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 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].

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

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

Fig. 1. Schematic of neural network–based direct transfer function identifier.
The norm of a complex number is defined as
\[ |z| = \sqrt{z \bar{z}} = \sqrt{z_0^2 + z_1^2}, \]
where \( z = z_0 + z_1i \) is the conjugate of the complex number \( z \).

Hyperbolic numbers consist of two complex numbers and an imaginary unit: \( \mathbb{D} := \{ z = z_0 + z_1i, z_2, z_3 \in \mathbb{R}, i_2^2 = 1 \} \).

The sum of two hyperbolic numbers \( z_1 \) and \( z_2 \) is given as
\[ z_1 + z_2 = (z_{10} + z_{20} + z_{11} + z_{21})i + (z_{12} + z_{22} + z_{13} + z_{23})h, \]
while the product of two hyperbolic numbers is given as
\[ z_1z_2 = \left( z_{10}z_{20} + z_{11}z_{21} + z_{12}z_{22} + z_{13}z_{23} \right) + i\left( z_{10}z_{21} + z_{11}z_{20} + z_{12}z_{23} + z_{13}z_{22} \right), \]
where the product of two quaternion numbers is given as
\[ |z| = \sqrt{z z^*} = \sqrt{\frac{1}{2} \left( z_0^2 + z_1^2 + z_2^2 + z_3^2 \right)}, \]
where \( z^* = z_0 - z_1i - z_2j - z_3k \) is the conjugate of the quaternion number \( z \).

The hypercomplex–valued neural network is trained to minimize the cost function \( J \) defined by the norm of the output error \( e(t) \) as follows:
\[ J(t) = \frac{1}{2} \sum_k |e_k(t)|^2, \]
where \( e_k(t) = d_k(t) - x_k^{(t)}(t) \), \( d_k(t) \) 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 \( \Delta J \) can be calculated as follows:
\[ \Delta J = J(t + 1) - J(t) \approx \left( \frac{\partial J(t)}{\partial \omega(t)} \right)^\top \Delta \omega(t) + \frac{\partial^2 J(t)}{\partial \omega(t) \partial \omega(t)^\top} \Delta \omega(t) \Delta \omega^\top(t). \]
Setting \( \Delta \omega(t) = -\eta \frac{\partial J(t)}{\partial \omega(t)^\top} \) and \( \Delta \omega^\top(t) = -\eta \frac{\partial J(t)}{\partial \omega(t)} \), yields
\[ \Delta J = -2\eta \left| \frac{\partial J(t)}{\partial \omega(t)} \right|^2 \leq 0, \]
where the factor \( \eta \) is positive. Increment of the parameters in the output layer is given as
\[ \Delta w_{kj}^{(o)}(t) = \eta \sum_{\sigma} \delta_k(t) x_{j}^{(n-1)}(t), \]
\[ \Delta \phi_k^{(o)}(t) = \eta \sum_{\sigma} \delta_k(t), \]
where \( \delta_k(t) = e_k(t) f_k'(y_k^{(t)}(t)) \), 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
\[ \Delta w_{ij}^{(n)}(t) = \eta \sum_{\sigma} \delta_j(t) x_i^{(n-1)}(t), \]
\[ \Delta \phi_j^{(n)}(t) = \eta \sum_{\sigma} \delta_j(t), \]
where \( \delta_j(t) = \sum_k \delta_k^{(n+1)}(t) w_{kj}^{(n)}(t) f_j'(y_j^{(n)}(t)) \).
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:

\[ y(t + d) = \sum_{s=1}^{n} A_{y_s} y(t - s + 1) + \sum_{s=0}^{m+d-1} A_{u_s} u(t - s) \quad (5) \]

where \( A_{y_s} = [\alpha_{y_1}, \ldots, \alpha_{y_q}] \) and \( A_{u_s} = [\alpha_{u_1}, \ldots, \alpha_{u_p}] \) are plant parameter matrices, \( n \) and \( m \) are plant orders, \( d \) is plant dead time and \( y(t) = [y_1(t), y_2(t), \ldots, y_q(t)]^T \), \( u(t) = [u_1(t), u_2(t), \ldots, u_p(t)]^T \). Considering the condition

\[ \lim_{t \to \infty} \{y(t + d) - \hat{y}(t + d)\} = 0 \]

yields

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

where

\[ \Lambda = [A_{y_1}, A_{y_2}, \ldots, A_{y_q}; A_{u_1}, A_{u_2}, \ldots, A_{u_{m+d-1}}], \]

and

\[ I(t) = [y(t), y(t - 1), \ldots, y(t - n + 1); u(t), u(t - 1), \ldots, u(t - m - d + 1)]^T. \]

Representing the mapping function of the hypercomplex–valued neural network as \( F_{nn}(\cdot) \) yields

\[ x^{(c)}(t) = F_{nn}(\omega(t), \xi(t)), \quad (7) \]

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

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, \( f(z) = f_0(z_0) + f_1(z_1) 1 + f_2(z_2) j + f_3(z_3) k \) in the quaternion neural network, where the function \( f_i(z) = 1/(1 + e^{-z^i}) (i = 0, 1, 2, 3) \) 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 \pm 0.5. Considering the dominant part of the plant for designing the direct transfer function identifier, the input of the hypercomplex–valued neural network was defined as \( \xi(t) = y(t - 1) + y(t - 2)j + u(t - 1)j + u(t - 2)k \). The estimated plant output comprised one of the imaginary parts of the quaternion neural network’s output, that is, \( \hat{y}(t) = \Im_m[x^{(c)}(t)] \), where \( \Im_m \) denotes the imaginary part of the imaginary unit k.

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 networks. 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
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/Jan./2009) and test set (3/Jan./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 identifier. Assuming that the dynamics of historical data can be represented by the autoregressive (AR) model \( y_t(t) = \sum_{s=1}^{m} a_{ts}y_{t-s} + u_t(t) \) \( (l = 1, 2, 3, 4) \), the input vector of the hypercomplex-valued neural network in the direct transfer function identifier was defined as \( \xi(t) = [ y(t-1) \ y(t-2) \ \ldots \ y(t-m) \ u(t) ]^T \). 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 \( u_t(t) \) 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 \( y = y_1 + y_2 + y_3 + y_4 \) and \( u = u_1 + u_2 + u_3 \), while the estimation results were given by \( \hat{y}(t) = \begin{bmatrix} \text{Re}[x^{(o)}(t)] \\ \text{Im}_1[x^{(o)}(t)] \\ \text{Im}_j[x^{(o)}(t)] \\ \text{Im}_k[x^{(o)}(t)] \end{bmatrix}^T \). 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 \( \xi(t) = \begin{bmatrix} y_h(t-1) \ y_w(t-1) \ y_h(t-2) \ y_w(t-2) \ u_h(t) \ u_w(t) \end{bmatrix}^T \), where \( y_j = x_i + y_i + z_i \) and \( u_l = u_{1i} + u_{2j} + u_{3k} \) \( (l = h, w) \) represent random noise. The \( x- \), \( y- \) and \( z- \) positions of the
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 Taijiquan movements. The experimental results confirmed the feasibility and effectiveness of the proposed hypercomplex–valued neural network–based direct transfer function identifier.

REFERENCES


Fig. 7. Experimental results of motion data estimation using quaternion neural network with 6–2–2 (∈) 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).

IV. CONCLUSIONS

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