Efficient Data-Driven Estimation of Passivity Properties

Masaya Tanemura and Shun-ichi Azuma

***Abstract*—In this letter, the estimation of passivity based on data is considered. We extend an existing estimation method that is based on iterative input–output experiments and can determine the passivity of systems, but requires many experiments for each iteration. Therefore, we propose a method that reduces the number of experiments. Some of the measurements for the update of the input is omitted in the method, and as a result, the number of experiments is halved.**

***Index Terms*—Data-driven estimation, passivity, gradient methods.**

I. Introduction

PASSIVITY is one of the most useful properties and plays an important role in control engineering. Many researchers have studied passivity [1]–[3], and methods based on passivity have been applied to a wide range of control areas [4]–[6]. When using passivity properties to design a control system, it is useful to know the passivity level of the controlled plant. However, generally, a mathematical model is required to determine the passivity level. If the passivity level could be directly determined from the input and output data, then the passivity-based approach would be used more extensively in the big data era.

Recently, the data-driven estimation approach for passivity has been studied, for example, in [7]–[9]. In [7] and [8], estimation methods for nonlinear systems were established. In this letter, it is assumed that the control input belongs to a limited space. Then, input sequences over the entire limited space are applied to the system and the passivity of the system is estimated using the input-output data. However, the methods require many input-output data tuples to estimate passivity. By contrast, a promising framework for linear systems was proposed in [9]. The method is based on the gradient of the passivity level and linearity of systems, and it estimates the passivity level more effectively using iterative input-output experiments. However, the update formula of the gradient method is composed of four experiment data tuples for each iteration. For example, in base-isolated buildings [10], excitation experiments have to be conducted in which the force exerted by actuators is added to based floor. However, it is costly to perform these large-scale experiments many times in terms of time and money. Therefore, a more efficient method, that is, with fewer experiments, would be practical and useful.

Manuscript received August 7, 2018; revised October 31, 2018; accepted November 23, 2018. Date of publication December 18, 2018; date of current version January 4, 2019. This work was supported in part by the Japan Science and Technology Agency, Core Research for Evolutional Science and Technology, Japan, under Grant JPMJCR15K1. Recommended by Senior Editor G. Yin. *(Corresponding author: Masaya Tanemura.)*

The authors are with the Graduate School of Engineering, Nagoya University, Nagoya 464-8603, Japan (e-mail: masaya.tanemura@mae.nagoya-u.ac.jp; shunichi.azuma@mae.nagoya-u.ac.jp).

Digital Object Identifier 10.1109/LCSYS.2018.2887241

In this letter, we propose estimation with fewer experiments. First, we establish an algorithm that requires only three-quarters of the time taken for experiments compared with the conventional method by focusing on the quadratic form in the update formula of the input. Next, we indicate that one measurement can be composed of previous input-output data tuples in each iteration, and develop an algorithm with half the number of experiments.

This letter is organized as follows: In Section II, we present the problem formulation and an iterative method. In Section III, we explain the update method based on the gradient method proposed in [9] and the number of measurements for the iteration. In Section IV, we propose a method that reduces the complexity of the update formula using the quadratic form. In Section V, we propose a method that uses previous input-output data to compose one measurement. In Section VI, we conclude this letter.

Notation: Let $\mathbf{R}$ and $\mathbf{Z}_+$ be the set of real numbers and set of non-negative integers, respectively. Let $\mathcal{L}_2$ denote the set of discrete-time signals $u(t) : \mathbf{Z}_+ \to \mathbf{R}$ that satisfies

$$\sum_{t=0}^{\infty} u^2(t) < \infty. \tag{1}$$

For vector x, let $\|x\|$ represent the Euclidean norm of x.

II. Finite-Time Passivity

A. Problem Formulation

We consider the discrete-time SISO linear system given by the minimal realization

$$G_d : \begin{cases} x(t+1) = Ax(t) + bu(t), \\ y(t) = cx(t) + du(t), \end{cases} \tag{2}$$

where $x(t) \in \mathbf{R}^n$ is the state variable, $u(t) \in \mathbf{R}$ is the input, and $y(t) \in \mathbf{R}$ is the output.

Now, we introduce the notion of passivity described in [11]. System G_d of (2) with $x(0) = 0$ is said to be input passive if

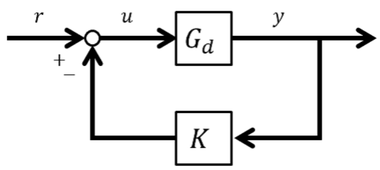

Fig. 1. Block diagram of feedback control.

there exists $\nu_\infty \in \mathbf{R}$ such that

$$\sum_{t=0}^{\infty} y(t)u(t) \geq \nu_\infty \sum_{t=0}^{\infty} u^2(t) \tag{3}$$

for all $(u(0), u(1), \ldots) \in \mathcal{L}_2$. The largest value of ν_∞ that satisfies (3) is called the input passivity index, and it indicates the passivity level. If the passivity index is known, then constant feedback stabilizes the closed loop system of Fig. 1 [1].

Next, we introduce the notion of finite-time input passivity. System G_d of (2) with $x(0) = 0$ is said to be *finite-time input passive* if there exists $\nu \in \mathbf{R}$ such that

$$\sum_{t=0}^{N-1} y(t)u(t) \geq \nu \sum_{t=0}^{N-1} u^2(t) \tag{4}$$

for a given $N \in \mathbf{Z}_+$ and all inputs $(u(0), u(1), \ldots, u(N-1)) \in \mathbf{R}^N$. When $N \to \infty$, this notion corresponds to the input passivity notion in (3).

In this letter, we consider the problem of estimating the largest value ν^* that satisfies (4) for all inputs $(u(0), u(1), \ldots, u(N-1)) \in \mathbf{R}^N$, which is formulated as follows:

Problem 1: Suppose that N is given and assume that

1) the mathematical model of system G_d is unknown, that is, matrices A, b, c, and d in (2) are unknown; and
2) we can apply any input sequence $u(0), u(1), \ldots, u(N-1)$ to system G_d and observe the corresponding output sequence $y(0), y(1), \ldots, y(N-1)$.

Then, determine the largest value ν^* that satisfies (4) for all inputs $(u(0), u(1), \ldots, u(N-1)) \in \mathbf{R}^N$. ■

B. Iterative Computation of Input Passivity With Information About G_d

Passivity is an input-output property, therefore, we introduce the input-output representation of (2) as follows:

$$G_d : y(t) = \sum_{\tau=0}^{t} g_\tau u(t-\tau), \tag{5}$$

subject to $x(0) = 0$, where

$$g_\tau = \begin{cases} d, & \tau = 0 \\ cA^{\tau-1}b, & \tau = 1,\ 2, \ldots \end{cases} \tag{6}$$

denotes the impulse response sequence. For time sequence $0,\ 1, \ldots,\ N-1$, system G_d is expressed as

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ g_{N-1} & g_{N-2} & g_{N-3} & \cdots & g_0 \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N-1) \end{bmatrix}. \tag{7}$$

This is simply denoted by

$$\bar{y} = G\bar{u}, \tag{8}$$

where

$$\bar{u} = \begin{bmatrix} u(0) & u(1) & \cdots & u(N-1) \end{bmatrix}^{\mathrm{T}}, \tag{9}$$

$$\bar{y} = \begin{bmatrix} y(0) & y(1) & \cdots & y(N-1) \end{bmatrix}^{\mathrm{T}}, \tag{10}$$

and $G \in \mathbf{R}^{N\times N}$ is the matrix composed of $g_0, g_1, \ldots, g_{N-1}$.

Next, we briefly introduce an iterative method to estimate ν^* when information about G is available. Applying (8) to (4) leads to

$$\nu \leq \frac{\bar{u}^{\mathrm{T}} G\bar{u}}{\bar{u}^{\mathrm{T}}\bar{u}} \tag{11}$$

subject to $\|\bar{u}\| \neq 0$. Thus,

$$\nu^* = \min_{\|\bar{u}\|\neq 0} \frac{\bar{u}^{\mathrm{T}} G\bar{u}}{\bar{u}^{\mathrm{T}}\bar{u}} \tag{12}$$

holds for the largest value ν^*. Because

$$\bar{u}^{\mathrm{T}} G\bar{u} = \frac{1}{2}\bar{u}^{\mathrm{T}} G\bar{u} + \frac{1}{2}(\bar{u}^{\mathrm{T}} G\bar{u})^{\mathrm{T}}, \tag{13}$$

$$= \frac{1}{2}\bar{u}^{\mathrm{T}}(G + G^{\mathrm{T}})\bar{u}, \tag{14}$$

Equation (12) is rewritten as

$$\nu^* = \min_{\|\bar{u}\|\neq 0} \frac{1}{2}\frac{\bar{u}^{\mathrm{T}}(G + G^{\mathrm{T}})\bar{u}}{\bar{u}^{\mathrm{T}}\bar{u}}, \tag{15}$$

which is represented as

$$\nu^* = \min_{\bar{u}\in \mathbf{S}^{N-1}} \rho(\bar{u}), \tag{16}$$

with Rayleigh quotient

$$\rho(\bar{u}) = \frac{1}{2}\frac{\bar{u}^{\mathrm{T}}(G + G^{\mathrm{T}})\bar{u}}{\bar{u}^{\mathrm{T}}\bar{u}} \tag{17}$$

and $\mathbf{S}^{N-1} = \{\bar{u} \in \mathbf{R}^N | \|\bar{u}\| = 1\}$. Note that Rayleigh quotient $\rho : \mathbf{R}^N \setminus \{0\} \to \mathbf{R}$ is scale-invariant, that is,

$$\rho(\alpha\bar{u}) = \rho(\bar{u}),\ \ \forall\alpha \in \mathbf{R}. \tag{18}$$

Using (16), we can construct an iterative method to solve Problem 1. More specifically, the method is given by the update formula

$$\bar{u}^{(k+1)} = R(\bar{u}^{(k)} - \delta^{(k)}\nabla\rho(\bar{u}^{(k)})), \tag{19}$$

where the mapping $R : \mathbf{R}^N \to \mathbf{R}^N$ is defined as

$$R(\bar{u}) = \frac{\bar{u}}{\|\bar{u}\|}, \tag{20}$$

and $\nabla\rho$ is the gradient vector of ρ with respect to $\bar{u}$, and $\delta^{(k)}$ is the step size. Superscript (k) denotes the number of the update (19). This yields a sequence $\bar{u}^{(0)}$, $\bar{u}^{(1)}, \ldots$ on the sphere $\mathbf{S}^{N-1}$, and ν^* is given by $\rho(\bar{u}^{(k)})$ for a sufficiently large number k.

Then, it follows from (17) that the gradient vector $\nabla\rho(\bar{u}^{(k)})$ is given by

$$\nabla\rho(\bar{u}^{(k)}) = (G + G^{\mathrm{T}})\bar{u}^{(k)} - 2\rho(\bar{u}^{(k)})\bar{u}^{(k)}. \tag{21}$$

By contrast, step size $\delta^{(k)}$ is chosen so as to increase the convergence rate of the estimation. For instance, the step size is given by

$$\delta^{(k)} = \arg\min_{\delta \in \mathbf{R}} \rho(R(\bar{u}^{(k)} - \delta \nabla \rho(\bar{u}^{(k)}))) \tag{22}$$

to minimize ρ in each iteration [9], where $\delta \in \mathbf{R}$ is a decision variable. Because $\rho(R(\bar{u}^{(k)} - \delta \nabla \rho(\bar{u}^{(k)}))) = \rho(\bar{u}^{(k)} - \delta \nabla \rho(\bar{u}^{(k)}))$ from the scale-invariance of $\rho(\bar{u})$, $\delta^{(k)}$ is computed by minimizing

$$\rho(\bar{u}^{(k)} - \delta \nabla \rho^{(k)}) = \frac{1}{2} \frac{\begin{bmatrix} 1 \\ \delta \end{bmatrix}^{\mathrm{T}} M_1(\bar{u}^{(k)}, \nabla \rho^{(k)}) \begin{bmatrix} 1 \\ \delta \end{bmatrix}}{\begin{bmatrix} 1 \\ \delta \end{bmatrix}^{\mathrm{T}} M_2(\bar{u}^{(k)}, \nabla \rho^{(k)}) \begin{bmatrix} 1 \\ \delta \end{bmatrix}}, \tag{23}$$

with respect to $\delta \in \mathbf{R}$, where

$$M_1(\bar{u}^{(k)}, \nabla \rho^{(k)}) = \begin{bmatrix} (\bar{u}^{(k)})^{\mathrm{T}}(G + G^{\mathrm{T}})\bar{u}^{(k)} & -(\bar{u}^{(k)})^{\mathrm{T}}(G + G^{\mathrm{T}})\nabla \rho^{(k)} \\ -(\nabla \rho^{(k)})^{\mathrm{T}}(G + G^{\mathrm{T}})\bar{u}^{(k)} & (\nabla \rho^{(k)})^{\mathrm{T}}(G + G^{\mathrm{T}})\nabla \rho^{(k)} \end{bmatrix}, \tag{24}$$

$$M_2(\bar{u}^{(k)}, \nabla \rho^{(k)}) = \begin{bmatrix} (\bar{u}^{(k)})^{\mathrm{T}}\bar{u}^{(k)} & -(\bar{u}^{(k)})^{\mathrm{T}}\nabla \rho^{(k)} \\ -(\nabla \rho^{(k)})^{\mathrm{T}}\bar{u}^{(k)} & (\nabla \rho^{(k)})^{\mathrm{T}}\nabla \rho^{(k)} \end{bmatrix}, \tag{25}$$

and $\nabla \rho^{(k)}$ represents $\nabla \rho(\bar{u}^{(k)})$. Note that (23) is the Rayleigh quotient, hence $\delta^{(k)}$ is obtained by solving the generalized eigenvalue problem.

C. Example

In this section, we demonstrate the update formula (19) for estimating ν^*. We consider the system

$$G_c : \begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -50 & -100 & -5 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} 40 & 50 & 5 \end{bmatrix} x(t) + 0.5u(t). \end{cases} \tag{26}$$

Let G_d be the discrete-time model given the zero-order hold with sampling period $t_s = 0.05$ s. We set $N = 200$ and

$$\bar{u}^{(0)} = \frac{1}{\|\bar{h}\|} \bar{h}, \tag{27}$$

$$\bar{h} = [h(0) \cdots h(N-1)]^{\mathrm{T}}, \tag{28}$$

$$h(i) = \sin(2\pi i t_s), \quad i = 0, \ldots, N-1. \tag{29}$$

Then, we perform 40 updates using the update formula (19). Fig. 2 shows the update of $\rho(\bar{u}^{(k)})$. We calculate the input passivity index in (3), and it is shown as the dashed line in Fig. 2. From this result, we see that $\rho(\bar{u}^{(k)})$ converges to the input passivity index.

III. Data-Driven Computation Method

In this section, we introduce the data-driven method proposed in [9] to execute the algorithm (19) under the two assumptions in Problem 1.

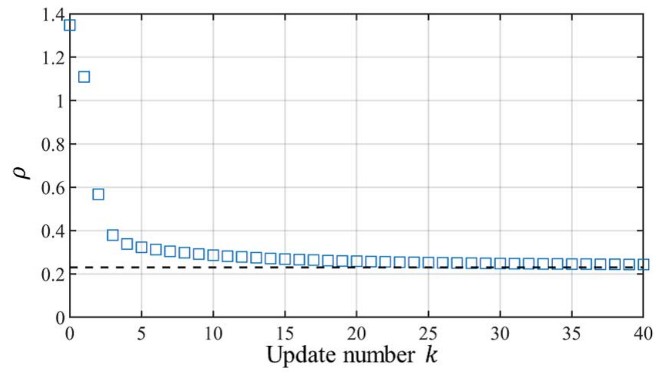

Fig. 2. Results of the iterative computation of $\rho(\bar{u}^{(k)})$.

A. Data-Driven Computation of the Gradient Vector

To execute the algorithm (19), it is necessary to compute the gradient vector of ρ with no information about G; that is, the gradient vector is given by (21), but $\nabla \rho(\bar{u}^{(k)})$ is not directly obtained using this formula. We show that $\nabla \rho(\bar{u}^{(k)})$ can be computed using two measurements of the system (5) in the following manner.

From (21), it is necessary to obtain the values of $G\bar{u}^{(k)}$ and $G^{\mathrm{T}}\bar{u}^{(k)}$ to compute $\nabla \rho(\bar{u}^{(k)})$. The former can be easily obtained as the output $\bar{y}$ for the input $\bar{u}^{(k)}$; however, the latter cannot. Therefore, we introduce the following involutory matrix:

$$P = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & 1 & 0 \\ 0 & \iddots & \iddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbf{R}^{N \times N}. \tag{30}$$

Because $G^{\mathrm{T}} = PGP$, $G^{\mathrm{T}}\bar{u}^{(k)}$ is obtained by applying the input $P\bar{u}^{(k)}$ to G and reversing the resulting output, that is, $PGP\bar{u}^{(k)}$. This idea is formalized as follows [9].

Lemma 1: Let $\bar{y}_1^{(k)}$ be the output measurement of the system (5) for the input sequence $\bar{u}^{(k)}$, that is, $\bar{y}_1^{(k)} = G\bar{u}^{(k)}$, and let $\bar{y}_2^{(k)}$ be that for the input sequence $P\bar{u}^{(k)}$, that is, $\bar{y}_2^{(k)} = GP\bar{u}^{(k)}$. Then, gradient vector $\nabla \rho(\bar{u}^{(k)})$ is given by

$$\nabla \rho(\bar{u}^{(k)}) = \bar{y}_1^{(k)} + P\bar{y}_2^{(k)} - \frac{(\bar{u}^{(k)})^{\mathrm{T}}(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)})}{(\bar{u}^{(k)})^{\mathrm{T}}\bar{u}^{(k)}} \bar{u}^{(k)}. \tag{31}$$

■

Note that (31) is composed of $\bar{u}^{(k)}$, $\bar{y}_1^{(k)}$, $\bar{y}_2^{(k)}$, and P. Thus, gradient vector $\nabla \rho(\bar{u}^{(k)})$ is computed using two sets of input-output data.

B. Data-Driven Computation of the Step Size

The step size is computed by minimizing (23). However, matrix M_1 in (23) consists of G, and thus M_1 cannot be computed without information about G. Lemma 2 provides a data-driven method for computing $\delta^{(k)}$ [9].

Lemma 2: Let $\bar{y}_3^{(k)}$ be the output measurement of the system (5) for the input sequence $\nabla \rho^{(k)}$, that is, $\bar{y}_3^{(k)} = G\nabla \rho^{(k)}$, and let $\bar{y}_4^{(k)}$ be that for the input sequence $P\nabla \rho^{(k)}$, that is,

Algorithm 1 Estimation of ν^*

$\bar{u}^{(0)}$ is arbitrarily determined.
$\bar{y}_1^{(0)} \leftarrow G\bar{u}^{(0)}$ ▷ Experiment
for $k = 0, 1, \cdots$ **do**
  $\bar{y}_2^{(k)} \leftarrow GP\bar{u}^{(k)}$ ▷ Experiment
  $\nabla\rho(\bar{u}^{(k)})$ is computed by (31).
  $\bar{y}_3^{(k)} \leftarrow G\nabla\rho(\bar{u}^{(k)})$ ▷ Experiment
  $\bar{y}_4^{(k)} \leftarrow GP\nabla\rho(\bar{u}^{(k)})$ ▷ Experiment
  $\delta^{(k)}$ is computed by (32).
  $\bar{u}^{(k+1)}$ is computed by (19).
  $\bar{y}_1^{(k+1)} \leftarrow G\bar{u}^{(k+1)}$ ▷ Experiment
end for
return $\rho_1(\bar{u}^{(k+1)}) = \dfrac{(\bar{y}_1^{(k+1)})^{\mathrm{T}}\bar{u}^{(k+1)}}{(\bar{u}^{(k+1)})^{\mathrm{T}}\bar{u}^{(k+1)}}$

$\bar{y}_4^{(k)} = GP\nabla\rho^{(k)}$. Then, $\delta^{(k)}$ is given by the solution of the generalized eigenvalue problem

$$M_1(\bar{u}^{(k)}, \nabla\rho^{(k)}, \bar{y}_1^{(k)}, \bar{y}_2^{(k)}, \bar{y}_3^{(k)}, \bar{y}_4^{(k)})\begin{bmatrix} 1 \\ \delta^{(k)} \end{bmatrix} = \lambda_{\min} M_2(\bar{u}^{(k)}, \nabla\rho^{(k)})\begin{bmatrix} 1 \\ \delta^{(k)} \end{bmatrix}, \tag{32}$$

where

$$M_1(\bar{u}^{(k)}, \nabla\rho^{(k)}, \bar{y}_1^{(k)}, \bar{y}_2^{(k)}, \bar{y}_3^{(k)}, \bar{y}_4^{(k)}) = \begin{bmatrix} (\bar{u}^{(k)})^{\mathrm{T}}(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)}) & -(\bar{u}^{(k)})^{\mathrm{T}}(\bar{y}_3^{(k)} + P\bar{y}_4^{(k)}) \\ -(\nabla\rho^{(k)})^{\mathrm{T}}(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)}) & (\nabla\rho^{(k)})^{\mathrm{T}}(\bar{y}_3^{(k)} + P\bar{y}_4^{(k)}) \end{bmatrix}, \tag{33}$$

and $\lambda_{\min} \in \mathbf{R}$ is the minimum eigenvalue of the generalized eigenvalue problem. ■

Note that (32) is composed of $\bar{u}^{(k)}$, $\nabla\rho^{(k)}$, $\bar{y}_1^{(k)}$, $\bar{y}_2^{(k)}$, $\bar{y}_3^{(k)}$, $\bar{y}_4^{(k)}$, and P. Thus, step size $\delta^{(k)}$ is computed using four sets of input-output data.

C. Number of Measurements in Each Iteration

From Lemmas 1 and 2, $\bar{u}^{(k+1)}$ is computed by the update formula (19) with (31) and (32), and as a result, we can obtain the estimated value of ν^* with input-output data $\bar{u}^{(k)}$ and $\bar{y}_1^{(k)}$, that is, $\rho(\bar{u}^{(k)})$ is computed from (14) as follows:

$$\rho(\bar{u}^{(k)}) = \frac{(\bar{y}_1^{(k)})^{\mathrm{T}}\bar{u}^{(k)}}{(\bar{u}^{(k)})^{\mathrm{T}}\bar{u}^{(k)}}, \tag{34}$$

which is shown in Algorithm 1. Then, the following result is obtained for the number of measurements in Algorithm 1.

Theorem 1: Algorithm 1 is executed for four measurements, $\bar{y}_1^{(k)}$, $\bar{y}_2^{(k)}$, $\bar{y}_3^{(k)}$, and $\bar{y}_4^{(k)}$, in each iteration; that is, the total number of measurements is $4k$ until the k-th iteration. ■

IV. Complexity Reduction of the Data-Driven Computation Method

A. Efficient Computation of the Step Size $\delta^{(k)}$

In this section, we provide an efficient calculation method for step size $\delta^{(k)}$ to reduce the total number of measurements.

Lemma 3: The definitions of the measurements $\bar{y}_1^{(k)}$, $\bar{y}_2^{(k)}$, and $\bar{y}_3^{(k)}$ are provided in Lemmas 1 and 2. Then, $\delta^{(k)}$ is given by the solution of the generalized eigenvalue problem

$$M_3(\bar{u}^{(k)}, \nabla\rho^{(k)}, \bar{y}_1^{(k)}, \bar{y}_2^{(k)}, \bar{y}_3^{(k)})\begin{bmatrix} 1 \\ \delta^{(k)} \end{bmatrix} = \lambda_{\min} M_2(\bar{u}^{(k)}, \nabla\rho^{(k)})\begin{bmatrix} 1 \\ \delta^{(k)} \end{bmatrix}, \tag{35}$$

where

$$M_3(\bar{u}^{(k)}, \nabla\rho^{(k)}, \bar{y}_1^{(k)}, \bar{y}_2^{(k)}, \bar{y}_3^{(k)}) = \begin{bmatrix} (\bar{u}^{(k)})^{\mathrm{T}}(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)}) & -(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)})^{\mathrm{T}}\nabla\rho^{(k)} \\ -(\nabla\rho^{(k)})^{\mathrm{T}}(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)}) & 2(\nabla\rho^{(k)})^{\mathrm{T}}\bar{y}_3^{(k)} \end{bmatrix}, \tag{36}$$

and $\lambda_{\min} \in \mathbf{R}$ is the minimum eigenvalue of the generalized eigenvalue problem.

Proof: Consider matrix M_1 in (24). First, the $(2, 1)$-th and $(1, 2)$-th elements are simply calculated using $\bar{y}_1^{(k)} = G\bar{u}^{(k)}$ and $\bar{y}_2^{(k)} = GP\bar{u}^{(k)}$ as

$$(\nabla\rho^{(k)})^{\mathrm{T}}(G + G^{\mathrm{T}})\bar{u}^{(k)} = (\nabla\rho^{(k)})^{\mathrm{T}}(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)}), \tag{37}$$

$$(\bar{u}^{(k)})^{\mathrm{T}}(G + G^{\mathrm{T}})\nabla\rho^{(k)} = (\bar{y}_1^{(k)} + P\bar{y}_2^{(k)})^{\mathrm{T}}\nabla\rho^{(k)}. \tag{38}$$

Next, the $(2, 2)$-th element is calculated using only $\bar{y}_3^{(k)} = G\nabla\rho^{(k)}$ as

$$(\nabla\rho^{(k)})^{\mathrm{T}}(G + G^{\mathrm{T}})\nabla\rho^{(k)} = 2(\nabla\rho^{(k)})^{\mathrm{T}}(G\nabla\rho^{(k)}), \tag{39}$$

$$= 2(\nabla\rho^{(k)})^{\mathrm{T}}\bar{y}_3^{(k)}, \tag{40}$$

because

$$(\nabla\rho^{(k)})^{\mathrm{T}}G\nabla\rho^{(k)} = \frac{1}{2}(\nabla\rho^{(k)})^{\mathrm{T}}G\nabla\rho^{(k)} + \frac{1}{2}((\nabla\rho^{(k)})^{\mathrm{T}}G\nabla\rho^{(k)})^{\mathrm{T}}, \tag{41}$$

$$= \frac{1}{2}(\nabla\rho^{(k)})^{\mathrm{T}}(G + G^{\mathrm{T}})\nabla\rho^{(k)}. \tag{42}$$

From the above, $M_1(\bar{u}^{(k)}, \nabla\rho^{(k)}, \bar{y}_1^{(k)}, \bar{y}_2^{(k)}, \bar{y}_3^{(k)}, \bar{y}_4^{(k)}) = M_3(\bar{u}^{(k)}, \nabla\rho^{(k)}, \bar{y}_1^{(k)}, \bar{y}_2^{(k)}, \bar{y}_3^{(k)})$ holds and step size $\delta^{(k)}$ can be computed using only three measurements: $\bar{y}_1^{(k)}$, $\bar{y}_2^{(k)}$, and $\bar{y}_3^{(k)}$. ■

Because matrix M_3 in (35) is composed of $\bar{u}^{(k)}$, $\nabla\rho^{(k)}$, $\bar{y}_1^{(k)}$, $\bar{y}_2^{(k)}$, $\bar{y}_3^{(k)}$, and P, the generalized eigenvalue problem (35) is computed using three sets of input-output data; that is, (37)–(40) enable us to solve the generalized eigenvalue problem in (35) without the measurement $\bar{y}_4^{(k)}$.

The method based on (19), (31), and (35) is referred to as Algorithm 2. For this algorithm, the following result is obtained.

Theorem 2: For Algorithm 2, the following statements hold.

(i) Let $\rho_1^{(k)}$ be $\rho(\bar{u}^{(k)})$ of Algorithm 1 with initial condition $\bar{u}^{(0)}$ and $\rho_2^{(k)}$ be $\rho(\bar{u}^{(k)})$ of Algorithm 2 with the same initial condition $\bar{u}^{(0)}$. Then,

$$\rho_1^{(k)} = \rho_2^{(k)} \tag{43}$$

for each $k \in \{0, 1, \ldots\}$.

Algorithm 2 Estimation of ν^*

1: $\bar{u}^{(0)}$ is arbitrarily determined.
2: $\bar{y}_1^{(0)} \leftarrow G\bar{u}^{(0)}$ ▷ Experiment
3: **for** $k = 0, 1, \cdots$ **do**
4: $\bar{y}_2^{(k)} \leftarrow GP\bar{u}^{(k)}$ ▷ Experiment
5: $\nabla\rho(\bar{u}^{(k)})$ is computed by (31).
6: $\bar{y}_3^{(k)} \leftarrow G\nabla\rho(\bar{u}^{(k)})$ ▷ Experiment
7: $\delta^{(k)}$ is computed by (35).
8: $\bar{u}^{(k+1)}$ is computed by (19).
9: $\bar{y}_1^{(k+1)} \leftarrow G\bar{u}^{(k+1)}$ ▷ Experiment
10: **end for**
11: **return** $\rho_2(\bar{u}^{(k+1)}) = \dfrac{(\bar{y}_1^{(k+1)})^{\mathrm{T}}\bar{u}^{(k+1)}}{(\bar{u}^{(k+1)})^{\mathrm{T}}\bar{u}^{(k+1)}}$

(ii) Algorithm 2 is executed for three values, $\bar{y}_1^{(k)}$, $\bar{y}_2^{(k)}$, and $\bar{y}_3^{(k)}$ in each iteration; that is, the total number of measurements is $3k$ until the k-th iteration.

Proof: The difference between Algorithms 1 and 2 is the calculation of step size $\delta^{(k)}$ at lines 7–8 in Algorithm 1 and line 7 in Algorithm 2. Therefore, we prove that the calculation result of $\delta^{(k)}$ of Algorithm 1 is equivalent to that of Algorithm 2 for the same input $\bar{u}^{(k)}$. Because $M_1(\bar{u}^{(k)}, \nabla\rho^{(k)}, \bar{y}_1^{(k)}, \bar{y}_2^{(k)}, \bar{y}_3^{(k)}, \bar{y}_4^{(k)}) = M_3(\bar{u}^{(k)}, \nabla\rho^{(k)}, \bar{y}_1^{(k)}, \bar{y}_2^{(k)}, \bar{y}_3^{(k)})$ from Lemma 3, the generalized eigenvalue problem of (32) is equal to that of (35). This means that step size $\delta^{(k)}$ of Algorithm 1 and that of Algorithm 2 are equivalent for the same input $\bar{u}^{(k)}$. Therefore, if Algorithms 1 and 2 start with the same initial condition $\bar{u}^{(0)}$, then the same estimation results are obtained. This proves statement (i). Meanwhile, statement (ii) is trivial from Algorithm 2. ■

From this theorem, we have shown that Algorithm 2 reduces the number of measurements to three-quarters, while preserving the estimation performance. The number of measurements corresponds to the number of experiments. Therefore, Algorithm 2 reduces the total time taken for the estimation experiments to three-quarters compared with Algorithm 1.

B. Simulation

We implement Algorithm 2 for the example in Section II-C. In particular, we assume that the output data are corrupted with noise. Although the algorithm was developed for noise-free systems, this example illustrates the estimation performance of the algorithm in a noisy case. Therefore, in this section, we consider the example in Section II-C with outputs contaminated by zero-mean white noise of variance 0.005. We calculate the gradient vector using

$$\nabla\rho(\bar{u}^{(k)}) = \frac{\bar{y}_1^{(k)} + P\bar{y}_2^{(k)} - \dfrac{(\bar{u}^{(k)})^{\mathrm{T}}(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)})}{(\bar{u}^{(k)})^{\mathrm{T}}\bar{u}^{(k)}}\bar{u}^{(k)}}{\left\| \bar{y}_1^{(k)} + P\bar{y}_2^{(k)} - \dfrac{(\bar{u}^{(k)})^{\mathrm{T}}(\bar{y}_1^{(k)} + P\bar{y}_2^{(k)})}{(\bar{u}^{(k)})^{\mathrm{T}}\bar{u}^{(k)}}\bar{u}^{(k)} \right\|} \tag{44}$$

instead of (31) to maintain the S/N ratio.

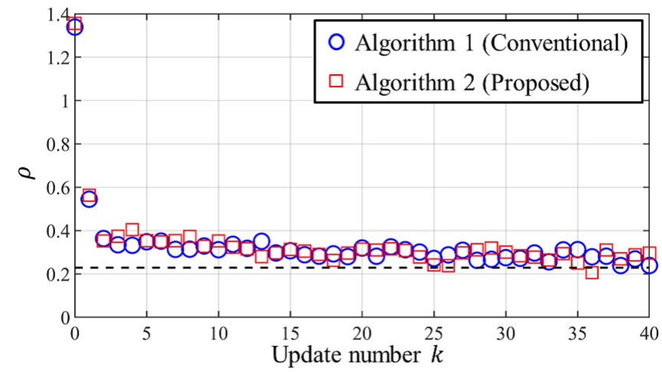

Fig. 3. Simulation results for $\rho(\bar{u}^{(k)})$ in a noisy case for Algorithms 1 and 2.

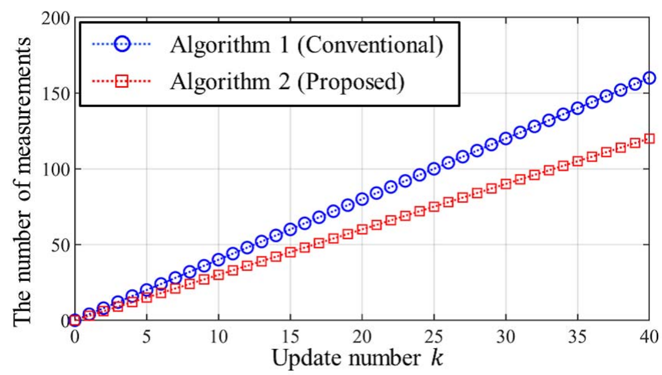

Fig. 4. Relation between the number of measurements and the update.

Fig. 3 shows the simulation results for $\rho(\bar{u}^{(k)})$ with noise in the output, where the circles and squares represent $\rho(\bar{u}^{(k)})$ of Algorithms 1 and 2, respectively. The dashed line indicates the passivity index in (3). The results show that the estimation performance of Algorithm 2 is almost equal to that of Algorithm 1.

Fig. 4 shows the relation between the number of measurements and the update number, where the circles and squares represent Algorithms 1 and 2, respectively. From this figure, the total number of measurements for Algorithm 1 is 160, and Algorithm 2 reduces the total number of measurements to 120. From these results, Algorithm 2 is useful for noisy data.

V. Composition of the Measurement Using Previous Input-Output Data

A. Efficient Composition of $\bar{y}_1^{(k)}$

In this section, we provide a more efficient method solution for Problem 1. The idea is based on the composition of $\bar{y}_1^{(k)}$ using previous input-output data. We provide a calculation of $\bar{y}_1^{(k)}$ as follows:

$$\bar{y}_1^{(k)} = G\frac{\bar{u}^{(k-1)} - \delta^{(k-1)}\nabla\rho(\bar{u}^{(k-1)})}{\|\bar{u}^{(k-1)} - \delta^{(k-1)}\nabla\rho(\bar{u}^{(k-1)})\|}, \tag{45}$$

$$= \frac{\bar{y}_1^{(k-1)} - \delta^{(k-1)}\bar{y}_3^{(k-1)}}{\|\bar{u}^{(k-1)} - \delta^{(k-1)}\nabla\rho(\bar{u}^{(k-1)})\|}. \tag{46}$$

From (46), the experiment to measure $\bar{y}_1^{(k)} = G\bar{u}^{(k)}$ is omitted in each iteration. The method based on (19), (31), (35),

Algorithm 3 Estimation of ν^*

$\bar{u}^{(0)}$ is arbitrarily determined.
$\bar{y}_1^{(0)} \leftarrow G\bar{u}^{(0)}$ ▷ Experiment
for $k = 0, 1, \cdots$ **do**
  $\bar{y}_2^{(k)} \leftarrow GP\bar{u}^{(k)}$ ▷ Experiment
  $\nabla\rho(\bar{u}^{(k)})$ is computed by (31).
  $\bar{y}_3^{(k)} \leftarrow G\nabla\rho(\bar{u}^{(k)})$ ▷ Experiment
  $\delta^{(k)}$ is computed by (35).
  $\bar{u}^{(k+1)}$ is computed by (19).
  $\bar{y}_1^{(k+1)}$ is computed by (46).
end for
return $\rho_3(\bar{u}^{(k+1)}) = \dfrac{(\bar{y}_1^{(k+1)})^{\mathrm{T}}\bar{u}^{(k+1)}}{(\bar{u}^{(k+1)})^{\mathrm{T}}\bar{u}^{(k+1)}}$

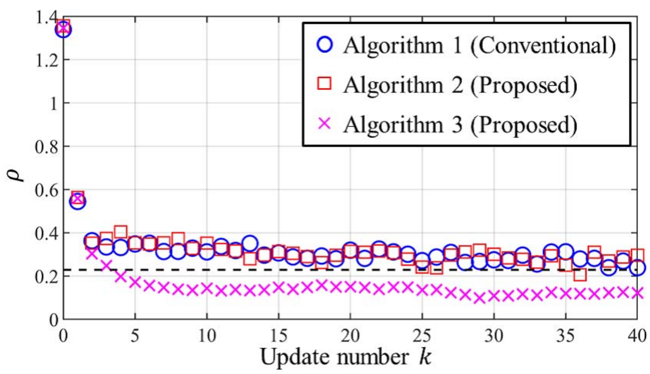

Fig. 5. Simulation results for $\rho(\bar{u}^{(k)})$ in a noisy case for Algorithm 3.

and (46) is referred to as Algorithm 3. For Algorithm 3, a similar result to Theorem 2 is obtained.

Theorem 3: For Algorithm 3, the following statements hold.

(i) Let $\rho_1^{(k)}$ be $\rho(\bar{u}^{(k)})$ of Algorithm 1 with initial condition $\bar{u}^{(0)}$ and $\rho_3^{(k)}$ be $\rho(\bar{u}^{(k)})$ of Algorithm 3 with the same initial condition $\bar{u}^{(0)}$. Then,

$$\rho_1^{(k)} = \rho_3^{(k)} \tag{47}$$

for each $k \in \{0, 1, \ldots\}$.

(ii) Algorithm 3 is executed for three values, $\bar{y}_1^{(k)}$, $\bar{y}_2^{(k)}$, and $\bar{y}_3^{(k)}$, in each iteration. Additionally, $\bar{y}_1^{(k)}$ is computed by (46); that is, the total number of measurements is $2k$ until the k-th iteration.

Proof: The proof of Theorem 3 is similar to that of Theorem 2. ■

From Theorem 3, Algorithm 3 halves the total time taken for experiments compared with Algorithm 1.

B. Simulation

In this section, Algorithm 3 is demonstrated using the example in Section IV-B. Fig. 5 shows the estimation results for $\rho(\bar{u}^{(k)})$, where the crosses represent $\rho(\bar{u}^{(k)})$ of Algorithm 3. The figure shows that, compared with the performance of the other algorithms, the performance of Algorithm 3 is degraded within a tolerable level and reduces the number of measurements to 80. The results demonstrate that Algorithm 3 is useful for estimating the passivity index with fewer experiments.

VI. Conclusion

We considered finite-time input passivity and focused on the estimation method for discrete-time LTI systems in [9], which is based on iterative input-output experiments and estimates the input passivity without information about systems. The method, however, requires four experiments for each iteration.

We proposed two algorithms to reduce the number of measurements. First, we proposed Algorithm 2 in which one measurement for determining the step size was omitted in each iteration. The algorithm reduced the total time taken for experiments to three-quarters compared with the conventional method. Next, we proposed Algorithm 3 that computed one measurement using previous input-output data tuples in each iteration. The algorithm halved the total time taken for experiments compared with the conventional method. We performed a simulation with noisy output. The simulation results demonstrated that the estimation results for Algorithm 2 were almost equal to those of the conventional method, and the deviation of the estimation results for Algorithm 3 was slightly larger than that of the other algorithms. However, Algorithm 3 reduced the total time taken for experiments significantly. From these results, Algorithm 2 provided good results, even for noisy data, and if the noise level was low Algorithm 3 was very useful.

In this letter, we considered the case of SISO systems. Note that, a data-driven estimation method for MIMO systems was recently proposed in [12], and our methods can also reduce the number of measurements in the estimation method for MIMO systems.

References

[1] A. van der Schaft, *L_2-Gain and Passivity Techniques in Nonlinear Control*, 2nd ed. Cham, Switzerland: Springer-Verlag, 2000.
[2] R. Ortega, A. van der Schaft, B. Maschke, and G. Escobar, “Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems,” *Automatica*, vol. 38, no. 4, pp. 585–596, 2002.
[3] P. Kokotović and M. Arcak, “Constructive nonlinear control: A historical perspective,” *Automatica*, vol. 37, no. 5, pp. 637–662, 2001.
[4] J. Bao and P. L. Lee, *Process Control: The Passive Systems Approach*. London, U.K.: Springer-Verlag, 2007.
[5] J. T. Wen and M. Arcak, “A unifying passivity framework for network flow control,” *IEEE Trans. Autom. Control*, vol. 49, no. 2, pp. 162–174, Feb. 2004.
[6] H. Berghuis and H. Nijmeijer, “A passivity approach to controller-observer design for robots,” *IEEE Trans. Robot. Autom.*, vol. 9, no. 6, pp. 740–754, Dec. 1993.
[7] J. M. Montenbruck and F. Allgöwer, “Some problems arising in controller design from big data via input–output methods,” in *Proc. IEEE 55th Conf. Decis. Control*, Las Vegas, NV, USA, 2016, pp. 6525–6530.
[8] A. Romer, J. M. Montenbruck, and F. Allgöwer, “Determining dissipation inequalities from input–output samples,” in *Proc. 20th IFAC World Congr.*, vol. 50, 2017, pp. 7789–7794.
[9] A. Romer, J. M. Montenbruck, and F. Allgöwer, “Sampling strategies for data-driven inference of passivity properties,” in *Proc. IEEE 56th Annu. Conf. Decis. Control*, Melbourne, VIC, Australia, 2017, pp. 6389–6394.
[10] J. M. Kelly, G. Leitmann, and A. G. Soldatos, “Robust control of base-isolated structures under earthquake excitation,” *J. Optim. Theory Appl.*, vol. 53, no. 2, pp. 159–180, 1987.
[11] N. Kottenstette, M. J. McCourt, M. Xia, V. Gupta, and P. J. Antsaklis, “On relationships among passivity, positive realness, and dissipativity in linear systems,” *Automatica*, vol. 50, no. 4, pp. 1003–1016, 2014.
[12] A. Romer, J. M. Montenbruck, and F. Allgöwer, “Some ideas on sampling strategies for data-driven inference of passivity properties for MIMO systems,” in *Proc. Amer. Control Conf.*, Milwaukee, WI, USA, 2018, pp. 6094–6100.