WORKING WITH DIFFERENTIAL, FUNCTIONAL AND DIFFERENCE EQUATIONS USING FUNCTIONAL NETWORKS

Enrique Castillo, Angel Cobo,
José Manuel Gutiérrez and Eva Pruneda

Department of Applied Mathematics and Computational Sciences,
University of Cantabria, SPAIN

ABSTRACT

In this paper we first analyze the problem of equivalence of differential, functional and difference equations and give methods to move between them. We also introduce functional networks, a powerful alternative to neural networks, which allow neural functions to be different, multidimensional, multi-argument and constrained by link connections, and use them for predicting values of magnitudes satisfying differential, functional and/or difference equations, and for obtaining the difference and differential equation associated with a set of data. The estimation of the differential or difference equation coefficients is done by simply solving systems of linear equations, in the cases of equally or unequally spaced or missing data points. Some examples of applications are given to illustrate the method.

Key Words : Functional networks, functional equations, difference equations, differential equations, neural networks, applications.

1 Introduction

In this paper we analyze the problem of equivalence of differential, functional and difference equations and give methods to move between these representations. We motivate the paper by giving the following illustrative example.

Consider the system in Figure 1 consisting of a mass m supported by two springs and a viscous damper or dashpot (see Richart, Hall and Woods [15]). The spring constants $k/2$ are defined as the change in force per unit change in length of the spring. The force in the dashpot is directly proportional with a constant m to velocity $z'(t)$ and has a value computed from the viscous coefficient.

The differential equation of motion of the system in Figure 1 may be obtained by making use of the Newton's second law and measuring displacement from the rest position. The equilibrium of vertical forces at position $z(t)$ leads to the differential equation

$$\boxed{mz''(t) + cz'(t) + kz(t) = f(t).} \tag{1}$$

As it will be shown in this paper, in the case of regular damping ($c^2 < 4km$), the differential equation (1) is equivalent (in the sense of having the same solutions) to the functional

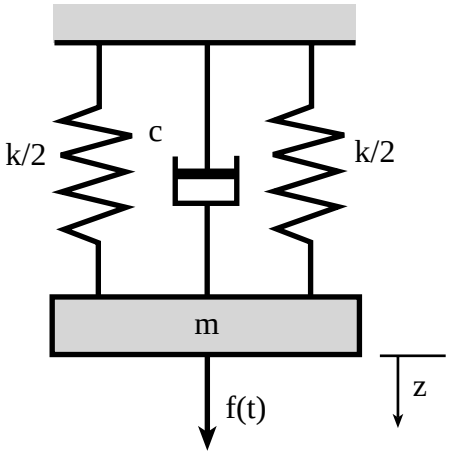

Figure 1: One degree of freedom system with springs and viscous damping.

equation

$$\boxed{z(t+u_2) = \alpha_0(u_1,u_2)z(t) + \alpha_1(u_1,u_2)z(t+u_1) + \delta(t;u_1,u_2),} \tag{2}$$

where

$$\begin{aligned} \alpha_0(u_1,u_2) &= \frac{exp(au_2)(\cos(b\,u_2)\,\mathrm{sen}\,(b\,u_1) - \cos(b\,u_1)\,\mathrm{sen}\,(b\,u_2))}{\mathrm{sen}\,(b\,u_1)}, \\ \alpha_1(u_1,u_2) &= \frac{\mathrm{sen}\,(b\,u_2)exp(a(u_2-u_1))}{\mathrm{sen}\,(b\,u_1)}, \end{aligned} \tag{3}$$

with a and b arbitrary constants, and $\delta(t;u_1,u_2)$ is a function associated with a particular solution.

Similarly, the differential equation (1) is equivalent to the difference equation

$$\boxed{z(t+2u) = \alpha_0(u)z(t) + \alpha_1(u)z(t+u) + \delta(t),} \tag{4}$$

where $\alpha_0(u)$ and $\alpha_1(u)$ are functions of u (constants if u is assumed constant) and function $\delta(t)$ is associated with a particular solution.

The important thing here is that Equations (2) and (4) are exact in the sense that they gives exact values of the solution at any point or the grid points $(t, t+u, t+2u, \ldots t+nu, \ldots)$, respectively.

Equation (2) can be represented by the network in Figure 2, where I is used to refer to the identity function. Similarly, Equation (4) can be represented by the network in Figure 3.

Any reader, who is familiar with artificial neural networks can immediately think of one of such networks to reproduce the two different, but equivalent, approaches to differential Equation (1) of the problem above.

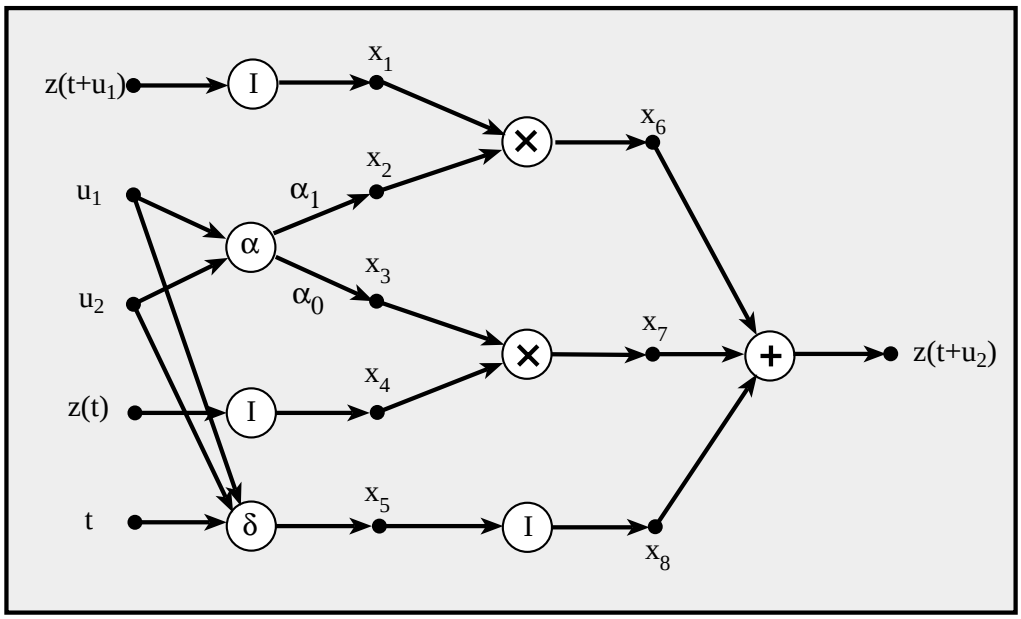

Figure 2: Functional network associated with functional equation (2).

Artificial neural networks have been recognized as a powerful tool to learn and reproduce systems in various fields of applications (see Kohonen (1984), McClelland and Rumelhart (1989), Freeman and Skapura (1991), Hertz, Krogh and Palmer (1991), Freeman (1994), Adeli and Hung (1996), etc.). Artificial neurons parallel the brain behavior and consist of one or several layers of neurons connected by links. Each artificial neuron computes a scalar output from a linear combination of inputs, using a given scalar function, which is assumed the same for all neurons. The differences between two neurons are due to either the number of input components or to their associated weights. Since the neural function is given, only

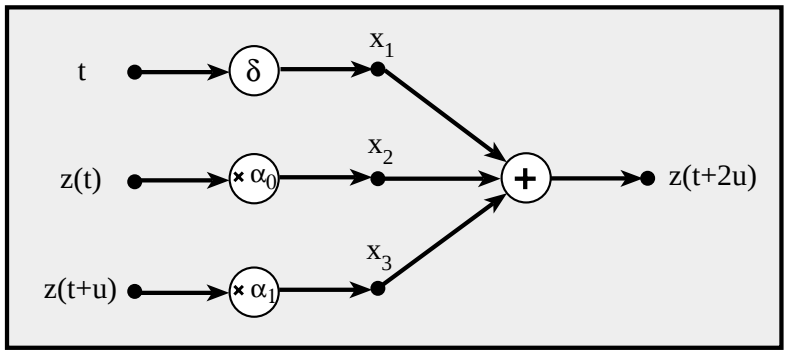

Figure 3: Functional network associated with difference equation (4).

the structure (links) and the weights are learned using well known learning methods, as the backpropagation method.

However, the networks in Figures 2 and 3 do not fit into the artificial neural paradigm.

Gómez-Nesterkín (1997), Castillo (1997a,b), and Castillo, Cobo and Gómez-Nesterkín (1997)), have introduced functional networks as a powerful generalization of artificial neural networks, which provide a new paradigm in which the above networks perfectly fit.

In this paper we use functional networks to approximate solutions of differential, functional and difference equations and to obtain the differential equation associated with a set of data.

In Section 2 we make an introduction to functional networks and show the main differences between neural and functional networks.

In Section 3 we show the equivalence between differential, functional and difference equations. More precisely, in Subsection 3.1 we give methods for obtaining a functional equation which is equivalent to a given differential equation, in Subsection 3.2 we give methods for obtaining a difference equation which is equivalent to a given functional equation, and in Subsection 3.3 we give methods for obtaining a differential equation which is equivalent to a given difference equation, thus, closing the cycle.

In Section 4 we give several illustrative examples of how to use the previous results for approximating differential, functional and difference equations by functional networks. The cases of equally and unequally spaced data are analyzed. Finally, in Section 5 we give some conclusions and recommendations.

2 Functional networks

In this section we make a short introduction to functional networks by: (a) describing its main components, (b) discussing differences between neural and functional networks, and (c) giving the steps to work with functional networks.

2.1 Components of a functional network

The main components of a functional network are:

1. *A layer of input units.* This layer contains the input information. In the network in Figure 2, this input layer consists of the units $z(t+u_1), u_1, u_2, z(t)$ and t.

2. *A set of intermediate layers.* It is an optional layer. In Figure 2 there are two intermediate layers that contain units x_1 to x_5 and x_6 to x_8, respectively.

3. *A layer of output units.* This layer consists of the output units, which contain the output information. In Figure 2 this output layer reduces to the unit $z(t+u_2)$.

4. *One or several layers of neurons.* A neuron is a computing unit which evaluates a set of input values and outputs one or several values. Neurons are represented by circles with their names inside. In Figure 2 the neurons, from top to bottom and left to right, are $I, \alpha, I, \delta, \times, \times, I$ and $+$.

5. *A set of directed links.* They connect input, intermediate and output units with neurons. They are represented by arrow, that indicate the information flow direction in the network (see Figure 2).

2.2 Differences between neural and functional networks

The main differences between functional networks and neural networks are (see Figure 2):

1. In functional networks the neural functions have *several arguments*, while in neural networks they have *only one*. See, for example, neurons denoted α and δ in Figure 2; they have two and three input arguments, respectively.

2. In functional networks the neural functions can be *multivariate*, while in neural networks they are *univariate*. See, for example, the bivariate neuron α in Figure 2.

3. In a given functional network the neural functions can be *different*, while in neural networks they are *identical*. Neurons $\alpha, \delta, I, \times$ and $+$ in Figure 2 are different.

4. In functional networks *there are no weights*, while in neural networks *there are weights*.

5. In functional networks *the neuron outputs can be coincident*, while in neural networks *neuron outputs are different*. For example we could connect the neuron δ with node x_3. Note that this is not possible with standard neural networks.

6. In functional networks *neural functions are learned*, while in neural networks *weights are learned*. For example, in the network in Figure 2 we learn neurons α and δ.

2.3 Working with functional networks

Next, we describe how functional networks can be used, describing step by step.

Step 1 *(Statement of the problem)*: Understanding of the problem to be solved. This is a crucial step.

Step 2 *(Initial topology)*: Based on the knowledge of the problem, the topology of the initial functional network is selected. For example, the vibrating functional (2) and difference (4)

equations of the mass problem has led to the two functional networks in Figures 2 and 3, respectively.

Step 3 *(Simplification)*: In this step, the initial functional network is simplified using functional equations. It must be noted that when there are coincident neural outputs, they must coincide in values, and this leads to a functional equation which allows simplifying the initial topology of the functional network. This is not the case of the functional networks in Figures 2 and 3. Thus, no simplification is possible here. For an illustration of this step, the reader is referred to Castillo, Cobo, Gutiérrez and Pruneda (1998).

Step 4 *(Uniqueness of representation)*: In this step, uniqueness conditions for the neural functions to be unique must be found. For example, in the case of the network in Figure 2, we can wonder about the existance of two sets of functions $\{\alpha_0, \alpha_1, \delta\}$ and $\{\alpha_0^*, \alpha_1^*, \delta^*\}$ such that

$$\begin{array}{rcl} z(t+u_2) & = & \alpha_0(u_1,u_2)z(t) + \alpha_1(u_1,u_2)z(t+u_1) + \delta(t;u_1,u_2) \\ & = & \alpha_0^*(u_1,u_2)z(t) + \alpha_1^*(u_1,u_2)z(t+u_1) + \delta^*(t;u_1,u_2), \end{array} \tag{5}$$

that is, about the existence of two different functional networks with the same structure leading to the same outputs for the same inputs.

Step 5 *(Data collection)*: For the learning to be possible we need some data. Consider, for example, that we have available the data given in Table 2, corresponding to the vibrating mass displacements z corresponding to different times t.

Step 6 *(Approximating neural functions)*: The neural functions are approximated by a parametric family (normally a finite linear combination of basic functions). In Section 4 we give some illustrative examples.

Step 7 *(Learning)*: At this point, the neural functions are estimated (learned), by using some minimization methods. In functional networks, this learning process consists of obtaining the neural functions based on a set of data $D = \{(I_i, O_i)|i = 1, \ldots, n\}$ given in a previous step, where I_i and O_i are the i-th inputs and outputs, respectively, and n is the sample size.

The learning process is based on minimizing the sum of squared errors of the actual and the observed outputs for the given inputs

$$Q = \sum_{i=1}^{n} \left(O_i - F(I_i)\right)^2, \tag{6}$$

where F is the compound function given the outputs, as a function of the inputs, for the given network topology.

t	z	t	z	t	z	t	z	t	z
0.0	0.100	0.04	0.177	0.08	0.241	0.12	0.277	0.16	0.278
0.2	0.238	0.24	0.155	0.28	0.036	0.32	-0.113	0.36	-0.277
0.4	-0.441	0.44	-0.590	0.48	-0.708	0.52	-0.781	0.56	-0.800
0.6	-0.760	0.64	-0.660	0.68	-0.507	0.72	-0.31	0.76	-0.086
0.8	0.151	0.84	0.381	0.88	0.586	0.92	0.752	0.96	0.867
1.0	0.924	1.04	0.921	1.08	0.861	1.12	0.754	1.16	0.612
1.2	0.450	1.24	0.284	1.28	0.130	1.32	0.001	1.36	-0.093
1.4	-0.145	1.44	-0.156	1.48	-0.130	1.52	-0.074	1.56	-0.000
1.6	0.077	1.64	0.144	1.68	0.188	1.72	0.199	1.76	0.170
1.8	0.098	1.84	-0.012	1.88	-0.153	1.92	-0.313	1.96	-0.478
2.0	-0.632	2.04	-0.758	2.08	-0.843	2.12	-0.876	2.16	-0.850
2.2	-0.764	2.24	-0.623	2.28	-0.435	2.32	-0.214	2.36	0.023
2.4	0.258	2.44	0.474	2.48	0.654	2.52	0.786	2.56	0.861
2.6	0.876	2.64	0.833	2.68	0.740	2.72	0.608	2.76	0.452
2.8	0.287	2.84	0.130	2.88	-0.005	2.92	-0.107	2.96	-0.169
3.0	-0.189	3.04	-0.170	3.08	-0.119	3.12	-0.047	3.16	0.032
3.2	0.105	3.24	0.158	3.28	0.180	3.32	0.163	3.36	0.105
3.4	0.006	3.44	-0.126	3.48	-0.282	3.52	-0.446	3.56	-0.603
3.6	-0.737	3.64	-0.834	3.68	-0.880	3.72	-0.870	3.76	-0.799
3.8	-0.671	3.84	-0.494	3.88	-0.280	3.92	-0.045	3.96	0.194
4.0	0.417								

Table 1: Observed displacements z of system in Figure 1 for different equally spaced times t.

In Section 4 we give several illustrative examples of the learning process.

Step 8 *(Model validation)*: The test for quality and/or the cross validation of the model is performed. Checking the obtained error is important to see whether or not the selected family of approximating functions are adequate. A cross validation of the model is also convenient.

Step 9 *(Use of the model)*: If the validation process is satisfactory, the model is ready to be used.

3 Differential, functional and difference equations. The equivalence problem

In this section we show the relations between differential and functional equations. To this end we follow the process described in the following diagram

$$Differential \Rightarrow Functional \Rightarrow Difference \Rightarrow Differential.$$

3.1 From differential equations to functional equations

First we show that given a linear differential equation with constant coefficients we can obtain an equivalent functional equation, in the sense of having the same sets of solutions.

The following theorem shows that if $z(t)$ satisfies a linear differential equation with constant coefficients it also satisfies a functional equation and, more important, gives a way for obtaining a functional equation equivalent to a given differential equation.

Theorem 1 *If $z(t)$ satisfies a linear differential equation of order n with constant coefficients, then it also satisfies the functional equation*

$$z(t+u_n) = \sum_{s=0}^{n-1} \alpha_s(u_1,\ldots,u_n) z(t+u_s) + \delta(t;u_1,\ldots,u_n), \forall t, u_1,\ldots,u_n, \tag{7}$$

where

$$\delta(t;u_1,\ldots,u_n) = h(t+u_n) - \sum_{s=0}^{n-1} \alpha_s(u_1,\ldots,u_n) h(t+u_s) \tag{8}$$

and $h(t)$ is a particular solution.

Proof: Since $z(t)$ satisfies a linear differential equation of order n with constant coefficients, it must be of the form:

$$z(t) = \sum_{i=1}^{m} P_i(t)\exp(w_i t) + h(t), \tag{9}$$

where

$$P_i(t) = \sum_{\ell=0}^{k_i-1} c_{\ell i} t^{\ell} \tag{10}$$

is a polynomial of degree $k_i - 1$ (where k_i is the order of multiplicity of the associated root of its characteristic equation), $w_i; i = 1,\ldots,m$ are the roots (real or imaginary) of its characteristic equation, and $h(t)$ is a particular solution.

Letting

$$z^*(t) = z(t) - h(t), \tag{11}$$

Expressions (9) and (10) lead to

$$z^*(t) = \sum_{i=1}^{m} \exp(w_i t) \sum_{\ell=0}^{k_i-1} c_{\ell i} t^{\ell}, \tag{12}$$

Using Lemma 2, in the appendix, we can write

$$P_i(t+u) = \sum_{s=1}^{k_i} f_{is}(t) g_{is}(u),$$

and then

$$\begin{aligned} z^*(t+u) &= \sum_{i=1}^{m} \exp(w_i(t+u)) \sum_{\ell=0}^{k_i-1} c_{\ell i}(t+u)^{\ell} \\ &= \sum_{i=1}^{m} \sum_{s=1}^{k_i} \left[\exp(w_i t) f_{is}(t)\right] \left[\exp(w_i u) g_{is}(u)\right] \\ &= \sum_{j=1}^{n} f_j^*(t) g_j^*(u), \end{aligned} \tag{13}$$

where $n = \sum_{i=1}^{m} k_i$, and $f_j^*(t)$ and $g_j^*(u)$ are functions of the form $\exp(w_i t) f_{is}(t)$ and $\exp(w_i u) g_{is}(u)$, respectively.

Using (13) for $u = 0, u_1, \ldots, u_n$ we get

$$\begin{aligned} z^*(t) &= \sum_{j=1}^{n} f_j^*(t) g_j^*(0) \\ z^*(t+u_1) &= \sum_{j=1}^{n} f_j^*(t) g_j^*(u_1) \\ \cdots &\quad \cdots \quad \cdots \\ z^*(t+u_n) &= \sum_{j=1}^{n} f_j^*(t) g_j^*(u_n), \end{aligned} \tag{14}$$

that is,

$$\begin{pmatrix} z^*(t) \\ z^*(t+u_1) \\ \cdots \\ z^*(t+u_n) \end{pmatrix} = f_1^*(t) \begin{pmatrix} g_1^*(0) \\ g_1^*(u_1) \\ \cdots \\ g_1^*(u_n) \end{pmatrix} + \ldots + f_n^*(t) \begin{pmatrix} g_n^*(0) \\ g_n^*(u_1) \\ \cdots \\ g_n^*(u_n) \end{pmatrix}, \tag{15}$$

which shows that the left hand side vector is a linear combination of the right hand side vectors, and then

$$D = \begin{vmatrix} z^*(t) & g_1^*(0) & \ldots & g_n^*(0) \\ z^*(t+u_1) & g_1^*(u_1) & \ldots & g_n^*(u_1) \\ \cdots & \cdots & \cdots & \cdots \\ z^*(t+u_n) & g_1^*(u_n) & \ldots & g_n^*(u_n) \end{vmatrix} = 0. \tag{16}$$

Calculating the determinant in (16) by its first column we get

$$D = \sum_{s=0}^{n} \gamma_s(u_1, \ldots, u_n) z^*(t + u_s) = 0, \tag{17}$$

where $u_0 = 0$. Without loss of generality, we can assume that $\gamma_n(u_1, \ldots, u_n) \neq 0$, and then

$$z^*(t + u_n) = -\sum_{s=0}^{n-1} \frac{\gamma_s(u_1, \ldots, u_n)}{\gamma_n(u_1, \ldots, u_n)} z^*(t + u_s) = \sum_{s=0}^{n-1} \alpha_s(u_1, \ldots, u_n) z^*(t + u_s), \tag{18}$$

where $\alpha_s(u_1, \ldots, u_n) = \dfrac{-\gamma_s(u_1, \ldots, u_n)}{\gamma_n(u_1, \ldots, u_n)}$.

Finally, from (11) and (18) the value of $z(t + u_n)$ becomes

$$z(t + u_n) = \sum_{s=0}^{n-1} \alpha_s(u_1, \ldots, u_n) z(t + u_s) + \delta(t; u_1, \ldots, u_n). \tag{19}$$

■

Example 1 (A simple example). *Consider the differential equation*

$$z''(x) + (a + b)z'(x) + abz(x) = 0 \tag{20}$$

with general solution

$$z(x) = c_1 \exp(-ax) + c_2 \exp(-bx). \tag{21}$$

Writing (21) for x, $x + u_1$ and $x + u_2$ we get

$$\begin{array}{rcl} z(x) & = & c_1 \exp(-ax) + c_2 \exp(-bx) \\ z(x + u_1) & = & c_1 \exp(-a(x + u_1)) + c_2 \exp(-b(x + u_1)) \\ z(x + u_2) & = & c_1 \exp(-a(x + u_2)) + c_2 \exp(-b(x + u_2)), \end{array} \tag{22}$$

and eliminating c_1 and c_2 we obtain

$$\boxed{z(x + u_2) \;=\; \alpha_0(u_1, u_2) z(x) + \alpha_1(u_1, u_2) z(x + u_1)} \tag{23}$$

where

$$\begin{array}{rcl} \alpha_0(u_1, u_2) & = & \dfrac{\exp(au_1 + bu_2) - \exp(bu_1 + au_2)}{\exp((a + b)u_2)(\exp(au_1) - \exp(bu_1))}, \\ \alpha_1(u_1, u_2) & = & \dfrac{\exp((a + b)u_1)(\exp(au_2) - \exp(bu_2))}{\exp((a + b)u_2)(\exp(au_1) - \exp(bu_1))}, \end{array} \tag{24}$$

which is the functional equation equivalent to differential equation (20).

Example 2 (The vibrating mass example). *Consider again the vibrating mass given at the introduction.*

The general solution of Equation (1) is

$$z(t) = z_h(t) + z_p(t), \tag{25}$$

where $z_h(t)$ is the general solution of the homogeneous equation and $z_p(t)$ is a particular solution. Suppose (case of regular damping $c^2 < 4km$) that the associated polynomial have two complex roots $a \pm bi$, then

$$z_h(t) = c_1 \exp(at)\cos(bt) + c_2 \exp(at)\, sen\,(bt). \tag{26}$$

Taking u_1 and u_2 arbitrary real numbers, we get

$$\begin{array}{rcl} z_h(t) & = & c_1 \exp(at)\cos(bt) + c_2 \exp(at)\, sen\,(bt), \\ z_h(t+u_1) & = & c_1 \exp(a(t+u_1))\cos(b(t+u_1)) + c_2 \exp(a(t+u_1))\, sen\,(b(t+u_1)), \\ z_h(t+u_2) & = & c_1 \exp(a(t+u_2))\cos(b(t+u_2)) + c_2 \exp(a(t+u_2))\, sen\,(b(t+u_2)). \end{array} \tag{27}$$

Eliminating c_1 and c_2 from (27), we obtain Equation (2) with (3). This proves the statement made at the introduction of the paper.

3.2 From functional equations to difference equations

Secondly, we show that given a functional equation we can obtain an equivalent difference equation, in the sense of having the same solutions at the grid points.

From Theorem 1 we immediately get the following corollary, which shows that if $z(t)$ satisfies a linear differential equation with constant coefficients it also satisfies a difference equation and, more important, it gives a way of obtaining one from the other and vice versa.

Corollary 1 *If $z(t)$ satisfies a linear differential equation of order n with constant coefficients, then it satisfies the difference equation*

$$z(t+nu) = \sum_{s=0}^{n-1} \alpha_s(u) z(t+su) + \delta(t,u), \tag{28}$$

where

$$\delta(t,u) = h(t+nu) - \sum_{s=0}^{n-1} \alpha_s(u) h(t+su). \tag{29}$$

Proof: Letting $u_j = ju; j = 1, \ldots, n$ in (7) and (8), we get (28) and (29). ∎

Since we use the functional equation (7) we show how to go from a functional equation to a difference equation.

Example 3 (The vibrating mass example). *Consider again the vibrating mass example.*

In the case of equally spaced data, making $u_1 = u$ *and* $u_2 = 2u$*, as indicated in the introduction of the paper, we obtain Equation (4):*

$$z(t+2u) = \alpha_0(u)z(t) + \alpha_1(u)z(t+u) + \delta(t), \tag{30}$$

where from (3) we get

$$\begin{array}{rcl} \alpha_0(u) & = & -\exp(2au), \\ \alpha_1(u) & = & 2\cos(bu)\exp(au). \end{array} \tag{31}$$

3.3 From difference to differential equations

Third, we show that given a difference equation we can obtain an equivalent differential equation, thas is, having the same solutions at the grid points. To this aim we need the following lemma.

Lemma 1 *The general solution of the linear difference equation*

$$z(t+nu) = \sum_{s=0}^{n-1} a_s z(t+su) \tag{32}$$

is

$$z(t) = \sum_{i=1}^{m} Q_i(\frac{t}{u}) w_i^{\frac{t}{u}}, \tag{33}$$

where $Q_i(t)$ *and* w_i *are the polynomials and their associated characteristic* m *different roots of the solution*

$$g(t) = \sum_{i=1}^{m} Q_i(t) w_i^t. \tag{34}$$

of the difference equation

$$g(t+n) = \sum_{s=0}^{n-1} a_s g(t+s). \tag{35}$$

Proof: Assume that the solution of (35) is of the form (34). Letting

$$z(t) = g(\frac{t}{u}) \Leftrightarrow g(t) = z(ut) \tag{36}$$

and

$$t^* = \frac{t}{u}, \tag{37}$$

(32) transforms to

$$g(t^* + n) = \sum_{s=0}^{n-1} a_s g(t^* + s), \tag{38}$$

which is of the form (35) and has solution (34).

Thus, considering (36) and (37) we finally get (33).

■

The following Theorem 2 and Algorithm 1 solve our problem.

Theorem 2 *For a differential equation with constant coefficients to have the same solution as a difference equation, the characteristic equation of the differential equation must have as roots the logarithms of the roots of the characteristic equation of the difference equation with the same multiplicities.*

Proof: Assume that we have a linear differential equation in $z(t)$ with constant coefficients, i.e., with general solution

$$z(t) = \sum_{i=1}^{m} P_i(t) \exp(w_i t) + h(t), \tag{39}$$

where

$$P_i(t) = \sum_{\ell=0}^{k_i - 1} c_{\ell i} t^{\ell} \tag{40}$$

is a polynomial of degree $k_i - 1$ (where k_i is the order of multiplicity of the associated root of its characteristic equation), $w_i; i = 1, \ldots, m$ are the roots (real or imaginary) of its characteristic equation, and $h(t)$ is a particular solution.

In order to have an equivalent difference equation both must have the same solution, so we take $Q_i(t) = P_i(ut)$ and $w_i = \exp(w_i u)$, $i = 1, \ldots, m$, since

$$z(t) = \sum_{i=1}^{m} P_i(t) \exp(w_i t) + h(t) = \sum_{i=1}^{m} Q_i(\frac{t}{u}) w_i^{\frac{t}{u}} + h(t). \tag{41}$$

That is, the characteristic equation of the differential equation must have as roots the logarithms of the roots of the difference equation with the same multiplicities. ■

This suggest the following algorithm for obtaining the equivalent differential equation associated with a given difference or functional equation.

Algorithm 1 Obtaining a differential equation equivalent to a linear difference equation with constant coefficients.

- **Input:** A linear difference equation with constant coefficients:

$$z(t+nu) = \sum_{s=0}^{n-1} a_s z(t+su) + h(t). \tag{42}$$

- **Output:** The equivalent differential equation.

1. **Step 1**: Find the roots $w_i : i = 1, \ldots, m$ and their associated multiplicities k_i of the characteristic equation of (42):

$$p^n - \sum_{s=0}^{n-1} a_s p^s = 0. \tag{43}$$

2. **Step 2**: Obtain the characteristic equation of the equivalent linear differential equation, using as roots the logarithms of the above roots divided by u and the same multiplicities:

$$\prod_{i=1}^{m} (q - \frac{1}{u}\log(w_i))^{k_i} = 0. \tag{44}$$

3. **Step 3**: Expand the characteristic equation and calculate the corresponding coefficients $b_s; s = 1, \ldots, n-1$:

$$\prod_{i=1}^{m} (q - \frac{1}{u}\log(w_i))^{k_i} = q^n + \sum_{s=0}^{n-1} b_s q^s = 0. \tag{45}$$

4. **Step 4**: Return the equivalent differential equation:

$$z^{(n)} + \sum_{s=0}^{n-1} b_s z^{(s)} = h(t). \tag{46}$$

Example 4 *If we consider the difference equation*

$$f(x+3) = 3f(x+2) - 3f(x+1) + f(x).$$

Its characteristic equation

$$r^3 - 3r^2 + 3r - 1 = 0,$$

has the root $w_1 = 1$ *with multiplicity* $k_1 = 3$.

According to (44), the characteristic equation of the equivalent differential equation is

$$(q - \log 1)^3 = q^3 = 0.$$

Thus, the equivalent differential equation becomes

$$f'''(x) = 0.$$

Remark 1 *It is well known that there are polynomials in which the roots are quite sensitive to small changes in the coefficients (see Atkinson (1989)). This represents an important problem in the process of finding a differential equation using a set of observed data, because the coefficients of the difference equation (28) are approximated, so the w_i roots in Algorithm 1 can content small errors leading to significant errors in the differential equation. In order to avoid this problem, it is necessary to analyze the stability of the difference equation.*

Suppose that $C(p) = 0$ is the characteristic equation of (42) and w_i, $i = 1, \ldots, m$ are the associated roots with multiplicities k_i, respectively. We define a perturbation $C(p) + \epsilon D(p)$, where $D(p)$ is a polynomial with $degree(D) \leq degree(C)$, then to estimate the modified roots we use

$$w_i(\epsilon) \approx w_i + \gamma_i \epsilon^{1/k_i}, \tag{47}$$

where

$$\gamma_i^{k_i} = \frac{-k_i! C(w_i)}{D^{(k_i)}(w_i)}. \tag{48}$$

Example 5 *Consider again the difference equation in Example 4. A small perturbation of value 0.01 in the $f(x)$ coefficient, leads to a stability coefficient $\gamma_1 = 0.107722 + 0.18658i$ and to the equation*

$$0.00996273 f(x) + 0.0149362 f'(x) - 0.00995033 f''(x) + f'''(x) = 0.$$

The new characteristic equation, instead of a real multiple solution of multiplicity three, has two complex roots. In other words, the new functional form (with sines and cosines) of the solution has nothing to do with the old solution (with polynomial functions).

4 Approximations using functional networks

In this section we use functional networks to approximate functional and difference equations. We consider two cases: (a) equally space data, and (b) unequally spaced data.

4.1 Equally spaced data

We start by analyzing the case with equally spaced data, as those in Table 1.

In the case of equally spaced data (constant u), we use Equation (4), where $\alpha_0(u)$ and $\alpha_1(u)$ for constant u are constants, and function $\delta(t)$ can be approximated by a linear combination of a set of linearly independent functions $\{\phi_i(t) | i = 1, \ldots, m\}$.

If $z(t_j)$ for $j = 0, \ldots, n$ are the observed data for equally spaced times t_j, according to Corollary 1, the solution of a differential equation of order k with constant coefficients in

$z(t)$ can be approximated using the model

$$z_{j+k} = \sum_{i=1}^{k} c_i z_{j+i-1} + \sum_{i=k+1}^{k+m} c_i \phi_{i-k}^{j}; j = 0, \ldots, n-k. \tag{49}$$

where $u = t_{j+1} - t_j$ for $j = 0, \ldots, n-1$, $z_j = z(t_0 + ju)$, $\phi_i^j = \phi_i(t_0 + ju)$ and $c_1, \ldots, c_{k+m}$ are constant coefficients.

The functional network associated with (49) is given in Figure 4.

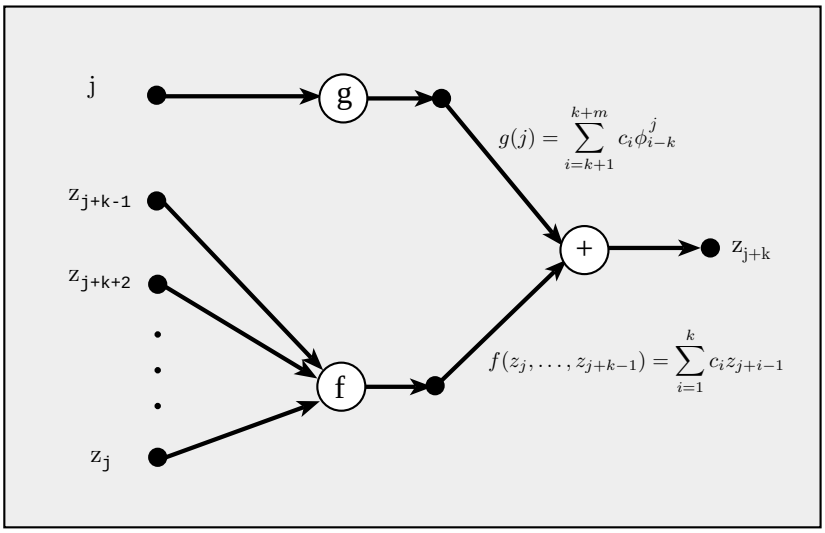

Figure 4: Functional network associated with Equation (49).

The error e_{j+k} at the point $t_{j+k} = t_0 + (j+k)u$ using this approximation becomes

$$e_{j+k} = z_{j+k} - \sum_{i=1}^{k} c_i z_{j+i-1} - \sum_{i=k+1}^{k+m} c_i \phi_{i-k}^{j}; j = 0, \ldots, n-k. \tag{50}$$

Thus, the parameters $c_1, \ldots, c_{k+m}$ can be estimated by minimizing

$$Q = \sum_{j=0}^{n-k} e_{j+k}^2 = \sum_{j=0}^{n-k} \left(z_{j+k} - \sum_{i=1}^{k} c_i z_{j+i-1} - \sum_{i=k+1}^{k+m} c_i \phi_{i-k}^{j} \right)^2. \tag{51}$$

The minimum is obtained for

$$\begin{aligned} -\frac{1}{2}\frac{\partial Q}{\partial c_r} &= \sum_{j=0}^{n-k} \left(z_{j+k} - \sum_{i=1}^{k} c_i z_{j+i-1} - \sum_{i=k+1}^{k+m} c_i \phi_{i-k}^{j} \right) z_{j+r-1} = 0; r = 1, \ldots, k \\ -\frac{1}{2}\frac{\partial Q}{\partial c_r} &= \sum_{j=0}^{n-k} \left(z_{j+k} - \sum_{i=1}^{k} c_i z_{j+i-1} - \sum_{i=k+1}^{k+m} c_i \phi_{i-k}^{j} \right) \phi_{r-k}^{j} = 0; r = k+1, \ldots, k+m \end{aligned} \tag{52}$$

This leads to the system of linear equations with $k+m$ unknowns

$$\mathbf{Ac} = \mathbf{b} \Leftrightarrow \begin{pmatrix} \mathbf{A}_{11} & | & \mathbf{A}_{12} \\ -- & + & -- \\ \mathbf{A}_{21} & | & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{c}_1 \\ -- \\ \mathbf{c}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ -- \\ \mathbf{b}_2 \end{pmatrix} \tag{53}$$

From (52) we can write the expressions for each element a_{rs} of $\mathbf{A}$ and b_r of $\mathbf{b}$:

$$\begin{aligned}
a_{rs} &= \sum_{j=0}^{n-k} z_{j+s-1} z_{j+r-1} \quad if \quad r = 1, \ldots, k; s = 1, \ldots, k \\
a_{rs} &= \sum_{j=0}^{n-k} \phi_{s-k}^{j} z_{j+r-1} \quad if \quad r = 1, \ldots, k; s = k+1, \ldots, k+m \\
a_{rs} &= \sum_{j=0}^{n-k} z_{j+s-1} \phi_{r-k}^{j} \quad if \quad r = k+1, \ldots, k+m; s = 1, \ldots, k \\
a_{rs} &= \sum_{j=0}^{n-k} \phi_{s-k}^{j} \phi_{r-k}^{j} \quad if \quad r = k+1, \ldots, k+m; s = k+1, \ldots, k+m \\
b_r &= \sum_{j=0}^{n-k} z_{j+k} z_{j+r-1} \quad if \quad r = 1, \ldots, k \\
b_r &= \sum_{j=0}^{n-k} z_{j+k} \phi_{r-k}^{j} \quad if \quad r = k+1, \ldots, k+m
\end{aligned} \tag{54}$$

Finally, from (53) we get

$$\mathbf{c} = \mathbf{A}^{-1}\mathbf{b}, \tag{55}$$

which gives the solution.

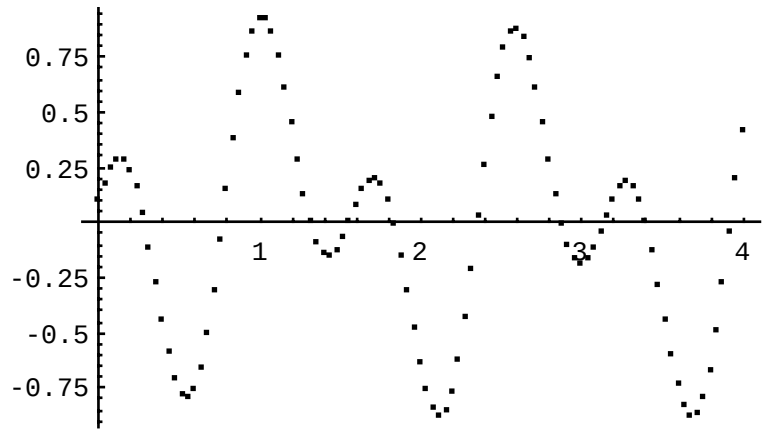

Figure 5: Observed data z for the displacement of system in Figure 1.

Returning to the system in Figure 1, if we use the functions

$$\{\phi_1(t), \phi_2(t), \phi_3(t), \phi_4(t), \phi_5(t)\} = \{1, \text{ sen }(t), \cos(t), \text{ sen }(2t), \cos(2t)\}.$$

t	z	t	z	t	z	t	z	t	z
0.000	0.100	0.001	0.102	0.009	0.119	0.028	0.155	0.040	0.177
0.258	0.105	0.308	-0.067	0.321	-0.117	0.327	-0.140	0.331	-0.155
0.413	-0.491	0.494	-0.739	0.534	-0.794	0.561	-0.800	0.588	-0.777
0.639	-0.664	0.640	-0.661	0.660	-0.588	0.666	-0.565	0.695	-0.436
0.741	-0.195	0.781	0.038	0.781	0.039	0.784	0.056	0.813	0.230
0.831	0.333	1.050	0.915	1.100	0.826	1.110	0.786	1.200	0.459
1.220	0.377	1.370	-0.109	1.450	-0.154	1.580	0.036	1.580	0.048
1.590	0.054	1.610	0.094	1.610	0.097	1.620	0.110	1.660	0.166
1.780	0.135	1.790	0.125	1.820	0.057	1.820	0.052	1.890	-0.188
1.950	-0.437	1.960	-0.487	1.970	-0.520	1.980	-0.542	2.040	-0.747
2.120	-0.876	2.140	-0.871	2.220	-0.707	2.230	-0.646	2.330	-0.178
2.340	-0.099	2.410	0.315	2.420	0.385	2.430	0.434	2.450	0.517
2.490	0.682	2.600	0.877	2.610	0.867	2.670	0.767	2.710	0.640
2.740	0.545	2.760	0.432	2.820	0.223	2.830	0.164	2.840	0.115
3.000	-0.189	3.001	-0.189	3.040	-0.173	3.090	-0.109	3.110	-0.066
3.110	-0.065	3.200	0.103	3.220	0.136	3.260	0.173	3.310	0.172
3.310	0.169	3.340	0.133	3.420	-0.047	3.480	-0.300	3.500	-0.345
3.520	-0.437	3.520	-0.453	3.630	-0.817	3.690	-0.884	3.700	-0.883
3.750	-0.823	3.750	-0.821	3.770	-0.769	3.770	-0.758	3.780	-0.750
3.850	-0.454	3.880	-0.294	3.940	0.057	3.950	0.127	3.970	0.238

Table 2: Observed displacements z of system in Figure 1 for different random times t.

and the equally spaced observed displacements for different times shown in Table 1 or Figure 5, we get:

$$\mathbf{A} = \begin{pmatrix} 24.4 & 23.6 & -3.67 & 5.08 & 3.56 & -4.46 & -1.63 \\ 23.6 & 24.4 & -3.58 & 4.8 & 3.52 & -4.12 & -2.1 \\ -3.67 & -3.58 & 99. & 41.7 & -19.1 & 13.8 & 11.9 \\ 5.08 & 4.8 & 41.7 & 43.5 & 6.91 & -6.85 & -20.1 \\ 3.56 & 3.52 & -19.1 & 6.91 & 55.5 & 21.6 & -12.2 \\ -4.46 & -4.12 & 13.8 & -6.85 & 21.6 & 50.4 & 6.18 \\ -1.63 & -2.1 & 11.9 & -20.1 & -12.2 & 6.18 & 48.6 \end{pmatrix}; \ \mathbf{b} = \begin{pmatrix} 21.3 \\ 23.7 \\ -3.34 \\ 4.35 \\ 3.25 \\ -3.52 \\ -2.62 \end{pmatrix}, \quad (56)$$

which leads to

$$\mathbf{c} = \begin{pmatrix} -1.0130 \\ 1.9442 \\ -0.0214 \\ 0.0335 \\ -0.0155 \\ 0.0154 \\ 0.0091 \end{pmatrix}. \quad (57)$$

Finally, using (49) with the values in (57) we can predict displacements which are visually indistinguishable from those in Figure 6. In fact, we get a maximum absolute prediction error of 0.0334 and a medium absolute prediction error of 0.0132.

To test the possibility of overfitting, we have obtained the RMSE (root mean squared error) for the training data and a set of 1000 test data points, obtaining the following results

$$RMSE_{training} = 0.018; \ RMSE_{testing} = 0.042,$$

which shows that the error increase is not very high.

4.2 Unequally spaced data

Table 2 shows 100 observed displacements of the system in Figure 1 for random times. In this section we use the expression (2) to predict the behavior of the system using these observed displacements and two different models approximating the functions α_0, α_1 and δ involved in it. Note that this approach is also valid for the case of missing data.

Model 1:

Suppose that functions α_0, α_1 and δ are approximated by

$$\alpha_0(u_1, u_2) = a_1 + a_2 u_1 + a_3 u_2 \quad (58)$$

$$\alpha_1(u_1, u_2) = b_1 + b_2 u_1 + b_3 u_2 \quad (59)$$
$$\delta(t, u_1, u_2) = c_1 + c_2 \operatorname{sen}(t) + c_3 \cos(t) + c_4 \operatorname{sen}(2t) + c_5 \cos(2t), \quad (60)$$

where a_i, b_i and c_i are parameters to be estimated. To this aim, we define the function

$$F(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \sum_{i=3}^{100} \left(z^*(t_i) - z(t_i)\right)^2, \quad (61)$$

where $z^*(t_i)$ is the predicted displacement for time t_i using expression (2). The minimum of this function is attained at:

$$\begin{array}{lll} a_1 = 1.603; & a_2 = -23.663; & a_3 = 15.514; \\ b_1 = -0.591; & b_2 = 26.331; & b_3 = -18.058; \\ c_1 = -0.007; & c_2 = 0.030; & c_3 = -0.013; \\ c_4 = 0.014; & c_5 = -0.015. & \end{array}$$

Using these parameters, the medium absolute prediction error is $E = 0.064$. Figure 6 shows the observed and predicted displacements of the system in Figure 1.

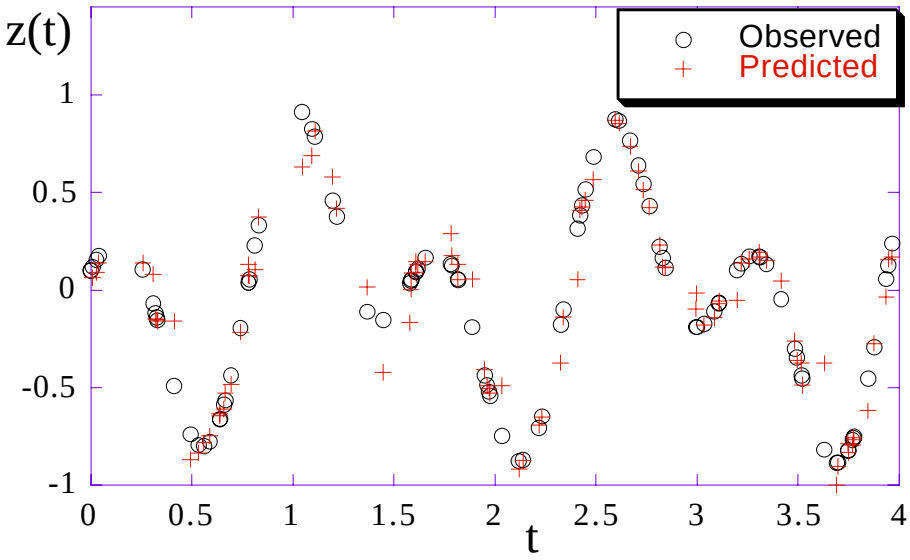

Figure 6: Observed and predicted displacements.

To test the possibility of overfitting, we have obtained the RMSE (root mean squared error) for the training data and a set of 1000 test data points, obtaining the following results

$$RMSE_{training} = 0.11; \; RMSE_{testing} = 0.15,$$

which shows that the error increase is small.

Model 2:

Suppose that functions α_0, α_1 and δ are approximated by

$$\begin{aligned}
\alpha_0(u_1,u_2) &= a_1 + a_2u_1 + a_3u_2 && (62)\\
\alpha_1(u_1,u_2) &= b_1 + b_2u_1 + b_3u_2 && (63)\\
\delta(t,u_1,u_2) &= c_1(u_1,u_2) + c_2(u_1,u_2)\,\mathrm{sen}\,(t) + c_3(u_1,u_2)\cos(t) + && (64)\\
&\quad c_4(u_1,u_2)\,\mathrm{sen}\,(2t) + c_5(u_1,u_2)\cos(2t), && (65)
\end{aligned}$$

where, $c_i(u_1,u_2) = c_{i1} + c_{i2}u_1 + c_{i3}u_2$, for $i = 1,\ldots,5$, and a_i, b_i and c_{ij} are parameters to be estimated. To this purpose, we define the function

$$F(\mathbf{a},\mathbf{b},\mathbf{c}) = \sum_{i=3}^{100} \left(z^*(t_i) - z(t_i)\right)^2, \tag{66}$$

where $z^*(t_i)$ is the predicted displacement for time t_i using expression (2). The minimum of this function is attained for:

$$\begin{array}{llllll}
a_1 = & 1.749; & a_2 = & -27.282; & a_3 = & 16.893;\\
b_1 = & -0.694; & b_2 = & 30.703; & b_3 = & -20.628;\\
c_{11} = & 0.042; & c_{12} = & 1.706; & c_{13} = & -1.999;\\
c_{14} = & -0.077; & c_{15} = & -2.620; & c_{21} = & 3.415;\\
c_{22} = & 0.006; & c_{23} = & 0.302; & c_{24} = & -0.717;\\
c_{25} = & -0.049; & c_{31} = & -0.189; & c_{32} = & 1.196;\\
c_{33} = & 0.014; & c_{34} = & -1.870; & c_{35} = & 0.929.
\end{array}$$

Using these parameters, the medium absolute prediction error is $E = 0.060$. This shows that it is not worthwhile including non-constant $c_i(u_1,u_2); i = 1,2,3,4,5$ functions.

5 Conclusions and Recommendations

The equivalence of differential, functional and difference equations has been analyzed for the case of linear differential equations with constant coefficients and some methods to find them have been given. Appart from critical cases, they allow obtaining good approximations to the exact difference and the differential equations associated with a set of data. If the data points are equally spaced, the problem leads to a linear system of equations. If the data are unequally spaced or there are missing data, the corresponding set becomes non-linear. Functional networks have been shown to be the natural solving tools for this kind of problems. Several examples of applications have shown its power. Though this methodology has been applied to one-dimensional problems, it can be extended to multidimensional problems, but its validity need to be proved.

6 Appendix

In this appendix we give the lemma used in Section 3.

Lemma 2 *Every polynomial of degree n in t and u can be written as a sum with $n+1$ summands with two factors each, one a function of t and one a function of u:*

$$P_n(u,t) = \sum_{i=1}^{n+1} f_i(t) g_i(u). \tag{67}$$

Proof: Given the polynomial of degree n, we group its monomials using the following rules:

Rule 1: If $\alpha \leq \beta$, the monomial in $u^\alpha t^\beta$ is included in the $2\alpha+1$ summand.

Rule 2: If $\alpha > \beta$, the monomial in $u^\alpha t^\beta$ is included in the $2\beta+2$ summand.

Note that the summand number is determined by $\min(\alpha, \beta)$.

Since all monomials in the $2\alpha+1$ summand have the common factor u^α, they can be written as $u^\alpha f_{2\alpha+1}(t)$. Similarly, since all monomials in the $2\beta+2$ summand have the common factor t^β, they can be written as $f_{2\beta+2}(u)t^\beta$.

Since we are dealing with polynomials of degree n, for all monomials whose summand number is assigned by Rule 1 we have $\alpha \leq n/2$, because if $\alpha > n/2$, then $\beta < \alpha$ and then we must use Rule 2 instead of Rule 1. Thus,

$$\alpha \leq n/2 \Rightarrow 2\alpha + 1 \leq n+1,$$

which implies that the summand number obtained by Rule 1 cannot be greater than $n+1$.

Similarly, any monomial whose summand number is assigned by Rule 2, must satisfy $\beta < n/2$, because if $\beta \geq n/2$, then $\alpha \leq \beta$ and then we must use Rule 1 instead of Rule 2. Then, we have

$$\beta < n/2 \Rightarrow 2\beta + 2 < n+2 \Rightarrow 2\beta + 2 \leq n+1,$$

which implies that the summand number obtained by Rule 2 cannot be greater than $n+1$.

Thus, we have

$$P_n(x,t) = \sum_{\alpha=0}^{\lfloor n/2 \rfloor} u^\alpha f_{2\alpha+1}(t) + \sum_{\beta=0}^{\lfloor (n-1)/2 \rfloor} f_{2\beta+2}(u) t^\beta, \tag{68}$$

where $\lfloor x \rfloor$ is the maximum integer less than or equal to x. ∎

References

[1] J. Aczél. *Lectures on functional equations and their applications*. Vol. 19. Mathematics in Science and Engineering. Academic Press 1966.

[2] H. Adeli and S. Hung *Machine learning. Neural networks, genetic algorithms, and fuzzy systems*. John Wiley and Sons, New York, 1996.

[3] K.E. Atkinson *An Introduction to Numerical Analysis*. John Wiley and Sons, New York, 1989.

[4] E. Castillo and R. Ruiz-Cobo. *Functional Equations in Science and Engineering.* Marcel Dekker, 1992.

[5] E. Castillo. Functional Networks. *Neural Processing Letters (in press)*, 1997a.

[6] E. Castillo. *Functional Networks. A Learnable Neuron Approach*, (to appear), 1997b.

[7] E. Castillo, A. Cobo and Ruslán Gómez-Nesterkín. *A General Framework for Functional Network*, (to appear), 1997.

[8] E. Castillo, A. Cobo, J. M. Gutiérrez and E. Pruneda. *Functional networks. A new neural networks based methodology*, (to appear), 1998.

[9] R. Gómez-Nesterkín. Modelación y Predicción mediante Redes Funcionales. Revista Electrónica Foro Red Mat, Facultad de Ciencias, UNAM, ISSN:1405-1745. *http* : $//www.red-mat.unam.mx/foro/vol002/voldos\_6.html$.

[10] J.A. Freeman and D.M. Skapura. *Redes Neuronales: Algoritmos, Aplicaciones y Técnicas de Programación*, Addison Wesley, 1991.

[11] J.A. Freeman. *Simulating Neural Networks with Mathematica*, Addison Wesley, 1994.

[12] J. Hertz, A. Krogh and R.G. Palmer. *Introduction to the Theory of Neural Computation*, Addison Wesley, 1991.

[13] T. Kohonen. *Self-organization and associative memory*, Springer Verlag, 1984.

[14] J.L. McClelland, D. K. Rumelhart. *Explorations in parallel distributed processing*, Ed. M.I.T. Press, 1989.

[15] Richart, F. E., Hall, J. R. and Woods, R. D. *Vibrations of Soils and Foundations*. Prentice Hall International Series in Theoretical and Applied Mechanics, Englewood Cliffs, New Jersey, 1970.