

# Neural network modelling for mean velocity and turbulence intensities of steep channel flows

Fi-John Chang,<sup>1\*</sup> Han-Chung Yang,<sup>2</sup> Jau-Yau Lu<sup>3</sup> and Jian-Hao Hong<sup>3</sup>

<sup>1</sup> Department of Bioenvironmental System Engineering, National Taiwan University, Taipei, Taiwan, ROC

<sup>2</sup> Department of Leisure, Recreation and Tourism Management, Shu-Te University, Yanchao, Taiwan, ROC

<sup>3</sup> Department of Civil Engineering, National Chung-Hsing University, Taichung, Taiwan, ROC

## Abstract:

The main purpose of this study is to evaluate the potential of simulating the profiles of the mean velocity and turbulence intensities for the steep open channel flows over a smooth boundary using artificial neural networks. In a laboratory flume, turbulent flow conditions were measured using a fibre-optic laser doppler velocimeter (FLDV). One thousand and sixty-four data sets were collected for different slopes and aspect ratios at different locations. These data sets were randomly split into two subsets, i.e. training and validation sets. The multi-layer functional link network (MFLN) was used to construct the simulation model based on the training data. The constructed MFLN models can almost perfectly simulate the velocity profile and turbulence intensity. The values of correlation coefficient ( $\gamma$ ) are close to one and the values of root mean square error (RMSE) are close to zero in all conditions. The results demonstrate that the MFLN can precisely simulate the velocity profiles, while the log law and Reynolds stress model (RSM) are less effective when used to simulate the velocity profiles close to the side wall. The simulated longitudinal turbulence intensities yielded by the MFLN were also fairly consistent with the measured data, while the simulated vertical turbulence intensities by the RSM were not consistent with the measured data. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS artificial neural network (ANN); velocity profile; turbulent open channel flow; log-law model; Reynolds stress model

Received 12 February 2006; Accepted 25 September 2006

## INTRODUCTION

Over the past decades, numerous researchers have attempted to construct the velocity profile and understand the structure of the turbulent in open channel flow. Both laboratory experiments and field studies have used hot-film anemometer, laser-Doppler anemometer, particle-tracking velocimeter (Nezu and Nakayama, 1997), and other anemometers (Rolland and Lemmin, 1997; Tariq and Masood, 2001; Shi and Hughes, 2002). These studies identified qualitative patterns of the turbulence in open channel flow and supported quantitative description turbulence regimes. This has resulted in a substantial increase in the understanding of form and process inter-relationships in channel geomorphology, including applications in the dynamic analysis of channels and sediment transport. Indeed, in recent years a considerable increase in the use of a process-based numerical approach to simulate hydraulic patterns in complex natural channels has been witnessed. Advances have included techniques to describe two- and three-dimensional velocity profiles and turbulence-driven secondary motions (Thomas and Williams, 1999; Czernuszenko and Rylov, 2002). Many researchers have emphasized the need for improved laboratory and field studies. There is also a need for

the development of alternative methodologies to better address the simulation of complex conditions in natural channels (Lane and Richards, 1998; Booker, 2003). The expectation is that better data from new instruments can enhance the development of the velocity profile models and turbulence simulation. The development of process-based simulation models has only just begun. Many non-linear complex characteristics of flow conditions, especially turbulent flows, still cannot be satisfactory simulated. For example, sophisticated formulas/models such as the log-law and Reynolds stress model (RSM) cannot well describe the velocity profile near the side wall, water surface and channel bed (Lu *et al.*, 2003).

The artificial neural network (ANN) is now gaining interest from researchers in many disciplines of science and engineering. ANNs have also provided promising results in the field of hydrology and water resources such as streamflow prediction (Chang and Chen, 2003; Chiang *et al.*, 2004), rainfall-runoff modelling (Tokar and Johnson, 1999; Lin and Chen, 2004), and reservoir operation (Chang and Chang, 2001). Although some attempts have been made to apply ANNs to problems in fluid dynamics (Milano and Koumoutsakos 2002; Hocevar *et al.*, 2004), this method is rarely used in open channel flow simulation (Omid *et al.*, 2005).

A previous study had demonstrated that the ANN can nicely fit the velocity profiles of a relatively flat

\* Correspondence to: Fi-John Chang, Department of Bioenvironmental System Engineering, National Taiwan University, No.1, Sec. 4, Roosevelt Road, Taipei, Taiwan, ROC. E-mail: changfj@ntu.edu.tw

(slope = 0.1%) open channel flow and can precisely predict the discharges for the investigated conditions (Yang and Chang, 2005). The main objective of this work is to further evaluate the potential of ANNs for simulating the profiles of the mean velocity and turbulence intensities of the steep open channel flows over a smooth boundary. Simulations will provide profiles of the mean velocity and turbulence intensities in the longitudinal and vertical directions using a neural network trained to data collected in a laboratory flume using a fibre-optic laser doppler velocimeter (FLDV). The simulation model is based on a multi-layer functional link neural network (MFLN). The modelling processes and performance of the MFLN are described and compared with the corresponding experimental data and two familiar semi-empirical models.

### VELOCITY PROFILE FORMULAS AND RSM

The velocity profiles of fully developed turbulent open channel flow are of great interest to engineers, particularly in the estimation of erosion and sediment transport in alluvial channels. A recent research result has also represented that the velocity profile is the driver of physical habitat quality for aquatic species (Booker, 2003). Due to the practical importance of the problem, many studies have been conducted to seek the velocity profiles in turbulent flow, experimentally using pitot tubes, hot film anemometer (Cardoso *et al.*, 1989) or a laser-doppler anemometer (Kirkgöz and Ardicioglu, 1997; Lu *et al.*, 2003). Various semi-empirical models have then been provided to simulate the velocity profile of fully developed turbulent channel flow (Sarma *et al.*, 1983; Hu and Hui, 1995). One of the most popular velocity distribution models is the log-law presented by von Kármán (1930) and Prandtl (1932):

$$\frac{u}{U_*} = \frac{1}{\kappa} \ln \frac{U_* y}{\nu} + C \quad (1)$$

where  $\kappa$  is the von Kármán constant;  $C$  is a constant;  $U_*$  is shear velocity ( $= \sqrt{\tau_0/\rho}$ );  $\tau_0$  is bed shear stress;  $\rho$  is mass density and  $\nu$  is the kinematic viscosity. Equation (1) was used to simulate the velocity profiles from FLDV measurements and provide a data set to compare with the results obtained from simulations using ANN.

RSMs are the basis for most traditional turbulence models. The RSM is not restricted by the Boussinesq assumption relating turbulent stresses to rates of mean strain. In principle, RSM is a high level scheme and is used to model the Reynolds stress transport equation. RSM discards the eddy viscosity and velocity gradient approach and the Reynolds stresses ( $-\rho \overline{u_i u_j}$ ) are computed directly by the following differential equation:

$$\frac{\partial}{\partial t}(\rho \overline{u_i u_j}) + C_{ij} = D_{ij}^T + D_{ij}^L + P_{ij} + \varphi_{ij} + \varepsilon_{ij} \quad (2)$$

where  $C_{ij}$ ,  $D_{ij}^T$ ,  $D_{ij}^L$ ,  $P_{ij}$ ,  $\varphi_{ij}$  and  $\varepsilon_{ij}$  are the convection, turbulent diffusion, molecular diffusion, stress production, pressure strain and dissipation terms, respectively.

The main disadvantage of RSM is computational complexity and corresponding calculation time. To explore the capabilities and limitations of the existing turbulence model, the sophisticated RSM was chosen to simulate the fully developed turbulent open-channel flow by Lu *et al.* (2003), and the computed mean velocities and turbulence statistical quantities from this model was compared with the MFLN.

In this study, the results of Lu *et al.* (2003) are quoted to carry on the comparison and briefly describe how the RSM is developed. The computational domain of the flow was discrete with  $60 \times 10 \times 36$  grid points. The initial velocities at the inlet section were set to be the measured mean velocity. For the pressures at the outlet section, the pressure at the water surface was assumed to be zero, the effective pressure below the water surface was assumed to be hydrostatic. The initial turbulence intensity was set to be 3%. The turbulence length scale was set to be 0.01 m. The two layer zonal model was chosen for the wall boundaries and the volume of fluid method was used at the free-surface.

### MULTI-LAYER FUNCTIONAL LINK NETWORK (MFLN)

Neural networks were inspired by biological neuron processing to perform brain-like computation through massively simple connective artificial neurons. After McCulloch and Pitts (1943) established the first neural network, many neural networks, such as the multi-layer perceptrons neural network, radial basis function neural network, self-organizing network, and fuzzy neural network were developed to solve different problems (Ham and Kostanic, 2001). Multi-layer perceptrons neural network is the most common type of neural network and has been successfully applied for adaptive identification of a variety of non-linear processes. In this network, the inputs to a node are linearly weighted before the sum is passed to the activation function of the hidden layer that gives the network its non-linear approximation ability. This process, however, might create problems in training the network weights, as non-linear learning rules must be used, the learning rate is often slow and local minima may cause problems (Chen and Billings, 1992). One way to avoid non-linear training is to use MFLNs. In these networks, a non-linear functional expansion of the network inputs is initially performed and the resulting terms are combined linearly. The structure obtained has a good non-linear approximation capability, and the estimation of network weights is linear. Among them, the MFLN has been used as a powerful pattern recognition tool (Yamany *et al.*, 1997), applied in modelling (Patra and van den Bos, 2000), and process control (Costa *et al.*, 2002).

Figure 1 shows the general structure of a MFLN with an error back-propagation algorithm. It includes input (including inputs transformation), hidden, and output layers. Each layer includes several neurons that are the fundamental building blocks for the network. The

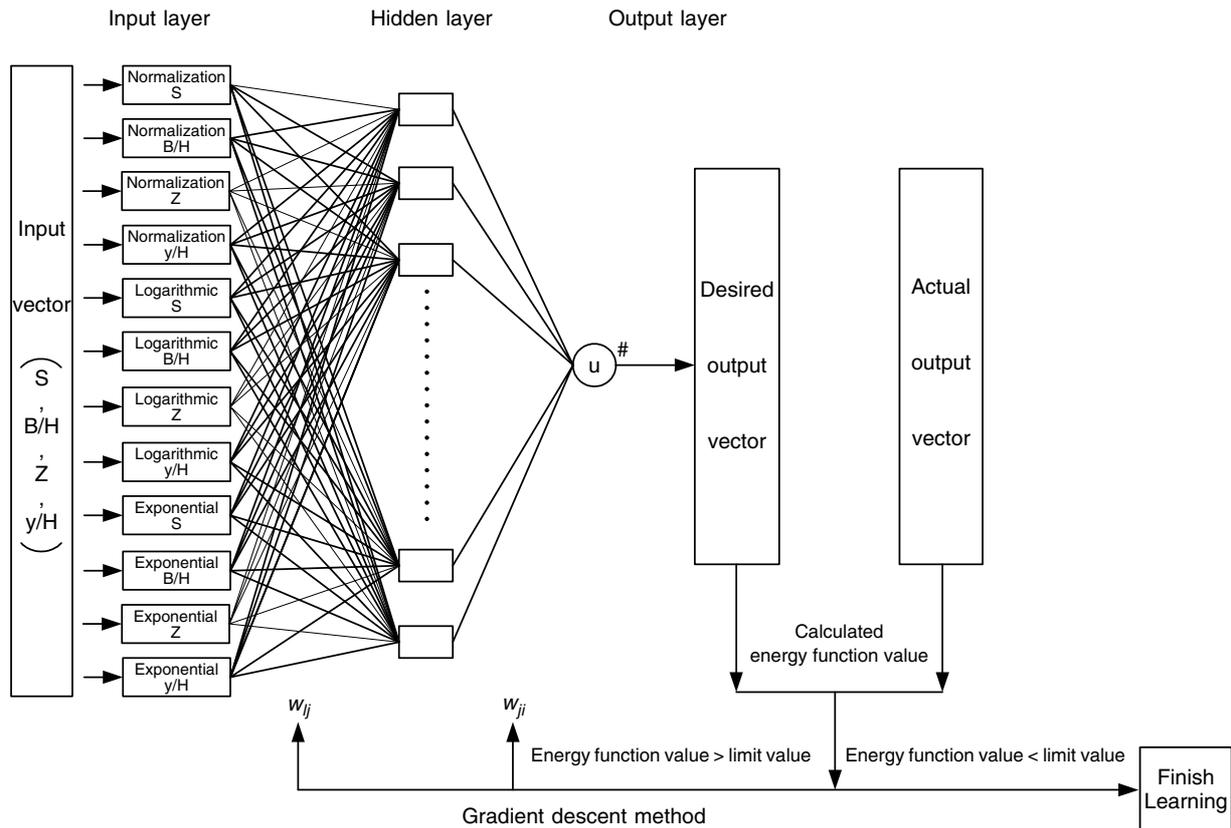


Figure 1. MFLN architecture and learning process (# replace  $u$  with  $u'$  or  $v'$  for simulating longitudinal or vertical turbulence intensities, respectively)

input layer receives the input vector. The hidden layer represents the interaction of the input variables. The network may have many or no hidden layers depending on the problem's degree of complexity. The output layer provides transportation for a result to output vectors. The learning process goes from input layer to hidden layer, and then to output layer. Learning involves modifying the connection weights and biases until the network is capable of reproducing the target output for the respective input. Learning takes place in an iterative fashion by the back propagation algorithm. When MFLN is implemented, one would have to decide on the number of neurons in the hidden layer that leads to the best results. MFLN is a variation of the basic backpropagation network to improve the ability to form complex relationships between inputs and outputs. It provides information to the network by adding inputs that are functions (ex. normalization) of the inputs that handle non-linearly non-separable classes. Therefore the dimensions of the inputs increase by using non-linear combinations of the input features. Generally, simple normalization, exponential and logarithmic terms may be appropriate for transforming inputs to the network (Tang *et al.*, 2003) The transforming inputs are able to enhance overall effect of mapping the input into a large space. MFLN usually can perform better than the multi-layer network and has a distinct advantage because of easier learning in the model MFLNs are better at approximating functions that contain both linear and

non-linear terms. But the computational complexity of network is increased as weights corresponding to direct interconnection between the input and output neurons are added.

MFLN is a feedforward network that also uses as standard a backpropagation training algorithm. Backpropagation adjusts the weights in a feedforward network, with differentiable activation function units, to learn (fit) a training set of input–output patterns. The standard backpropagation algorithm for training the network is based on the gradient-descent search algorithm that computes the first derivative of an objective (cost) function with respect to the parameters (weights) of the network. A supervised learning rule for adjusting the weights of the network follows the error function being minimized. By using the Delta rule directly, the targets for the output units are explicitly specified for updating the  $w_{lj}$  weights

$$\Delta w_{lj} = w_{lj}^{\text{new}} - w_{lj}^c = \eta_0 (d_l - y_l) f'_0(\text{net}_l) x_l \quad (3)$$

$$\text{net}_l = \sum w_{lj} x_l \quad (4)$$

where  $w_{lj}^{\text{new}}$  and  $w_{lj}^c$  are the updated and current weights;  $d_l$  and  $y_l$  are the  $l$ th component of target and output unit;  $\eta_0$  is the learning rate;  $f'_0$  is the derivative of the non-linear function  $f_0$  with respect to net; net is the weighted sum for the  $l$ th output unit. Learning takes place in an iterative fashion: each cycle consists of a forward propagation step with the weights being modified by the method to reduce the cost function. The standard

backpropagation algorithm can be found in Ham and Kostanic (2001).

#### EXPERIMENTAL EQUIPMENT AND PROCEDURE

A schematic diagram of the experimental equipment is shown in Figure 2. The data were collected in a recirculating flume 0.25 m wide, 0.25 m depth and 12 m long. The flow can be varied from 0 to  $0.03 \text{ m}^3 \text{ s}^{-1}$  by using two pumps and a valve control on the discharge line. The slope of the glass channel can be adjusted from 0 to 10% by screw jacks. A settling honeycomb was installed at the entrance of the channel to prevent large-scale turbulence and achieve uniform entrance flow.

The average depth of flow was determined by observing the difference in elevation between the water surface

and the channel bed with an ultrasonic water level sensor, which was mounted on a four-wheeled carriage traveling on rails of the flume. The water level was measured every 0.5 m along the centreline of the flume. Only those depths observed in the part of the flume having uniform slope were used to establish the average depth.

The velocities were measured with a two-component (two-colour, four beams) FLDV operated in the backward-scattering system. To understand the variation of the velocity close to the channel bed, more observations were made. The mean velocity of flow in the flume ( $U_m$ ) was determined from the observed discharge and water depth by use of the continuity equation,

$$U_m = \frac{Q}{HB} \quad (8)$$

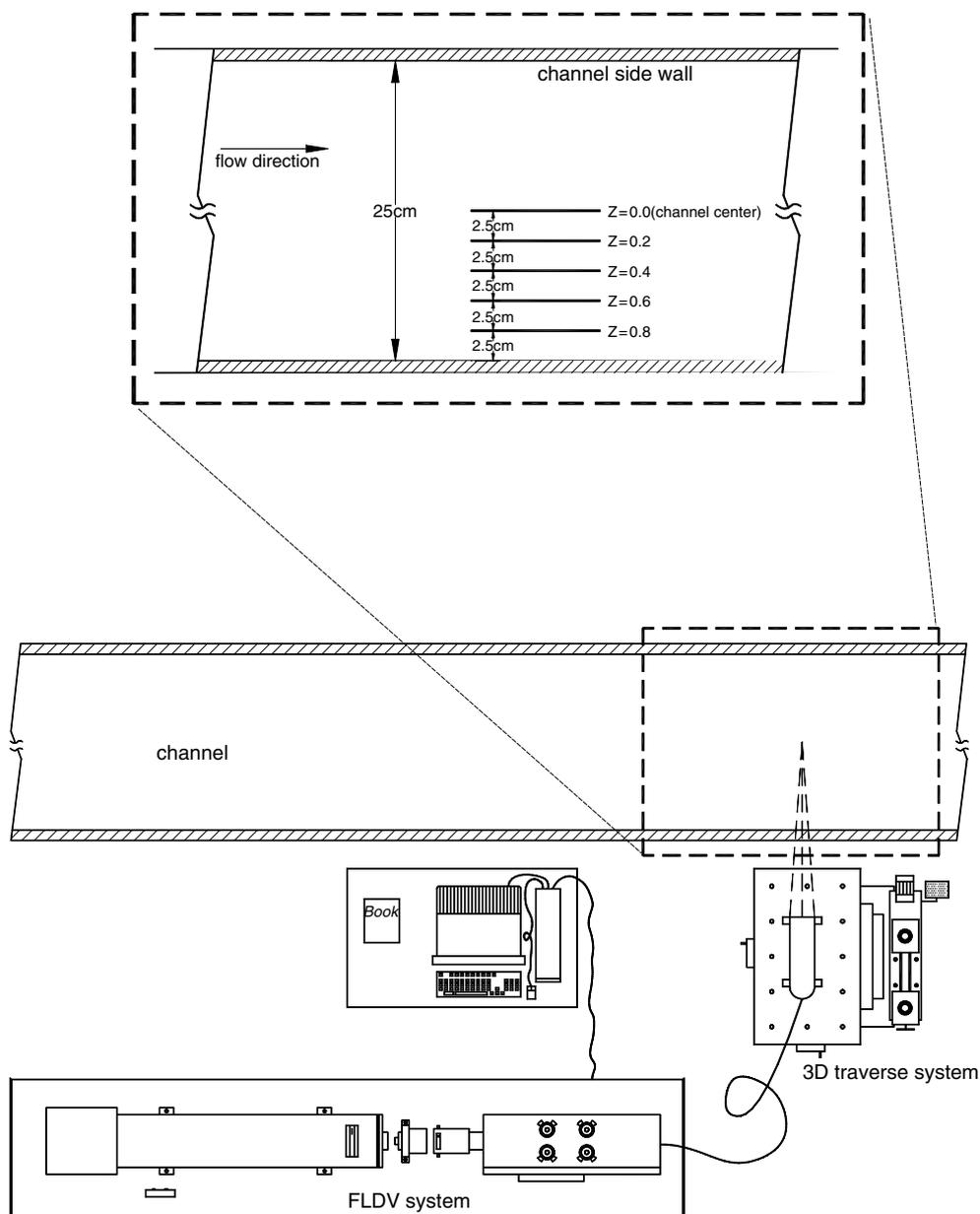


Figure 2. Set-up of experimental equipment and velocity profile measuring locations

Table I. Summary of available data

Run	$S$ (%)	$H$ (mm)	$T$ (°C)	$Q$ (m <sup>3</sup> s <sup>-1</sup> )	$U_m$ (m s <sup>-1</sup> )	$U_*$ (m s <sup>-1</sup> )	$B/H$	Re	Fr
S1H22	1	22	27	0.0049	0.891	0.0428	11.36	19 475	1.92
S1H30	1	30	28	0.008	1.067	0.0487	8.33	30 824	1.97
S1H45	1	45	27	0.0139	1.236	0.0570	5.56	47 771	1.86
S1H54	1	54	27	0.0178	1.319	0.0608	4.63	58 098	1.81
S2H22	2	22	25	0.0069	1.255	0.0606	11.36	26 282	2.70
S2H30	2	30	26	0.0108	1.440	0.0689	8.33	39 843	2.65
S2H54	2	54	25.5	0.0263	1.948	0.0862	4.63	82 352	2.64

where  $Q$  is discharge,  $H$  is water depth,  $B$  is channel width equalling to 0.25 m. For measuring the fully developed turbulent flow, velocity profiles were measured 8.2 m from the flume entrance where the water depth of the test section of flume was uniform and fully developed (Yang, 1998). Shear velocities are defined as:

$$U_* = \sqrt{\tau_0/\rho} = \sqrt{gRS} \quad (9)$$

where  $g$  is acceleration due to gravity,  $R$  is hydraulic radius and  $S$  is channel bed slope.

The aspect ratio, ratio of width to depth, varied from 4.63 to 11.36. Channel bed slopes of 1 and 2% were selected. The Reynolds number ( $Re = U_m R/\nu$ ) varied approximately from 20 000 to 80 000, and the Froude number ( $Fr = U_m/\sqrt{gH}$ ) ranged from 1.81 to 2.70. The velocity profile data were taken in five verticals with different  $Z$  values ( $Z = 0$  and 1 represent the channel centreline and the side wall, respectively). The details of the experimental conditions are given in Table I.

Flows with different channel bed slope and aspect ratio were adopted in this experimental study to investigate the characteristics of the steep open channel flow. It was found that the maximum velocity often occurred not at the free surface. The experimental observations indicated that the depression of the point of maximum velocity increased with a decrease of the aspect ratio, an increase of the bed slope, or an increase of the  $Z$  value (closer to the side wall). It appears that the flow condition can be influenced by the values of  $S$ ,  $B/H$  and  $Z$ .

## MODELLING RESULTS

### Mean velocity

As stated earlier, one of the primary objectives of the present study is to evaluate the potential of neural networks to simulate the velocity profiles from a readily available experiment data set. The mean velocity of the turbulent open channel flow over a smooth boundary is a function of  $S$ ,  $B/H$ ,  $Z$  and  $y/H$ . Therefore, the vertical velocity can be expressed as:

$$u = f(S, B/H, Z, y/H) \quad (10)$$

where  $f()$  represents the non-linear function.

There are 35 available vertical velocity profiles, and each vertical profile includes 22 to 34 data points. The total number of data points is 1064. To build a MFLN for

simulating the velocity profiles, this quantity of data was sufficient. It is common practice to split the available data set into two subsets: training and validation sets. In MFLN, it typically begins with a training sample and uses the backpropagation algorithm to adjust the synaptic weights of multi-layer perceptrons by using as many of the training examples as possible, so that the trained neural network will generalize the input–output mapping. A network is said to generalize well when the input–output mapping computed by the network is correct for validation data that was never used in training the network. Accordingly, 25 vertical velocity profiles with 756 velocities were chosen to train the network, and the remaining 10 vertical velocity profiles (runs S1H45 and S2H30) with 308 velocities were used for validation. This subdivision into training and validation data sets is to assess the generalization of the properties of the trained network.

As shown in Figure 1, the input layer includes 12 variables, and the output layer has only one node representing the velocity value in the specific location of  $y/H$  on the vertical. The eight neurons in the hidden layer were determined through a trial and error process based on the best fitness of the training data sets. The learning rate could affect the convergence speed of the MFLN. The learning rate was set to unity in the beginning and multiplying it by a number, which was depending upon the error function increasing or decreasing during the training process. The sigmoid function was set as the activation function. After a number of trials, the values of the network's weights range from 2.05 to  $-5.78$  and the biases vary from 1.17 to  $-1.48$  were obtained. The network training was continued until the error goal was reached and/or stopped after 5400 iterations. An iteration is one complete pass through a set of inputs and target patterns while training the network.

The root mean square error (RMSE), which can be used for an assessment of the performance, is given by:

$$RMSE = \sqrt{\frac{\sum (u_{obs} - u_{sim})^2}{n}} \quad (11)$$

where  $u_{obs}$  and  $u_{sim}$  are observed and simulated velocities and  $n$  is the number of observations. The correlation coefficient ( $\gamma$ ) is given by:

$$\gamma = \frac{\sum (u_{obs} - \bar{u}_{obs}) \sum (u_{sim} - \bar{u}_{sim})}{\sqrt{\sum (u_{obs} - \bar{u}_{obs})^2 \sum (u_{sim} - \bar{u}_{sim})^2}} \quad (12)$$

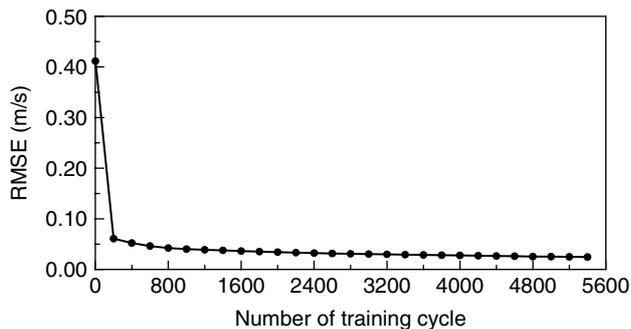


Figure 3. Degree of convergence for the training

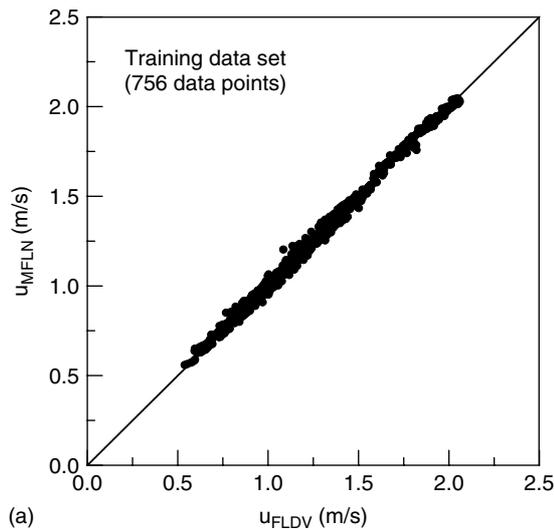
Table II. MFLN results

Subset	$\gamma$	RMSE ( $\text{m s}^{-1}$ )
Training	0.998	0.024
Validation	0.991	0.031

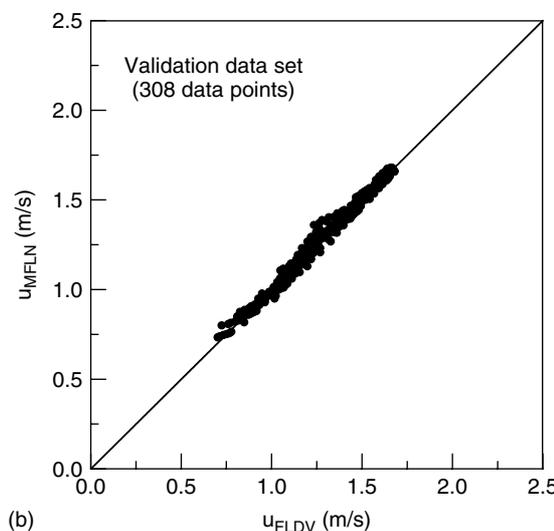
where  $\bar{u}_{\text{obs}}$  and  $\bar{u}_{\text{sim}}$  represent the mean of observed and simulated velocities, respectively. The error graph for the neural network model during training is shown in Figure 3. It shows that the network has converged after about 4000 iterations and remains fairly stable following convergence. Table II gives the value of the  $\gamma$  and RMSE of the training and validation sets. The  $\gamma$  values are close to unity and the RMSEs are quite small. Figure 4 shows the simulated results for both training and verification cases, respectively. It appears that all the data points (observation versus model simulation) nicely fall onto the line of agreement. The results in both Table II and Figure 4 indicate that MFLN can simulate the velocity profile accurately. For the purpose of comparison (reference), the velocities of profiles simulated by RSM presented by Lu *et al.* (2003) and log-law are also provided. The measured mean velocity profiles exhibit a clear retardation effect near the free surface, a feature absent from the numerical simulations (ex. log-law and RSM). Figures 5 and 6 provide a visual impression of the good performance of the MFLN. Table III summarizes the results and shows the performance of MFLN, log-law, and RSM method for  $Z = 0.8$  (near side wall, with significant depression of the point of maximum velocity). It appears that the  $\gamma$  values are close to one and the values of RMSE are very small (close to zero) in all conditions modelled by the MFLN. The results demonstrate that the MFLN can precisely simulate the velocity profiles while the log-law and RSM are less effective when used to simulate the velocity profiles close to the side wall.

*Turbulence intensities*

The analysis and simulation method for the longitudinal and vertical turbulence intensities using MFLN is identical to that for the mean flow velocity. The structure of MFLN is similar to Figure 1. However, the mean velocity  $u$  is replaced with  $u'$  or  $v'$  for simulating the longitudinal or vertical turbulence intensities, respectively.



(a)



(b)

Figure 4. The observation ( $u_{\text{FLDV}}$ ) versus simulation ( $u_{\text{MFLN}}$ ) results: (a) training set; (b) validation set

Four input factors ( $S$ ,  $B/H$ ,  $Z$  and  $y/H$ ) are used as input vector, and the output layer has only one node representing the longitudinal turbulence intensity ( $u'$ ) or vertical turbulence intensity ( $v'$ ) in the specific location of  $y/H$  on the vertical. Twelve neurons in the hidden layer are determined through a trail and error process. The number of neurons in this case is larger than the previous mean-flow case because the physical process influencing the turbulence intensities are more complex near the channel bed. As with the previous case, each set of turbulence intensity data is divided into training and validation subsets. The initial learning rate is set to unity. The network training is continued until either the goal of the RMSE is reached or the number of iterations exceeds 5400.

Figures 7 and 8 show the comparisons of the observed and simulated longitudinal dimensionless turbulence intensities ( $u'/U_*$ ) for runs S1H54 and S2H54, respectively. The simulated longitudinal turbulence intensities yielded by the MFLN were fairly consistent with the measured data (FLDV), while the simulated vertical

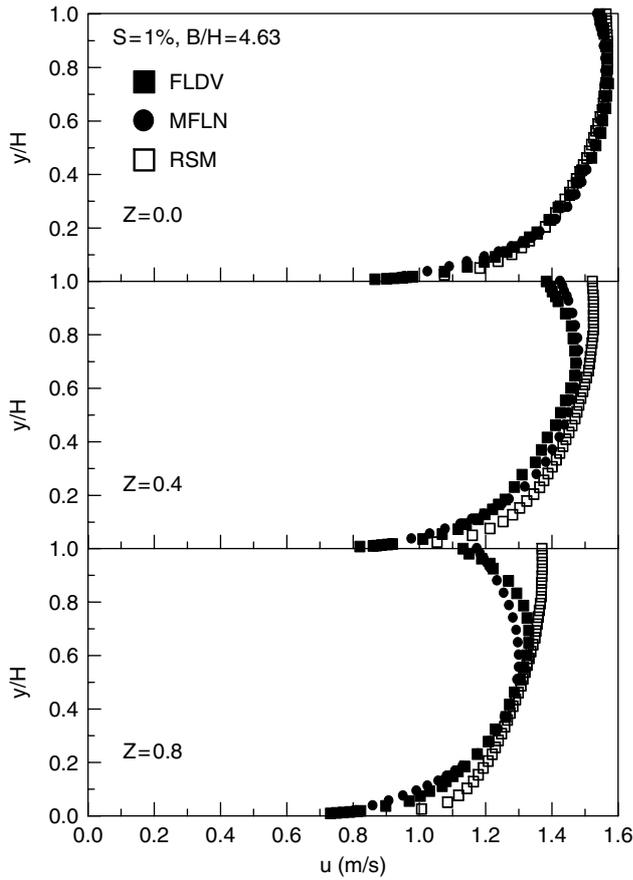


Figure 5. Comparisons of observed and simulated velocity profiles by MFLN and RSM for run S1H54

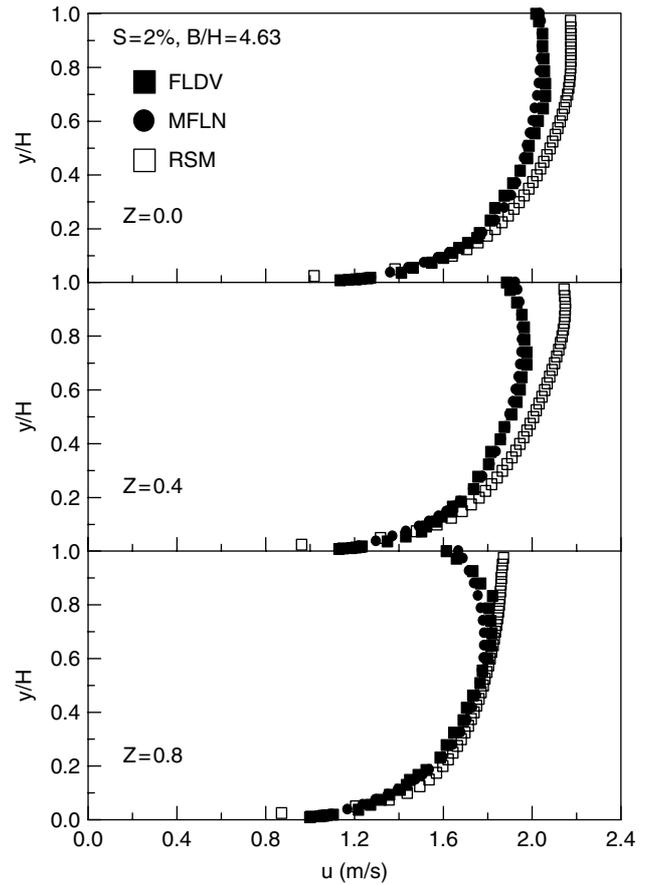


Figure 6. Comparisons of observed and simulated velocity profiles by MFLN and RSM for run S2H54

turbulence intensities by the RSM were not consistent with the measured data. The results indicate that the FLDV data points are simulated better by MFLN than RSM. The comparisons of the observed and simulated vertical dimensionless turbulence intensities ( $v'/U_*$ ) for runs S1H54 and S2H54 are given in Figures 9 and 10.

Again, the results also demonstrate that the MFLN can better simulate the experimental data when compared to RSM, especially near the water surface, channel bed and the side wall (e.g.  $Z = 0.8$ ). From these observations it was found that both the longitudinal and vertical turbulence intensities simulated from MFLN were consistent

Table III. Summary of results ( $Z = 0.8$ )

Run	Method	Maximum $u$ ( $m s^{-1}$ )	Minimum $u$ ( $m s^{-1}$ )	RMSE ( $m s^{-1}$ )	$\gamma$
S1H22	FLDV	0.884	0.539	0.000	1.000
	MFLN	0.892	0.560	0.007	0.998
	log-law	0.985	0.580	0.080	0.985
	RSM	0.769	0.317	0.104	0.934
S1H54	FLDV	1.33	0.730	0.000	1.000
	MFLN	1.327	0.797	0.029	0.990
	log-law	1.589	0.877	0.236	0.942
	RSM	1.371	0.364	0.139	0.926
S2H22	FLDV	1.245	0.775	0.000	1.000
	MFLN	1.224	0.781	0.019	0.998
	log-law	1.441	0.867	0.169	0.970
	RSM	1.233	0.637	0.050	0.968
S2H54	FLDV	1.823	0.963	0.000	1.000
	MFLN	1.791	1.030	0.030	0.994
	log-law	2.317	1.260	0.438	0.975
	RSM	1.876	0.258	0.279	0.956

Note:  $\kappa = 0.4$  and  $C = 5.5$  were given for log-law formula.

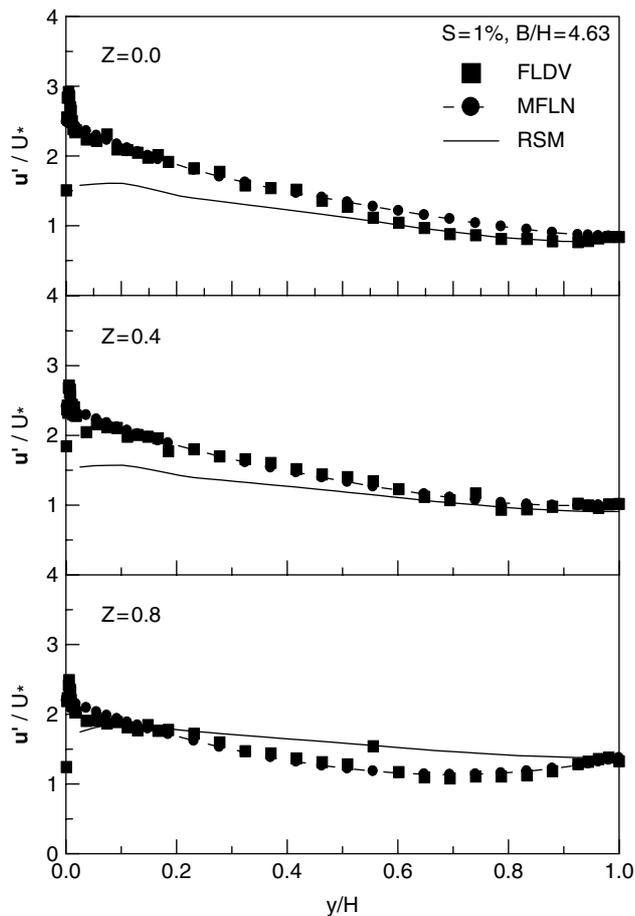


Figure 7. Comparisons of observed and simulated longitudinal dimensionless turbulence intensities by MFLN and RSM for run S1H54

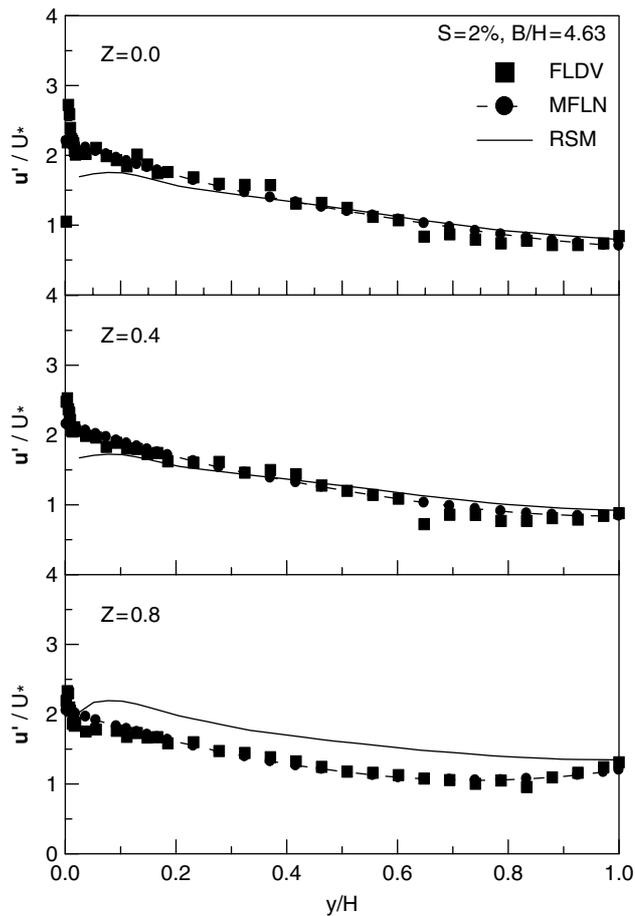


Figure 8. Comparisons of observed and simulated longitudinal dimensionless turbulence intensities by MFLN and RSM for run S2H54

with that for FLDV measurements under steep slope conditions (i.e. 1 and 2%).

The results obtained in this study give nice evidence that the constructed neural network models can be used for estimating or generating the profiles of mean velocity and turbulence intensities for turbulent open channel flows over a smooth boundary.

### CONCLUSIONS

The ANN technique for modelling the characteristics of turbulent flows in a steep open-channel with a smooth boundary, for different channel bed slopes and aspect ratios ranging from 4.63 to 11.36 measured by a FLDV, is presented. Due to the complexity and non-linearity of the profiles for the mean velocity and turbulence intensities, there are no simple relationships or functions that can effectively describe and/or simulate these profiles, especially near the boundaries (including water surface, side walls and channel bed). In this study, the intent has been to investigate the potential of neural networks to simulate the experimental data measuring profiles and turbulence.

It has been confirmed that the flow characteristics, including the mean velocity and turbulence intensities, are a function of the channel bed slope, aspect ratio,

and location, for turbulent open channel flows over a smooth boundary. Therefore, four factors— $S$ ,  $B/H$ ,  $Z$ , and  $y/H$ —are used as input vectors, while the corresponding value of mean velocity (or turbulence intensity) is used as the desired output. The data set was split into two-subsets, a training set and an independent validation set. MFLN is used to construct the simulation model based on the training set (a total of 756 input–output patterns). The results show that the constructed MFLN models do agree well with the velocity profile or turbulence intensity for the training and validation patterns. The MFLN can precisely simulate the turbulent flow characteristics for the given input conditions. The  $\gamma$  values are close to one and the values of RMSE are close to zero in all conditions. The constructed neural network model can be embedded as a module for estimating or generating the profiles of mean velocity and turbulence intensities for turbulent open channel flows over a smooth boundary. The model can be used as a powerful tool to simulate the flow profiles for similar flow conditions and can also be used to correct the flow profiles with missing data. Furthermore, because the velocity profiles could be almost perfectly simulated by the MFLN in the given conditions, an integral method could be used to obtain the corresponding discharges correctly.

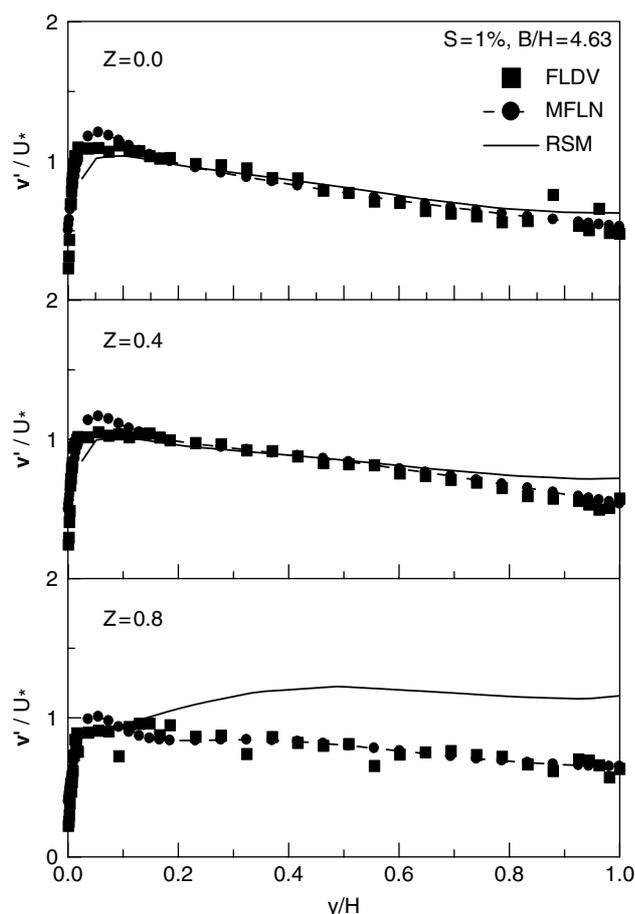


Figure 9. Comparisons of observed and simulated vertical dimensionless turbulence intensities by MFLN and RSM for run S1H54

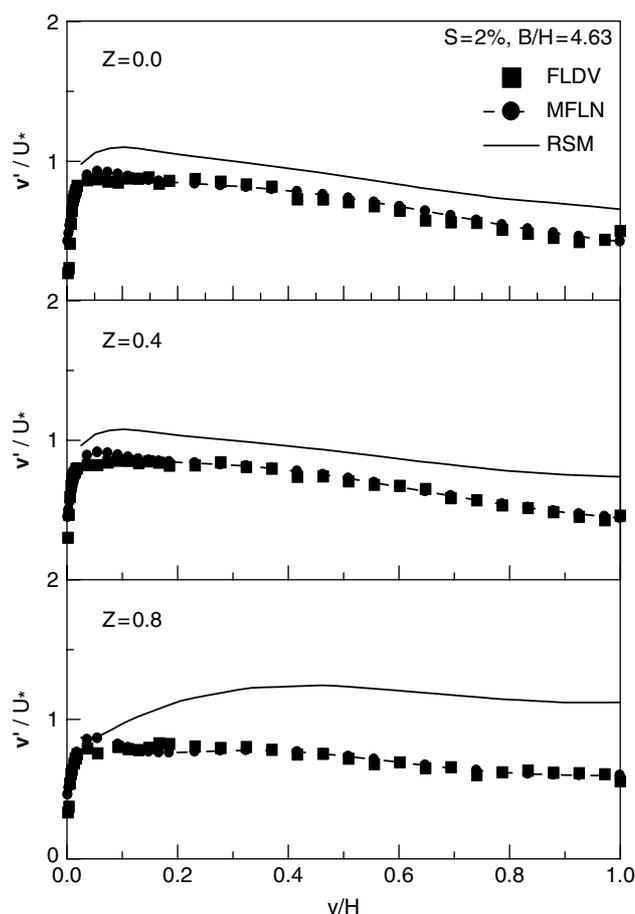


Figure 10. Comparisons of observed and simulated vertical dimensionless turbulence intensities by MFLN and RSM for run S2H54

#### REFERENCES

- Booker DJ. 2003. Hydraulic modelling of fish habitat in urban rivers during high flows. *Hydrological Processes* **17**(3): 577–599.
- Cardoso AH, Graf WH, Gust G. 1989. Uniform flow in a smooth open channel. *Journal of Hydrological Research* **27**(5): 603–616.
- Chang LC, Chang FJ. 2001. Intelligent control for modelling of real-time reservoir operation. *Hydrological Processes* **15**: 1621–1634.
- Chang FJ, Chen YC. 2003. Estuary water-stage forecasting by using radial basis function neural network. *Journal of Hydrology* **270**(1–2): 158–166.
- Chen S, Billings SA. 1992. Neural networks for non-linear dynamic system modelling and identification. *International Journal of Control*, **56**(2): 319–346.
- Chiang YM, Chang LC, Chang FJ. 2004. Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling. *Journal of Hydrology* **290**(3–4): 297–311.
- Costa AC, Meleiro LAC, Maciel Filho R. 2002. Non-linear predictive control of an extractive alcoholic fermentation process. *Process Biochemistry* **38**(5): 743–750.
- Czernuszenko W, Rylow A. 2002. Modeling of three-dimensional velocity field in open channel flows. *Journal of Hydrology Research* **40**(2): 135–143.
- Ham MF, Kostanic I. 2001. *Principles of Neurocomputing for Science & Engineering*. McGraw-Hill: New York.
- Hu CH, Hui YJ. 1995. *Fluid Mechanics and Statistical Regularity for Sediment-laden Open Channel Flow*. Science Publisher: Beijing.
- Hocevar M, Sirok B, Grabec I. 2004. Experimental turbulent field modeling by visualization and neural networks. *Journal of Fluids Engineering Transactions of the ASME* **126**(3): 316–322.
- Kirkgöz MS, Ardiclioglu M. 1997. Velocity profiles of developing and developed open channel flow. *Journal of Hydrologic Engineering, ASCE* **123**(2): 1009–1105.
- Lane SN, Richards KS. 1998. High resolution, two-dimensional spatial modelling of flow processes in a multi-thread channel. *Hydrological Processes* **12**(8): 1279–1298.
- Lu JY, Hong JH, Lee KZ, Wang CY, Yang HC. 2003. Measurement and simulation of turbulent flow in a steep open-channel with smooth boundary. *Journal of the Chinese Institute of Engineers* **26**(2): 201–210.
- Lin G, Chen L. 2004. A non-linear rainfall-runoff model using radial basis function network. *Journal of Hydrology* **289**: 1–8.
- McCulloch WS, Pitts W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* **5**: 115–133.
- Milano M, Koumoutsakos P. 2002. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics* **182**(1): 1–26.
- Nezu I, Nakayama T. 1997. Space-time correlation structures of horizontal coherent vortices in compound open-channel flows by using particle-tracking velocimetry. *Journal of Hydrology Research* **35**(2): 191–208.
- Omid MH, Omid M, Varaki ME. 2005. Modelling hydraulic jumps with artificial neural networks. *Proceedings of the Institution of Civil Engineers—Water Management* **158**(2): 65–70.
- Patra JC, van den Bos A. 2000. Modeling of an intelligent pressure sensor using functional link artificial neural networks. *ISA Transactions* **39**(1): 15–27.
- Prandtl L. 1932. Zur turbulenten Strömung in Röhren und längs Platten. *Ergebnisse der Aerodynamischen Versuchsanstalt zu Göttingen* **4**: 18–29.
- Rolland T, Lemmin U. 1997. A two-component acoustic velocity profiler for use in turbulent open-channel flow. *Journal of Hydrology Research* **3**(4): 545–561.
- Sarma KVN, Lakshminarayana P, Rao NSL. 1983. Velocity distribution in smooth rectangular open channels. *Journal of Hydrologic Engineering, ASCE* **109**(2): 270–289.
- Shi Z, Hughes JMR. 2002. Laboratory flume studies of microflow environments of aquatic plants. *Hydrological Processes* **16**(16): 3279–3289.
- Tang CW, Chen HJ, Yen T. 2003. Modeling confinement efficiency of reinforced concrete columns with rectilinear transverse steel using

- artificial neural networks. *Journal of Structural Engineering, ASCE* **129**(6): 775–783.
- Tariq AUR, Masood M. 2001. Deflecting velocity rod for flow measurements in small channels. *Journal of Irrigation and Drainage Engineering* **127**(5): 311–317.
- Thomas TG, Williams JR. 1999. Large-eddy simulation of flow in a rectangular open-channel. *Journal of IAHR* **37**(3): 345–361.
- Tokar AS, Johnson PA. 1999. Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering, ASCE* **4**(3): 232–239.
- von Kármán T. 1930. Mechanische Ähnlichkeit und Turbulenz. *Göttinger Nachrichten, Math. Phys. Klasse* 58–60.
- Yamany SM, Khiani KJ, Farag AA. 1997. Application of neural networks and genetic algorithms in the classification of endothelial cells. *Pattern Recognition Letters* **18**(11–13): 1205–1210.
- Yang HC. 1998. *Flow characteristics for steep channel flow with smooth boundary*. Master Thesis, National Chung-Hsing University, Taichung, Taiwan (in Chinese).
- Yang HC, Chang FJ. 2005. Modelling combined open channel flow by artificial neural network. *Hydrological Processes* **19**: 3747–3762.