Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

# Finite Abstractions of Discrete-time Linear Systems and Its Application to Optimal Control

**Yuichi Tazaki** * **Jun-ichi Imura** **

* Department of Mechanical and Environmental Informatics, Tokyo Institute of Technology, Tokyo, Japan, (e-mail: tazaki@cyb.mei.titech.ac.jp)
** Department of Mechanical and Environmental Informatics, Tokyo Institute of Technology, Tokyo, Japan, (e-mail: imura@mei.titech.ac.jp)

**Abstract:** Optimal control and reachability analysis of continuous-state systems often require computational algorithms with high complexity. The use of finite abstractions of continuous-state systems reduces such problems to path-planning problems on directed graphs with a finite number of nodes, which can be computed efficiently. In this research, we propose a method to design an approximately bisimilar finite abstraction of stabilizable discrete-time linear systems, considering the minimization of the complexity of the resultant finite automaton. Moreover, we show that a suboptimal solution to optimal control problems with a known error bound is obtained by simulating the optimal path of an approximately bisimilar finite abstraction.

## 1. INTRODUCTION

Control problems and verification problems on complex systems require numerical methods, whose computational complexity often grows rapidly as the state dimension and the number of time steps increase. Instead of computing the exact solutions to such problems with the original complex model, one can think of creating a simpler model, that is much easier to analyze and to control, while preserving the essential characteristics of the original model. Such simplified models are called an abstraction of the original model.

The notion of bisimulation is a powerful mathematical framework for addressing systems abstraction. Bisimulation originated in the field of labeled transition systems [1]. The major difference between labeled transition systems and dynamical systems in control theory is that the latter may consist of both continuous and discrete variables, while the former is purely discrete. In the latter case, the original definition of bisimulation, which requires precise coincidence of observations (measurement signals), are often too restrictive. In [2] the notion of bisimulation was extended to metric space and called approximate bisimulation. Approximate bisimulation requires the distance of measurement signals to be within a specified precision. Based on this notion, abstraction problems of various classes of dynamical systems were discussed [3][5].

Among various abstraction problems, the problem of deriving a finite automaton that abstracts a given continuous-state system is called finite abstraction problem. A finite-state system is suitable for abstraction since many control problems and verification problems can be solved by numerical algorithms whose computational complexities are in polynomial order. In [4], a procedure for constructing an approximately bisimilar finite abstraction of stable discrete-time linear systems was derived. In [7] and [8], a design of approximately (bi-)similar finite abstraction of continuous-time nonlinear dynamical systems under so-called incremental stability assumption were addressed.

In most of the past researches, the application of bisimilar abstraction has been limited to safety verification problems. Our problem of interest, on the other hand, is application to optimal control problems. To date, this topic is not well explored, although some results are reported [9][10][11]. In this paper, we discuss the finite abstraction problem of stabilizable discrete-time linear systems, as well as its application to optimal control problems. In general, it is quite difficult to obtain the global optimal solution to optimal control problems with non-convex constraints and cost functions, even if the state-dynamics is linear. In the past researches, this problem was tackled by the discretization of the state-space [12][13][14], but the relation between the resolution of the discretization and the performance of the approximate solution has not been clarified. The result of this paper provides an upper-bound of the performance of the approximate solution as a function of the precision parameter of approximate bisimulation.

The rest of this paper is organized as follows: In Section 2, we give a definition of approximate bisimulation for a class of discrete-time dynamical systems. In Section 3, at first we derive a sufficient condition for a class of finite automata to be approximately bisimilar with stabilizable discrete-time linear systems, with desired precision. Next, we propose a design procedure of finite abstraction considering the minimization of the number of states. In Section 4, we show that a suboptimal solution to the finite-horizon optimal control problem is obtained by simulating the optimal trajectory of approximately bisimilar abstraction of the original model. Section 6 concludes this paper with some remarks for future works.

Notations: The symbol $[\boldsymbol{v}_1; \boldsymbol{v}_2; ...; \boldsymbol{v}_N]$ denotes the vertical concatenation of vectors or that of matrices, which is equivalent to $[\boldsymbol{v}_1^{\mathrm{T}}\, \boldsymbol{v}_2^{\mathrm{T}} \, \dots \, \boldsymbol{v}_N^{\mathrm{T}}]^{\mathrm{T}}$. Throughout the paper, the symbol $\|\cdot\|$ denotes the 2-norm unless otherwise stated. Moreover, the symbol $\|\boldsymbol{v}\|_M$ is defined as $\sqrt{\boldsymbol{v}^{\mathrm{T}} M \boldsymbol{v}}$ .

## 2. APPROXIMATE SIMULATIONS AND BISIMULATIONS OF DISCRETE-TIME DYNAMICAL SYSTEMS

In this section, we introduce the definition of approximate (bi-)simulation on a class of discrete-time dynamical systems.

*Definition 1.* Discrete-time dynamical system
A discrete-time dynamical system is a 5-tuple $\langle X, U, Y, f, h\rangle$, where $X \subset \mathbb{R}^n$ is the set of states, $U \subset \mathbb{R}^m$ is the set of inputs, $Y \subset \mathbb{R}^l$ is the set of outputs, $f : X \times U \mapsto X$ is the state transition function, and $h : X \mapsto Y$ is the measurement function. The state, input, and output of the system at time $t \in \mathcal{T} = \{0\} \cup \mathbb{N}$ are expressed as $\boldsymbol{x}_t \in X$, $\boldsymbol{u}_t \in U$, and $\boldsymbol{y}_t \in Y$, respectively. The state transition and the measurement at time $t$ are expressed as

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \tag{1}$$
$$\boldsymbol{y}_t = h(\boldsymbol{x}_t), \tag{2}$$

respectively.

Throughout this paper, we use the symbol $\Sigma\langle X, U, Y, f, h\rangle$ or simply $\Sigma$ to express a discrete-time system.

Let us introduce the notion of approximate simulation and approximate bisimulation on the class of systems just defined.

*Definition 2.* Approximate simulation
Let $\Sigma\langle X, U, Y, f, h\rangle$ and $\hat{\Sigma}\langle \hat{X}, U, Y, \hat{f}, \hat{h}\rangle$ be discrete-time systems, and let $\epsilon_u$ and $\epsilon_y$ be positive constants. A binary relation $R \subset X \times \hat{X}$ is called an $(\epsilon_u, \epsilon_y)$-approximate simulation relation if and only if for every $(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in R$, the following holds.

$$\|h(\boldsymbol{x}) - \hat{h}(\hat{\boldsymbol{x}})\| \le \epsilon_y \tag{3}$$

$$\begin{aligned}&\text{for any } \boldsymbol{u} \in U, \text{ there exists } \hat{\boldsymbol{u}} \in U \text{ such that}\\ &\quad \|\boldsymbol{u} - \hat{\boldsymbol{u}}\| \le \epsilon_u \text{ and } (f(\boldsymbol{x}, \boldsymbol{u}),\ \hat{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})) \in R\end{aligned} \tag{4}$$

Moreover, if such an $R$ exists, $\hat{\Sigma}$ is said to be $(\epsilon_u, \epsilon_y)$-approximately similar to $\Sigma$ with respect to $R$.

*Definition 3.* Approximate bisimulation
Let $\Sigma\langle X, U, Y, f, h\rangle$ and $\hat{\Sigma}\langle \hat{X}, U, Y, \hat{f}, \hat{h}\rangle$ be discrete-time systems, and let $\epsilon_u$ and $\epsilon_y$ be positive constants. A binary relation $R \subset X \times \hat{X}$ is called an $(\epsilon_u, \epsilon_y)$-approximate bisimulation relation between $\Sigma$ and $\hat{\Sigma}$ if and only if $R$ is an $(\epsilon_u, \epsilon_y)$-approximate simulation relation from $\Sigma$ to $\hat{\Sigma}$ and its inverse relation $R^{-1} = \{(\hat{\boldsymbol{x}}, \boldsymbol{x}) \mid (\boldsymbol{x}, \hat{\boldsymbol{x}}) \in R\}$ is an $(\epsilon_u, \epsilon_y)$-approximate simulation relation from $\hat{\Sigma}$ to $\Sigma$. Moreover, if such an $R$ exists, $\Sigma$ and $\hat{\Sigma}$ are said to be $(\epsilon_u, \epsilon_y)$-approximately bisimilar with respect to $R$.

The major difference between the above definitions and those introduced in the literature is that our definitions require not only measurements but also control inputs of both systems to be close enough to each other. This extra condition is needed to apply bisimilar abstractions to optimal control problems with input-dependent criteria, such as input constraints, and input-energy minimization.

## 3. FINITE ABSTRACTIONS OF DISCRETE-TIME LINEAR SYSTEMS

### 3.1 Problem setting

In this section, we consider the problem of finding a finite automaton that is approximately bisimilar with a given discrete-time linear system. Discrete-time linear systems are denoted by

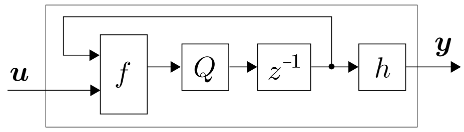



Fig. 1. system with state quantizer

$$\Sigma_{\mathrm{L}}\langle X, U, Y, A, B, C\rangle, \tag{5}$$

where the state transition and the measurement are expressed as follows.

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t \tag{6}$$
$$\boldsymbol{y}_t = C\boldsymbol{x}_t \tag{7}$$

On the other hand, finite automata are expressed as

$$\Delta\langle S, U, Y, \mathcal{U}, \mathcal{Y}\rangle. \tag{8}$$

The state set $S = \{1, 2, \ldots, |S|\}$ is a finite set of symbols, and the state at time $t$ is denoted by $s_t$. The symbol $\mathcal{U} = \{\mathcal{U}_{ij}\}\ \ (i \in S,\ j \in S)$ is a collection of subsets in $U$, where, for each $i \in S$, $\{\mathcal{U}_{ij}\}_{j\in S}$ forms a partition of $U$. The symbol $\mathcal{Y} = \{\boldsymbol{y}_i\}\ \ (i \in S)$ is a finite set composed of points on $Y$. The state transition and the measurement of $\Delta$ are defined as follows.

$$s_t = i \ \wedge\ \boldsymbol{u}_t \in \mathcal{U}_{ij} \ \ \Rightarrow\ \ s_{t+1} = j \tag{9}$$
$$s_t = i \ \ \Rightarrow\ \ \boldsymbol{y}_t = \boldsymbol{y}_i \tag{10}$$

Clearly, the set $\mathcal{U}_{ij}$ consists of control inputs that moves the state from $i$ to $j$. Since $\{\mathcal{U}_{ij}\}_{j\in S}$ is a partition on $U$, the state transition is deterministic.

Roughly speaking, our purpose is to design, for a given plant model $\Sigma_{\mathrm{L}}$, an approximately bisimilar finite automaton $\Delta$. Note that in order for the finite abstraction to be applicable to some optimal control problems (or some verification problems), an extra condition should be imposed to the approximate bisimulation relation $R$; that is, for any possible initial state $\boldsymbol{x}_0$ of $\Sigma_{\mathrm{L}}$, there should exist its approximately bisimilar pair $s_0$ in the states of $\Delta$. Under the assumption that the initial state $\boldsymbol{x}_0$ is chosen arbitrarily on $X$, this condition is written as

$$\pi^X(R) = X \tag{11}$$

where $\pi^X(\cdot)$ denotes projection of a subset of $X \times S$ onto $X$.

Based on the above considerations, the problem of concern is stated as follows.

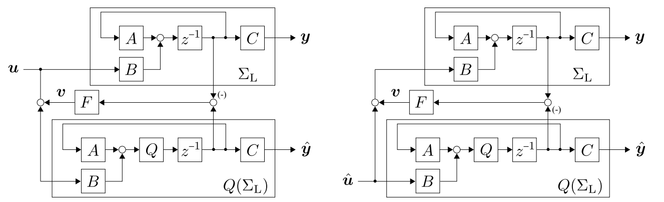
*Problem 1. Finite abstraction of a discrete-time linear system*
For a given discrete-time linear system (5) and a pair of positive constants $\epsilon_u, \epsilon_y$, find a finite automata (8) and a binary relation $R \subset X \times S$, where $R$ is an $(\epsilon_u, \epsilon_y)$-approximate bisimulation between $\Sigma_{\mathrm{L}}$ and $\Delta$ satisfying the condition (11).

From now on, we assume $U = \mathbb{R}^m$. Now, let us mention that it is quite unrealistic to regard all the parameters of $\Delta$ ($S, \mathcal{U}$, and $\mathcal{Y}$) as independent design parameters, meaning that we should restrict our attention to smaller class of finite automata. For this purpose, we focus on the following fact: *state-quantization of a continuous-state system results in a finite state system.* Let us consider a quantization function defined by

$$\begin{aligned}&Q : X \mapsto \mathcal{X}\\ &Q(\boldsymbol{x}) = \boldsymbol{x}_i \ \text{ if } \ \boldsymbol{x} \in \mathcal{S}_i\end{aligned} \tag{12}$$

where $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ is a finite set of points on $X$ and $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2 \ldots, \mathcal{S}_N\}$ is a partition of $X$. Using $Q$, we introduce the following new state equation.

$$\boldsymbol{x}_{t+1} = Q(A\boldsymbol{x}_t + B\boldsymbol{u}_t) \tag{13}$$




(a) $Q(\Sigma_{\mathrm{L}})$ simulating $\Sigma_{\mathrm{L}}$ (b) $\Sigma_{\mathrm{L}}$ simulating $Q(\Sigma_{\mathrm{L}})$

Fig. 2. Two systems simulating one another

Notice that, in this state equation, the state transition is closed on $\mathcal{X}$ as long as the initial state is chosen from $\mathcal{X}$. This leads us to define a finite automaton, induced by state quantization of a continuous-state system:

$$\begin{aligned}&\Delta\langle \mathcal{X}, U, Y, \mathcal{U}, \mathcal{Y}\rangle,\\ &\mathcal{U}_{ij} = \{\boldsymbol{u} \in U \mid Q(A\boldsymbol{x}_i + B\boldsymbol{u}) = \boldsymbol{x}_j\},\\ &\boldsymbol{y}_i = C\boldsymbol{x}_i.\end{aligned} \tag{14}$$

We denote by $Q(\Sigma_{\mathrm{L}})$ the finite automaton obtained by state-quantization of $\Sigma_{\mathrm{L}}$ with $Q$. Note that, by state quantization, the control input is also discretized (in the sense of partitioning of the control input space) without introducing explicit control input quantization. This characteristics differs from existing researches (like [4][7]), where explicit input quantization or originally discrete input systems are considered. In the rest of this paper, we deal with the design problem of the quantization function $Q$, whose resultant state-quantized system $Q(\Sigma_{\mathrm{L}})$ is approximately bisimilar with $\Sigma_{\mathrm{L}}$.

Let us mention that the term "quantised system" is used in [15], and a similar expression to (13) is used in [17][18]. The essential difference is that, in their definitions, quantization operation is a map from the state space to a finite set of symbols, whereas in our definition it is a projection of the state space onto its finite subset.

### 3.2 *Condition for Approximate Bisimulation*

Roughly speaking, if two systems are approximately bisimilar, one system can be driven in such a way that its state tracks the state of the other system under a certain error bound on control inputs and measurements. This is illustrated in Fig. 2. Fig. 2(a) shows $Q(\Sigma_{\mathrm{L}})$ tracking $\Sigma_{\mathrm{L}}$, and Fig. 2(b) shows the opposite case, $\Sigma_{\mathrm{L}}$ tracking $Q(\Sigma_{\mathrm{L}})$. Let us take a closer look into Fig. 2(a). Here, an arbitrary control input $\boldsymbol{u}_t$ is applied to the system $\Sigma_{\mathrm{L}}$ at each time $t$. On the other hand, the control input applied to $Q(\Sigma_{\mathrm{L}})$ is given by $\boldsymbol{u}_t + \boldsymbol{v}_t$. The variable $\boldsymbol{v}_t$ is the difference between the control inputs applied to both systems, and it plays a crucial role for $Q(\Sigma_{\mathrm{L}})$ to simulate $\Sigma_{\mathrm{L}}$. Let us denote the state of $\Sigma_{\mathrm{L}}$ by $\boldsymbol{x}_t$ and that of $Q(\Sigma_{\mathrm{L}})$ by $\hat{\boldsymbol{x}}_t$, respectively. Then, the state transition of each system is expressed as follows.

$$\boldsymbol{x}_{t+1} = A\boldsymbol{x}_t + B\boldsymbol{u}_t \tag{15}$$

$$\hat{\boldsymbol{x}}_{t+1} = A\hat{\boldsymbol{x}}_t + B(\boldsymbol{u}_t + \boldsymbol{v}_t) + \boldsymbol{d}_t \tag{16}$$

Here, the variable $\boldsymbol{d}_t$ is a quantization error signal given by

$$\boldsymbol{d}_t = Q(A\hat{\boldsymbol{x}}_t + B(\boldsymbol{u}_t + \boldsymbol{v}_t)) - (A\hat{\boldsymbol{x}}_t + B(\boldsymbol{u}_t + \boldsymbol{v}_t)). \tag{17}$$

Taking the difference of the above state equations, we obtain the following error system;

$$\boldsymbol{e}_{t+1} = A\boldsymbol{e}_t + B\boldsymbol{v}_t + \boldsymbol{d}_t \tag{18}$$

where $\boldsymbol{e}_t = \hat{\boldsymbol{x}}_t - \boldsymbol{x}_t$.

Now, let us define an invariant set of the error system (18) as a set $E \subset \mathbb{R}^n$ satisfying the following conditions.

$$\begin{aligned}&\forall \boldsymbol{e} \in E,\ ||C\boldsymbol{e}|| \le \epsilon_y \wedge\\ &\quad \exists \boldsymbol{v} \text{ s.t. } (||\boldsymbol{v}|| \le \epsilon_u \ \wedge\ (\forall \boldsymbol{d} \in \mathcal{D},\ A\boldsymbol{e} + B\boldsymbol{v} + \boldsymbol{d} \in E))\end{aligned} \tag{19}$$

Here, the set $\mathcal{D}$ is defined as $\mathcal{D} = \bigcup_{\boldsymbol{x}\in X}(Q(\boldsymbol{x}) - \boldsymbol{x})$. In the case of Fig. 2(b), following the same line as above results in the error system

$$\boldsymbol{e}_{t+1} = A\boldsymbol{e}_t - B\boldsymbol{v}_t + \boldsymbol{d}_t. \tag{20}$$

It is easy to verify that an invariant set of (18) is an invariant set of (20) and vice versa. Here, the following lemma holds.

*Lemma 1.* Consider $E \subset \mathbb{R}^n$ and $R \subset X \times X$ related by

$$(\hat{\boldsymbol{x}} - \boldsymbol{x}) \in E \ \Leftrightarrow\ (\boldsymbol{x}, \hat{\boldsymbol{x}}) \in R. \tag{21}$$

If the set $E$ is an invariant set (19) of the error system (18), then the relation $R$ is an $(\epsilon_u, \epsilon_y)$-approximate bisimulation relation between $\Sigma_{\mathrm{L}}$ and $Q(\Sigma_{\mathrm{L}})$.

Now, let us assume that the control input difference is given in the following explicit form.

$$\boldsymbol{v}_t = \begin{cases} F(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t) & (\text{when } Q(\Sigma_{\mathrm{L}}) \text{ simulates } \Sigma_{\mathrm{L}})\\ F(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t) & (\text{when } \Sigma_{\mathrm{L}} \text{ simulates } Q(\Sigma_{\mathrm{L}}))\end{cases} \tag{22}$$

Here, $F$ is a matrix making $(A + BF) =: A_F$ asymptotically stable. This makes the error system become a asymptotically stable autonomous system with disturbances, which is written as $\boldsymbol{e}_{t+1} = A_F\boldsymbol{e}_t + \boldsymbol{d}_t$. Moreover, there exists a positive definite matrix $M$ satisfying the following conditions.

$$\begin{aligned}&M \ge \frac{1}{(1-\lambda)^2\epsilon_y^2}C^{\mathrm{T}}C,\ M \ge \frac{1}{(1-\lambda)^2\epsilon_u^2}F^{\mathrm{T}}F,\\ &A_F^{\mathrm{T}}MA_F \le \lambda^2 M\end{aligned} \tag{23}$$

Here, $\lambda$ is a constant satisfying $0 < \lambda < 1$, The reader is referred to [4] for the proof to a similar statement. Based on the above arguments, the next theorem provides a sufficient condition to approximate bisimulation between $\Sigma_{\mathrm{L}}$ and $Q(\Sigma_{\mathrm{L}})$.

*Theorem 1.*
Let $\Sigma_{\mathrm{L}}\langle X, U, Y, A, B, C\rangle$ be an $(A, B)$ stabilizable discrete-time linear system. Further, choose $F$, $M$ and $\lambda$ that satisfy (23). Then, for a quantization function $Q$ satisfying the condition

$$||\boldsymbol{x} - Q(\boldsymbol{x})||_M \le 1 \ \ \forall \boldsymbol{x} \in X, \tag{24}$$

the systems $\Sigma_{\mathrm{L}}$ and $Q(\Sigma_{\mathrm{L}})$ are $(\epsilon_u, \epsilon_y)$-approximately bisimilar with respect to the relation

$$R = \{(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in X \times X \mid ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_M \le 1/(1-\lambda)\} \tag{25}$$

and $R$ satisfies the condition (11).

Proof) The following holds for the error system:

$$\begin{aligned}||\boldsymbol{e}_{t+1}||_M &= ||A_F\boldsymbol{e}_t + \boldsymbol{d}_t||_M\\ &\le \sqrt{\boldsymbol{e}_t^{\mathrm{T}}A_F^{\mathrm{T}}MA_F\boldsymbol{e}_t} + ||\boldsymbol{d}_t||_M \le \lambda||\boldsymbol{e}_t||_M + ||\boldsymbol{d}_t||_M.\end{aligned}$$

Define the set $E$ as shown below.

$$E = \{\boldsymbol{e} \in \mathbb{R}^n \mid ||e_t||_M \le 1/(1-\lambda)\} \tag{26}$$

Then, by (23), for every element $\boldsymbol{e} \in E$ the conditions $||C\boldsymbol{e}|| \le \epsilon_y$ and $||F\boldsymbol{e}|| \le \epsilon_u$ hold, and since $||\boldsymbol{d}_t||_M \le 1$, $E$ is an invariant set (19) of the error system. Therefore it follows from Lemma 1 that $\Sigma_{\mathrm{L}}$ and $Q(\Sigma_{\mathrm{L}})$ are $(\epsilon_u, \epsilon_y)$-approximately bisimilar with respect to the relation $R$ given by (25). Finally, the relation $R$ satisfies (11) since $(\boldsymbol{x}, Q(\boldsymbol{x})) \in R$ for any $\boldsymbol{x}$. □

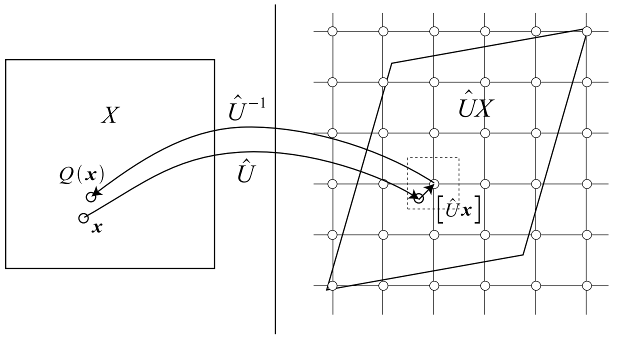



Fig. 3. Implementation of quantizer

*3.3 Design of Quantization Function*

Theorem 1 in the previous subsection gives a sufficient condition to approximate bisimulation, and in general, arbitrarily many $Q$ may satisfy this condition. In practice, in addition to approximate bisimulation, there is a requirement that the resultant finite automaton should consist of as small number of states as possible. This is due to the fact that the computational complexities of many numerical algorithms on finite automata heavily depends on the number of states. This section discusses a design procedure of the quantization function $Q$ making the number of finite states as small as possible. To begin with, we derive the explicit expression of $Q$ satisfying the condition (24). The simplest way of doing this is giving $Q$ by

$$Q(\boldsymbol{x}) = \hat{U}^{-1}[\hat{U}\boldsymbol{x}], \;\; \hat{U} = (\sqrt{n}/2)U \tag{27}$$

where the matrix $U$ is given by the decomposition of $M$, $M = U^{\mathrm{T}}U$. The operator $[\cdot]$ maps each element of a vector to its nearest integer. Here, one can see that the number of the states of the finite automaton $Q(\Sigma_{\mathrm{L}})$ is approximately proportional to the volume of the set $\hat{U}X = \{\hat{U}\boldsymbol{x} \,|\, \boldsymbol{x} \in X\}$. Moreover, since the state set $X$ is given, it is proportional to $\prod_i \sigma_i(\hat{U})$ ($\sigma_i(\hat{U})$ is the $i$th singular value of $\hat{U}$) and equivalently, to $(\sqrt{n}/2)^{(n/2)}\sqrt{\det(M)}$. Therefore, by finding the positive definite matrix $M$ that minimizes $\det(M)$ subject to the conditions (23), one can design a suboptimal finite abstraction (suboptimality is due to the fact that (23) is a sufficient condition). This problem is transformed into a tractable class of mathematical programming problem by a procedure described below. First, we show that the conditions (23) can be transformed into linear matrix inequality (LMI) conditions as long as $\lambda$ is fixed. First, define

$$N = M^{-1}, \, G = FM^{-1} \tag{28}$$

and by multiplying each equation of (23) with $N$ from both sides, we obtain

$$\begin{aligned} &\lambda^2 N - (AN+BG)^{\mathrm{T}}M(AN+BG) \geq O, \\ &N - (CN)^{\mathrm{T}}(CN)/((1-\lambda)^2\epsilon_y^2) \geq O, \\ &N - G^{\mathrm{T}}G/((1-\lambda)^2\epsilon_u^2) \geq O. \end{aligned}$$

Taking Schur complements, we obtain

$$\begin{aligned} &\begin{bmatrix} \lambda^2 N & (AN+BG)^{\mathrm{T}} \\ (AN+BG) & N \end{bmatrix} \geq O, \\ &\begin{bmatrix} N & (CN)^{\mathrm{T}} \\ CN & (1-\lambda)^2\epsilon_y^2 I \end{bmatrix} \geq O, \\ &\begin{bmatrix} N & G^{\mathrm{T}} \\ G & (1-\lambda)^2\epsilon_u^2 I \end{bmatrix} \geq O, \end{aligned} \tag{29}$$

which are LMI conditions of $N$ and $G$. Therefore, for a fixed $\lambda$, the optimization of F and M reduces to the following determinant-maximization problem under LMI constraints.

$$\text{maximize } \det(N) \;\text{ sub.to }\; (29).$$

This problem can be solved by a numerical solver SDPT3 [16]. So far, the problem is dependent on the scalar parameter $\lambda$, which also must be optimized. In order to obtain the optimal value of $\lambda$, we perform a one-dimensional search in the interval $(0, 1)$, iteratively solving the corresponding determinant-maximization problem, and find the value whose associated $\det(M)$ is the smallest.

## 4. APPLICATION TO OPTIMAL CONTROL PROBLEM

In this section, we show how to construct a suboptimal solution to a class of optimal control problem, taking advantage of an approximately bisimilar abstraction of the original plant model.

*4.1 Construction of suboptimal control*

To begin with, let us define a finite horizon optimal control problem on the discrete-time system $\Sigma$.

*Problem 2.* (Finite horizon optimal control problem). Suppose that for the system $\Sigma\langle X, U, Y, f, h\rangle$, the initial state $\boldsymbol{x}_0 \in X$ and the horizon length $N$ are given. Then, find the control input sequence $\bar{\boldsymbol{u}} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N-1}\}$ that minimizes the cost function

$$J(\bar{\boldsymbol{u}}) = \sum_{t=0}^{N-1} \phi_t(\boldsymbol{u}_t, \, \boldsymbol{y}_t) + \phi_N(\boldsymbol{y}_N) \tag{30}$$

while satisfying the constraints

$$\begin{aligned} &\boldsymbol{u}_t \in \mathcal{C}_u \quad (t \in \{0, 1, \ldots, N-1\}) \\ &\boldsymbol{y}_t \in \mathcal{C}_y \quad (t \in \{0, 1, \ldots, N\}) \end{aligned} \tag{31}$$

where $\{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ is given by (1)(2).

We denote this problem by $\mathcal{P}\langle \Sigma, \boldsymbol{x}_0, \mathcal{C}_u, \mathcal{C}_y, J\rangle$. Further, we assume that $J(\bar{\boldsymbol{u}})$ is smooth. For later use, we define

$$\begin{aligned} L_t^u &:= \max_{\boldsymbol{u}\in U, \, \boldsymbol{y}\in Y} \left\| \frac{\partial}{\partial \boldsymbol{u}} \phi_t(\boldsymbol{u}, \boldsymbol{y}) \right\| \quad (t = 0, 1, \ldots, N-1), \\ L_t^y &:= \max_{\boldsymbol{u}\in U, \, \boldsymbol{y}\in Y} \left\| \frac{\partial}{\partial \boldsymbol{y}} \phi_t(\boldsymbol{u}, \boldsymbol{y}) \right\| \quad (t = 0, 1, \ldots, N), \end{aligned}$$

$L^u := \sum_{t=0}^{N-1} L_t^u$, and $L^y := \sum_{t=0}^{N} L_t^y$.

Our objective is to construct a suboptimal solution to the problem

$$\mathcal{P} := \mathcal{P}\langle \Sigma, \boldsymbol{x}_0, \mathcal{C}_u, \mathcal{C}_y, J\rangle. \tag{32}$$

This can be done by the following procedure:

First, we design an approximately bisimilar abstraction $\hat{\Sigma}\langle \hat{X}, U, Y, \hat{f}, \hat{h}\rangle$ of the original plant model $\Sigma$, together with its associated relation $R$. As mentioned before, the relation $R$ must satisfy (11). Next, we define a new optimal control problem on $\hat{\Sigma}$, defined as

$$\mathcal{P}_1 := \mathcal{P}\langle \hat{\Sigma}, \hat{\boldsymbol{x}}_0, \mathrm{int}(\mathcal{C}_u, \epsilon_u), \mathrm{int}(\mathcal{C}_y, \epsilon_y), J\rangle. \tag{33}$$

Here, $\hat{\boldsymbol{x}}_0$ is an element of $\hat{X}$ that satisfies $(\boldsymbol{x}_0, \hat{\boldsymbol{x}}_0) \in R$. The condition (11) on $R$ guarantees the existence of such $\hat{\boldsymbol{x}}_0$. Moreover, the symbol $\mathrm{int}(\mathcal{C}, \epsilon)$ denotes set operation defined by

$$\mathrm{int}(\mathcal{C}, \epsilon) := \{\boldsymbol{v} \in \mathcal{C} \,|\, \mathcal{B}(\boldsymbol{v}, \epsilon) \subseteq \mathcal{C}\}, \tag{34}$$

which returns a subset of $\mathcal{C}$ obtained by shrinking $\mathcal{C}$ by the amount $\epsilon$. Here, $\mathcal{B}(\boldsymbol{v}, \epsilon)$ is a closed ball with the radius $\epsilon$

centered at $\boldsymbol{v}$. We denote by $\bar{\boldsymbol{u}}_1^*$ the optimal solution to $\mathcal{P}_1$. It goes without saying that $\mathcal{P}_1$ should be easier to solve than the original problem $\mathcal{P}$.

As a final step, we generate a control input sequence of $\Sigma$ that simulates $\bar{\boldsymbol{u}}_1^*$ starting from $\boldsymbol{x}_0$. Let us denote this by $\tilde{\boldsymbol{u}}_1^*$. In the case of $\Sigma_{\rm L}$ and $Q(\Sigma_{\rm L})$ described in Section 3, the simulating input is given by $\boldsymbol{u}_t = \hat{\boldsymbol{u}}_t + F(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t)$, where $\hat{\boldsymbol{u}}_t$ is an element of $\bar{\boldsymbol{u}}_1^*$ corresponding to the control input at time $t$, and $\hat{\boldsymbol{x}}_t$ is the state of $\hat{\Sigma}$ driven by $\bar{\boldsymbol{u}}_1^*$ from $\hat{\boldsymbol{x}}_0$.

The next theorem provides a condition for $\tilde{\boldsymbol{u}}_1^*$ to be a feasible solution to $\mathcal{P}$ and an upper bound of the cost $J(\tilde{\boldsymbol{u}}_1^*)$. As a preparation, we need to define yet one more optimal control problem on $\Sigma$:

$$\mathcal{P}_2 := \mathcal{P}\langle \Sigma, \boldsymbol{x}_0, \mathrm{int}(\mathcal{C}_u, 2\epsilon_u), \mathrm{int}(\mathcal{C}_y, 2\epsilon_y), J\rangle. \tag{35}$$

We denote by $\bar{\boldsymbol{u}}_2^*$ the optimal solution to $\mathcal{P}_2$.

*Theorem 2.*
Suppose the two systems $\Sigma\langle X, U, Y, f, h\rangle$ and $\hat{\Sigma}\langle \hat{X}, U, Y, \hat{f}, \hat{h}\rangle$ are $(\epsilon_u, \epsilon_y)$-approximately bisimilar with respect to $R \subset X \times \hat{X}$, and $R$ satisfies (11). Moreover, consider the three optimal control problems $\mathcal{P}$, $\mathcal{P}_1$, and $\mathcal{P}_2$ defined by (32), (33), and (35), respectively. Then the following statements hold:
i) If $\mathcal{P}_2$ is feasible, then $\mathcal{P}_1$ is feasible.
ii) If $\mathcal{P}_1$ is feasible, then $\mathcal{P}$ is feasible.
iii) The cost of simulating trajectory $\tilde{\boldsymbol{u}}_1^*$ is upper-bounded by the following inequality;

$$J(\tilde{\boldsymbol{u}}_1^*) \le J(\bar{\boldsymbol{u}}_2^*) + 2(L^u\epsilon_u + L^y\epsilon_y). \tag{36}$$

Proof) The statement i) and ii) are straightforward from the definition of approximate bisimulation.

To prove the statement iii), define $\tilde{\boldsymbol{u}}_2^*$ as a trajectory simulating $\bar{\boldsymbol{u}}_2^*$ on $\hat{\Sigma}$ from $\hat{\boldsymbol{x}}_0$. From approximate bisimilarity, $\tilde{\boldsymbol{u}}_1^*$ is a feasible solution to $\mathcal{P}$ and $\tilde{\boldsymbol{u}}_2^*$ is a feasible solution to $\mathcal{P}_1$. Moreover,

$$J(\tilde{\boldsymbol{u}}_1^*) - J(\bar{\boldsymbol{u}}_1^*) \le L^u\epsilon_u + L^y\epsilon_y,$$
$$J(\tilde{\boldsymbol{u}}_2^*) - J(\bar{\boldsymbol{u}}_2^*) \le L^u\epsilon_u + L^y\epsilon_y$$

hold. On the other hand, the optimality of $\bar{\boldsymbol{u}}_1^*$ implies

$$J(\bar{\boldsymbol{u}}_1^*) \le J(\tilde{\boldsymbol{u}}_2^*).$$

Therefore (36) holds.

□

### 4.2 Computation of optimal control on finite abstraction

We briefly show that for a finite automaton, the solution to the optimal control problem can be efficiently computed. Let us consider a finite automaton $Q(\Sigma)$ induced by the state quantization of a discrete-time system $\Sigma$. We use the notation for quantization function given in (12).

Our goal here is to solve the optimal control problem

$$\mathcal{P}\langle Q(\Sigma), \boldsymbol{x}_0 \in \mathcal{X}, \mathcal{C}_u, \mathcal{C}_y, J\rangle.$$

Recall that although the state of $Q(\Sigma)$ is discrete, the input is still continuous. First, consider a "small" continuous optimization problem as defined below.

*Problem 3. 1-step Optimal Control Problem*
For given $i, j \in [1 : |\mathcal{X}|]$ and $t \in [0 : N]$,
minimize $\phi_t(h(\boldsymbol{x}_i, \boldsymbol{u}))$ subject to $\boldsymbol{u} \in \mathcal{U}_{ij}$, $\boldsymbol{u} \in \mathcal{C}_u$.

The solution to this problem, if exists, moves the state from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ at time $t$ in one step, with the minimal cost while respecting the constraints. Let us denote the optimal solution to this problem by $\boldsymbol{u}_t^{ij}$ and its corresponding cost by $J_t^{ij}$. The number of all possible combinations of $(i, j, t)$, which is $N|\mathcal{X}|^2$, could be considerably large but finite. If the state transition relation is sparse, which is often the case, the actual number of combinations could be much smaller. Therefore, we shall compute Problem 3 for every combination of $(i, j, t)$ and store the result (feasibility, and if feasible, $\boldsymbol{u}_t^{ij}$ and $J_t^{ij}$) into a database. Using these expressions, the original problem can be regarded as the problem finding a state sequence $\bar{s} = \{s_0, s_1, \ldots, s_N\}$ which satisfies the constraint

$$\boldsymbol{y}_{s_t} \in \mathcal{C}_y \quad (t = 0, 1, \ldots, N) \tag{37}$$

and minimizes the cost function

$$J(\bar{s}) = \sum_{t=0}^{N-1} J_t^{s_t\, s_{t+1}} + \phi_N(\boldsymbol{y}_{s_N}). \tag{38}$$

This problem is a path-planning problem over a directed graph with finite number of nodes, and therefore it is solvable by various efficient searching algorithms.

It should be noted that not every kind of control problem is transformed into a tractable planning problem. One such example is a problem with "liveness" constraints, in which the system is required to visit every state infinitely often. Also, extra treatment might be needed for infinite-horizon cases.

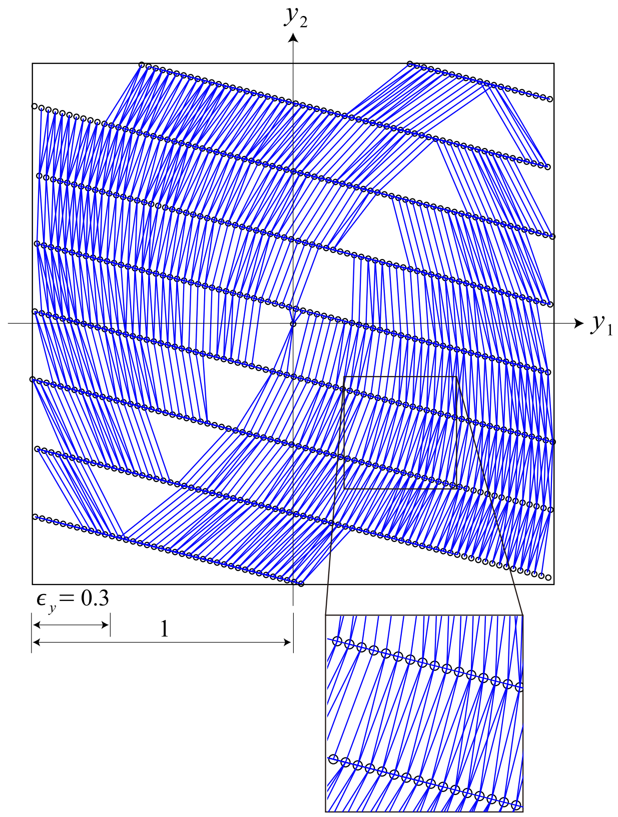



Fig. 4. Finite abstraction of sample system.

## 5. NUMERICAL EXAMPLES

This section shows simple examples. Consider a 2-state 1-input discrete-time linear system $\Sigma_{\rm L}$ with the following parameters:

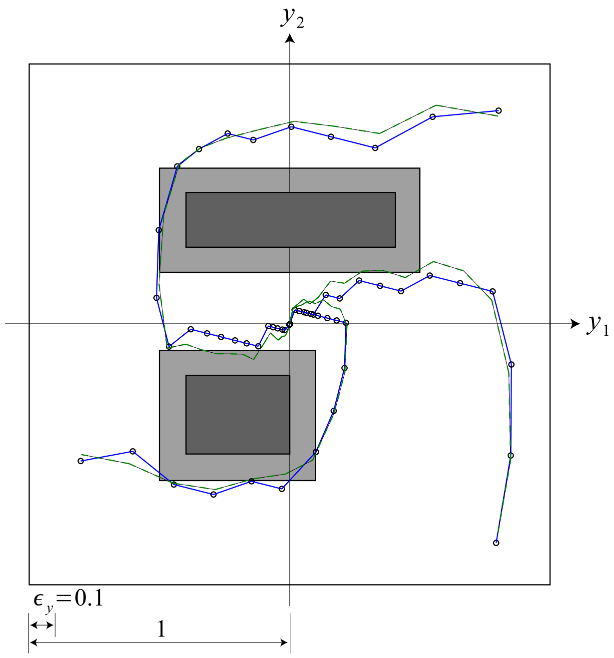



Fig. 5. Optimal trajectory on $Q(\Sigma_\mathrm{L})$ (solid), and simulated trajectory on $\Sigma_\mathrm{L}$ (dash).

$$X = [-1,\ 1] \times [-1,\ 1],\ U = [-1,\ 1],\ Y = X$$
$$A = \begin{bmatrix} 0.88 & -0.17 \\ 0.17 & 0.88 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

For this system, we designed an approximately bisimilar finite abstraction of $Q(\Sigma_\mathrm{L})$ with $\epsilon_u = 0.3$, $\epsilon_y = 0.3$ using the method described in the previous section. Fig. 4 shows the obtained finite automaton $Q(\Sigma_\mathrm{L})$. In Fig. 4, Small circles represent the measurement values ($\boldsymbol{y}_i$s) corresponding to each of the states of $Q(\Sigma_\mathrm{L})$, and lines connecting circles depicts possible state transitions.

The next example demonstrates an application to finite-horizon optimal control problem with non-convex constraints (Fig. 5). The output-constraint set is given by a non-convex set $\mathcal{C}_y = Y \backslash \mathcal{D}$ where

$$\mathcal{D} = [-0.4, 0.4] \times [0.3, 0.5] \cup [-0.4, 0.0] \times [-0.5, -0.2]$$

denotes the unsafe (entering-prohibited) region (drawn as gray areas in the figure). The input constraint is given by

$$\mathcal{C}_u = [-0.1,\ 0.1].$$

The cost function is simply set as

$$J(u, \boldsymbol{y}) = \sum_{t=0}^{N-1} \left[ u_t^2 + ||\boldsymbol{y}_t||_2^2 \right] + ||\boldsymbol{y}_N||_2^2$$

where $N = 25$. In this experiment, the finite abstraction is computed with the precisions $\epsilon_y = 0.1$ and $\epsilon_u = 0.1$. Three different initial states: $[-0.8; -0.5]$, $[0.8; 0.8]$, and $[0.8; -0.8]$ are specified. The simulation result is shown in Fig. 5. In the figure, solid lines represent the optimal paths computed on the finite automaton, and dashed lines represent the trajectories obtained by simulating the optimal paths (solid lines) on the original system.

## 6. CONCLUSION

This paper discussed the finite abstraction problem of stabilizable discrete-time linear systems, using the framework of approximate bisimulation. The main idea is to express the finite abstraction as a state-quantized form of the original continuous-state system. This recasts approximate bisimulation into trajectory tracking of two systems under disturbances due to quantization error.

## REFERENCES

[1] R. Milner : Communication and Concurrency, Prentice-Hall, 1989.

[2] A. Girard and G.J. Pappas : Approximation metrics for discrete and continuous systems, IEEE Transactions on Automatic Control, 52(5), pp782-798, 2007.

[3] A. Girard and G.J. Pappas : Approximate bisimulations for constrained linear systems, 44th IEEE Conference on Decision and Control and European Control Conference, pp 4700-4705, Seville, Spain, December 2005.

[4] A. Girard : Approximately Bisimilar Finite Abstractions of Stable Linear Systems, Hybrid Systems: Computation and Control, vol 4416 in LNCS, pp 231-244, Springer, 2007.

[5] A.A. Julius, A. Girard and G.J. Pappas : Approximate bisimulation for a class of stochastic hybrid systems, in the Proc. American Control Conference, pp 4724-4729, Minneapolis, USA, 2006.

[6] A.A. Julius and G.J. Pappas: Approximate Equivalence and Approximate Synchronization of Metric Transition Systems, in the Proc. 45th IEEE Conf. Decision and Control 2006, San Diego, USA.

[7] P. Tabuada : Approximate Simulation Relations and Finite Abstractions of Quantized Control Systems, Hybrid Systems: Computation and Control, vol 4416 in LNCS, pp 529-542, Springer, 2007.

[8] G. Pola, A. Girard and P. Tabuada: Symbolic models for nonlinear control systems using approximate bisimulation, 46th IEEE Conference on Decision and Control, New Orleans, Lousiana, December 2007, to appear.

[9] M.E. Broucke : A geometric approach to bisimulation and verification of hybrid systems, Hybrid Systems: Computation and Control (HSCC99), Lecture Notes in Computer Science 1569, Springer-Verlag, pp.61-75, 1999.

[10] M.E. Broucke, M.D. Di Benedetto, S. Di Gennaro, and A. Sangiovanni-Vincentelli : Theory of optimal control using bisimulations, Hybrid Systems: Computation and Control (HSCC00), Springer-Verlag, LNCS 1790, pp. 89-102, 2000.

[11] M.E. Broucke, M.D. Di Benedetto, S. Di Gennaro, and A. Sangiovanni-Vincentelli : Optimal control using bisimulations: Implementation, Hybrid Systems: Computation and Control (HSCC01), Springer-Verlag, LNCS 2034, pp. 175-188, 2001.

[12] H.L. Hagenaars, J. Imura and H. Nijmeijer : Approximate continuous-time optimal control in obstacle avoidance by time/space discretization of non-convex state constraints, Proc. of IEEE Conf. on Control Applications, pp.878-883, 2004.

[13] J. Imura and H. Matsushima : Simultaneous optimization of continuous control inputs and discrete state waypoints, 9th International Workshop on Hybrid Systems: Computation and Control (HSCC 2006), J.Hespanha and A.Tiwari (Eds.) LNCS 3927 Springer-Verlag, pp.302-317, 2006.

[14] Y. Tazaki, J. Imura : Graph-based Model Predictive Control of a Planar Bipedal Walker, 17th International Symposium on Mathematical Theory of Networks and Systems, pp.128-133, Kyoto, Japan, 2006.

[15] J. Lunze, B. Nixdorf and J. Schroder : Deterministic Discrete-event Representations of Linear Continuous-variable Systems, Automatica, Vol. 35, pp. 395-406, 1999.

[16] K.C. Toh, M.J. Todd and R.H. Tutuncu: SDPT3 - a MATLAB software for semidefinite-quadratic-linear programming, `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`

[17] K. Tsumura : Approximation of Discrete Time Linear Systems via Bit Length of Memory, Proc. of the 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan, July 24-28, 2006.

[18] K. Tsumura : Stabilization of Linear Systems by Bit-Memory Controllers under Constraints of Bit-Length, 45th IEEE Conference on Decision and Control, pp5507-5512 , San Diego, CA, USA, December, 2006.