Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

Dynamics Modeling of Musical String by Linear Scattering Recurrent Network*

S.F. Liang and Alvin W.Y. Su*

Depart. of Control Eng., National Chiao-Tung University, Hsin-Chu, Taiwan.
*Depart of CSIE, Chung-Hwa Polytechnic Inst., Hsin-Chu, Taiwan.
email: alvin@cello.ccl.itri.org.tw

Abstract

Music synthesis by physical modeling methods becomes a major research topic in the related area when FM synthesis and Wavetable synthesis cannot satisfy the demanding users. Combining the property of wave propagation and the associate discrete-time implementation, it is possible to generate realistic and dynamic musical tones. In this paper, we first start from the modeling of a musical string by proposing a class of neural networks called Linear Scattering Recurrent Network (LSRN) which employs the measurement of the response of a plucked string as the learning data such that the model can be trained to be a counterpart of the string in the synthesis domain. The correspondent learning algorithm and computer simulations are given to demonstrate the encouraging modeling results.

1. Introduction

With the introduction of electronic music, many techniques for generating musical tones have been proposed such as *FM* (frequency modulation) synthesis and *Wavetable* synthesis which are two most popular methods used nowadays. However, the sound quality cannot meet the requirements of the most demanding users, especially the reproduction of the musical dynamics of most instruments. In order to have the synthesis result closer to the sound generated by real instruments, Smith proposed the so-called *Digital Waveguide Filter* technique [3][13]. This synthesis algorithm starts from the equation of a 1-D traveling wave and implements the solution to the equation on a discrete-time model. In his later efforts in the physical modeling synthesis, algorithms of simulating the sounding mechanisms such as the reed-driven ones and bowed-driven ones were studied [2][3]. These techniques make some very realistic sounds such as those of an oboe and become more and more popular in the music synthesis business.

DWF uses digital filters to simulate simple wave propagation on a discrete-time system. For example, the model of an ideal string can be converted into a discrete-time system by using the DWF. Although the traveling waves can be explicitly simulated in the waveguide model, it is difficult to find the correct parameters of the DWF such that a real instrument can be closely modeled.

In our experience with the above physical modeling techniques for music synthesis, we found that the analysis of the musical instruments themselves had never been addressed. Being a universal approximator, the Artificial Neural Network has been widely used in many applications such as pattern recognition, time series analysis, system identification and so on. Our question is that "if the responses at various positions of a real musical instrument can be measured, can a neural network be trained such that it can reproduce the sounds of the very instrument under the identical excitations other than the training sets? "

In order to simplify our first attempt, we start from the modeling of a plucked string. A string can be approximately regarded as a one-dimensional instrument. This allows a simpler implementation on a recurrent network and requires less computation to train the network. The training data is obtained by the following method. A 1-D Digital Waveguide Filter technique is used to produce some computer simulated strings with arbitrarily assigned parameters for each virtual strings. 'Plucks' are applied to the virtual strings and the responses of the correspondent digital waveguide filters are used as our training data. We propose a model called *Linear Scattering Recurrent Network* (LSRN) which is closely mapped into the DWF in order to simulate a plucked string and the *Back-Propagation-Through-Time* (BPTT) is used for the training algorithm of our recurrent network [7][8]. By using this model, even the overall response of the virtual string are not available, it can nevertheless adjust the parameters based on the partial response. The resultant recurrent network model can response almost identically to that of the virtual string under various 'plucks'.

In section 2, we relate the solution of the ideal lossy wave equation to the recurrent network. In section 3, we advance the recurrent network for the modeling of the nonuniform musical strings and propose our *Linear Scattering Recurrent Network* (LSRN). The training algorithm by *Back-Propagation-Through-Time* (BPTT) is also discussed. In section 4, the computer simulation is given. Conclusion is given in section 5.

2. The Recurrent Neural Network Model for the Lossy Plucked String

2.1 The Lossy Vibrating String

The wave equation for an ideal vibrating string was fully derived by Morse[5]. "Ideal" means lossless, linear, uniform, volumeless and flexible. Consider the uniform string with linear mass density ε (kg/m) stretched to a ten-

*This project is supported in part by N.S.F., R.O.C., under Grant # NSC 86-2213-E-216-018

sion K (newtons). The well-known wave equation for traveling waves within an ideal vibrating string can be represented by

$$Ky'' = \epsilon\ddot{y} \quad (1)$$

where $\dot{y} \equiv \partial y/\partial t$ and $y' \equiv \partial y/\partial x$. The solution of the above wave equation was first published by d'Alembert in 1747. The general solution of Eq. (1) can be written as:

$$y(x,t) = y_r(t-x/c) + y_l(t+x/c) \quad (2)$$

where that the right-going traveling wave is denoted by $y_r(t-x/c)$, with a velocity c and $y_l(t+x/c)$ represents the left-going traveling wave with the identical velocity. The transverse wave velocity c is equal to $\sqrt{K/\epsilon}$.

Practically, there is no lossless vibrating string in the world. In order to simulate the realistic dynamics of the vibrating string, it is necessary to consider the loss in the string. In any real vibrating string, there is energy loss due to the friction by surrounding air, yielding terminations, and internal friction. Many kinds of loss present an approximated resistive force proportional to transverse velocity of the string. If the constant for resistive force is denoted by u, the resistive force $u\dot{y}$ is involved in Eq. (1). Therefore, we can obtain the modified wave equation as follows.

$$Ky'' = u\dot{y} + \epsilon\ddot{y}. \quad (3)$$

The solution to the above equation can be easily derived to be

$$y(t,x) = e^{-(u/2\epsilon)(x/c)}y_r(t-x/c) + e^{(u/2\epsilon)(x/c)}y_l(t+x/c) \quad (4)$$

According to Shannon's Sampling Theorem[6], the traveling wave can be fully expressed by a discrete-time system as long as the sampling interval is small enough. For the DWF, the sampling is performed along the longitudinal direction on the string, instead of time. The magnitude of the vibration at a sampled position is then sampled with period equal to T (sec). To be specific, let the sampling interval corresponding to the string with velocity c be Δx, we have the sampling period with respect to time as $T = \Delta x/c$. By replacing the variable x in Eq. (4) with x_m and t with t_n, we have

$$y(t_n, x_m) = e^{-(u/2\epsilon)(x_m/c)}y_r(t_n - x_m/c) + e^{(u/2\epsilon)(x_m/c)}y_l(t_n + x_m/c)$$
$$= e^{-(u/2\epsilon)mT}y_r[(n-m)T] + e^{(u/2\epsilon)mT}y_l[(n+m)T] \quad (5)$$

where $t_n = n\cdot T$, $x_m = m\cdot\Delta x = m\cdot c\cdot T$.

The left-going and right-going traveling waves decay exponentially in their respective traveling directions. The discrete time signal representation of Eq. (4) is given by

$$y(t_n, x_m) = g^m f_r(n-m) + g^{-m} f_l(n+m) \quad (6)$$

where $g \equiv e^{-uT/2\epsilon}$, $f_r(n) \equiv y_r(nT)$, and $f_l(n) \equiv y_l(nT)$.

Usually, a string is fixed at the two ends where the displacements of the string are zero. Let the length of the string is L, by carrying the constraint into the general solution of Eq. (2), and assuming that the string is fixed at $x = 0$ and $x = L$, we have

$$y(0,t) = 0 = y_r(ct-0) + y_l(ct+0) \quad (7)$$

and

$$y(L,t) = 0 = y_r(ct-L) + y_l(ct+L). \quad (8)$$

From Eq. (7) and Eq. (8), the relationship between the right-going and the left-going traveling waves at the end positions becomes

$$y_r(ct) = -y_l(ct), \text{ and } y_l(ct+L) = -y_r(ct-L). \quad (9)$$

Carrying Eq. (9) into the discrete-time signal representation and let $L = M\cdot c\cdot T$, we have

$$f_r(n) = -f_l(n) \text{ and } f_l(n-M) = -f_r(n+M). \quad (10)$$

2.2 The Recurrent Neural Network Model for the Lossy Plucked String

Although the DWF simulates the wave propagation on a plucked string, the applications are limited to the ideal strings. It is very difficult to find the correct wave equation for a real string so that a correspondent DWF can be built. In order to have a closer imitation of a real plucked string, we need a model which has the ability to simulate the wave propagation and provides a general methodology to find the correspondent model parameters for any particular string.

In the Digital Waveguide Model of the plucked-string, the transverse displacement of the string at any position is the summation of the right-going and left-going traveling waves. A recurrent neural network for the solution of the wave equation with fixed ends can be expressed in Figure 1. Let the recurrent network have N *displacement nodes* denoted by y_i, which represents the displacement of the i-th sampling position for $i = 1, 2, \ldots, N$, and the traveling wave from the j-th node to the i-th node be $f_{i,j}$ through the weight $w_{i,j}$, and each represents the corresponding loss factor according to Eq. (6). Each branch with a loss factor also contains a unit-sample delay. If i is larger than j, $f_{i,j}$ represents the right-going wave, which is the upper track in Figure 1. On the contrary, if i is less than j, $f_{i,j}$ represents the left-going wave, which is the lower track in Figure 1. For any displacement node i except the two end points, there are two traveling waves, flowing into this node from the adjacent nodes, the $(i-1)$-th node and the $(i+1)$-th node. The displacement of the i-th node at any time instant is the sum of these two traveling waves multiplied by the corresponding loss factors one-sampling interval before this time instant.

The initial displacement of the string is normalized such that the largest magnitude within the string are bounded by unity. In practice, the magnitude at any positive throughout the period of vibration cannot exceed the largest magnitude in the initial condition since the string is assumed to be lossy. Then, the displacements of various positions within the plucked string at time-$(t+1)$ can be simply represented by

$$\begin{cases} y_i(t+1) = \begin{cases} a\left[net_i^y(t)\right] & , i = 2, \ldots, N-1 \\ 0 & , i = 0 \text{ or } i = N \end{cases} \\ net_i^y(t) = w_{i,i-1}\cdot f_{i,i-1}(t) + w_{i,i+1}\cdot f_{i,i+1}(t) \end{cases} \quad (11)$$

For the displacement nodes, the right-going and the left-going traveling waves at the identical time instant are shown in the followings, respectively.

$$\begin{cases} f_{i+1,i}(t+1) = a\left(net^f_{i+1,i}(t+1)\right) \\ net^f_{i+1,i}(t+1) = y_i(t+1) - w_{i,i+1} \cdot f_{i,i+1}(t) \end{cases} \quad (12)$$

for $i = 1, \ldots, N\text{-}1$, and

$$\begin{cases} f_{i-1,i}(t+1) = a\left(net^f_{i-1,i}(t+1)\right) \\ net^f_{i-1,i}(t+1) = y_i(t+1) - w_{i,i-1} \cdot f_{i,i-1}(t) \end{cases} \quad (13)$$

for $i = 2, \ldots, N$.

For those nodes which represent the right-going and left-going traveling waves whose magnitudes are not measurable externally, we called them the *departure nodes* shown in Figure 1. We choose the bipolar ramp function to be the activation function for Eq. (11), (12) and (13) as follows:

$$a(x) = \begin{cases} 1 & , 1 < x \\ x & , -1 \le x \le 1 \\ -1 & , x < -1 \end{cases} \quad (14)$$

3. Linear Scattering Recurrent Network Model for the 1-D Transverse Wave of A String

3.1 LSRN Model

In practical situation, the wave traveling in a real string cannot be modeled by the method shown in above since most strings do not satisfy the uniform-impedance constraint. Therefore, a traveling wave may reflect partially at a position and also allow part of the wave to pass through it if the acoustic impedances from the two sides of the position are not identical. The concept of the scattering junction which deals with the nonuniform condition of strings has been used in the application of the acoustic tube.

Assume that there is a scattering junction in a vibrating string with two potions of different characteristic impedances on the sides of this junction. The displacement of the junction can be represented as [12]

$$y^J(t,x) = (1-\rho)y^1_r(t,x) + (1+\rho)y^2_l(t,x) \quad (15)$$

where

$$\rho = \frac{Z_2 - Z_1}{Z_1 + Z_2} \quad (16)$$

Z_1 and Z_2 are the acoustic impedances belonging to the two sides of the junction and ρ is the reflection coefficient. Similar to the procedure in Eq. (5), the discrete-time representations of the traveling waves within the vibrating string are

$$\begin{cases} f^1_l(n+m) = -\rho f^1_r(n-m) + (1+\rho)f^2_l(n+m) \\ f^2_r(n-m) = (1-\rho)f^1_r(n-m) + \rho f^2_l(n+m) \end{cases} \quad (17)$$

Figure 2 shows the structure of a node for the scattering junction based on the unit architecture for the displacement node shown in Figure 1. According to the proposed model, the junction displacement can be calculated as

$$y^J = (1-\rho)f^1_r + (1+\rho)f^2_l \quad (18)$$

which is discrete-time representation of Eq. (15).

The right-going and the left-going traveling waves departing from the junction are

$$\begin{cases} f^1_l = y^J - f^1_r = -\rho f^1_r + (1+\rho)f^2_l \\ f^2_r = y^J - f^2_l = (1-\rho)f^1_r + \rho f^2_l \end{cases} \quad (19)$$

We can find that Eq. (19) is identical to Eq. (17) together with Eq. (15). The new model means that the traveling wave departing from the junction can be computed by subtracting the traveling wave belonging to the same segment flowing into the junction from the displacement magnitude of the junction. According to the scattering junction model and the recurrent network shown in Figure 1, *Linear Scattering Recurrent Network* is proposed. This is shown in Figure 3.

The *departure nodes* and *displacement nodes*, similar to those used in our previous proposed model, as well as the additional nodes, called the *arrival nodes*, are employed to build the new model. Each arrival node in the upper track or the low track represents the traveling wave flowing into the junction and each departure node represents the traveling wave departing from the junction. In order to make the model suitable for broad and complex applications, we allow the characteristic impedance of each segment which is represented by a unit delay to be different from its adjacent segments such that all of the displacement nodes except for the two ends can be regarded as the scattering junction. It is easy to show that if the characteristic impedance of each segment is uniform, the reflection coefficients of the LSRN are zero such that the new model is identical to the previous model in Figure 1.

According to Figure 3, the arrival nodes for the upper and lower tracks of the model can be computed by

$$\varphi_{i,i-1}(t+1) = a\left[net^{\varphi}_{i,i-1}(t)\right] = a\left[w_{i,i-1} \cdot f_{i,i-1}(t)\right] \quad (20)$$

and

$$\varphi_{i,i+1}(t+1) = a\left[net^{\varphi}_{i,i+1}(t)\right] = a\left[w_{i,i+1} \cdot f_{i,i+1}(t)\right] \quad (21)$$

The magnitude of the displacement nodes of the plucked string at time-$(t+1)$ can be represent by

$$y_i(t+1) = \begin{cases} a\left[net^y_i(t+1)\right] & , i = 2, \ldots, N-1 \\ 0 & , i = 1 \text{ or } i = N \end{cases} \quad (22)$$

where

$$net^y_i(t+1) = (1-\rho_i)\varphi_{i,i-1}(t+1) + (1+\rho_i)\varphi_{i,i+1}(t+1) \quad (23)$$

The right-going and left-going departure waves at a certain time instant are shown in the following, respectively.

$$f_{i+1,i}(t+1) = a\left(net^f_{i+1,i}(t+1)\right), \text{ for } i = 1, \ldots, N\text{-}1 \quad (24)$$

where

$$net^f_{i+1,i}(t+1) = y_i(t+1) - \varphi_{i,i+1}(t+1) \quad (25)$$

and

$$f_{i-1,i}(t+1) = a\left(net^f_{i-1,i}(t+1)\right), 2, \ldots, N \quad (26)$$

where

$$net^f_{i-1,i}(t+1) = y_i(t+1) - \varphi_{i,i-1}(t+1) \quad (27)$$

3.2 Training of LSRN

We can unfold the temporal operation of the LSRN into the multilayer-feedforward architecture with synchronous update. Figure 4 illustrates the training for the LSRN shown in Figure 3 by the *BackPropagation Through Time* (BPTT) method [7][8].

In Figure 4, we assign for each time instant a neural network layer called a *time layer*. Each time layer consists of three sublayers, *displacement layer*, *departure layer* and *arrival* layer. The displacement layers, the departure layers and the arrival layers contain the displacement nodes, the departure nodes and the arrival nodes, respectively. Because only the displacements of the displacement nodes are measurable, the training vector is actually the sampled magnitudes of the string under excitations at various preset positions. It is unlikely to measure the displacement for each physical position on the string in order to supply the training data of the correspondent displacement node. There are only a number of positions that are measured. As a matter of fact, we measure only 9 positions in our simulations. For those positions with measured data, we call them the *visible nodes*. For those positions without measured data, we call them the *invisible nodes*. It is noted that the visible nodes and invisible nodes are also the displacement nodes expressed by Eq. (22). Assuming that $d_i(t)$ denotes the desired displacement of the i-th sampling position at time t and $A(t)$ denotes the set of visible nodes, the training task employs the sampled data to train this network such that the generated outputs of the visible nodes are as close as possible to the sampled data. The remaining nodes of this network, the departure nodes, arrival nodes and the invisible nodes, are called *hidden nodes*. The error signals at any time instant, t, are defined as

$$e_i(t) = \begin{cases} d_i(t) - y_i(t) & \text{, if } i \in A(t) \\ 0 & \text{,otherwise} \end{cases} \quad (28)$$

where

$e_i(t)$: the error signal of the i-th displacement node at time t.

$d_i(t)$: the desired response of the i-th displacement node at time t.

$y_i(t)$: the actual output of the i-th displacement node at time t.

The error function at time t is defined as

$$E(t) = 1/2 \sum_{i \in A(t)} e_i^2(t) \quad (29)$$

and we have the total cost function

$$E^{total}(t_0, t_1) = \sum_{t=t_0+1}^{t_1} E(t) \quad (30)$$

to be minimized over one epoch $[t_0, t_1]$, where t_1 is the last time step and t_0 is the initial time step. In order to adjust the weights of the LSRN to approach the desired loss factors and the desired reflection coefficients, the weights correspondent to the loss factors should change along the negative gradient of the cost function as follows,

$$\Delta w_{i+1,i} = -\eta \frac{\partial E^{total}(t_0, t_1)}{\partial w_{i+1,i}} = \sum_{t=t_0+1}^{t_1} \Delta w_{i+1,i}(t)$$

$$= \eta \sum_{t=t_0+1}^{t_1} \delta^{\varphi}_{i+1,i}(t) \cdot f_{i+1,i}(t-1) \quad (31)$$

for $i = 1, \ldots, N$-1.

$$\Delta w_{i-1,i} = -\eta \frac{\partial E^{total}(t_0, t_1)}{\partial w_{i-1,i}} = \sum_{t=t_0+1}^{t_1} \Delta w_{i-1,i}(t)$$

$$= \eta \sum_{t=t_0+1}^{t_1} \delta^{\varphi}_{i-1,i}(t) \cdot f_{i-1,i}(t-1) \quad (32)$$

for $i = 2, \ldots, N$.

The weights correspondent to the reflection coefficients should change along the negative gradient of the same cost function as follow.

$$\Delta \rho_i = -\eta \frac{\partial E^{total}(t_0, t_1)}{\partial \rho_i} = \sum_{t=t_0+1}^{t_1} \Delta \rho_i(t)$$

$$= \eta \sum_{t=t_0+1}^{t_1} \delta^{y}_i(t) \cdot (\varphi_{i,i+1}(t) - \varphi_{i,i-1}(t)) \quad (33)$$

for $i = 2, \ldots, N$-1. Where η is the learning constant and

$$\delta^{y}_i(t) = -\frac{\partial E^{total}(t_0, t_1)}{\partial net^{y}_i(t)} \quad (34)$$

$$\begin{cases} \delta^{f}_{i-1,i}(t) = -\dfrac{\partial E^{total}(t_0, t_1)}{\partial net^{f}_{i-1,i}(t)} \\ \delta^{f}_{i+1,i}(t) = -\dfrac{\partial E^{total}(t_0, t_1)}{\partial net^{f}_{i+1,i}(t)} \end{cases} \quad (35)$$

$$\begin{cases} \delta^{\varphi}_{i+1,i}(t) = -\dfrac{\partial E^{total}(t_0, t_1)}{\partial net^{\varphi}_{i+1,i}(t-1)} \\ \delta^{f}_{i+1,i}(t) = -\dfrac{\partial E^{total}(t_0, t_1)}{\partial net^{f}_{i+1,i}(t)} \end{cases} \quad (36)$$

Repeating the *epoch-wise backpropagation* procedure such that the cost function is minimized, we obtain the correspondent model parameters for the LSRN to imitate the simulated linear non-ideal string. The followings are some computer generated plucked-string examples which are made based on the DWFs and we use the simulated data to train the LSRN by BPTT such that the network parameters are as close as possible to those of the DWFs.

4. Computer simulation

A non-ideal plucked-string simulation produced by DWF with 21 sampling positions are used to produce the desired data to train the LSRN model. The two fixed ends reside at the position of $x = 0$ and $x = 20$. Let the DWF be plucked at the position of $x = 16$, from the initial time step, $t = 0$, and the sampling operation stops at the last time step, $t = 61$. Although there are 21 correspondent displacement nodes in LSRN, but only nine of them are visible nodes. The training of LSRN employs the error signals produced by accumulating the absolute difference between the displacement of each visible node and the desired displacement at the associated sampling position in the time interval [0, 61] to adjust the weights. Each delay unit of the DWF is accompanied by an attenuation unit whose loss factor is 0.98 for both right-going and left-going traveling waves. There is a scattering junction at the 10-th sampling position with the reflection coefficient equal to -0.15.

Since the cost function is the accumulation of the errors of those nine sampling position over the period of 61 time steps, the learning constant η is therefore set to be a low 0.0002. The weights for the loss factors of LSRN are initialized by using random-positive numbers between 0 and 1, and the weights for the reflection coefficients are initialized by using random numbers between -1 and 1. The *Total Mean Square Error* (TMSE) converges to 0.026381 after 40000 epochs of learning.

Figure 5 shows the desired and the actual displacements (y-axis) of the sampling positions $x = 4$, 8, 10, 12, 16, and 18 against time steps (x-axis), respectively. We can find that the 'o' dots and the solid curves almost overlap with each other although there are only 9 visible nodes.

After the learning procedure, we pluck the string at a different position, say $x = 8$. It is found that the 'o' dots and the solid curves almost overlap with each other in Figure 6. This result demonstrates that the output of each displacement node in the LSRN can imitate the desired outputs very well for each sampling position at every time step after sufficient learning.

It was found that the system parameters might vary due to different initial condition of the network. However, the result networks with different parameters all response very close to each other as long as the training vector is identical. It seems that the model is not sensitive to the initial condition and is not bothered by the problem of local minima all too common to ANN, either.

5. Conclusion

In this paper, we have demonstrated a technique of modeling a musical string by a new class of neural networks called Linear Scattering Recurrent Networks (LSRN). By measuring the response of a plucked string, we can employ the sampled data to train the proposed network such that it can response similarly to the original acoustic string. This is to say that we can synthesize musical tones which is very close to those of their acoustic instrument counterparts. Based on this approach, it is possible to analyse many musical instruments and produce many more interesting sounds in addition to the plucked-string presented in this context. Our future efforts includes the modeling of other vibrating parts of instruments such as the bridge and top plate of a guitar, their coupling phenomenon and the reduction of the computation of music synthesis by using the proposed techniques.

References

[1] Alvin.Su, K.S.Meng, and S.F.Liang, "Physical Modeling of Acoustic Instrument and its Application on Music Synthesis", 工研院電通所委託 學術 機構研究 Final Report 1995.

[2] Smith,J.O., "Efficient Synthesis of Stringed Musical Instruments", ICMC 1993.

[3] Smith,J.O., "Physical Modeling Using Digital Waveguides", Computer Music Journal, Vol. 16, No.4, p74-87, 1992.

[4] P.R.Cook, "Identification of Control Parameters in an Articulatory Vocal Tract Model with Applications to the Synthesis of Singing", Ph.D. dissertation, Stanford U., 1990.

[5] Morse, P.M, "Vibration and Sound", Published by the American Institude of Physics for the Acoustical Society of America, Woodbury, New York, 1936.

[6] A.J.Terri., "The Shannon Sampling Theorem-Its Various Extensions and Applications: A Tutorial Review", Proceeding of IEEE. Vol.65, p1565-1596, Nov, 1977.

[7] Ronald J. Williams, and Jing Peng, "An Efficient Gradient-Based Algorithm for On-Line Training of Recurrent Network Trajectories", Neural Computation, Vol. 2, p490-501, 1990.

[8] Fernando J. Pineda, "Recurrent Backpropagation and the Dynamical Approach to Adaptive Neural Computation", Neural computation 1, p161-172, 1989.

[9] Max V. Mathews, John R. Pierce, Current Directions in Computer Music Research, MIT Press, 1991.

[10] Fletcher, Neville H., Rossing, Thomas D. The Physics of Musical Instruments, Springer-Verlag, New York,1991.

[11] A.V. Oppenhim., and R.W. Schafer., Discrete-Time Signal Processing, Prentice Hall, New Jersey, 1989.

[12] S.F. Liang,"Dynamics Modeling of Musical String by ANN", Master thesis, Depart. of Control Eng.,

Chiao-Tung University, 1996.

[13] V. Valimaki, J. Huopaniemi, M. Karjalainen, Z. Janosy, "Physical Modeling of Plucked String Instruments with Application to Real-Time Sound Synthesis", J. Audio Eng. Soc., Vol. 44, No. 5, p331-353, 1996.

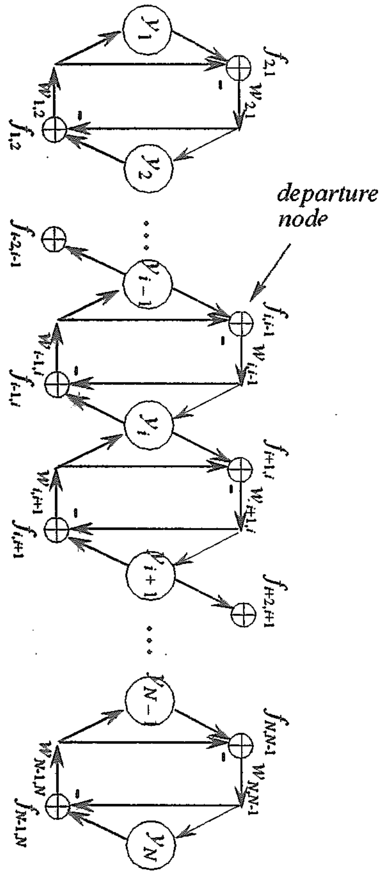

Figure 1 Recurrent network mapping of the plucked string with both fixed ends.

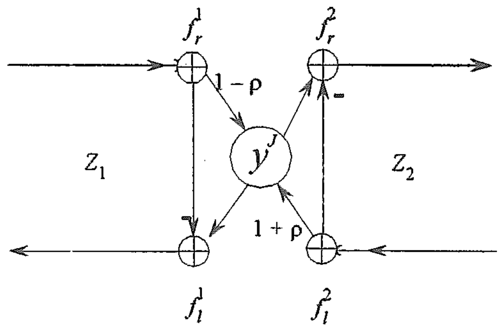

Figure 2 The model for the scattering junction within a string.

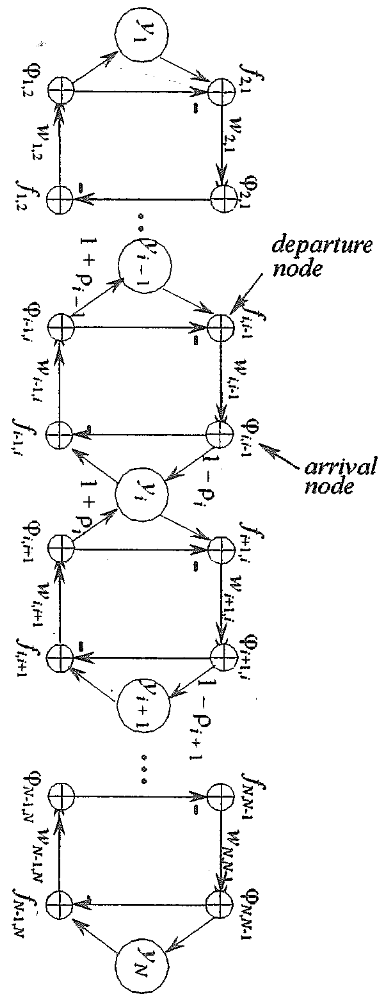

Figure 3 LSRN model of the plucked string with fixed ends.

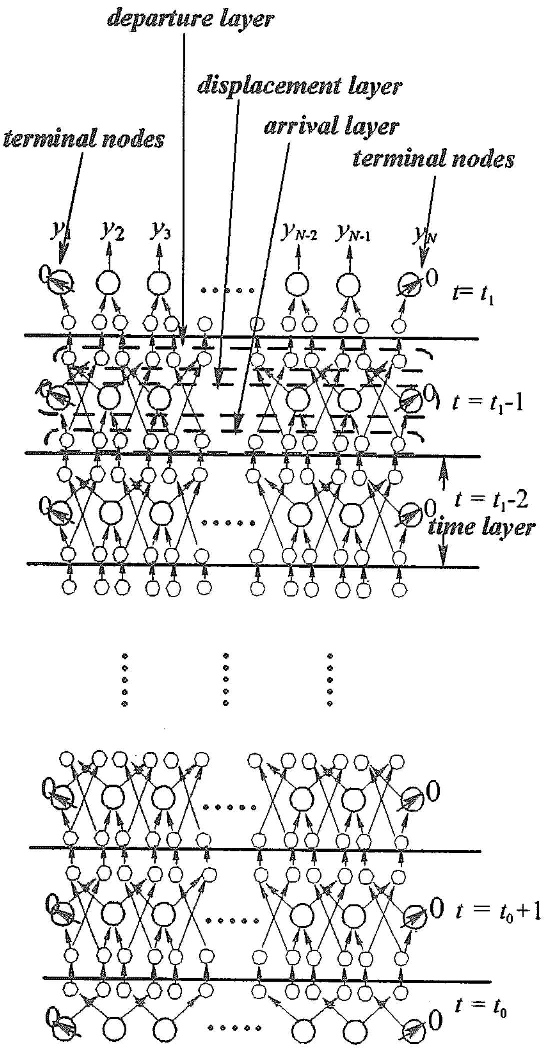

Figure 4 The feedforward architecture behaves as the LSRN of a plucked string shown in Figure 3.

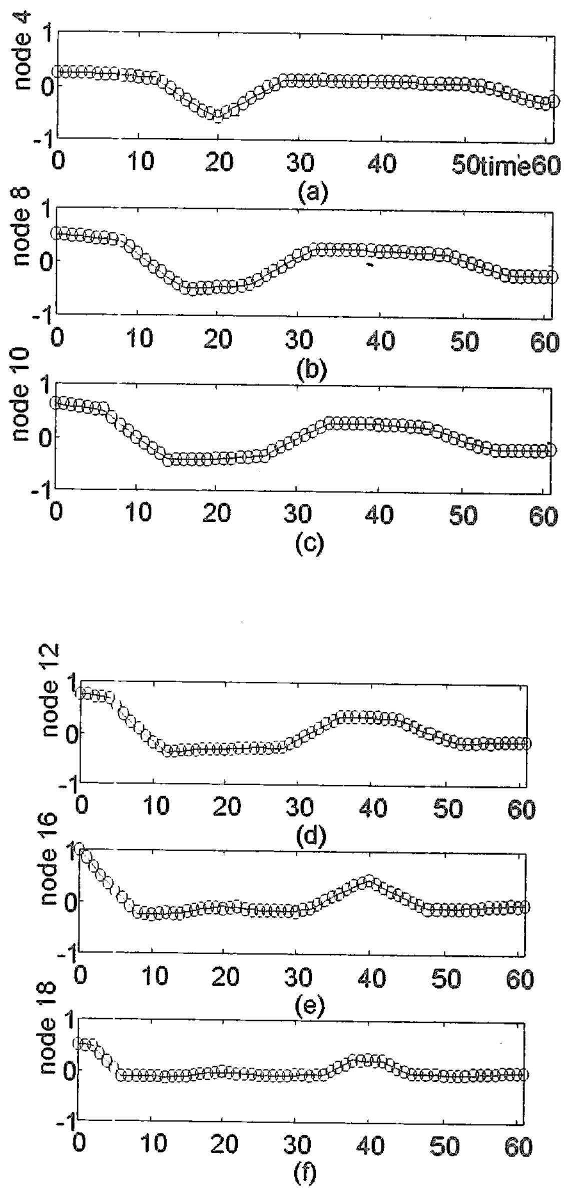

Figure 5 The comparison of desired outputs and the trained outputs of LSRN after 40000-epoch learning. The 'o' dots represent the trained outputs of LSRN and the solid curves represent the desired outputs generated by DWF, at different sampling positions.

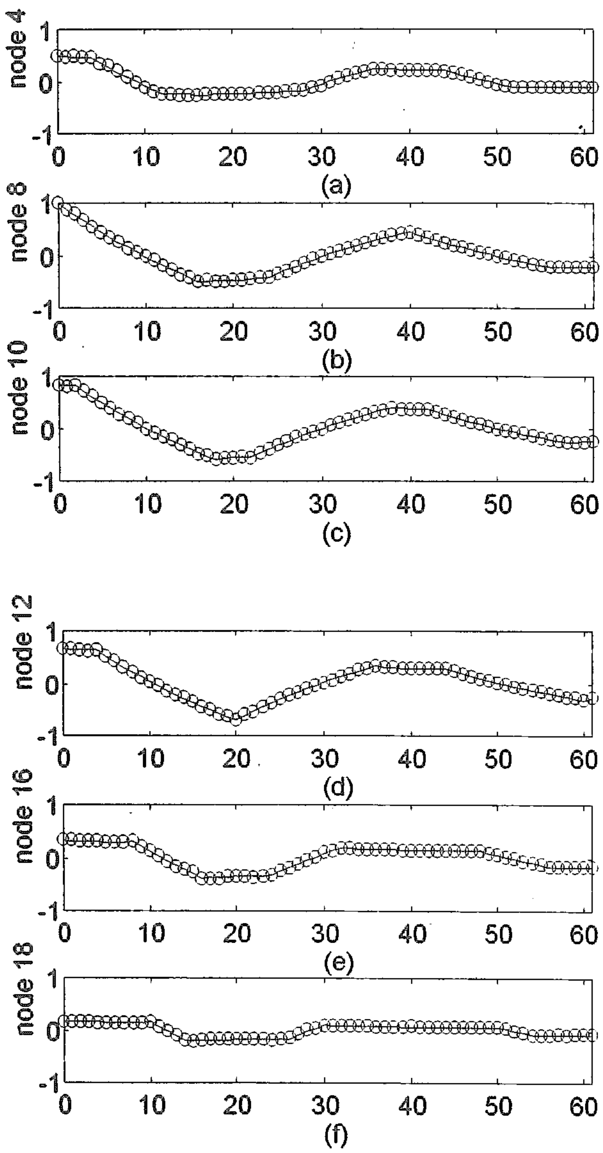

Figure 6 The vibration of simulated plucked string which is plucked at node 8.