# Wavelet Neural Network based Predictive Control for Mobile Robots

Dongbing Gu and Huosheng Hu

Dept. of Computer Science, University of Essex Wivenhoe Park, Colchester CO4 3SQ, UK Email: dgu@essex.ac.uk, hhu@essex.ac.uk

Abstract: This paper presents a predictive control scheme for mobile robots that possess complexity, non-linearity and uncertainty. A multi-layer back-propagation neural network is employed as a model for nonlinear dynamics of the robot. The control variables are produced by optimizing the performance index on-line using the steepest gradient decent algorithm. The neural network is constructed by the wavelet orthogonal decomposition to form a wavelet neural network that can overcome the problems caused by local minima of optimization. The wavelet network is also helpful to determine the number of the hidden nodes and the initial value of weights. The sparse train data in our path tracking case can reduce the effect of the "curse of dimensionality" on the network size in high dimensional function learning caused by the orthogonal wavelet base function.

Key words: Predictive control, Wavelet neural network, Robot motion control

### I. INTRODUCTION

The difficulties encountered in designing controls for many dynamic systems such as robotic systems can be broadly classified under three headings: 1) complexity, 2) non-linearity, and 3) uncertainty. Generalized Predictive Control (GPC) proposed by D.W. Clarke in 1987 [2] has been an effective tool for this purpose, playing a very important role in the control areas. As we know, most GPC control applications are based on the linear models of dynamic systems to predict the output over a certain horizon, and to evaluate future sequences of control signals in order to minimize a cost index that takes account of the future output prediction errors over a reference trajectory, as well as control efforts. However, when the systems are non-linear and are operated over a large dynamic range, the use of linear models become impractical, and the identification of non-linear models for control becomes absolutely necessary.

Neural networks have become an attractive tool to model the complex non-linear systems due to its inherent ability to approximate arbitrary continuous functions. During the 1980's and the early 1990's, conclusive proofs were given by numerous authors that feed-forward neural networks with one hidden layer are capable of

approximating any continuous function on a compact set in a very precise and satisfactory sense [5]. There were some researchers who have successful applied the feedforward neural networks as the model predictors to GPC [1][8][9][10][12]. In their research work, the control problems of GPC were changed into the non-linear optimization problems with the minimization of the cost index, but suffer from undesirable local minima and slow convergence of the error surface for back-propagation training algorithm and non-linear optimization. Other algorithms such as random search techniques and genetic algorithms can be used to overcome the local minima caused by the gradient ascent rule [4][7]. But the computational cost of these algorithms will degrade the performance of real-time control systems. Moreover, the implementation of multiple feed-forward neural networks suffers from the lack of efficient constructive approaches, both for determining the parameters of neurons and for choosing network structures.

Recently the wavelet decomposition emerges as a new powerful tool for function approximation in a manner that readily reveals properties of the arbitrary L2 function (energy-finite and continuous or discontinuous) [3][6][11]. Combining wavelets and neural networks can result in a wavelet neural network with efficient constructive approach [6][14][15]. Some application results in different areas have shown that the wavelet neural networks can approximate arbitrary functions belonging to L<sup>2</sup> space, specially with the evidence results for energy-finite and discontinuous function compared with the other neural networks [8]. The wavelet neural networks can further result in a convex cost index to which simple iterative solutions such as gradient descent rules are justifiable and are not in danger of being trapped in local minima when choosing the orthogonal wavelets as the activation functions in the nodes [13].

Motion control of mobile robots is a typical nonlinear tracking control issue and have been discussed with different control schemes such as PI, GPC based EKF model [16][17]. In this paper, we will implement a GPC controller for motion control with the neural network

model based on the orthogonal wavelets. In section II, a motion control model is presented for a car-like mobile robot. The wavelet neural network for motion dynamics modeling is shown in section III. Section IV provides Generalized Predictive Control (GPC) scheme based on neural networks. The simulation results are given in section V. Finally section VI presents a brief conclusion.

# II. MOTION DYNAMICS FOR A CAR-LIKE MOBILE ROBOTS

The robot used in this research is a car-like mobile robot with four pneumatic tyre wheels. Two front wheels serve for steering and two rear wheels serve for tracking, as shown in figure 1. The robot has a modular structure such that it can be readily extended and easily configured to allow the use of a range of sensor modules or actuator modules. Two 500 count/revolution rotary encoders are fixed onto two motor shafts for servo control by a PID motion controller. Another two 500 count/revolution rotary encoders have been connected to the rear axes near the rear wheels using 3:1 ratio pulley and belts. The reason we use these two extra encoders is to compensate for the position errors caused by inaccuracy of the steering angle. The robot has a weight of 100 Kg and a maximum speed 1 m/s. The sensors equipped include optical encoders, a sonar array, a laser scanner, an infrared proximity, and a stereo head. As a preliminary step, we currently use optical encoders and a rotating laser scanner to implement the navigation task.

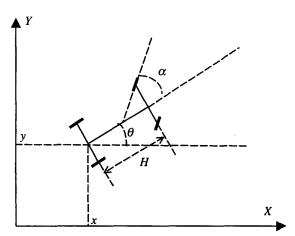


Figure 1 the global frame for motion tracking

The motion control system consists of two main feedback loops. The inner loop is a 2-axis motion controller to maintain stable speed and steering servo control, based on data from encoders. In contrast, the outer loop is a position control loop to guide the robot to follow the planned trajectory accordingly, relied on position

estimated from an Extended Kalman Filter (EKF) [17]. It compensates for any errors caused by disturbance from floor and non-perfected actuators. A path generator is used to produce a continuous curvature trajectory for the robot to travel.

The position (x(k), y(k)) and orientation  $\theta(k)$  (pose) of the robot can be expressed as the state x(k) in a global frame shown in figure 1. We have the equation for motion dynamics as follows:

$$\mathbf{x}(k+1) = \begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix}$$

$$= \begin{bmatrix} x(k) + v(k)T\cos(\theta(k))\cos(\alpha(k)) \\ y(k) + v(k)T\sin(\theta(k))\cos(\alpha(k)) \\ \theta(k) + v(k)T\sin(\alpha(k))/H \end{bmatrix} + \mathbf{w}(k)$$
(1)

where  $\mathbf{u}(k) = [v(k), a(k)]^T$  is the control variables for motion tracking,  $\mathbf{w}(k)$  is Gaussian noise with zero mean value, and T is the sample period. Currently the PI control and GPC control are employed for motion tracking based on EKF estimation for  $\mathbf{x}(k)$  [16][17].

# III. WAVELET NEURAL NETWORK MODELING OF MOTION DYNAMICS

In multi-resolution analysis, for the square integrable function space  $L^2(R)$ , there exists a nested chain of closed subspaces:

$$\{0\}\subset\cdots\subset V_{-1}\subset V_0\subset V_1\subset V_2\subset\cdots\subset L^2(R)$$

such that

$$\bigcap_{m\in \mathbb{Z}} V_m = \{0\}; \bigcup_{m\in \mathbb{Z}} V_m = L^2(R)$$

where  $V_m$  is the subspace spanned by the dilation and translation of a scaling function  $\varphi(t)$ :

$$\varphi_{m,n}(t) = 2^{m/2} \varphi(2^m t - n)$$

An orthogonal complement space  $W_m$  is existed for each of  $V_m$  in  $V_{m+1}$  and they meet:

$$V_{m+1} = V_m \oplus W_m, V_m \bot W_m$$

So we have

$$L^2(R)= \mathop{\oplus}_m W_m$$

where  $W_m$  is the subspace spanned by orthogonal wavelet basis  $\psi_{m,n}(t) = 2^{m/2} \psi(2^m t - n)$ . Two schemes for decomposing a  $L^2(R)$  function f(t) can be presented as:

$$f(t) = \sum_{m,n} \langle f, \psi_{m,n} \rangle \psi_{m,n}(t)$$
 (2)

and

$$f(t) = \sum_{n} \langle f, \varphi_{M,n} \rangle \varphi_{M,n}(t) + \sum_{m > M,n} \langle f, \psi_{m,n} \rangle \psi_{m,n}(t)$$
 (3)

What is more important for the function decomposition is that for sufficiently large M, any  $f(t) \in L^2(R)$  can be approximated arbitrarily closely in  $W_M$  [18]. That is for any  $\varepsilon > 0$ 

$$\left\| f(t) - \sum_{n} \left\langle f, \varphi_{M,n} \right\rangle \varphi_{M,n}(t) \right\| < \varepsilon$$

The approximation by the truncated wavelet decomposition can be expressed as:

$$f(t) \approx \sum_{n} \langle f, \varphi_{M,n} \rangle \varphi_{M,n}(t) = \sum_{n} C_n \varphi_{M,n}(t)$$
 (4)

It means some fine components (high frequency) that belong to wavelet space  $W_m$  for the function are neglected and coarse components (low frequency) that belong to scaling space  $V_m$  are preserved to approximate the originate function under M scale. This expression has similar structure for a 3-layer neural network as shown in figure 2. The number of hidden nodes is decided by wavelet translation n that depends on the support set of f(t). Here we can assume a positive integer N to form a range [-N, N] that can cover the support set of f(t). For the multiple dimension case, the scaling functions or wavelets are generated by the tensor products of one dimensional scaling functions or wavelets.

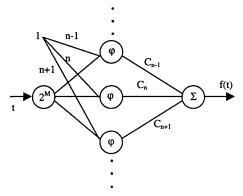


Figure 2 The wavelet network where  $C_n = \langle f, \varphi_{M,n} \rangle$ 

The proposed wavelet network with inputs and outputs for motion dynamics stated in the section 1 is shown in figure 3. This is an one-step-ahead predictive model. The function  $\varphi_d$  in the hidden layer is a multidimensional scaling function.

Other network parameters are:

- the number of input p=5, Input vector:  $\mathbf{z}(k-1) = [v(k-1), \alpha(k-1), x(k-1), y(k-1), \theta(k-1)]^T$ .
- the number of output q=3, Output vector:  $\mathbf{x}(k) = [x(k), y(k), \theta(k)]^T$ .
- the number of the hidden nodes r=2N+1.
- the weight matrix from input to hidden layer is W<sup>1</sup>
- the weight matrix from input to hidden layer is W<sup>2</sup>

The network output vector is

$$\mathbf{x}(k) = W^2 \Phi(W^1 \mathbf{z}(k-1)) \tag{5}$$

where  $\Phi = [\varphi_d, \dots, \varphi_d]^T$ . The weight matrix  $W^l$  is invariable during training cycle.

$$W^{1} = \begin{bmatrix} 2^{M} & \cdots & 2^{M} \\ \vdots & \vdots & \vdots \\ 2^{M} & \cdots & 2^{M} \end{bmatrix}_{(2N+1) \times 5}$$
 (6)

From wavelet decomposition, initial values for weight matrix  $W^2$  are generated by (4):

$$W^{2}(0) = \begin{bmatrix} C_{1,-N} & C_{1,-N+1} & \cdots & C_{1,N} \\ C_{2,-N} & C_{2,-N+1} & \cdots & C_{2,N} \\ C_{3,-N} & C_{3,-N+1} & \cdots & C_{3,N} \end{bmatrix}_{3\times(2N+1)}$$
(7)

The weight matrix can be updated by the steepest gradient decent algorithm.

$$W^{2}(k) = W^{2}(k-1) - \mu \Phi e^{T}$$
 (8)

where  $e = [x_d(k-1) - x(k-1)]$ , and  $\mu$  is the update rate.

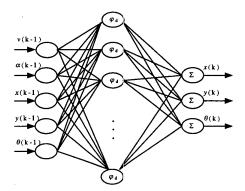


Figure 3 Wavelet network for motion dynamics

### IV. GPC CONTROL SCHEME BASED ON NEURAL NETWORK

The equation (5) of the network output presented in previous section can be rewritten in the state space form as follows:

$$\mathbf{x}(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{z}(k-1)) \\ f_2(\mathbf{z}(k-1)) \\ f_3(\mathbf{z}(k-1)) \end{bmatrix}$$

$$= F[\mathbf{z}(k-1)] = F[\mathbf{x}(k-1), \mathbf{u}(k-1)]$$
(9)

where  $\mathbf{x}(k) = [x(k), y(k), \theta(k)]^T$  and  $\mathbf{u}(k) = [v(k), \alpha(k)]^T$ .

The cost index of the generalized predictive control for motion tracking is

$$\min_{\mathbf{u}(k)} J(\mathbf{x}(k), \mathbf{u}(k))$$

The basic idea of GPC is that the current control variables are chosen to minimize the cost index over several steps in the future so that the path tracking of the robot is smooth and stable. Therefore, the cost index can be expressed as

$$J = \frac{1}{2} \sum_{i=N_1}^{N_2} \left\| \mathbf{x}_d(k+i) - \mathbf{x}(k+i) \right\|^2 + \frac{1}{2} \sum_{i=1}^{N_\mu} \lambda_i \left\| \Delta \mathbf{u}(k+i-d) \right\|^2$$
(10)

where  $\lambda(i)$  is a weighting matrix (2 by 2 positive definite symmetric), penalizing the control effort. This cost index reflects the desire to drive the next state  $\mathbf{x}(k)$  of the robot close to the desired state  $\mathbf{x}_d(k)$  generated by the path planner without excessive expenditure of control effort. The cost index is conditioned on data up to time k, assuming no future measurements are available. At each sampling instant, an optimal control sequence is calculated, but only the first one is applied to the system. This process will be repeated at the next sampling instant to form a receding horizon optimization procedure.

The other notations for equation are

- N<sub>1</sub> and N<sub>2</sub> are the minimum and maximum output horizon respectively.
- $N_{\mu}$  is called the *control horizon*.
- $\Delta \mathbf{u}(k)$  is a set of control increment.

$$\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$$

• d is the dead-time of system.

In general,  $N_2$  is chosen to encompass all the responses that are significantly affected by the current control. In practice, it is more typically set to approximate the rise-

time of the system.  $N_1$  is selected according to the deadtime of the robot system, and is often taken as I. In our tracking case, we choose

$$N_2 = L, N_1 = 1, d = 1, N_u = K$$

In each control cycle k, we need to compute a control increment matrix  $\Delta U(k)(\Delta U(k) = [\Delta u(k), \cdots, \Delta u(k+K-1)]^T)$  that has N vectors and choose the first one as actual control increments for output. For the non-linear system presented in (9), there are no analytic expressions available. We have to implement the non-linear optimal algorithm to find optimal solution of control variables during the interval k and k+I on line. The constrains on the cost index (10) are a convex expression shown in (5) because of the wavelet orthogonal decomposition. Therefor, these K control increments can be calculated recursively from the steepest gradient decent algorithm during the interval between k and k+I without local dilemma.

Let j be the iteration step for the nonlinear optimal programming (j=0, ..., B) and B is the number of the optimizing times, we have  $\Delta \mathbf{U}^{j} = -\mu \delta \mathbf{J}_{\mathbf{U}}^{j}$  according to the gradient decent algorithm, where  $\mu$  is the optimizing rate, and

$$\delta \mathbf{J}_{\mathbf{U}}^{j} = -\delta \mathbf{X}_{\mathbf{U}}^{j} \mathbf{E}^{j} + \Gamma \Delta \mathbf{U}^{j} \tag{11}$$

from (10), where

$$\delta \mathbf{X}_{\mathbf{U}}^{j} = \begin{bmatrix} \frac{\partial \mathbf{x}^{j}(k+1)}{\partial \mathbf{u}^{j}(k)} & \dots & \frac{\partial \mathbf{x}^{j}(k+K)}{\partial \mathbf{u}^{j}(k)} & \dots & \frac{\partial \mathbf{x}^{j}(k+L)}{\partial \mathbf{u}^{j}(k)} \\ 0 & \dots & \frac{\partial \mathbf{x}^{j}(k+K)}{\partial \mathbf{u}^{j}(k+K-1)} & \dots & \frac{\partial \mathbf{x}^{j}(k+L)}{\partial \mathbf{u}^{j}(k+K-1)} \end{bmatrix}$$

$$\mathbf{E}^{j} = \begin{bmatrix} \mathbf{x}_{d}^{j}(k+1) - \mathbf{x}^{j}(k+1) & \cdots & \mathbf{x}_{d}^{j}(k+L) - \mathbf{x}^{j}(k+L) \end{bmatrix}^{T}$$
$$\Gamma = diag[\lambda_{1}, \dots, \lambda_{K}]$$

So, iteration procedure for optimizing control increments during one control cycle is:

$$\Delta \mathbf{U}^{j} = (I + \mu \mathbf{\Gamma})^{-1} \mu \delta \mathbf{X}_{\mathbf{U}}^{j} \mathbf{E}^{j} \tag{12}$$

After M iterations, an optimal control increment  $\Delta U^*(k)$  is produced based on the current predictive states  $\mathbf{x}(k+i)$  (i=0,...,L) at k that can be estimated from  $\mathbf{x}(k-1)$  and  $\mathbf{N}$  optimal control increments  $\Delta U^*(k-1)$  at k-l. The main computation during this optimizing procedure is the

matrix  $\frac{\partial \mathbf{x}(k+i)}{\partial \mathbf{u}(k+l)}$  (i=1, ..., L; l=0, ..., B-1). One of the matrix elements is presented according to the neural network model (5) as:

$$\frac{\partial x(k+i)}{\partial \alpha(k+l)} = \frac{\partial f_1(\mathbf{z}(k+i-1))}{\partial \alpha(k+l)} = \frac{\partial \sum_{m=1}^r w_{1m}^2 \varphi_d}{\partial \alpha(k+l)}$$

$$= \begin{cases} \sum_{m=1}^r w_{1m}^2 \varphi_d' 2^M & (i=l+1) \\ 0 & (i \neq l+1) \end{cases}$$
(13)

### V. SIMULATION RESULTS

In our simulation, Lemarie-Meyer's wavelets are chosen as the function approximation base since they are orthogonal and posses good frequency and time localization. The simple closed form expressions are presented as shown in figure 4 [18]. The scaling function expression is

$$\varphi(t) = \frac{\sin \pi (1 - \beta)t + 4\beta t \cos \pi (1 + \beta)t}{\pi (1 - (4\beta t)^2)}$$
(14)

The dilation and translation of scaling function is centered in  $n/2^M$ . The ranges of network inputs and output are transformed into [-1/2, 1/2]. So, the number of hidden nodes can be determined roughly  $(2^M + 1)^p$ . For different outputs, p is selected from (1). To choose a proper M, a trial and error approach is used [18]:

- To start from a small M.
- To computed the mean square error (MSE).
- To increase M by 1 and repeat from first step if MSE is bigger than a threshold.

The sparse data in our path motion case can further decrease N, e.g. the steer angle is ranged in a smaller region.

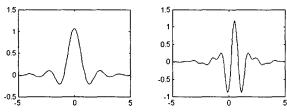


Figure 4 The scaling function and its wavelet.

The training data are generated from the simulation of the dynamics (1) where a path generator is employed for different path generation [22]. We collect 514 pairs of data for training the network that is a (4, 64, 3) neural model with only those output layer weights need to train.

The results of network approximation and dynamic output for a path starting from (19000, 12600, 0) to (23000, 14600,  $\pi$ /2) are shown in figure 5 where (a) is a plot of x, (b) is a plot of x error, (c) is a plot of y, and (d) is a plot of  $\theta$  after 500 epoches training.

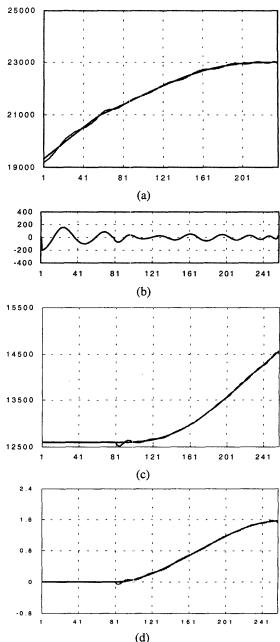


Figure 5 The network approximation and dynamic output, (a) x, (b) x error, (c) y, and (d)  $\theta$ . The unit of x, y, and x error is mm and of  $\theta$  is radians. The number in the horizontal axis is moving step.

The simulation result for GPC motion control moving along the path shown above is presented in figure 6. In the simulation testing, we use L=10, K=10, B=5. Maximum error in x and y are 118mm and 48mm respectively.

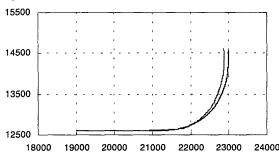


Figure 6 Path Trajectory

### VI. CONCLUSION

The control approach proposed in this paper has a potential for controlling any dynamic system that usually possesses complexity, non-linearity and uncertainty. The benefits resulted from wavelet neural network will be the convex cost index without local minima dilemma, the appropriate initial values of weights, and a constructive approach for the hidden layer of feed-forward neural networks. Simulation results show the ability of approximation of wavelet neural network to the nonlinear dynamics and of adaptive tracking control of the wavelet networks based GPC. Our future work will focus on the further exploitation for the wavelet network approximation with few hidden nodes, training network on line and adaptively selecting the hidden nodes.

### REFERENCES

- J. Canderle and D. Caldwell, Generalized Predictive Control of Pneumatic Muscle Actuator, European Advanced Robotics Systems Masterclass and Conference-Robotics 2000, Salford, UK, April 2000.
- [2] D. Clarke, C. Mohtadi, and P. Tuffs, Generalized Predictive Control-Part I. The Basic Algorithm, Automatica, Vol. 23, No. 2, pp137-148, 1987.
- [3] B. Delyon, A. Juditsky, and A. Benveniste, Accuracy Analysis for Wavelet Approximations, IEEE Trans. on Neural Network, Vol. 6, No. 2, March 1995, pp332-348.
- [4] M. Martinez, J. S. Senent, and X. Blasco, Generalized Predictive Control Using Genetic Algorithms (GAGPC), Artificial Intelligence, Vol. 11, 1998, pp355-367.
- [5] K. S. Narendra and K. Parthasarathy, Identification and Control of Dynamical Systems, IEEE Trans. on Neural Network, Vol. 1, No. 1, March 1990, pp4-27.

- [6] Y. C. Pati and P. S. Krishnaprasad, Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transformations, IEEE Trans. on Neural Network, Vol. 4, No. 1, Jan. 1993, pp73-85.
- [7] D. Ramirez, et al, Nonlinear MBPC for Mobile Robot Navigation Using Genetic Algorithms, Proceedings of the 1999 IEEE Int. Conf. on Robotics & Automation, Detroit, Michigan, May, 1999, pp2452-2457.
- [8] R. M. Sanner and J. E. Slotine, Structurally Dynamic Wavelet Networks for the Adaptive Control of Uncertain Robotic Systems, Proceedings of the 34<sup>th</sup> Conf. on Decision & Control, New Orland, LA, Dec. 1995, pp2460-2467.
- [9] B. Song and A. Koivo, Neural Network Model Based Control of a Flexible Link Manipulator, Proc. of the IEEE Int. Conf. on Robotics & Automation, Leuven, Belgium, May, 1998, pp812-817.
- [10] Y. Tan and R. Keyser, Neural Network Based Adaptive Predictive Control, Advances in Model Based Predictive Control, Edited by D. Clarke, pp358-369, 1994.
- [11] S. Yao, C. J. Wei, and Z. Y. He, Evolving Wavelet Neural Networks for Function Approximation, Electronics Letters, Vol. 32, No. 4, Feb. 1996, pp360-361.
- [12] J. M. Zamarreno and P. Vega, Neural Predictive Control: Application to a Highly Non-linear System, Artificial Intelligence, Vol. 12, 1999, pp149-158.
- [13] J. Zhang, G. Walter, Y. Miao, and W. Lee, Wavelet Neural Networks for Function Learning, IEEE Trans. on Signal Processing, Vol. 43, No. 6, June 1995, pp1485-1495.
- [14] Q. Zhang, Using Wavelet Network in Nonparametric Estimation, IEEE Trans. on Neural Network, Vol. 8, No. 2, March 1997, pp227-236.
- [15] Q. Zhang and A. Benveniste, Wavelet Networks, IEEE Trans. on Neural Network, Vol. 3, No. 6, Nov. 1992, pp889-898
- [16] Huosheng Hu and Dongbing Gu, Generalized Predictive Control of an Industrial Mobile Robot, IASTED International Conference on Intelligent Systems and Control, Santa Barbara, CA, USA, 28-30 October 1999.
- [17] D. Gu, H. Hu, M. Brady, F. Li, Navigation System for Autonomous Mobile Robots at Oxford, Proceedings of International Workshop on Recent Advances in Mobile Robots, Leicester, U.K., pp. 24-33, 1-2 July 1997
- [18] G. Walter and J. Zhang, Orthonormal Wavelets with Simple Closed-Form Expressions, IEEE Trans. on Signal Processing, Vol. 46, No. 8, 1998, pp2248-2251.