IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY SECTION

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 29, 2020, accepted October 4, 2020, date of publication October 8, 2020, date of current version October 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029756

Estimation of ARMA Model Order via Artificial Neural Network for Modeling Physiological Systems

MD-BILLAL HOSSAIN, (Graduate Student Member, IEEE), JIHYE MOON, (Graduate Student Member, IEEE), AND KI H. CHON, (Senior Member, IEEE)

Department of Biomedical Engineering, University of Connecticut, Storrs, CT 06269, USA

Corresponding author: Ki H. Chon (ki.chon@uconn.edu)

This work was supported by the Office of Naval Research under Grant N00014-20-1-2708.

ABSTRACT Model order estimation is the most important but challenging step for system identification using an autoregressive moving average (ARMA) model. In this paper, we propose an artificial neural network (ANN) structure to estimate the best model order for ARMA modeling of linear, time-invariant systems using the system's input and output data. The proposed algorithm creates an equivalent ANN structure corresponding to an ARMA model and chooses the best model order using the neural network's mean squared error (MSE) loss function. The proposed method is validated on simulated ARMA model data and the performance is compared with the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). We considered three hypothetical linear systems and performed 100 Monte Carlo simulations for each model, with different data lengths, and with additive noise. For each of the three simulation models, the proposed method significantly outperformed the AIC and BIC in terms of the correct model order selection. Finally, the proposed ANN-based model order estimator was successfully applied to determine the dynamic relationship between heart rate (HR) and instantaneous lung volume (ILV) using an ARMA model. The results indicate that physiological and biological systems can be modeled with appropriate ARMA models obtained by the proposed algorithm to better understand the system dynamics.

INDEX TERMS ARMA, artificial neural network, model order, AIC, BIC.

I. INTRODUCTION

Proper mathematical description of a physiological system is often sought in order to analyze the system's overall behavior and to predict its output, given input data. Given the popularity of deep learning approaches, our study has attempted to determine a model order for a parametric-based autoregressive moving average (ARMA) representation with a convolution neural network (CNN) configuration [1]. A key challenge is that CNN-based deep learning requires a vast amount of training data, which is not readily available for most cases.

An ARMA model is often preferred for linear system identification because of its compact representation of the system's response based on the input and output data. For example, ARMA modeling from multiple inputs and delays has been proposed for biological systems in [2]. The highly flexible structure of the ARMA model has been widely used to find the dynamics of systems and found application in wide-ranging areas including brain potential identification [3], measuring the transient response of a subscale sailplane [4], and predicting heart rate (HR) from arterial blood pressure (ABP) and instantaneous lung volume (ILV) [2], [5], for example.

Two important steps for ARMA modeling are: (i) model order estimation and (ii) coefficient estimation. The first and the most important step in ARMA modeling is to determine the correct model order (i.e. determining the orders of AR and MA polynomials). There have been several methods proposed for ARMA model order estimation. The most popular methods are the final prediction error [6], Akaike Information Criterion (AIC) [7], Bayesian Information Criterion (BIC) [8], minimum description length (MDL) [9], and minimum eigenvalue criterion (MRV) [10]. Most of these methods rely on overdetermined model orders for AR and MA polynomials (i.e. $0 \leq p \leq p_{max}, 0 \leq q \leq q_{max}$)

The associate editor coordinating the review of this manuscript and approving it for publication was Kin Fong Lei.

and calculate the corresponding loss function for each of the combinations of the predetermined model orders. Finally, the combination of AR and MA orders that gives the least loss value is considered the correct model order.

In [11], the authors presented an accurate technique for estimating linear and nonlinear ARMA model parameters using the optimal parameter search (OPS) algorithm [12]. This approach has been shown to be more accurate than the above-noted model order determination techniques. The primary reason the OPS method works better than AIC, for example, is that the former is able to handle missing terms whereas the latter would assign some value since it is based on the traditional least-squares approach for model order estimation.

There have been some efforts using artificial intelligence including genetic algorithms [13], [14] and artificial neural networks [15]–[17] for model order determination. However, the model order estimation accuracy is not much better than it is for most of the aforementioned methods.

In this paper, we will describe the use of the artificial neural network (ANN) structure described in [18] for automatic ARMA model order identification. In [18], the authors showed the equivalence of the ANN structure and the ARMA model equation and estimated linear and nonlinear ARMA parameters using the neural network's weights. However, it was assumed that the model order was known, as the accuracy of the parameter estimation degraded when the model order was incorrectly chosen. To overcome this limitation, in this work, we used the feedforward ANN structure to determine the correct model order based on the neural network loss function criterion.

II. METHODOLOGY

A. SYSTEM IDENTIFICATION AND ARMA MODEL

System identification refers to the methodology of building a mathematical model structure optimizing the error criterion between a model output and a system's desired output based on some experimental data. A linear, time invariant system can be modeled by an ARMA model with the following equation:

$$y\,(n) = \sum_{i=1}^{p} a_i s\,(n-i) + \sum_{j=0}^{q} b_j x\,(n-j) + e(n) \qquad (1)$$

where $x\,(n)$ is the input to the system, $y\,(n)$ is the noiseless output and $e\,(n)$ is the residual error [19]. The first term of the right-hand side of equation (1) represents the AR polynomials, where p is the maximum AR order. Similarly, the second term is the MA polynomials and q represents the maximum MA order.

$$A\,(z)\,y\,(z) = B\,(z)\,x\,(z) \qquad (2)$$

where

$$A\,(z) = 1 - \sum_{i=1}^{p} a_i z^{-i}, \quad B\,(z) = \sum_{j=1}^{q} b_j z^{-j}. \qquad (3)$$

From equation (2), we can define the following transfer function:

$$H\,(z) = \frac{B\,(z)}{A\,(z)}. \qquad (4)$$

This system can be approximated using the ARMA model provided that $A\,(z)$ and $B\,(z)$ can be approximated with finite polynomials in the Z-domain. Fig. 1 shows a general block diagram for system identification.

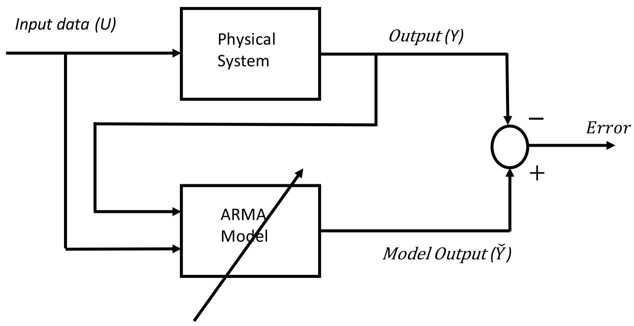

FIGURE 1. General system identification block diagram.

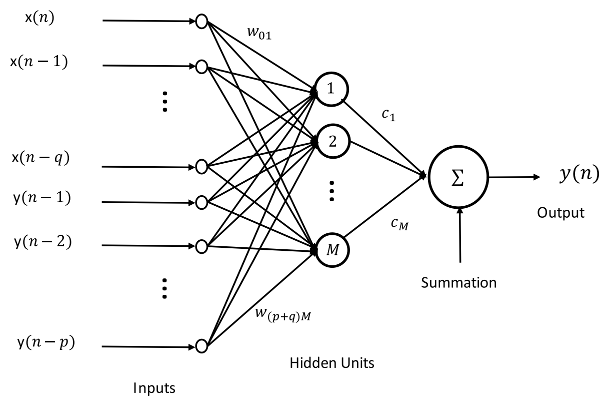

FIGURE 2. The proposed ANN topology.

B. NEURAL NETWORK

We propose a feedforward neural network where information flows in one direction (input to output) to create an ARMA model. Fig. 2 shows the proposed ANN structure with delayed input and output samples which are fed to the neural network inputs. Therefore, the neural network's output can be represented as a function f of present and past values of input and output samples, as described in the following equation:

$$y\,(n) = f(x(n), x(n-1), \ldots, x(n-q), \\ y(n-1), y(n-2), \ldots, y(n-p)), \qquad (5)$$

where p and q represent the orders of autoregressive and moving average polynomials, respectively.

Since all of the inputs in this neural network are composed of delayed input and output signals, this network can be considered as a dynamic network where each input is multiplied by its corresponding weight such that w_{jk} connects input

$x(n-j)$ with the k_{th} hidden unit and $w_{(q+i)k}$ connects $x(n-i)$ with the k_{th} hidden unit. The M hidden outputs are multiplied by output weights $c_1, c_2, \ldots, c_M$ and are summed together to estimate the present output, $y(n)$. We did not use any bias in the network. The output of the proposed ANN structure can be written as follows:

$$\check{y}(n) = \sum_{i=1}^{M} c_i \phi(v_i) + e(n), \tag{6}$$

where ϕ is the activation function used, c_i are the weights of the hidden unit i to the output, M is the number of hidden units, and v_i are the weighted sum of inputs to the hidden unit i. Therefore, v_i can be rewritten as follows:

$$v_i = \sum_{j=0}^{q} w_{ji} x(n-j) + \sum_{j=1}^{p} w_{(j+q)i} y(n-j), \tag{7}$$

where w is the neural network weight matrix with dimension of $(p+q+1) \times M$. Now, if we consider the polynomial representation of the activation $\phi(v_i)$ such that:

$$\phi(v_i) = \beta_0 + \beta_1 v_i + \beta_2 v_i^2 \ldots + \beta_n v_i^n + \ldots \tag{8}$$

then combining Eqs. (6) and (8) yields:

$$\begin{aligned}
&\check{y}(n) \\
&= c_1 \left(\beta_0 + \beta_1 v_1 + \beta_2 v_1^2 + \ldots + \beta_n v_1 + \ldots\right) \\
&\quad + c_2 \left(\beta_0 + \beta_1 v_2 + \beta_2 v_2^2 + \ldots + \beta_n v_2 + \ldots\right) \\
&\quad + \ldots + c_m \left(\beta_0 + \beta_1 v_M + \beta_2 v_M^2 + \ldots + \beta_n v_M + \ldots\right) \\
&\quad + e(n).
\end{aligned} \tag{9}$$

Replacing v_i from equations (7) and (9) and gathering the like terms, the following expression can be derived (up to 2^{nd} order nonlinearity for brevity):

$$\begin{aligned}
\check{y}(n) &= c_1\beta_0 + c_2\beta_0 + \ldots + c_M\beta_0 \\
&+ \sum_{j=0}^{q} (c_1\beta_1 w_{j1} + c_2\beta_1 w_{j2} \\
&+ \ldots + c_M\beta_1 w_{jM}) x(n-j) \\
&+ \sum_{j=1}^{p} (c_1\beta_1 w_{(j+q)1} + c_2\beta_1 w_{(j+q)2} \\
&+ \ldots + c_M\beta_1 w_{(j+q)M}) y(n-j) \\
&+ \sum_{j=1}^{p}\sum_{k=1}^{p} (c_1\beta_2 w_{(j+q)1} w_{(k+1)1} \\
&+ c_2\beta_2 w_{(j+q)1} w_{(k+q)2} \\
&+ \ldots + c_M\beta_2 w_{(j+q)M} w_{(k+q)M}) x(n-j) \\
&\times x(n-k) + \sum_{j=1}^{p}\sum_{k=1}^{p} (c_1\beta_2 w_{(j+q)1} w_{(k+1)1} \\
&+ c_2\beta_2 w_{(j+q)1} w_{(k+q)2} \\
&+ \cdots + c_M\beta_2 w_{(j+q)M} w_{(k+q)M}) \\
&\times y(n-j) y(n-k) + \cdots + e(n).
\end{aligned} \tag{10}$$

In cases where the neural network has an activation function of first order only, the higher order terms in Eq. (10) vanish (since $\beta_i = 0$, for $i \geq 2$). Therefore, Eq. (11) follows the same structure as Eq. (1)

$$\begin{aligned}
\check{y}(n) &= c_1\beta_0 + c_2\beta_0 + \ldots + c_M\beta_0 \\
&+ \sum_{j=0}^{q} (c_1\beta_1 w_{j1} + c_2\beta_1 w_{j2} \\
&+ \ldots + c_M\beta_1 w_{jM}) x(n-j) \\
&+ \sum_{j=1}^{p} (c_1\beta_1 w_{(j+q)1} + c_2\beta_1 w_{(j+q)2} \\
&+ \ldots + c_M\beta_1 w_{(j+q)M}) y(n-j) + e(n).
\end{aligned} \tag{11}$$

Comparing the coefficients from Eqs. (11) and (1), ARMA parameters a and b can be represented in terms of the ANN weights.

$$a(i) = \beta_1 \sum_{k=1}^{M} c_k w_{(i+q)k} \tag{12}$$

$$b(i) = \beta_1 \sum_{k=q+1}^{M} c_k w_{ik} \tag{13}$$

This approach has been used in [18] for ARMA coefficients estimation and it produced better results when compared to the traditional least squares methods. However, it was also shown that the accuracy of parameter estimation degraded with incorrect model order selection [18].

We have shown how ANN can be used to derive ARMA parameters using equations 12-13. Since these parameters are affected by the model order selection, it is critical to find an approach to best determine the model order. Similar to AIC and BIC model order criteria, we have used the minimization of the mean square error (MSE) as the criterion to find the optimal model order for a given system.

C. ARMA MODEL ORDER ESTIMATION

For a particular input and output time series data, the neural networked structure described in Fig. 1 is trained for a fixed number of epochs using a series of different orders ($1 \leq p \leq p_{max}, 0 \leq q \leq q_{max}$). It can be easily seen from the ANN structure that depending on the order combination, the input layer uses different lags of the input and output time series which are determined by the AR and MA order. It is expected that for the correct model order, the mean square error (MSE) of the neural network should be minimum. However, with higher order combinations as well as noise corruption, the loss value can still decrease but at a very slow rate. Therefore, selection of correct order based on minimum MSE loss gives over estimation in certain instances. In order to address this issue, we utilized both loss value and the rate change of loss to decide the correct model order.

For each combination of the model orders we sort the MSE values of different epochs in ascending order and calculated the average of the first five MSEs. Finally, we compute the percent change in the loss function as we increase the model order. Let us consider the average MSE loss in the last n epochs for kth combination of AR and MA orders is $L(k)$, where k is any number between 1 to the total number of combinations of AR and MA orders, N. Then, we defined the percent change in the loss for changes in the model order as:

$$\check{L}(k) = \frac{L(k-1) - L(k)}{L(k)} \times 100\% \tag{14}$$

where, $k = 2, 3, \ldots N$.

We compute the model orders' loss values using the log transformation. The order of the loss is defined as:

$$r(k) = integer\,(\log(L(k))). \tag{15}$$

For example, $r = \log(10^r)$. Finally, we defined the lowest model order combination (p, q) as the correct model order for which the order of the loss is the minimum, and the percent change in the loss function is above a certain positive threshold. There are some cases when there are only a few (one or two) or most of the combinations of model orders for which the order of the loss (r value) is minimum. In those cases, we selected the lowest model order combination for which the loss value was less than 1.5 times the minimum loss and the rate change of loss was greater than a certain positive threshold as the correct model orders. It should be noted that for $k = 1$, there is no percentage change in the loss value. Therefore, we considered this as the correct order if the order of the loss was the minimum.

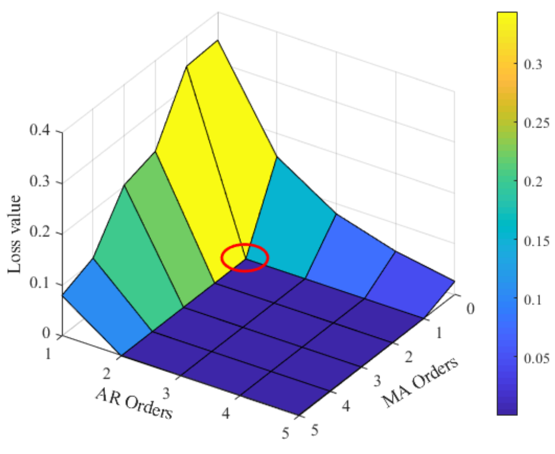

FIGURE 3. MSE loss for different combinations of AR and MA orders.

Fig. 3 shows the 3D loss function plot for an ARMA (2,1) model, in which the red circle marks the true model orders. It can be observed from the figure that the loss value is also very low for orders beyond (2,1). However, the rate change of loss is also very low, which suggests that increasing orders beyond (2,1) does not significantly add extra contribution to the model.

For the input and output time series data, the overall procedure of the algorithm for model order determination is summarized in Table 1.

TABLE 1. Model order estimation algorithm.

```
Count=0
for p = 1: p_max
   for q = 0: q_max
      Orders(count,:) = (p,q)
      data = format_data(y,x,p,q)
      Loss(count,:) = train_NN(data,nepoch)
      avg_loss(count) = mean_minimum_5(Loss)
   count = count + 1
end both loops
     percent_change
     = get_percent_loss_change(avg_loss)
Loss_order = get_order_loss(avg_loss)
min_order  = min(loss_order)
correct_index = get_index_where(order
                  = min _order and percent_change
                  > threshold)
actual_order = Orders(correct_index,:)
```

III. SIMULATION AND RESULTS

A. PERFORMANCE ON SIMULATED DATA

The proposed method for ARMA model order identification was validated on three different simulated models. As most real world systems can be modeled with ARMA(5,5), as described in [20], in this paper we fixed the maximum AR and MA orders at 5 (AR(1-5) and MA (0-5)).

For the neural network, we used ($M = 20$) hidden units (based on trials of 5, 10, 20, 50, and 100) and we trained the model up to 25 epochs (with a batch size of 20) for each of the combinations of the AR and MA orders. For updating the neural parameter, we used the Adam optimizer [21] with a moderate learning rate of 0.008 which was chosen empirically to provide an optimal training. The Adam optimizer is a popular first order gradient-based optimization algorithm which is widely used in deep learning applications [22]. Since learning rates do affect the loss function [23], we fixed the empirically-derived optimal learning rate throughout the entire analysis. Fig.4. shows a representative example of training the neural network. We segmented the data sequence into training (700 samples) and validation (300 samples) to observe the loss characteristics during the training process. As shown in Fig. 4, the MSE loss values for both training and validation datasets follow the same decreasing trend, which indicates non-overfitting of the data. We found that the rate of MSE reduction becomes minimal after ~15 epochs. Since we only considered linear time invariant systems, we used the rectified linear unit (ReLU) activation function [24] for the hidden units. However, other non-linear functions can also be used with linear approximations. For example, non-linear functions can be approximated using the first order Taylor's expansion as described in [25]. The computation was performed in Python 3.7 version with PyTorch package [26].

For each combination of the model orders, we computed the average training MSE loss value of the last 10 epochs

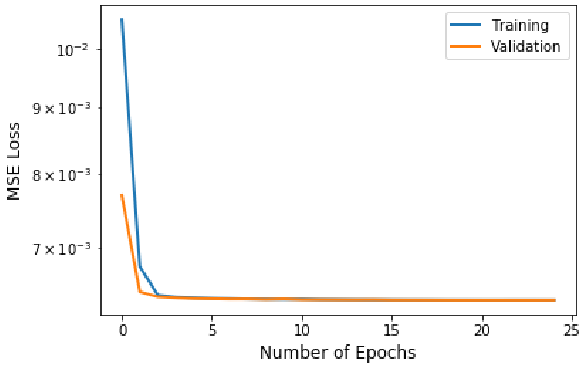

FIGURE 4. Training neural network.

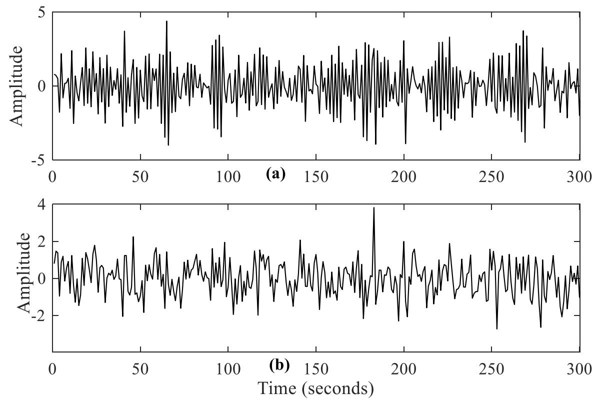

FIGURE 5. Input signal (a) and corresponding ARMA model output for eq. (15) (b).

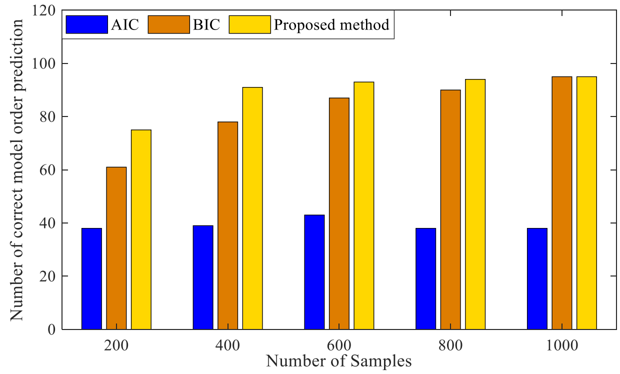

FIGURE 6. Comparison of correct order estimation using different order estimation techniques for an ARMA (2,0) model.

where loss value reached a steady state. For each of the systems to be identified, we performed 100 Monte Carlo simulations at each of the different lengths (200, 400, 600, 800, and 1000 samples) of time series and performed model order estimation using the proposed method, AIC, and BIC criteria for comparison of these methods' performance. Model identification using the AIC and BIC criteria was performed using the R software described in [27]. This approach requires the time series data to be stationary and invertible.

In order to make the time series data stationary and invertible, the coefficients a and b were chosen such that all the roots of AR and MA coefficients are outside the unit circle. At first, we started with random initial coefficients and then we adjusted them until they satisfied the stationarity and invertibility criteria.

Example 1: We simulated the following ARMA (2,0) model and performed 100 Monte Carlo simulations.

$$H(z) = \frac{1}{1 - 0.2688z^{-1} - 0.9169z^{-2}}. \tag{16}$$

While the simulation example has the exact known model order, in practical situation, based on the known input and output data, we do not know the precise model order. Hence, the purpose of the work was to determine the correct model order given the initial incorrect (overdetermined) model order assumption using only the input and output data using the neural network. For example, in simulation example involving Eq. (15), while the correct model order is ARMA (2,0), we initially assumed the model order to be ARMA (5,5). From this overdetermined ARMA model order the goal was to determine the correct ARMA (2,0) model order using our proposed NN and compare its results with AIC and BIC. The same strategy was applied to other simulation examples to follow.

Fig. 5 shows a segment of the input signal (generated with Gaussian white noise) and corresponding ARMA model output signal of Eq. (16). We simulated 100 different realizations of the system's response with time series lengths of 200, 400, 600, 800, and 1000

The comparison of the order estimation results is shown in Fig. 6. The proposed ANN-based method provides accurate order estimation for every length of the signals. It can also be seen that the proposed method is less sensitive to the data length than is the BIC criterion. The performance of the AIC criterion is low compared to both the BIC criterion and the proposed method.

Example 2: We considered a second hypothetical system simulated using the following ARMA (2,1) model:

$$H(z) = \frac{1 + 0.2910z^{-1}}{1 - 0.7047 - 0.5553z^{-2}}. \tag{17}$$

Similar to example 1, we performed 100 Monte Carlo simulation of the system's response at five different time series lengths. The model order estimation results are presented in Fig. 7. The plots indicate that the performance of the proposed ANN-based model order selection is better than the AIC and BIC criteria. In addition, the proposed method provided accurate order estimation almost irrespective of the data lengths, whereas the model order estimation performance based on AIC and BIC degrades for smaller lengths of time series.

Example 3: We considered a third hypothetical system with a relatively higher order (ARMA (4, 2)) model:

$$H(z) = \frac{1+0.2823z^{-1}-0.4690z^{-2}}{1+0.3107z^{-1}+0.3251z^{-2}+0.0561z^{-3}-0.8154z^{-4}} \tag{18}$$

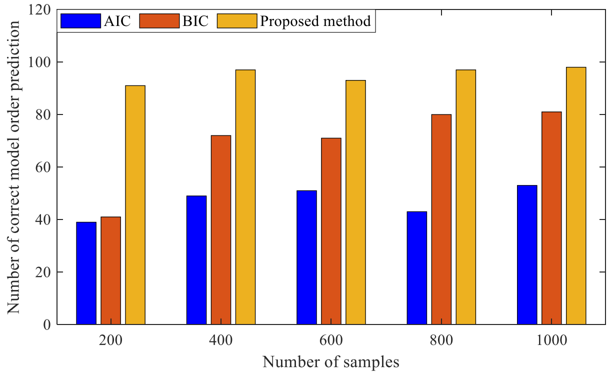

FIGURE 7. Comparison of correct order estimation using different order estimation techniques for ARMA (2,1) model.

TABLE 2. Model order detection performance of the system of eq. (19).

Method	Time series samples				
	200	400	600	800	1000
AIC (%)	55	52	43	46	32
BIC (%)	71	67	60	58	44
ANN(%)	99	99	100	99	100

Similar to examples 1 and 2, we performed 100 Monte Carlo realizations of Eq. (19) at different time series lengths (200, 400, 600, 800, and 1000). The model order estimation results for this system are shown in Table 2. Again, the proposed method outperforms the AIC and the BIC selection criteria in terms of the number of correctly detected model orders. Moreover, the proposed method could identify the correct model order almost all times, irrespective of the time series length, whereas the AIC and BIC criteria are more sensitive to the data length and their performances decreased for longer time series lengths.

We compared the effect of additive noise on the model order estimation for Eqs. (17-18). We added Gaussian white noise that is independent from the input to the output data (of 1000 samples) at three different SNR levels consisting of 50 dB, 30 dB, and 10 dB, and compared the model order estimation performances for these two simulation examples. As shown in Table 3, the proposed ANN criterion provided significantly better performance when compared to AIC and BIC criteria at 50 dB and 30 dB SNR levels. At the lowest SNR level (e.g. 10 dB) the performance of the ANN based order selection is again better than BIC and AIC but its performance is not as good as the higher SNR as expected [28]. This is because the output data are corrupted with significant noise, all methods including the ANN have trouble estimating the correct model order.

B. APPLICATION TO EXPERIMENTAL DATA

In this subsection, we will demonstrate that the proposed ANN-based model order selection method can be used to analyze experimentally obtained instantaneous lung volume (ILV) and heart rate (HR) data. Since respiration affects fluctuations in HR, it is interesting to understand the dynamic

TABLE 3. Effect of measurement noise on model order estimation.

	Method	SNR		
		50 dB	30 dB	10 dB
Eqn. (18)	AIC(%)	53	49	3
	BIC(%)	67	68	11
	ANN(%)	100	100	29
	AIC(%)	51	43	2
Eqn. (17)	BIC(%)	80	76	9
	ANN(%)	98	95	21

relationship between ILV and HR. In the past, several linear system methods such as the power spectrum [29], transfer function [30], and impulse response [31] have been performed on ILV and HR data. In this paper, our purpose is to examine if the proposed model order selection method can find proper model orders to obtain physiologically interpretable impulse response functions [18], [31].

We used experimentally obtained human subject ILV and HR data published in [30], [32]. While detailed descriptions of the data collection can be found in [30], we provide a brief summary here. This dataset consists of surface electrocardiogram (S-ECG) and changes in ILV from five subjects. A 13-minute data sample was collected for the supine position. The S-ECG and ILV signals were recorded at a sampling frequency of 360 Hz, which is high enough to allow accurate QRS detection [33], [34]. The HR and ILV data were then down-sampled to 3 Hz, since the dynamics of HR fluctuations are located at frequencies below 0.5 Hz [18], [29]–[31]. For the model order estimation, the input (ILV data) and output (HR data) pair of a 500 data point segment were used. The neural network was trained for a wide range of model orders (AR (1-15) and MA (0-15)). The best model order was chosen using proposed ANN based model order selection and the BIC criterion as well. We only considered BIC criterion since it has shown to be more accurate than AIC. Finally, the ARMA parameters were estimated using the least squares method for the model order estimated by the ANN and the BIC criterion. The impulse response functions obtained for five different subjects were then averaged to obtain an overall impulse response function, which is shown in Fig. 8. The upper and lower panels show the averaged impulse response obtained using the model order estimated by ANN based proposed model and BIC criterion, respectively. The averaged impulse responses in both panels have a fast positive peak followed by an underdamped wave, which has also been shown in previous studies [18], [30], [31]. Physiologically, the abrupt rise in HR is due to respiration modulating the increase due to the autonomic nervous system. The red and yellow lines mark the standard deviation bounds at each of the sampled values. It can be also seen that the impulse response functions obtained using BIC criterion has larger deviations especially around the first positive peak value.

IV. DISCUSSION

The results presented in this paper suggest that the proposed ANN-based model order detection algorithm can determine

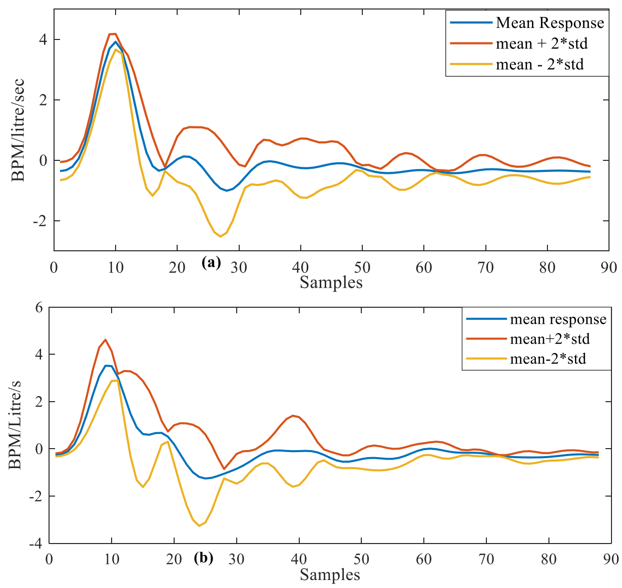

FIGURE 8. **Averaged impulse response obtained using ILV and HR data. (a) Using the proposed model order selection. (b) Using BIC criterion.**

the correct ARMA model order to model a physical system using experimentally obtained input and output data. In the process of obtaining the correct model order, the ARMA coefficients can also be obtained for the best model order from the neural network weight matrix using Eqs. (12-13). Therefore, the proposed method can be used for identification of any linear time-invariant system.

The performance of the proposed ANN-based model order selection technique was validated on three hypothetical systems. We performed 100 Monte Carlo simulations for each of the systems, using different time series lengths. The model order estimation performance when using AIC and BIC selection criteria was found to be sensitive to the data length, as shown in another research paper [35]. In contrast, the proposed ANN approach shows low dependency on the data length and provides accurate model order almost irrespective of the data length. Moreover, the application of the proposed ANN based model order selection method on the ILV and HR data resulted in more consistent impulse response function. While five subjects may not be enough to represent the 95% confidence bounds, comparing the pattern of the impulse response with that of the previously published studies [18], [30], [31], we can conclude that the proposed ANN based model order selection shows more similar performance in obtaining physiologically interpretable impulse response function than via BIC criterion. This suggests that the proposed method can be applied on the physiological data to obtain more consistent estimates of the system's dynamics.

While providing a better order estimation accuracy, the proposed ANN approach also requires higher computational time. For our study, the ANN required 9 sec (data length 400 samples) to 16 sec (data length 1000 samples) to the estimate the model order, for example, the Eq, (18). This time is calculated using a Windows 10 computer with intel(R) Core (TM) i7-8700k CPU @3.70 GHz and memory of 32 GB. The number of parameters of the ANN varied between a minimum of 60 (for ARMA (1,0)) to a maximum of 240 (for ARMA (5,5)). However, for time-invariant systems, once the model order is known, it can be used in the future without re-estimation. This approach can also be extended to time-varying dynamic systems. In addition, we considered only linear ARMA models in this paper. This work can be easily generalized for any nonlinear system as well, by incorporating non-linearity in the activation function.

V. CONCLUSION

We have shown that an ANN structure can be used for accurate model order estimation of the ARMA model using the input and the output data. We considered three different arbitrary linear time invariant systems and performed 100 Monte Carlo simulations at each of the different data lengths. Finally, we compared the model order estimation performance of the proposed method with that of the AIC and BIC selection criteria on the 100 realizations. For each of the systems, the proposed model order selection method showed significantly better performance than did either AIC or BIC. Even though we considered systems with one input and one output, this approach can be generalized for multiple inputs and outputs. In future work, we can generalize the method for nonlinear ARMA models to include the nonlinear system characteristics as well. It should be mentioned that even though in this paper we considered the AR and MA order up to 5, this is not a limitation of this approach. The algorithm can be generalized for any model orders but at the cost of increased computational time.

REFERENCES

[1] W. Hoh Tang and A. Röllin, "Model identification for ARMA time series through convolutional neural networks," 2018, *arXiv:1804.04299*. [Online]. Available: http://arxiv.org/abs/1804.04299

[2] M. H. Perrott and R. J. Cohen, "An efficient approach to ARMA modeling of biological systems with multiple inputs and delays," *IEEE Trans. Biomed. Eng.*, vol. 43, no. 1, p. 1, Jan. 1996, doi: 10.1109/10.477696.

[3] S. Cerutti, G. Chiarenza, D. Liberati, P. Mascellani, and G. Pavesi, "A parametric method of identification of single-trial event-related potentials in the brain," *IEEE Trans. Biomed. Eng.*, vol. 35, no. 9, pp. 701–711, Sep. 1988, doi: 10.1109/10.7271.

[4] J. J. Hollkamp and S. M. Batill, "Automated parameter identification and order reduction for discrete time series models," *AIAA J.*, vol. 29, no. 1, pp. 96–103, Jan. 1991, doi: 10.2514/3.10550.

[5] S. Lu, K. Hwan Ju, and K. H. Chon, "A new algorithm for linear and nonlinear ARMA model parameter estimation using affine geometry [and application to blood flow/pressure data]," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 10, pp. 1116–1124, Oct. 2001, doi: 10.1109/10.951514.

[6] H. Akaike, "Fitting autoregressive models for prediction," *Ann. Inst. Stat. Math.*, vol. 21, no. 1, pp. 243–247, Dec. 1969, doi: 10.1007/BF02532251.

[7] H. Akaike, "A Bayesian analysis of the minimum AIC procedure," *Ann. Inst. Stat. Math.*, vol. 30, no. 1, pp. 9–14, Dec. 1978, doi: 10.1007/BF02480194.

[8] H. Akaike, "Statistical predictor identification," *Ann. Inst. Stat. Math.*, vol. 22, no. 1, pp. 203–217, Dec. 1970, doi: 10.1007/BF02506337.

[9] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978, doi: 10.1016/0005-1098(78)90005-5.

[10] G. Liang, D. M. Wilkes, and J. A. Cadzow, "ARMA model order estimation based on the eigenvalues of the covariance matrix," *IEEE Trans. Signal Process.*, vol. 41, no. 10, pp. 3003–3009, Oct. 1993, doi: 10.1109/78.277805.

[11] R. Zou, H. Wang, and K. H. Chon, "A robust time-varying identification algorithm using basis functions," *Ann. Biomed. Eng.*, vol. 31, no. 7, pp. 840–853, Jul. 2003, doi: 10.1114/1.1584683.

[12] H. Zhao, S. Lu, R. Zou, K. Ju, and K. H. Chon, "Estimation of time-varying coherence function using time-varying transfer functions," *Ann. Biomed. Eng.*, vol. 33, no. 11, pp. 1582–1594, Nov. 2005, doi: 10.1007/s10439-005-7045-4.

[13] Z. S. Abo-Hammour, O. M. K. Alsmadi, A. M. Al-Smadi, M. I. Zaqout, and M. S. Saraireh, "ARMA model order and parameter estimation using genetic algorithms," *Math. Comput. Model. Dyn. Syst.*, vol. 18, no. 2, pp. 201–221, Apr. 2012, doi: 10.1080/13873954.2011.614068.

[14] B. C. Ervural, O. F. Beyca, and S. Zaim, "Model estimation of ARMA using genetic algorithms: A case study of forecasting natural gas consumption," *Procedia—Social Behav. Sci.*, vol. 235, pp. 537–545, Nov. 2016, doi: 10.1016/j.sbspro.2016.11.066.

[15] J. Sjöberg, H. Hjalmarsson, and L. Ljung, "Neural Networks in System Identification," *IFAC Proc.*, vol. , vol. 27, no. 8, pp. 359–382, Jul. 1994, doi: 10.1016/S1474-6670(17)47737-8.

[16] J. K. Lee and W. C. Jhee, "A two-stage neural network approach for ARMA model identification with ESACF," *Decis. Support Syst.*, vol. 11, no. 5, pp. 461–479, Jun. 1994, doi: 10.1016/0167-9236(94)90019-1.

[17] W. C. Jhee, K. C. Lee, and J. K. Lee, "A neural network approach for the identification of the box-jenkins model," *Network: Comput. Neural Syst.*, vol. 3, no. 3, pp. 323–339, Jan. 1992, doi: 10.1088/0954-898X_3_3_005.

[18] K. H. Chon and R. J. Cohen, "Linear and nonlinear ARMA model parameter estimation using an artificial neural network," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 3, pp. 168–174, Mar. 1997, doi: 10.1109/10.554763.

[19] S. L. M. Li, *Digital Spectral Analysis*, 2nd ed. New York, NY, USA: Dover, 2019.

[20] K. C. Lee and S. B. Oh, "An intelligent approach to time series identification by a neural network-driven decision tree classifier," *Decis. Support Syst.*, vol. 17, no. 3, pp. 183–197, Jul. 1996.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[22] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On optimization methods for deep learning," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, Jun. 2011, pp. 265–272.

[23] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*. [Online]. Available: http://arxiv.org/abs/1609.04747

[24] A. Fred Agarap, "Deep learning using rectified linear units (ReLU)," 2018, *arXiv:1803.08375*. [Online]. Available: http://arxiv.org/abs/1803.08375

[25] T. A. Tutunji, "Parametric system identification using neural networks," *Appl. Soft Comput.*, vol. 47, pp. 251–261, Oct. 2016, doi: 10.1016/j.asoc.2016.05.012.

[26] A. Paszke, "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv:1912.01703*. [Online]. Available: http://arxiv.org/abs/1912.01703

[27] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: TheforecastPackage forR," *J. Stat. Softw.*, vol. 27, no. 3, pp. 1–22, 2008, doi: 10.18637/jss.v027.i03.

[28] A. Al-Smadi and D. Mitchell Wilkes, "Robust and accurate ARX and ARMA model order estimation of non-Gaussian processes," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 759–763, Mar. 2002, doi: 10.1109/78.984778.

[29] S. Akselrod, D. Gordon, F. Ubel, D. Shannon, A. Berger, and R. Cohen, "Power spectrum analysis of heart rate fluctuation: A quantitative probe of beat-to-beat cardiovascular control," *Science*, vol. 213, no. 4504, pp. 220–222, Jul. 1981, doi: 10.1126/science.6166045.

[30] J. P. Saul, R. D. Berger, M. H. Chen, and R. J. Cohen, "Transfer function analysis of autonomic regulation. II. Respiratory sinus arrhythmia," *Amer. J. Physiol.-Heart Circ. Physiol.*, vol. 256, no. 1, pp. 153–161, Jan. 1989, doi: 10.1152/ajpheart.1989.256.1.H153.

[31] K. Yana, J. P. Saul, R. D. Berger, M. H. Perrott, and R. J. Cohen, "A time domain approach for the fluctuation analysis of heart rate related to instantaneous lung volume," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 74–81, Jan. 1993, doi: 10.1109/10.204773.

[32] K. H. Chon, T. J. Mullen, and R. J. Cohen, "A dual-input nonlinear system analysis of autonomic modulation of heart rate," *IEEE Trans. Biomed. Eng.*, vol. 43, no. 5, pp. 530–544, May 1996, doi: 10.1109/10.488800.

[33] T. F. O. T. E. S. Electrophysiology, "Heart rate variability," *Circulation*, vol. 93, no. 5, pp. 1043–1065, Mar. 1996, doi: 10.1161/01.CIR.93.5.1043.

[34] M. B. Hossain, S. K. Bashar, A. J. Walkey, D. D. Mcmanus, and K. H. Chon, "An accurate QRS complex and p wave detection in ECG signals using complete ensemble empirical mode decomposition with adaptive noise approach," *IEEE Access*, vol. 7, pp. 128869–128880, 2019, doi: 10.1109/ACCESS.2019.2939943.

[35] L. Fenga, "Bootstrap order determination for ARMA models: A comparison between different model selection criteria," *J. Probab. Statist.*, vol. 2017, pp. 1–12, 2017.