

Remarks on Direct System Identification Using Hypercomplex-Valued Neural Network with Application to Time-Series Estimation

Kazuhiko Takahashi, Junpei Ozaki, Yasunao Nakahara, Yuka Wakaume,
Takahiro Hirayama, and Masafumi Hashimoto

Abstract—This study discusses a design method for system identification using multilayer hypercomplex-valued neural networks, such as complex, hyperbolic, bicomplex and quaternion neural networks, and investigates its characteristics. The direct transfer function identifier, which employs a multilayer hypercomplex-valued neural network, is used to estimate the output of the target system. A computational experiment involving estimation of a discrete-time dynamic plant is conducted to evaluate the feasibility of the hypercomplex-valued neural network-based direct transfer function identifier. As a practical application of dynamic system identification, the direct transfer function identifier is used to estimate historical foreign exchange data and motion data of *Taijiquan* in addition to investigating its characteristics. The experimental results show the effectiveness of the proposed direct transfer function identifier.

Index Terms—Hypercomplex-valued neural network, System identification, Dynamic systems, Historical data, Motion data.

I. INTRODUCTION

THE use of hypercomplex numbers for neural networks has attracted increasing attention from engineering research fields because it helps to learn to handle a wide variety of geometric objects and their transformation in the form of hypercomplex numbers. To overcome classically hard-to-treat intractable problems with real-number neural networks, high-dimensional neural networks based on hypercomplex numbers, such as complex numbers and quaternions, have been investigated [1] and their effectiveness in solving multidimensional problems has been demonstrated [2]. Complex neural networks have been used successfully in various engineering applications, for example, filtering, time-series signal processing, communications, image processing and video processing [3]. As an extension of complex numbers, quaternions decrease computational complexity significantly in three- or four-dimensional problems compared to real numbers. Therefore, quaternion neural networks can cope with multidimensional issues more efficiently by directly employing quaternions. Several studies have used quaternion neural networks in applications requiring multidimensional signal processing, for instance, rigid body attitude control [4], colour image processing [5], signal processing [6][7], filtering [8], pattern classification [9] and inverse problems [10]. In previous studies, we have presented control schemes for robot manipulators [11] and servo-controllers [12][13].

Manuscript received December 8, 2016; revised December 20, 2016.

K. Takahashi, J. Ozaki, Y. Nakahara, Y. Wakaume and T. Hirayama are with Information Systems Design, Doshisha University, Kyoto, Japan e-mail: {katakaha@mail, bun1063@mail4, bum1040@mail4, bun1084@mail4, bun1016@mail4}.doshisha.ac.jp.

M. Hashimoto is with Intelligent Information Engineering and Science, Doshisha University, Kyoto, Japan e-mail: mhashimo@mail.doshisha.ac.jp.

In this study, a method of system identification based on a hypercomplex-valued neural network is discussed from the viewpoint of its applicability to time-series signal processing of dynamic systems. A method of designing a direct transfer function identifier in which the neural network output serves as an estimate of the output of a target system is presented. Computational experiments for estimating the output of a discrete-time dynamic plant, historical foreign exchange data and motion data of *Taijiquan* movements are conducted to evaluate the feasibility of the proposed neural network-based direct transfer function identifier.

II. DIRECT TRANSFER FUNCTION IDENTIFIER BASED ON HYPERCOMPLEX-VALUED NEURAL NETWORKS

Figure 1 shows a schematic of a direct transfer function identifier using a neural network, where the output of the neural network converges with the plant output after the neural network is trained and the plant transfer functions are composed in the neural network. Here $y \in \mathbb{R}^q$ is the plant output, $u \in \mathbb{R}^p$ is the input to the plant and $\hat{y} \in \mathbb{R}^q$ is the output of the identifier that consists of the output from the hypercomplex-valued neural network. Assuming a multilayer network topology that has one input layer, one output layer and N hidden layers ($N \geq 0$), the input-output relationship of the hypercomplex-valued neural network is defined. Because all signals, weights and thresholds of a hypercomplex-valued neural network are hypercomplex numbers, the algebra of hypercomplex numbers should be followed to derive a training algorithm for hypercomplex-valued neural networks. In this study, complex, hyperbolic, bicomplex and quaternion numbers are considered so as to consist of the neural networks.

Complex numbers consist of two real numbers and an imaginary unit: $\mathbb{C} := \{z = z_0 + z_1 i | z_0, z_1 \in \mathbb{R}, i^2 = -1\}$. The sum of two complex numbers z_1 and z_2 is given as

$$z_1 \pm z_2 = (z_{10} \pm z_{20}) + (z_{11} \pm z_{21})i,$$

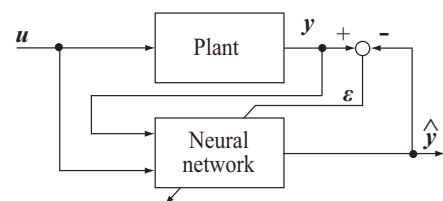


Fig. 1. Schematic of neural network-based direct transfer function identifier.

while the product of two complex numbers is given as

$$z_1 z_2 = (z_{10} z_{20} - z_{11} z_{21}) + (z_{10} z_{21} + z_{11} z_{20})i.$$

The norm of a complex number is defined as

$$|z| = \sqrt{z z^*} = \sqrt{z_0^2 + z_1^2},$$

where $z^* = z_0 - z_1 i$ is the conjugate of the complex number z .

Hyperbolic numbers consist of two complex numbers and an imaginary unit: $\mathbb{D} := \{z = z_0 + z_1 h | z_0, z_1 \in \mathbb{R}, h^2 = 1\}$. The sum of two hyperbolic numbers z_1 and z_2 is given as

$$z_1 \pm z_2 = (z_{10} \pm z_{20}) + (z_{11} \pm z_{21})h,$$

while the product of two hyperbolic numbers is given as

$$z_1 z_2 = (z_{10} z_{20} + z_{11} z_{21}) + (z_{10} z_{21} + z_{11} z_{20})h.$$

The norm of a hyperbolic number is defined as

$$|z| = \sqrt{z z^*} = \sqrt{|z_0^2 - z_1^2|},$$

where $z^* = z_0 - z_1 h$ is the conjugate of the hyperbolic number z .

Bicomplex numbers consist of two complex numbers and an imaginary unit, that is four real numbers and three imaginary units: $\mathbb{T} := \{z = z_1 + z_2 i_2 | z_1, z_2 \in \mathbb{C}, i_2^2 = -1\} = \{z = z_0 + z_1 i_1 + z_2 i_2 + z_3 h | z_0, z_1, z_2, z_3 \in \mathbb{R}, i_1^2 = i_2^2 = -1, i_1 i_2 = h, h^2 = 1\}$. The sum of two bicomplex numbers z_1 and z_2 is given as

$$z_1 \pm z_2 = (z_{10} \pm z_{20}) + (z_{11} \pm z_{21})i_1 + (z_{12} \pm z_{22})i_2 + (z_{13} \pm z_{23})h,$$

while the product of two bicomplex numbers is given as

$$z_1 z_2 = (z_{10} z_{20} - z_{11} z_{21} - z_{12} z_{22} + z_{13} z_{23} h) + (z_{11} z_{20} + z_{10} z_{21} - z_{13} z_{22} - z_{12} z_{23})i_1 + (z_{12} z_{20} - z_{13} z_{21} + z_{10} z_{22} - z_{11} z_{23})i_2 + (z_{13} z_{20} + z_{12} z_{21} + z_{11} z_{22} + z_{10} z_{23})h.$$

The norm of a bicomplex number is defined as

$$|z| = \sqrt{\frac{z z^\dagger + z^*(z^\dagger)^*}{2}} = \sqrt{z_0^2 + z_1^2 + z_2^2 + z_3^2},$$

where $z^* = z_0 - z_1 i_1 + z_2 i_2 - z_3 h$, $z^\dagger = z_0 + z_1 i_1 - z_2 i_2 - z_3 h$ and $z^\ddagger = z_0 - z_1 i_1 - z_2 i_2 + z_3 h$ are the conjugates of the bicomplex number z .

Quaternion numbers consist of four real numbers and three imaginary units: $\mathbb{H} := \{z = z_0 + z_1 i + z_2 j + z_3 k | z_0, z_1, z_2, z_3 \in \mathbb{R}, i^2 = j^2 = k^2 = ijk = -1, ij = -ji = k, jk = -kj = i, ki = -ik = j\}$. The sum of two quaternion numbers z_1 and z_2 is given as

$$z_1 \pm z_2 = (z_{10} \pm z_{20}) + (z_{11} \pm z_{21})i + (z_{12} \pm z_{22})j + (z_{13} \pm z_{23})k,$$

while the product of two quaternion numbers is given as

$$z_1 z_2 = (z_{10} z_{20} - z_{11} z_{21} - z_{12} z_{22} - z_{13} z_{23}) + (z_{11} z_{20} + z_{10} z_{21} - z_{13} z_{22} + z_{12} z_{23})i + (z_{12} z_{20} + z_{13} z_{21} + z_{10} z_{22} - z_{11} z_{23})j + (z_{13} z_{20} - z_{12} z_{21} + z_{11} z_{22} + z_{10} z_{23})k.$$

The norm of a quaternion number is defined as

$$|z| = \sqrt{z z^*} = \sqrt{z_0^2 + z_1^2 + z_2^2 + z_3^2},$$

where $z^* = z_0 - z_1 i - z_2 j - z_3 k$ is the conjugate of the quaternion number z .

The output of the m -th neuron unit in the n -th layer $x_m^{(n)}$ is defined as follows:

$$\begin{cases} x_m^{(n)} = f_m(v_m^{(n)}) \\ v_m^{(n)} = \sum_l w_{ml}^{(n)} x_l^{(n-1)} + \phi_m^{(n)} \end{cases}, \quad (1)$$

where $w_{ml}^{(n)}$ is the weight between the l -th neuron unit in the $(n-1)$ -th layer and the m -th neuron unit in the n -th layer, $\phi_m^{(n)}$ is the threshold of the m -th neuron unit in the n -th layer, and $f(\cdot)$ is an activation function of the neuron unit. Here the activation function requires analyticity of the function in the hypercomplex number domain [14].

The hypercomplex-valued neural network is trained to minimize the cost function J defined by the norm of the output error $\epsilon(t)$ as follows:

$$J(t) = \frac{1}{2} \sum_\sigma \sum_k |\epsilon_k(t)|^2, \quad (2)$$

where $\epsilon_k(t) = d_k - x_k^{(o)}(t)$, d_k is the desired output of the k -th neuron unit in the output layer and the superscript n of o indicates the output layer and t is the iteration number. Assuming increment of the network parameters in the t -th iteration $\Delta\omega(t) = \omega(t+1) - \omega(t)$ where the vector $\omega(t)$ is composed of network parameters such as weights and thresholds, variation of the cost function ΔJ can be calculated as follows:

$$\begin{aligned} \Delta J &= J(t+1) - J(t) \\ &\approx \left(\frac{\partial J(t)}{\partial \omega(t)} \right)^T \Delta\omega(t) + \left(\frac{\partial J(t)}{\partial \omega^*(t)} \right)^T \Delta\omega^*(t). \end{aligned}$$

Setting $\Delta\omega(t) = -\eta \frac{\partial J(t)}{\partial \omega^*(t)}$ and $\Delta\omega^*(t) = -\eta \frac{\partial J(t)}{\partial \omega(t)}$ yields

$$\Delta J = -2\eta \left| \frac{\partial J(t)}{\partial \omega(t)} \right|^2 \leq 0,$$

where the factor η is positive. Increment of the parameters in the output layer is given as

$$\begin{cases} \Delta w_{kj}^{(o)}(t) = \eta \sum_\sigma \delta_k(t) x_j^{*(n-1)}(t) \\ \Delta \phi_k^{(o)}(t) = \eta \sum_\sigma \delta_k(t) \end{cases}, \quad (3)$$

where $\delta_k(t) = \epsilon_k(t) f'_k(v_k^{(o)}(t))$, and the superscript $(n-1)$ indicates the hidden layer links to the output layer, while increment of the parameters in the hidden layer is given by

$$\begin{cases} \Delta w_{ji}^{(n)}(t) = \eta \sum_\sigma \delta_j(t) x_i^{*(n-1)}(t) \\ \Delta \phi_j^{(n)}(t) = \eta \sum_\sigma \delta_j(t) \end{cases}, \quad (4)$$

where $\delta_j(t) = \sum_k \delta_k^{(n+1)}(t) w_{kj}^{(n)}(t) f'_j(v_j^{(n)}(t))$.

To define the input of the identifier simply, we assume that the plant is represented by a linear combination of single-input single-output discrete-time linear systems as follows:

$$\mathbf{y}(t+d) = \sum_{s=1}^n \mathbf{A}_{y_s} \mathbf{y}(t-s+1) + \sum_{s=0}^{m+d-1} \mathbf{A}_{u_s} \mathbf{u}(t-s) \quad (5)$$

where $\mathbf{A}_{y_s} = [\alpha_{y_{s1}l'}]$ and $\mathbf{A}_{u_s} = [\alpha_{u_{s1}l'}]$ ($l = 1, 2, \dots, q$, $l' = 1, 2, \dots, p$) are plant parameter matrices, n and m are plant orders, d is plant dead time and $\mathbf{y}(t) = [y_1(t) \ y_2(t) \ \dots \ y_q(t)]^T$, $\mathbf{u}(t) = [u_1(t) \ u_2(t) \ \dots \ u_p(t)]^T$. Considering the condition $\lim_{t \rightarrow \infty} \{\mathbf{y}(t+d) - \hat{\mathbf{y}}(t+d)\} = \mathbf{0}$ yields

$$\hat{\mathbf{y}}(t+d) = \mathbf{\Lambda} \mathbf{I}(t), \quad (6)$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{A}_{y_1} & \mathbf{A}_{y_2} & \dots & \mathbf{A}_{y_n} \\ \mathbf{A}_{u_0} & \mathbf{A}_{u_1} & \dots & \mathbf{A}_{u_{m+d-1}} \end{bmatrix},$$

and

$$\mathbf{I}(t) = \begin{bmatrix} \mathbf{y}(t) & \mathbf{y}(t-1) & \dots & \mathbf{y}(t-n+1) \\ \mathbf{u}(t) & \mathbf{u}(t-1) & \dots & \mathbf{u}(t-m-d+1) \end{bmatrix}^T.$$

Representing the mapping function of the hypercomplex-valued neural network as $\mathbf{F}_{nn}(\cdot)$ yields

$$\mathbf{x}^{(o)}(t) = \mathbf{F}_{nn}(\boldsymbol{\omega}(t), \boldsymbol{\xi}(t)), \quad (7)$$

where $\boldsymbol{\xi}(t)$ is the input to the hypercomplex-valued neural network. By comparing Eqs. (6) and (7), we choose the vector $\mathbf{I}(t-d)$ as input to the hypercomplex-valued neural network $\boldsymbol{\xi}(t)$.

III. COMPUTATIONAL EXPERIMENTS

To investigate the feasibility of the direct transfer function identifier using the hypercomplex-valued neural network, we conducted several computational experiments. In the experiments, we employed a split function as the activation function [15] [16], for example, $f(z) = f_0(z_0) + f_1(z_1)\mathbf{i} + f_2(z_2)\mathbf{j} + f_3(z_3)\mathbf{k}$ in the quaternion neural network, where the function $f_l(x) = 1/(1 + e^{-x})$ ($l = 0, 1, 2, 3$) is a real-valued function. The split function is not analytic in each hypercomplex number domain; however, we use it for the sake of computational convenience.

First, the quaternion neural network was applied to the direct transfer function identifier. The following discrete-time nonlinear plant was considered the target system:

$$y(t) = 1.3y(t-1) - 0.3y(t-2) + u(t-1) + 0.2u(t-2) + 0.03y(t-3) + 0.2y^2(t-1),$$

where the second-order system is dominant. In the experiment, the plant input $u(t)$ was synthesized by a digital proportional controller to make the plant output $y(t)$ follow a reference model $y_m(t)$. Here, the reference model was a first-order linear system $y_m(t) = 0.6y_m(t-1) + 0.4r(t)$, and the reference input $r(t)$ was a rectangular wave. The number of samples within one cycle of the rectangular wave was 100, and the amplitude of the wave was ± 0.5 . Considering the dominant part of the plant for designing the direct transfer function identifier, the input of the hypercomplex-valued

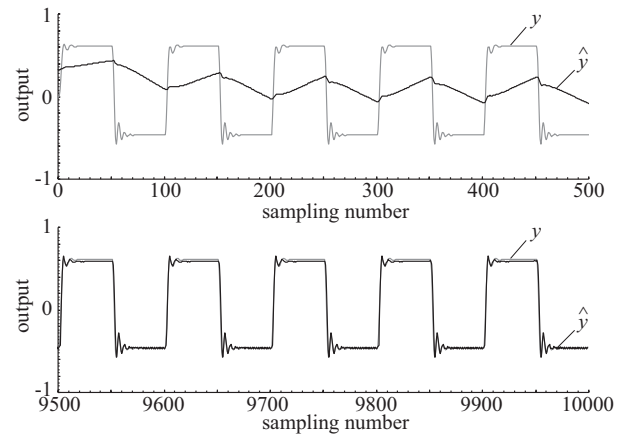


Fig. 2. Experimental results of system identification using quaternion neural network with 1-3-1 (\mathbb{H}) network topology, where the grey line indicates plant output and the black line indicates output of the quaternion neural network (Top: initial adaptation stage of quaternion neural network; bottom: final adaptation stage of quaternion neural network).

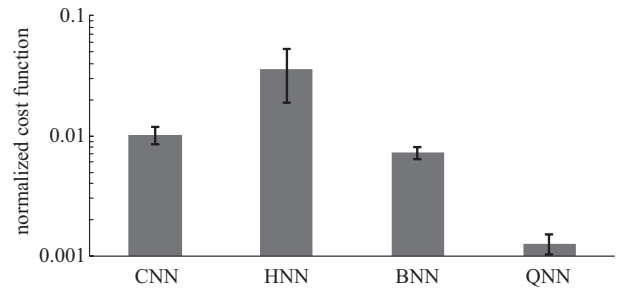


Fig. 3. Comparison of normalized cost function obtained by the hypercomplex-valued neural network-based direct transfer function identifiers (CNN: complex neural network with 2-5-1 (\mathbb{C}) network topology; HNN: hyperbolic neural network with 2-5-1 (\mathbb{D}) network topology; BNN: bicomplex neural network with 1-3-1 (\mathbb{T}) network topology; QNN: quaternion neural network with 1-3-1 (\mathbb{H}) network topology).

neural network was defined as $\boldsymbol{\xi}(t) = y(t-1) + y(t-2)\mathbf{i} + u(t-1)\mathbf{j} + u(t-2)\mathbf{k}$. The estimated plant output comprised one of the imaginary parts of the quaternion neural network's output, that is, $\hat{y}(t) = \text{Im}_{\mathbf{k}}[\mathbf{x}^{(o)}(t)]$, where $\text{Im}_{\mathbf{k}}$ denotes the imaginary part of the imaginary unit \mathbf{k} . Figure 2 shows an example of the system response when the direct transfer function identifier consisted of the quaternion neural network with a 1-3-1 network topology. As shown in Fig. 2, the identifier output converges with the plant output. Figure 3 shows the normalized cost function averaged within one period of the reference input. Here, the normalized cost function is averaged using 100 results of each neural network within the range [9000, 10000]. The topology of each neural network was defined so that the number of parameters, such as weights and thresholds, corresponded to the number in the quantum neural network. The normalized cost function of the quaternion neural network is smaller than that of the other neural network. These results indicate the feasibility of the proposed direct transfer function identifier using the hypercomplex-valued neural network.

Second, the characteristics of the direct transfer function identifier were investigated with historical foreign exchange data for currency pairs of the Japanese Yen and four major currencies (US Dollar, British Pound, Euro and Swiss

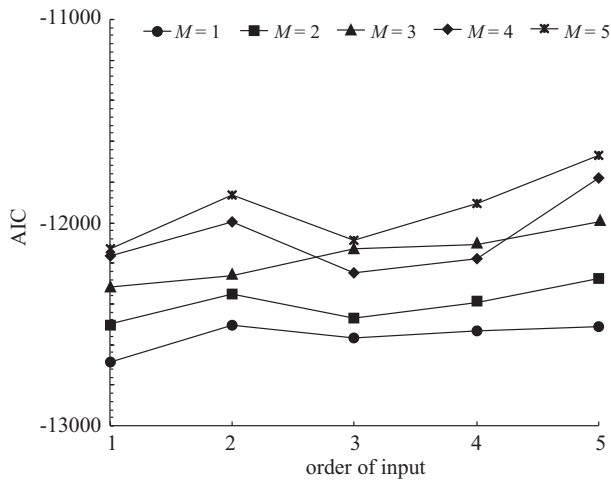


Fig. 4. Experimental results of quaternion neural network for finding optimal network topology based on AIC, where M indicates the number of neurons in the hidden layer.

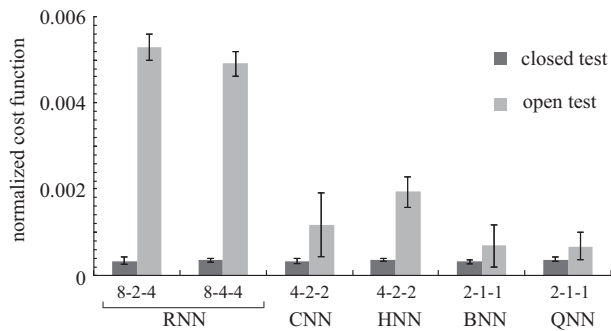


Fig. 5. Comparison of estimation performance obtained by the hypercomplex-valued neural network-based direct transfer function identifiers (RNN: real number neural network; CNN: complex neural network; HNN: hyperbolic neural network; BNN: bicomplex neural network; QNN: quaternion neural network).

Franc) as the target system. Here, the total number of data points was 3523 (4/Jan./2002 ~ 5/Aug./2016). We split the data into two sets, namely, training set (4/Jan./2002 ~ 2/Jun./2009) and test set (3/Jun./2009 ~ 5/Aug./2016). The training set was used in closed tests for evaluating the training accuracy, while the test set was used in open tests for evaluating the estimation accuracy of the proposed direct transfer function identifiers. Assuming that the dynamics of historical data can be represented by the autoregressive

$$(AR) \text{ model } y_l(t) = \sum_{s=1}^m a_{ls}y_l(t-s) + u_l(t) \quad (l = 1, 2, 3, 4),$$

the input vector of the hypercomplex-valued neural network in the direct transfer function identifier was defined as $\xi(t) = [y(t-1) \ y(t-2) \ \dots \ y(t-m) \ u(t)]^T$. In the experiments, y_1, y_2, y_3 and y_4 denoted the currency pairs USD/JPY, GBP/JPY, EUR/JPY and CHF/JPY, respectively, and $u_l(t)$ was random noise generated in the interval $[-0.02, 0.02]$. Figure 4 shows the result of network topology optimization of the quaternion neural network by using the Akaike Information Criterion (AIC) in the closed test. Here the components of the input vector were $y = y_1 + y_2i + y_3j + y_4k$ and $u = u_1 + u_2i + u_3j + u_4k$, while the estimation results were given by $\hat{y}(t) = [\text{Re}[x^{(o)}(t)] \ \text{Im}_i[x^{(o)}(t)] \ \text{Im}_j[x^{(o)}(t)] \ \text{Im}_k[x^{(o)}(t)]]^T$.

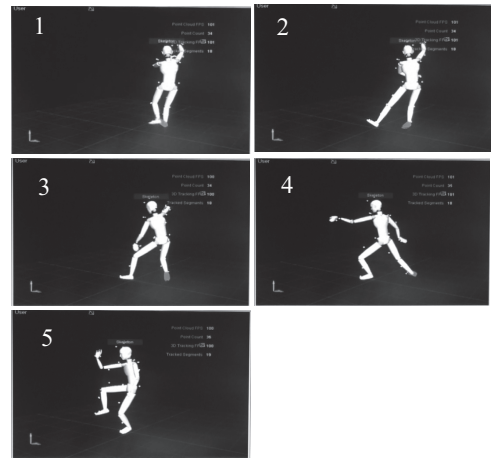


Fig. 6. Representation of movement ‘Push down and stand on one leg’ in 24-form *Taijiquan* with CG model based on captured motion data.

As shown in Fig. 4, the optimal order of the input m is one and the optimal number of neurons in the hidden layer M is one because of the lowest number of the AIC. Figure 5 shows the normalized cost function averaged within the range of each test. Here, the normalized cost function is averaged using 20 results of each neural network. Although the normalized cost functions in the closed test are almost identical, the normalized cost function of each hypercomplex-valued neural network in the open test is smaller than that of the real number neural network. These results indicate the effectiveness of the hypercomplex-valued neural network in applications involving multiple-input multiple-output system identification.

Next, the direct transfer function identifier was applied to estimate human movements acquired by an optical motion capture system. Here, the basic routine of 24-form *Taijiquan* was considered as the movement and the motion data of the movement ‘Push down and stand on one leg’, shown in Fig. 6, in which one arm is raised and the other is lowered while simultaneously standing on one leg – a crouching step changes the high position to a low one and then one leg is raised with a weight supported by the other one – was used as the target system. A subject (Japanese male, age: 22, height: 177 cm) who learned *Taijiquan* for six months from a *Taijiquan* instructor performed the movement twice, and the 3D motion data of 36 body parts were captured at a sampling rate of 30 fps [17]. In the motion capture system, the x - and the z -axes comprise the horizontal plane, and the y -axis is perpendicular to the aforementioned plane. In the computational experiment, 3D motion data of the head $(x_h(t), y_h(t), z_h(t))$ and centre of the waist $(x_w, y_w(t), z_w(t))$ were considered as the target data. The total number of data points was 2646, and the data was split up into two sets – training set covering the range $[0, 1322]$ and test set covering the range $[1323, 2646]$. Given that the motion had a constant acceleration characteristic, the second-order AR model was assumed to define the input vector of the hypercomplex-valued neural network $\xi(t) = [y_h(t-1) \ y_w(t-1) \ y_h(t-2) \ y_w(t-2) \ u_h(t) \ u_w(t)]^T$, where $y_l = x_l i + y_l j + z_l k$ and $u_l = u_{l1} i + u_{l2} j + u_{l3} k$ ($l = h, w$) represent random noise. The x -, y - and z - positions of the

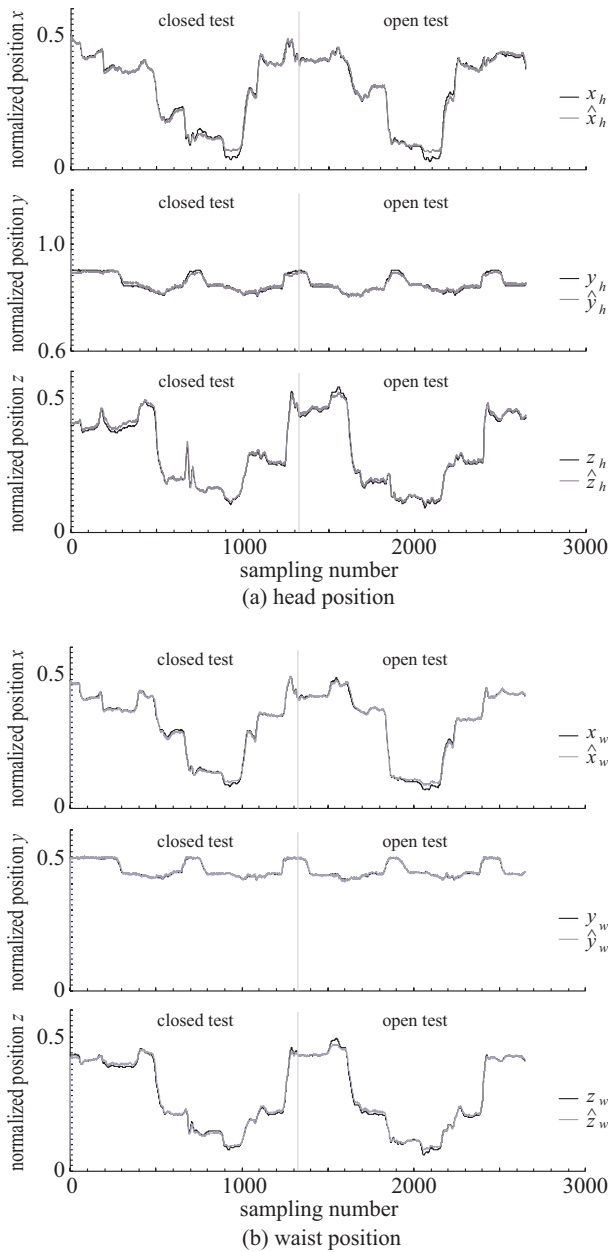


Fig. 7. Experimental results of motion data estimation using quaternion neural network with 6–2–2 (\boxplus) network topology; the black line indicates the measured data and the grey line indicates the output of the quaternion neural network (Top: 3D position of head, bottom: 3D position of waist).

head and waist were estimated from the imaginary parts of the quaternion neural network's output: $\hat{\mathbf{y}}(t) = [\text{Im}_1[\mathbf{x}^{(o)}(t)] \text{Im}_j[\mathbf{x}^{(o)}(t)] \text{Im}_k[\mathbf{x}^{(o)}(t)]]^T$. Figure 7 shows the system response for the case in which the direct transfer function identifier consisted of a quaternion neural network with a 6–2–2 network topology. As shown in the figure, the output of the quaternion neural network could approximate the motion data in the closed and the open tests. This result indicates the usefulness of the proposed direct transfer function identifier using the hypercomplex-valued neural network.

IV. CONCLUSIONS

In this study, the capability of hypercomplex-valued neural networks was investigated and their application to system

identification was explored. A direct transfer function identifier in which the output of a target system is estimated by the multilayer hypercomplex-valued neural network was designed and its characteristics were evaluated. Computational experiments using a complex neural network, hyperbolic neural network, bicomplex neural network and quaternion neural network were conducted to estimate the output of a discrete-time dynamic plant, historical foreign exchange data and the motion data of *Taijiqian* movements. The experimental results confirmed the feasibility and effectiveness of the proposed hypercomplex-valued neural network-based direct transfer function identifier.

REFERENCES

- [1] B. K. Tripathi, High Dimensional Neurocomputing - Growth, Appraisal and Applications -, Springer India, 2015.
- [2] T. Nitta (ed.), Complex-Valued Neural Networks - Utilizing High-Dimensional Parameters -, Information Science Reference, 2009.
- [3] A. Hirose (ed.), Complex-Valued Neural Networks - Advances and Applications -, IEEE Press, WILEY, 2013.
- [4] L. Fortuna, G. Muscato and M. G. Xibilia, "A Comparison Between HMLP and HRBF for Attitude Control", *IEEE Transactions on Neural Networks*, Vol. 12, No. 2, pp. 318–328, 2001.
- [5] H. Kusamichi, T. Isokawa, N. Matsui, Y. Ogawa and K. Maeda, "A New Scheme Color Night Vision by Quaternion Neural Network", in *Proceedings of the 2nd International Conference on Autonomous Robots and Agents*, pp. 101–106, 2004.
- [6] P. Arena, R. Caponetto, L. Fortuna, G. Muscato and M. G. Xibilia, "Quaternionic Multilayer Perceptrons for Chaotic Time Series Prediction", *IEICE Transactions Fundamentals*, Vol. E79-A, No. 10, pp. 1682–1688, 1996.
- [7] S. Buchholz and N. Le Bihan, "Optimal Separation of Polarized Signals by Quaternionic Neural Networks", in *Proceedings of 14th European Signal Processing Conference*, pp. 1–5, 2006.
- [8] B. C. Ujang, C. C. Took and D. P. Mandic, "Quaternion-Valued Non-linear Adaptive Filtering", *IEEE Transactions on Neural Networks*, Vol. 22, No. 8, pp. 1193–1206, 2011.
- [9] F. Shang and A. Hirose, "Quaternion Neural-Network-Based PolSAR Land Classification in Poincare-Sphere-Space", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 52, No. 9, pp. 5693–5703, 2014.
- [10] T. Ogawa, "Neural Network Inversion for Multilayer Quaternion Neural Networks", *Computer Technology and Applications*, Vol. 7, pp. 73–82, 2016.
- [11] Y. Cui, K. Takahashi and M. Hashimoto, "Design of Control Systems Using Quaternion Neural Network and Its Application to Inverse Kinematics of Robot Manipulator", in *Proceedings of 2013 IEEE/SICE International Symposium on System Integration*, pp.527–532, 2013.
- [12] K. Takahashi, Y. Hasegawa and M. Hashimoto, "Remarks on Quaternion Neural Network-based Self-tuning PID Controller", in *Proceedings of the 3rd International Symposium on Computer, Consumer and Control*, pp.231–234, 2016.
- [13] K. Takahashi, Y. Wakaume, T. Hirayama and M. Hashimoto, "Remarks on Hypercomplex-valued Neural Network-based Direct Adaptive Controller", in *Proceedings of the 2nd International Conference on Robotics and Automation Sciences / International Conference on Multimedia Systems and Signal Processing*, S18, 2016.
- [14] T. Isokawa, H. Nishimura and N. Matsui, "Quaternionic Multilayer Perceptron with Local Analyticity", *Information*, Vol. 3, pp. 756–770, 2012.
- [15] B. C. Ujang, C. C. Took and D. P. Mandic, "Split Quaternion Nonlinear Adaptive Filtering", *Neural Networks*, Vol. 23, No. 3, pp. 426–434, 2010.
- [16] E. Hitzler, "Algebraic Foundations of Split Hypercomplex Nonlinear Adaptive Filtering", *Mathematical Methods in the Applied Sciences*, Vol. 36, No. 9, pp. 1042–1055, 2013.
- [17] K. Takahashi, T. Koroyasu, Y. Kameda and M. Hashimoto, "Evaluation of Taijiqian Skill Acquisition Process using Motion Capture System and Simplified NIRS", *WIT Transactions on Information and Communication Technologies - Advances in Sport Science and Computer Science*, Vol. 57, pp.787–794, 2013.