SUGISAKA MASANORI

Journal of Robotics, Networking and Artificial Life
Vol. 9(4); March (2023), pp. 309–315
ISSN (Online):2352-6386; ISSN (Print):2405-9021
https://alife-robotics.org/jrnal.html

Research Article

Modeling of Finger Motions Measured by Leap Motion Controller Using State-Space Model with Step Input

Ryuichi Usami, Hideyuki Tanaka
Graduate School of Humanities and Social Sciences, Hiroshima University, 1-1-1 Kagamiyama, Higashi-Hiroshima city, Hiroshima, 739-8524 Japan

ARTICLE INFO

Article History
Received 24 November 2021
Accepted 24 September 2022

Keywords
Leap motion controller
Modeling of finger motions
State-space representation
Step response
N4SID

ABSTRACT

Modeling the motion of experts in sports is important in demonstrating their motion to beginners. A primary approach to this objective is to study the dynamics of a transient response from one position to another. Hence, finger-motion analysis is a primitive but crucial step to be considered. The state-space representation with a step input is used to eliminate the personal habits of experts from their motion and to present simple models to learners according to their learning stages. Based on the deterministic realization and N4SID algorithms, two algorithms are developed to demonstrate simple motions according to the order of the state-space representation. They are applied to the finger motion captured by the Leap Motion Controller.

© 2022 *The Author*. Published by Sugisaka Masanori at ALife Robotics Corporation Ltd.
This is an open access article distributed under the CC BY-NC 4.0 license
(http://creativecommons.org/licenses/by-nc/4.0/).

1. Introduction

In sports, player movements have been measured using sensors. For example, motion sensors were used to capture the movements of the golf swing [1], and pressure sensor systems were presented to monitor the ankle supination torque during sport motions [2]. A wearable inertial sensor network and its associated activity recognition algorithm were proposed to accurately recognize human activities in daily life and sports [3]. Motion capture has also been used to analyze sports performance; see a systematic review on the recent developments of motion capture systems for the analysis of sport performance [4]. Moreover, a system applicable to the marker-less sport-movement analysis has been presented [5].

The demonstration of models for learners has proven particularly effective in enhancing motor learning [6]. Presenting a model of an expert to beginners is therefore expected to be effective in learning the motion. However, in sports, advanced techniques and skills, including individual habits, are extremely complex for beginners. Hence, we believe that a simple model that eliminates these factors would facilitate motion learning by beginners. We therefore introduce the different levels of simplification by means of the order of the system.

In this study, we develop mathematical models to provide a simplified model for beginners and a complicated one for advanced learners. We introduce the state-space representation driven by a step input, often used for modeling in control engineering [7]. We use the state-space representation, because it can handle dynamics of multiple inputs and multiple outputs, and a simplified model can be obtained by reducing the order of the state-space representation. Moreover, using a step input response of the state-space representation, it will be possible to develop models of a transient response from one position to another such as a crouching start of a

Corresponding author's E-mail: tanakalpha@hiroshima-u.ac.jp

short-distance race. As a simple example of motion from one position to another, we will model a single grasping motion of a hand, because the motions of finger joints are related to each other and so are the motions of human-body joints in the transient response of sports movements.

We will apply the deterministic realization [8] and N4SID algorithms [9] to the problem. They are suitable for this study, because the order of the model can be systematically determined by computing the singular value decomposition (SVD) [7]. Since the deterministic realization algorithm was developed to determine a state-space model from an impulse response, we modify the algorithm to obtain a state-space model of finger motion. Since N4SID is applicable only to data with persistently excited inputs, we reconfigure the input and output data. We compare these algorithms by applying them to finger-motion data.

We utilize the Leap Motion Controller (LMC), an optical hand tracking module that captures hand movements [10]. The accuracy and robustness of the LMC have been analyzed [11], and its performance was evaluated using a professional, high-precision, fast motion tracking system [12].

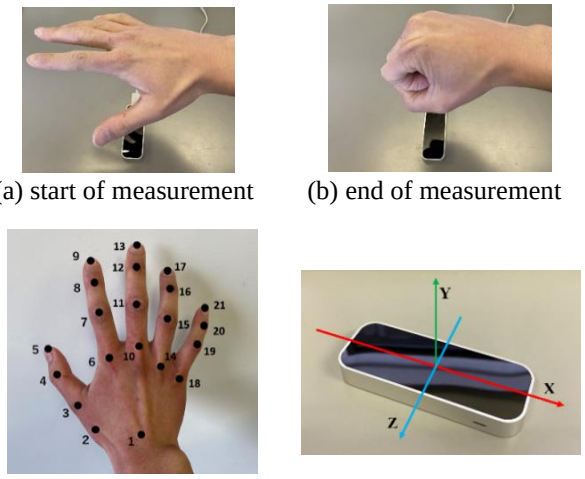
The remainder of this paper is organized as follows: Section 2 describes how the finger motions are measured, and Section 3 formulates the problem setting. Sections 4 and 5 solve the problems using deterministic realization and N4SID, respectively. Section 6 presents experimental results, and Section 7 concludes the paper.

2. Measuring Data

We describe how the finger motions are measured. Fig. 1(a) and Fig. 1(b) illustrate the start (open hand) and end (grasped hand) of transient motion, respectively. The LMC can measure 21 finger joints in a three-dimensional Cartesian coordinate system, which implies that the LMC obtains 63 elements for each sample. Fig. 1(c) and Fig. 1(d) present the measuring points and the coordinate system, respectively. We define the output $y(t)$ as

$$y(t) = \begin{bmatrix} p_x(t)^\top & p_y(t)^\top & p_z(t)^\top \end{bmatrix}^\top \in \mathbf{R}^{63},$$

where $p_x(t)$, $p_y(t)$, and $p_z(t) \in \mathbf{R}^{21}$ denote the positions in the X-Y-Z coordinates of the 21 joints of the fingers, respectively. The positions of finger joints are measured over the same sampling period between the movement from the open hand to the grasped hand.



(a) start of measurement (b) end of measurement

(c) measuring points (d) coordinate system

Fig. 1 Finger motion and measurement system

We measure the finger positions during the grasping motion, using Processing [13] [14]. The time interval from the start to end is 1(s), and 100 data points are sampled during 1(s). The countdown from 5(s) before the start time is displayed on the console.

3. Problem Setting

We formulate the problem settings. Suppose that the sampling time is h and let y_k be the coordinate $y(t)$ at $t = kh$:

$$y_k = y(kh). \tag{1}$$

We model y_k by means of the discrete-time state-space representation:

$$x_{k+1} = Ax_k + Bu_k, \tag{2a}$$
$$y_k = Cx_k + \zeta, \tag{2b}$$

where x_k is a state of the system, $x_0 = 0$, ζ is a constant vector, and u_k is as follows:

$$u_k = \begin{cases} 0 & (k = -1, -2, \ldots) \\ 1 & (k = 0, 1, 2, \ldots) \end{cases} \tag{3}$$

We suppose that $k = 0$ when the transient motion starts and that there is no explicit delay in starting the finger movement after the start time is displayed. We consider the following problems:

Problem 1: Find (A, B, C, ζ) within the degrees of freedom of similarity transformations, given y_k in (2) and (3).

Problem 2: Suppose that the positions of the finger joints are measured as presented in (1). Find a mathematical model for y_k in (2) and (3).

The complexity of the finger-motion model in (2) is determined by the size of the matrix A or the order of the system. The lower the order, the simpler the model is. Conversely, the higher the order, the more accurate the finger motion is, and the closer the motion is to the original one.

By selecting the order of the model (2), we can generate samples of the motion of experts for different levels of learners. It is expected that beginners will learn the motions by simpler dynamical models, whereas advanced learners will learn the motions by more complex models. It should be noted that the dynamical system (2) can model the transition from one point to another, although we model finger motions in this study.

4. Solution via Deterministic Realization

We solve Problem 1 using the deterministic realization algorithm [8] and apply the algorithm to Problem 2. We consider Problem 1. Let us describe y_k for (2) as

$$y_k = \sum_{j=1}^{k} CA^{j-1}B + \zeta, \tag{4}$$

and let v_k be as follows ($k \geq 1$):

$$v_k = y_k - y_{k-1}. \tag{5}$$

The signal v_k is the difference of the output y_k. From (4), v_k satisfies the following equation:

$$v_k = CA^{k-1}B. \tag{6}$$

For positive integers $\tau > n$ and $N > n$, define the Hankel matrix $H \in \mathrm{R}^{\tau p \times N}$ from v_k as follows:

$$H = \begin{bmatrix} v_1 & v_2 & \cdots & v_{N+1-\tau} \\ v_2 & v_3 & \cdots & v_{N+2-\tau} \\ \vdots & \vdots & \vdots & \vdots \\ v_\tau & v_{\tau+1} & \cdots & v_N \end{bmatrix}. \tag{7}$$

We also define the extended observability and reachability matrices as follows:

$$\mathcal{O}_\tau = [C^\top \quad (CA)^\top \quad \cdots \quad (CA^{\tau-1})^\top]^\top, \tag{8a}$$
$$\mathcal{C}_\mathcal{N} = [B \quad AB \quad \cdots \quad A^{N-1}B]. \tag{8b}$$

From (6), H satisfies the following equation:

$$H = \mathcal{O}_\tau \mathcal{C}_\mathcal{N}. \tag{9}$$

We compute the singular value decomposition (SVD):

$$\begin{aligned} H = U\Sigma V^\top &\approx [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} \\ &= U_1 \Sigma_1 V_1^\top, \end{aligned} \tag{10}$$

where Σ is a diagonal matrix, and U and V are orthogonal matrices satisfying $U^\top U = I$ and $V^\top V = I$. From (9), $\mathcal{O}_\tau$and $\mathcal{C}_\mathcal{N}$ are expressed as

$$\mathcal{O}_\tau = U_1 \Sigma_1^{\frac{1}{2}}, \qquad \mathcal{C}_\mathcal{N} = \Sigma_1^{\frac{1}{2}} V_1^\top. \tag{11}$$

From (8), C and B can be obtained as follows:

$$C = \mathcal{O}_\tau(1{:}p, :), \qquad B = \mathcal{C}_\mathcal{N}(:, 1), \tag{12}$$

where we use the colon notation [15]. Let us define $\mathcal{O}_\tau^\downarrow$ and $\mathcal{O}_\tau^\uparrow$ as follows:

$$\mathcal{O}_\tau^\downarrow = \mathcal{O}_\tau(1{:}p(\tau-1), :), \tag{13a}$$
$$\mathcal{O}_\tau^\uparrow = \mathcal{O}_\tau(\tau+1{:}p\tau, :). \tag{13b}$$

From (8a), we have

$$\mathcal{O}_\tau^\downarrow = [C^\top \quad (CA^\top) \quad \cdots \quad (CA^{\tau-2})^\top]^\top, \tag{14a}$$
$$\mathcal{O}_\tau^\uparrow = [(CA)^\top \quad (CA^2)^\top \quad \cdots \quad (CA^{\tau-1})^\top]^\top, \tag{14b}$$

and hence

$$\mathcal{O}_\tau^{\downarrow} A = \mathcal{O}_\tau^{\uparrow}. \tag{15}$$

We can thus obtain A by solving the least-squares method. Let the estimates of A, B, C and ζ be denoted as $\hat{A}$, $\hat{B}$, $\hat{C}$, and $\hat{\zeta}$, respectively. By setting the initial state as $\hat{x}_0 = 0$, we compute η_k for $k = 0, \cdots, N$ as follows:

$$\hat{x}_{k+1} = \hat{A}\hat{x}_k + \hat{B}u_k, \tag{16a}$$
$$\eta_k = \hat{C}\hat{x}_k, \tag{16b}$$

where u_k is the step input in (3). We then have $\eta_k = \sum_{j=1}^{k} \hat{C}\hat{A}^{j-1}\hat{B} \approx \sum_{j=1}^{k} CA^{j-1}B$ and hence from (4):

$$\zeta = y_k - \sum_{j=1}^{k} CA^{j-1}B \approx y_k - \eta_k$$

We therefore estimate ζ, by averaging $y_k - \eta_k$:

$$\hat{\zeta} = \frac{1}{N+1}\sum_{k=0}^{N}(y_k - \eta_k). \tag{17}$$

Thus, estimates of (A, B, C, ζ) for Problem 1 are obtained by $(\hat{A}, \hat{B}, \hat{C}, \hat{\zeta})$, and those of $\hat{y}_k$ in (2) are given by

$$\hat{x}_{k+1} = \hat{A}\hat{x}_k + \hat{B}u_k, \tag{18a}$$
$$\hat{y}_k = \hat{C}\hat{x}_k + \hat{\zeta}. \tag{18b}$$

We summarize the above procedure for Problem 1 as the following algorithm:

Algorithm based on deterministic realization:
Step 1: Calculate v_k in (5).
Step 2: Construct the block Hankel matrix H in (7).
Step 3: Compute the SVD in (10).
Step 4: Determine $\mathcal{O}_\tau$ and $\mathcal{C}_\mathcal{N}$ as in (11).
Step 5: Compute the estimate $(\hat{A}, \hat{B}, \hat{C})$ of (A, B, C) from (12) and (15).
Step 6: Obtain an estimate $\hat{\zeta}$ of ζ as (17) and calculate $\hat{y}_k$ in (18).

By applying this algorithm to Problem 2, we have mathematical models for finger motions. The model is simplified by the SVD of the Hankel matrix in (10), and the order is determined by the number of the dominant singular values. We select the order by choosing the number of the non-zero diagonal elements of H in (10). Thus, we simplify the state-space model (3) and obtain a simplified transient movement from the open hand to the grasped hand. The lower the order of the state-space model is, the simpler the transient motion becomes.

5. Solution via N4SID

We solve Problem 1 by using the N4SID algorithm [9]. The N4SID algorithm can also systematically determine the order of the model by computing the SVD. However, N4SID cannot be directly applied to the system in (2). Moreover, the input data in (3) do not satisfy the assumptions of N4SID. We therefore demonstrate how to apply the input-output data to N4SID.

Let the number N of given data be as follows:

$$N = 2\tau + l - 1 \quad (\tau > n). \tag{19}$$

Using the input u_t and output y_t, define

$$U_p = \begin{bmatrix} u_0 & u_1 & \cdots & u_{l-1} \\ u_1 & u_2 & \cdots & u_l \\ \vdots & \vdots & \vdots & \vdots \\ u_{\tau-1} & u_\tau & \cdots & u_{\tau+l-2} \end{bmatrix}, \tag{20a}$$

$$U_f = \begin{bmatrix} u_\tau & u_{\tau+1} & \cdots & u_{\tau+l-1} \\ u_{\tau+1} & u_{\tau+2} & \cdots & u_{\tau+l} \\ \vdots & \vdots & \vdots & \vdots \\ u_{2\tau-1} & u_{2\tau} & \cdots & u_{2\tau+l-2} \end{bmatrix}. \tag{20b}$$

In applying N4SID, the following assumption is made:

$$\operatorname{rank}\left(\begin{bmatrix} U_p \\ U_f \end{bmatrix}\right) = 2\tau m, \tag{21}$$

where $m = 1$ is the size of input. However, the input data $u_t = 1$ in (3) satisfy

$$\operatorname{rank}\left(\begin{bmatrix} U_p \\ U_f \end{bmatrix}\right) = 1.$$

We reconfigure the data to satisfy (21). Defining

$$u_k = 0 \quad (k = -1, \cdots, -M),$$
$$y_k = y_0 \ (k = -1, \cdots, -M),$$

we determine new input and output $(k = 0, 1, \ldots, N + M - 1)$, respectively:

$$\bar{u}_k = u_{k-M} - \mu_u, \tag{22a}$$

$$\bar{y}_k = y_{k-M} - \mu_y, \tag{22b}$$

where μ_u and μ_y are computed as follows:

$$\mu_u = \frac{1}{N+M}\sum_{t=0}^{N-1} u_t, \quad \mu_y = \frac{1}{N+M}\sum_{t=0}^{N-1} y_t,$$

since N4SID is applied to the input-output data with mean zero. Thus, we define the following matrices:

$$\overline{U}_p = \begin{bmatrix} \bar{u}_0 & \bar{u}_1 & \cdots & \bar{u}_{\bar{l}-1} \\ \bar{u}_1 & \bar{u}_2 & \cdots & \bar{u}_{\bar{l}} \\ \vdots & \vdots & \vdots & \vdots \\ \bar{u}_{\tau-1} & \bar{u}_{\tau} & \cdots & \bar{u}_{\tau+\bar{l}-2} \end{bmatrix}, \tag{23a}$$

$$\overline{U}_f = \begin{bmatrix} \bar{u}_{\tau} & \bar{u}_{\tau+1} & \cdots & \bar{u}_{\tau+\bar{l}-1} \\ \bar{u}_{\tau+1} & \bar{u}_{\tau+2} & \cdots & \bar{u}_{\tau+\bar{l}} \\ \vdots & \vdots & \vdots & \vdots \\ \bar{u}_{2\tau-1} & \bar{u}_{2\tau} & \cdots & \bar{u}_{2\tau+\bar{l}-2} \end{bmatrix}, \tag{23b}$$

where $\bar{l} = N + M - 2\tau + 1$. We construct $\overline{Y_p}$ and $\overline{Y_f}$ in the same way, using $\bar{y}_t$. Then, we have:

$$\operatorname{rank}\left(\begin{bmatrix}\overline{U}_p \\ \overline{U}_f\end{bmatrix}\right) = 2\tau, \tag{24}$$

which satisfies the assumption of N4SID in (21).

We apply N4SID to the input and output data $\overline{U}_p$,$\overline{U}_f$, $\overline{Y_p}$ and $\overline{Y_f}$ and explain how to select the order. We define the block Toeplitz matrix as follows:

$$\mathcal{T}_\tau = \begin{bmatrix} D & & & 0 \\ CB & D & & \\ \vdots & \ddots & \ddots & \\ CA^{\tau-2}B & \cdots & CB & D \end{bmatrix}. \tag{25}$$

We have the following equations:

$$\bar{Y}_p = \mathcal{O}_\tau \bar{X}_p + \mathcal{T}_\tau \bar{U}_{p,} \tag{26a}$$
$$\bar{Y}_f = \mathcal{O}_\tau \bar{X}_f + \mathcal{T}_\tau \bar{U}_f, \tag{26b}$$

where $\bar{X}_p$ and $\bar{X}_f$ are as follows:

$$\bar{X}_p = [\bar{x}_0 \quad \bar{x}_1 \quad \cdots \quad \bar{x}_{\bar{l}-1}], \tag{27a}$$
$$\bar{X}_f = [\bar{x}_\tau \quad \bar{x}_{\tau+1} \quad \cdots \quad \bar{x}_{\tau+\bar{l}-1}], \tag{27b}$$

and where $\bar{x}_k = 0$ $(k = 0, \ldots, M-1)$ and $\bar{x}_k = x_{k-M}$ $(k = M, \ldots, N+M-1)$. We estimate $\mathcal{O}_\tau \bar{X}_f$ based on N4SID. Describing the estimate of $\mathcal{O}_\tau \bar{X}_f$ as $O_{\bar{X}}$, we calculate the SVD as follows:

$$O_{\bar{X}} \approx [U_1 \quad U_2]\begin{bmatrix}\Sigma_1 & 0 \\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^\top \\ V_2^\top\end{bmatrix} = U_1 \Sigma_1 V_1^\top. \tag{28}$$

Approximating $\mathcal{O}_\tau \bar{X}_f \approx \hat{\mathcal{O}}_\tau \hat{X}_f$, $\hat{\mathcal{O}}_\tau$ and $\hat{X}_f$ are as follows:

$$\hat{\mathcal{O}}_\tau \approx U_1 \Sigma_1^{\frac{1}{2}}, \quad \hat{X}_f \approx \Sigma_1^{\frac{1}{2}} V_1^\top. \tag{29}$$

Using $\hat{X}_f$, define the followings:

$$\tilde{X}_\tau = \hat{X}_f(:, 1:\bar{l}-1), \tag{30a}$$
$$\tilde{X}_{\tau+1} = \hat{X}_f(:, 2:\bar{l}). \tag{30b}$$

The matrices A, B and C are estimated as follows.

$$[\hat{A} \quad \hat{B}] = \tilde{X}_{\tau+1}\begin{bmatrix}\hat{X}_\tau \\ \bar{U}_\tau\end{bmatrix}^\dagger, \qquad \hat{C} = \bar{Y}_\tau \tilde{X}_\tau^\dagger, \tag{31}$$

where $(\cdot)^\dagger$ expresses the pseudo-inverse. We compute $\hat{\zeta}$ from (17) using $(\hat{A}, \hat{B}, \hat{C})$.

We summarize the above procedure for Problem 1:

Algorithm based on N4SID:
Step 1: Calculate $\{\bar{u}_k, \bar{y}_k\}$ as (22).
Step 2: Construct $\overline{U}_p, \overline{U}_f, \overline{Y}_p$ and $\overline{Y}_f$ as (23).
Step 3: Compute the SVD in (28).
Step 4: Find an estimate of the state as (29).
Step 5: Compute the estimate $(\hat{A}, \hat{B}, \hat{C})$ from (31).
Step 6: Obtain an estimate $\hat{\zeta}$ of ζ as (17).

By applying this algorithm to Problem 2, we also have models for finger motions. The model is simplified by computing the SVD of the Hankel matrix in (28). It should be noted that both algorithms provide simplified model based on the SVDs; however, the models obtained from them are different, because different matrices are used in (10) and (28). Thus, in Section 6, we compare different simplifications, using both the algorithms based on deterministic realization and N4SID.

6. Experimental Results

We demonstrate the algorithms by experimental results.We present $p_y(t)$ at the measuring point 13, which is the tip of the middle finger with one of the largest motions among the measuring points of fingers.

In Fig. 2 and Fig. 3, the horizontal and vertical axes represent the time (s) and position (mm) of $p_y(t)$, respectively. The red lines in Fig. 2 and Fig. 3 indicate the position of the original motion, whereas the blue ones show models with the order of 4, 7, 12, and 18. Fig. 2 and Fig. 3 demonstrate that the output of models (blue lines) becomes closer to the original motion (red lines) as the order is higher. By comparing blue lines for $n = 4$ and $n = 18$, it is observed that the motion drawn by the blue line for $n = 4$ is much simpler than that for $n = 18$ in both Fig. 2 and Fig. 3.

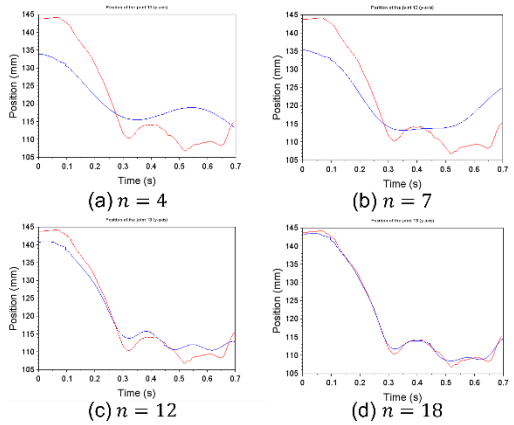


Fig. 2 Modeling based on deterministic realization for the different sizes of $\Sigma_1 \in \mathbf{R}^{n\times n}$ in (10) ($n = 4, 7, 12,$ and 18).

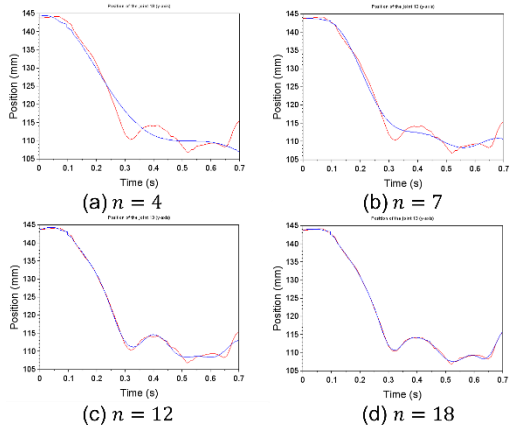


Fig. 3 Modeling based on N4SID for the different sizes of $\Sigma_1 \in \mathbf{R}^{n\times n}$ in (28) ($n = 4, 7, 12,$ and 18).

We compare the motions provided by the algorithms based on deterministic realization and N4SID. The reduced-order models ($n = 4, 7$) computed by N4SID capture the motion of the finger almost exactly. In contrast, the high-order models ($n = 12, 18$) given by the deterministic realization provide the motion close to the original one, although the motion given by the reduced-order models ($n = 4, 7$) is far away from the original one. From this experiment, we infer that the N4SID-based algorithm gives simple models that more accurately capture the motion of fingers than the deterministic-realization-based algorithm.

We compute the SVDs to obtain simplified models and select the order. We show singular values of the matrices in (10) and (28) in Fig. 4 and Fig. 5, respectively.

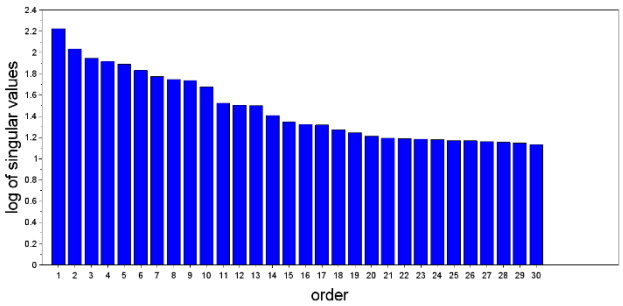


Fig. 4 Singular values (deterministic realization)

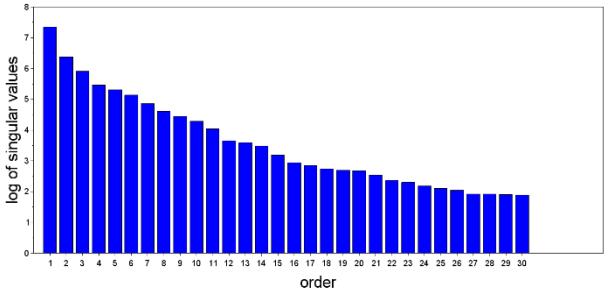


Fig. 5 Singular values (N4SID)

7. Conclusions

We developed mathematical models of finger motions to simplify the motion of experts in sports. We solved the problem of modeling finger motions based on the step response of a state-space representation, by modifying the deterministic realization and N4SID algorithms. The experimental results demonstrated that the modeling methods satisfied the purpose of simplifying complex finger motions by reducing the order of the state-space representation. By selecting the order, we could select the

model of finger motion at different levels including simple and accurate motions. The N4SID-based algorithm provided simple models that more accurately capture the motion of fingers than the deterministic-realization-based algorithm.

Acknowledgment: This work is partly supported by JSPS KAKENHI Grant Number 20K04535.

References

1. H. Ghasemzadeh, V. Loseu, E. Guenterberg, and R. Jafari, "Sport training using body sensor networks: A statistical approach to measure wrist rotation for golf swing", Proceedings of the Fourth International Conference on Body Area Networks, 2009.
2. D. T.-P. Fong, Y.-Y. Chan, Y. Hong, P. S.-H. Yung, K.-Y. Fung, and K.-M. Chan, "A three-pressure-sensor (3PS) system for monitoring ankle supination torque during sport motions", Journal of Biomechanics, Volume 41, Issue 11, pp. 2562-2566, 2008.
3. Y.-L. Hsu, S.-C. Yang, H.-C. Chang, and H.-C. Lai, "Human daily and sport activity recognition using a wearable inertial sensor network", IEEE Access, vol. 6, pp. 31715-31728, 2018.
4. B. Pueo and J. M. Jimenez-Olmedo, "Application of motion capture technology for sport performance analysis", editor: Federación Española de Asociaciones de Docentes de Educación Física (FEADEF), Retos, 32, pp. 241-247, 2017.
5. B. Rosenhahn, T. Brox, U. G. Kersting, A. W. Smith, J. K. Gurney, and R. Klette, "A system for marker-less motion capture", Künstliche Intelligenz, vol. 1, pp 46-52, 2006.
6. G. H. Sage, "Motor learning and control -A neuro-psychological approach-", Wm. C. Brown Publishers, 1984.
7. T. Katayama, "Subspace methods for system identi-fication", Springer-Verlag London Limited, 2005.
8. B. L. Ho and R. E. Kalman, "Effective construction of linear state-variable models from input/output functions", Regelungstechnik, vol. 14, no.12, pp. 545-548, 1966.
9. P. Van Overschee and B. De Moor, "N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems", Automatica, Vol.30, No. 1, pp. 75-93, 1994
10. ultraleap, "Leap Motion Controller", https://www.ultraleap.com/product/leap-motion-controller/.
11. F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the Accuracy and Robustness of the Leap Motion Controller", Sensors, 13, pp. 6380-6393, 2013.
12. J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik, "An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking", Sensors, 14, pp. 3702-3720, 2014.
13. Processing, https://processing.org.
14. ultraleap, "API Overview", https://developer-archive.leapmotion.com/documentation/java/devguide/Leap_Overview.html.
15. G. H. Golub and C. F. Van Loan, "Matrix computations, 4th Edition", The Johns Hopkins University Press, 2013.

Authors Introduction

Mr. Ryuichi Usami

He received his bachelor's degree from the School of Education, Hiroshima University, Japan in 2021. He is currently a Master's Program student of Graduate School of Humanities and Social Sciences in Hiroshima University, Japan.

Dr. Hideyuki Tanaka

He graduated master course at graduate school of engineering in Kyoto University and received Dr. (Eng.) from Kyoto University. He is a member of Division of Educational Sciences, Graduate School of Humanities and Social Sciences in Hiroshima University. He is a member of IEEE, SICE, and ISCIE.