisimulation

For the discrete-time Markov decision process (2), the notion of bisimulation is defined. Let \( A \) denote a set of atomic propositions. Suppose that an atomic proposition is assigned to each element of \( S \) by \( I : S \rightarrow 2^A \). Then the following definition is given.

**Definition 3:** Suppose that for the discrete-time Markov decision process (2), \( A \) and \( I : S \rightarrow 2^A \) are given. Then the equivalence relation \( \sim \) is called a bisimulation if for any two enlarged states \( s, t \) satisfying \( s \sim t \), the relation \( I(s) = I(t) \) holds, and for any element \( c \) of any equivalence class \( C \in S/\sim \), the relation \( p(s, c) = p(t, c) \) holds.

Let \( \sim_B \) denote a bisimilar equivalence relation. Consider the quotient transition system \( \mathcal{H}/\sim_B \). Then the following result [21] on verification of PCTL (Probabilistic Computation Tree Logic) formulas [11] is known.

**Theorem 1:** Suppose that for the discrete-time Markov decision process \( \mathcal{H} \) of (2), \( \sim_B \) and a PCTL formula \( \phi \) are given. The discrete-time Markov chain \( \mathcal{H} \) satisfies \( \phi \) if and only if the quotient transition system \( \mathcal{H}/\sim_B \) satisfies \( \phi \).

From this result, we see that even if \( S \) is a finite set, if \( S/\sim_B \) is a finite set, that is, \( \mathcal{H}/\sim_B \) is a finite-state system, then the problem for verifying PCTL formulas is feasible. However, from [3], it is obvious that \( S/\sim_B \) is not a finite set in general.

C. Bounded Bisimulation

In Definition 3, the behavior for an infinite time interval is considered for discrete-time Markov decision processes. On the other hand, in some cases, it is enough to consider the behavior for the finite time interval. From this viewpoint, bounded model checking [5] has been proposed so far. Also in the field of control engineering, model predictive control has been proposed. In model predictive control, the finite-time optimal control problem is solved at each time. If the
Several control problems can be considered by using discrete abstract models such as Fig. 1. For example, in model predictive control, the finite-time optimal control problem is solved at each time. Then, a discrete abstract model such as Fig. 1 is useful in computation of the finite-time optimal control problem. See also [18] for further details.

IV. COMPUTATIONAL PROCEDURE OF DISCRETE ABSTRACTION

In this section, we propose a procedure for computing a discrete abstract model. As a preparation, we define the notation. Let \( \pi \) denote a finite partition of the enlarged state, that is, \( \pi := \{ S_1, S_2, \ldots \}, |\pi| < \infty, \cup_i S_i = S, \cap_i S_i = \emptyset, i \neq j \). For a set \( Y \subseteq Y_1 \times Y_2 \times \cdots \), \( \mathcal{R}^n \times \cdots \times \mathcal{R}^n \), let \( \text{Pro}_{ Y_1}(Y) \) denote a projection of \( Y \) to \( Y_1 \).

A. Outline

The proposed procedure consists of four components \( \text{Pre}_X(S, I, i) \), \( \text{Pre}(S) \), \( \text{Partition}(\pi) \), and \( k\)-bounded bisimulation(\( \pi_0, k \)). The outline of each component is explained below.

- \( \text{Pre}_X(S, I, i) \): Consider the case of using the continuous dynamics determined by given \( I, i \). Let \( \text{Pre}_X(S, I, i) \subseteq S \) denote the set of enlarged states such that the system transits to \( S \) at one step.
- \( \text{Pre}(S) \): Let \( \text{Pre}(S) \subseteq S \times [0,1] \) denote the set of pairs of the enlarged state set such that the system transits to \( S \) at one step and its transition probability.
- \( \text{Partition}(\pi) \): A given partition \( \pi \) is refined by using \( \text{Pre}(S), S_i \in \pi, i = 1, 2, \ldots, |\pi| \). In addition, the transition probability from each element in the refined partition to each element in \( \pi \) is also derived.
- \( k\)-bounded bisimulation(\( \pi_0, k \)): For the initial partition \( \pi_0 \), which is given in advance, computation of \( \text{Partition}(\pi) \) is executed \( k \) times.

By using \( k\)-bounded bisimulation(\( \pi_0, k \)), we can obtain the partition on the enlarged state space and the transition probability at time \( 0, 1, \ldots, k \), that is, we can make a discrete abstract model such as Fig. 1. The proposed procedure can be implemented by using Multi-Parametric toolbox [26].

B. Proposed Computational Procedure

The proposed computational procedure is explained in detail.

First, \( \text{Pre}_X(S, I, i) \) is defined as

\[
\text{Pre}_X(S, I, i) := \{ (x, u) \in S | x' = A_{I, i} x + B_{I, i} u, \ x' \in \text{Pro}_{X}(S), x \in \mathcal{X}, u \in \mathcal{U} \}.
\]

From this definition, \( \text{Pre}_X(S, I, i) \) can be computed by manipulating convex polyhedra.

Next, \( \text{Pre}(S) \) can be computed by Algorithm 1. In this algorithm, \( \text{Pre}_X(S, I, i) \) is computed for all \( I, i \). In lines 6 and 7, based on \( \text{Pre}_X(S, I, i) \), \( S' \) is split to \( E_1 \) and \( E_2 \). In the case of \( E_1 = \emptyset \) and \( p > 0 \), there exists multiple linear
systems, which can transit from $E_1$ to $S$. In this case, the transition probability from $E_1$ to $S$ is derived as a sum of probabilities assigned to these linear systems.

Third, the computation procedure of $\text{Partition}(\pi)$ is shown by Algorithm 2. In lines 2–13, a given state partition $\pi$ is refined to $\pi_{\text{pre}}$. In line 5, the transition probability is also derived based on $\text{Pre}(S_j)$, $S_j \in \pi$. If $\pi$ is a finite partition, then the while loop in lines 7–11 terminates. This is because $S_1$ and $S_2$ do not satisfy the condition $\exists S'' \in \pi_{\text{pre}} : \emptyset \neq (S'' \cap S') \neq S''$ in this while loop, and the number of regions satisfying this condition monotonically decreases. In lines 14–24, the transition probability from the enlarged state set including in $\pi_{\text{pre}}$ to that including in $\pi$ is computed using $T_j$.

Finally, the computation procedure of $k$-bounded bisimulation is shown by Algorithm 3.

From the proposed computational procedure, we can obtain the following theorem.

**Theorem 2:** If the initial partition $\pi_0$ is given as a finite partition, then the partitions $\pi_j$, $j = 1, 2, \ldots, k$ are derived as a finite partition, and a procedure for computing $\pi_j, T_j$, $j = 1, 2, \ldots, k$ terminates.

**C. Numerical Examples**

Consider the following bimodal SPWL system:

$$x(t + 1) = \begin{cases} A(\alpha(t))x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ \text{with the prob. } pr(I(t), 1), \\ A_{I(t), 2}x(t) + B_{I(t), 2}u(t) \\ \text{with the prob. } pr(I(t), 2), \end{cases}$$

where $x(t) \in X = [-1, 1] \times [0, 2]$, $u(t) \in U = [-1, 1]$, $S_I = [0, 1] \times [0, 2]$, $S_2 = [-1, 0] \times [0, 2]$, $A(\alpha(t)) = 0.8 \begin{bmatrix} \cos \alpha(t) & -\sin \alpha(t) \\ \sin \alpha(t) & \cos \alpha(t) \end{bmatrix}$, $\alpha(t) = \begin{cases} +\pi/3 & \text{if } x(t) \in S_1, \\ -\pi/3 & \text{if } x(t) \in S_2 \end{cases}$

and the probability is given as $pr(1, 1) = 0.8$, $pr(1, 2) = 0.2$, $pr(2, 1) = 0.9$, $pr(2, 2) = 0.1$. For $A_{I(t), 2}$ and $B_{I(t), 2}$, we consider the following three cases:

- **Case (i):**
  $$A_{I, 2} = I_2, \; B_{I, 2} = 0_{2 \times 2}, \; I \in \{1, 2\}, \quad (3)$$

- **Case (ii):**
  $$A_{I, 2} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \; B_{I, 2} = 0_{2 \times 2}, \; I \in \{1, 2\},$$

- **Case (iii):**
  $$A_{I, 2} = 0.8 \begin{bmatrix} \cos \beta(k) & -\sin \beta(k) \\ \sin \beta(k) & \cos \beta(k) \end{bmatrix}, \; B_{I, 2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$\beta(k) = \begin{cases} -\pi/27 & \text{if } x(t) \in S_1, \\ +\pi/27 & \text{if } x(t) \in S_2. \end{cases}$$

In addition, the initial partition is given as Fig. 2 (left), where only the state space is shown, and for each set of the state, $U$ is assigned. The number of regions is four.
The computation result is explained. First, we show the result on Case (i). In Algorithm 3, $\pi_1 = \pi_2$ is obtained. That is, in this case, partitioning is terminated. Fig. 2 (right) shows a projection of $\pi_1$ to the state space. For example, we focus on the region F. By deciding the control input appropriately, the system can transit from the region F to the region A or C or E with the probability 0.8 and the region F with the probability 0.2. The transition from F to C can be realized by using the control input satisfying

$$
\begin{bmatrix}
-0.5410 & -0.3123 & -0.7809 \\
0.5 & -0.8660 & 0 \\
0.6786 & 0 & 0.7346 \\
-1 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
u
\end{bmatrix}
\leq
\begin{bmatrix}
-0.7809 \\
1.0602 \\
0 \\
0 \\
0 \\
1 \\
1
\end{bmatrix}.
$$

Since the enlarged state is used in the proposed method, the transition condition can be obtained by a set of linear inequalities with respect to the state and the control input. Also for other transitions, the linear inequality expressing the enlarged-state set is assigned. Furthermore, we focus on the region H. By deciding the control input appropriately, the system can transit from the region H to (i) the region B with the probability 0.8, and the region H with the probability 0.2, (ii) the region D with the probability 0.8, and the region H with the probability 0.2, (iii) the region F with the probability 0.8, and the region H with the probability 0.2, and (iv) the region H with the probability 0.8, and the region H with the probability 0.2, that is, the region H with the probability 1.0. It is obvious that from $x(t+1) = x(t)$ (see also (3)), the system can transit from H to H with the probability 0.2 (note that the region H is included in $S_I$). Also, by using another linear system $x(t+1) = A(\pi/3)x(t) + [0\ 1]^T u(t)$, the system can also transit from H to H with the probability 0.8. Therefore, by deciding the control input appropriately, the system can transit from H to H with the probability 1.0.

Next, three cases are compared. For $k = 0, 1, 2, 3, 4$, the number of regions in partitions of enlarged-state partitions is shown in Table I. From Table I, we see that in Case (ii) and Case (iii), partitioning is not terminated until $k = 4$ (probably, partitioning will be not globally terminated). However, in all cases, a discrete abstract model such as Fig. 1 can be constructed, and can be used to optimal control and model predictive control.

V. APPLICATION TO ANALYSIS OF GENETIC TOGGLE SWITCHES

In this section, we consider a genetic toggle switch [13] as one of the applications. A genetic toggle switch is a switch implemented by gene regulatory networks. A method to implement logic circuits by biological systems such as genetic networks has been extensively studied in the field of synthetic biology [14]. Biological logic circuits are expected as one of the methods to realize parallel computing. A genetic toggle switch is important as one of small-scale genetic devices in such a circuit. In this paper, according to numerical simulations in [25], we consider a simpler situation, that is, we consider finding a combination of modes such that a given switching pattern in the mode is realized.

A. Model of Genetic Toggle Switches

The PWA (piecewise affine) model of a genetic toggle switch has been proposed in [25], [27]. By replacing the linear system $x(t+1) = A_{0}x(t) + B_{0}u(t)$ in the SPWL system (1) with $x(t+1) = A_{0}x(t) + B_{0}u(t) + a_{I0}$, (where $a_{I0}$ is a given vector), the SPWL system (1) can be easily extended to the SPWA system. Hence, we consider the following SPWA model obtained by adding the stochastic behavior to the PWA model:

$$
x(t+1) = \begin{cases} 
A_{18}(t)x(t) + u(t) + a_{18}(t) & \text{with the prob. } pr(I(t), 1) = 0.9, \\
x(t) & \text{with the prob. } pr(I(t), 2) = 0.1, \\
& \text{if } x(t) \in S_{I(t)}
\end{cases}
$$

where $x(t) \in X = [0, 100] \times [0, 100]$, $u(t) \in U = [-15, 15] \times [-18, 18]$, $I(t) \in [1, 2, \ldots , 9]$. Matrices $A_{0}$ and $B_{0}$ are omitted due to the limited space. The region $S_{I}$ is shown by Fig. 3 (left). This partition is also used as the initial partition.

B. Computation Result and Discussion

The discrete abstract model based on 2-bounded bisimulation was computed, where the number of regions in each partition is 1667 ($t = 0$), 77 ($t = 1$), and 9 ($t = 2$). Fig. 3 (right) shows the obtained state partition at $t = 1$, i.e., a projection of $\pi_1$ to the state space.

By using the obtained discrete abstract model, we discuss the construction of a genetic toggle switch. Consider the following problem: assign ‘ON’ and ‘OFF’ to two modes, such that the pattern ‘ON ’ ‘OFF’ with some probability is realized by suitably adjusting the initial state and the control input. As an example, we focus on modes 5 and 6 (of course, we may focus on other nodes). In the obtained model, $S_{5}$ is split to 9 subregions, and $S_{6}$ is also split to 9 subregions. These facts can be also observed from Fig. 3 (right). In

<table>
<thead>
<tr>
<th>$k$</th>
<th>Case (i)</th>
<th>Case (ii)</th>
<th>Case (iii)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>14</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>34</td>
<td>77</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>131</td>
<td>3273</td>
</tr>
</tbody>
</table>
addition, from the obtained model, we can obtain the following facts: (i) the nodes corresponding to some subregions included in $S_0$ is connected to the node corresponding to $S_0$ with the probability 0.9, (ii) the nodes corresponding to some subregions included in $S_0$ is connected to the node corresponding to $S_0$ with the probability 0.9. Hence, there is a possibility that the pattern ‘ON’ ⇐ ‘OFF’ can be realized by modes 5 and 6. In fact, from the discrete abstract model and Fig. 3 (right), we can obtain two regions $W_1, W_2$ shown in Fig. 4. Then, there exists a control input such that the state included in $W_1$ reaches $W_2$ with the probability 0.9, and conversely, there exists a control input such that the state included in $W_2$ reaches $W_1$ with the probability 0.9. Therefore, the pattern ‘ON’ ⇐ ‘OFF’ can be realized.

VI. CONCLUSION

In this paper, for a class of stochastic hybrid systems, we proposed a discrete abstraction method based on the notion of bounded bisimulation. Although the time interval that the behavior is evaluated is limited to the finite time interval, the discrete abstract model obtained by the proposed method can be used in model predictive control and verification. Furthermore, we considered genetic toggle switches as an application. The proposed method provides us the basic in analysis and control of stochastic hybrid systems.

The proposed method can be basically applied to any SPWL system and any $k < \infty$. However, for a large $k$, there is a possibility that the discrete abstract model cannot be computed due to several issues such as memory consumption. The computability depends on the computer environment. This problem is a common problem in discrete abstraction. Then, it is important to develop an approximate method such as pruning. Details are one of the future works.

This work was partially supported by Grant-in-Aid for Young Scientists (B) 23760387.

REFERENCES

[23] conPAS/conPAS2: http://hyness.bu.edu/software/conPAS2.html