International Journal of Computer and Communication Technology

Volume 4 | Issue 4 Article 1

October 2013

Development of a fuzzy logic based software for automation of a single pool irrigation canal

Dr. R. Gopakumar
Government Engineering College, Sreekrishnapuram, Palakkad, Kerala – 678 633, India.,
gopak02@yahoo.co.in

Reena Nair
Government Engineering College, Sreekrishnapuram, Palakkad, Kerala – 678 633, India,
reenanair@gmail.com

Vinuraj R.
Government Engineering College, Sreekrishnapuram, Palakkad, Kerala – 678 633, India.,
vinurajr@gmail.com

Sony Davis
Government Engineering College, Sreekrishnapuram, Palakkad, Kerala – 678 633, India.,
sonydavis@gmail.com

Bijeesh V.
Government Engineering College, Sreekrishnapuram, Palakkad, Kerala – 678 633, India.,
bijeeshv@gmail.com

See next page for additional authors

Recommended Citation

Gopakumar, Dr. R.; Nair, Reena; R., Vinuraj; Davis, Sony; V., Bijeesh; and Jacob, Junil (2013) "Development of a fuzzy logic based software for automation of a single pool irrigation canal," *International Journal of Computer and Communication Technology*: Vol. 4 : Iss. 4 , Article 1.
Available at: https://www.interscience.in/ijcct/vol4/iss4/1

Development of a fuzzy logic based software for automation of a single pool irrigation canal

Authors

Dr. R. Gopakumar, Reena Nair, Vinuraj R., Sony Davis, Bijeesh V., and Junil Jacob

This article is available in International Journal of Computer and Communication Technology: https://www.interscience.in/ijcct/vol4/iss4/1

Development of a fuzzy logic based software for automation of a single pool irrigation canal

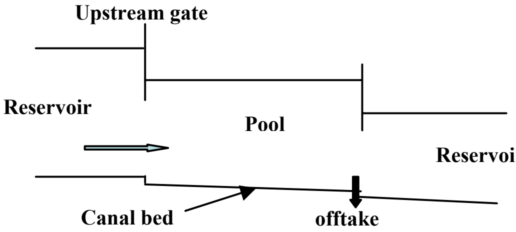
Dr. R. Gopakumar[1], Reena Nair[1,] Vinuraj R.[2], Sony Davis[3], Bijeesh V.[4], Junil Jacob[5]
Government Engineering College, Sreekrishnapuram, Palakkad, Kerala – 678 633, India.
E-mail: gopak02@yahoo.co.in

Abstract— **A fuzzy logic based software for automation of a single pool irrigation canal is presented. Purpose of the software is to control downstream discharge and water level of the canal, by adjusting discharge release from the upstream end and upstream gate settings. The software is developed on a fuzzy control algorithm proposed by the first author during his doctoral research work and published in literature. Details of the algorithm are given. The algorithm was originally developed using fuzzy logic tool box of MATLAB, which is proprietary software not available freely and hence cannot be adopted for general use. Present study describes development of a canal automation software based on this algorithm using open source tools, which are freely available. The software is transparent and intuitive, which can be easily applied by field engineers. The effort required in tuning the fuzzy model has been reduced by including an optimization technique. Also, a new procedure has been introduced for fuzzy inference based on the Mamdani Implication method. The software is tested by applying it to water level control problem in a canal with a single pool, as reported in literature, and satisfactory results are obtained.**

Keywords- Irrigation canals; canal automation software; open source tools; Fuzzy systems

I. INTRODUCTION

Irrigation is the largest user of water, which comes more than 80 % of world water consumption. Hence it is essential that irrigation canals are to be operated to their maximum efficiency by minimizing wastage of water and maximizing flexibility in operations. Also, many irrigation canals are under dilapidated conditions and their rehabilitation can be economically achieved by operating them effectively (and not by rebuilding them as originally designed, which will be highly expensive). Effective operation of the canals in accordance with these principles can be achieved through canal automation, which means application of automatic devises or logic to assist in the operation of the canal [1]. In the present context, canal automation implies development and application of a fuzzy logic based software to control the downstream discharge and water level of a single pool canal (Fig. 1). Here the term “pool” implies the stretch of the canal between two check gates. The canal is under known steady flow condition initially with water level at the downstream end at its specified target value. Perturbations in the system are due to changes in the discharges through the offtake, which is located at the downstream end of the pool (Fig. 1). Under the effect of these perturbations, the water level at the downstream end starts deviating from its target value. Purpose of the fuzzy control software is to determine the inflow discharges and the gate settings at the upstream end, at every regulation time step, so that stable control at the downstream end is achieved in a minimum of time.



Different types of canal automation algorithms have been proposed in the past as described in [2]. Some of them are developed on control systems theory. Examples are classical (e.g., Proportional plus Integral, PI), predictive (e.g., Model Predictive Control, MPC) and optimal (e.g., Linear Quadratic Regulator, LQR) controllers. Difficulties associated with application of these controllers have been discussed in literature. Tuning of PI controller, to determine the controller parameters, is a tedious task [3]. Also, both LQR and MPC assume that the system can be described by linear differential equations, where as the underlying unsteady flow equations are non-linear. The optimization involved in MPC is a difficult task and consume significant amount of computing power [4]. Also, for LQR controllers, the relationship between the measured water levels and the control actions is mathematically complex and hence the controller appears as a “black box” [5]. Such “black box” type controllers are less likely to be accepted by the canal operators than ones that are intuitive [6].

There is another class of canal automation algorithms, which make use of inversion of the following dynamic wave equations (Saint-Venant equations) [7], as the design technique:

Continuity equation

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}=0 \tag{1}$$

Momentum equation

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left[\frac{Q^2}{A}\right]+gA\frac{\partial h}{\partial \mathrm{x}}-gAS_0+gAS_f=0 \qquad (2)$$

where A is flow cross sectional area, h is flow depth, Q is discharge, S_0 is bed slope, S_f is friction slope, g is acceleration due to gravity, t is time variable and x is space variable. Gate Stroking [8] and CLIS [9] are examples for the algorithms coming under this class.

The Gate Stroking is a feed forward control algorithm, suitable for scheduled offtake discharge changes, where the offtake discharge changes are known a priori. It is based on the method of characteristics, which is more complex. The inherent difficulties and impracticality of Gate Stroking have been demonstrated in literature [10, 11]. Finite difference alternatives for the Gate Stroking method were also proposed [12, 13]. These algorithms involve inversion of the governing equations (1) and (2) in both time and space. But the information from initial conditions and transient boundary conditions degrades with time due to friction and hence backward computation in time can cause instability [14]. However, the problem of instability is not applied to inverse computations in space and a feed back control algorithm of this type, suitable for unscheduled offtake discharge changes, was reported later [15]. CLIS [9] is a modified version of this algorithm, made suitable for scheduled flow changes also. For all these algorithms, the recovery time, which is the time period in which the controlled variables are expected to attain their target values, is taken as the upstream wave traveling time, which is much more than the physically correct downstream wave traveling time. Additionally, CLIS makes use of linearized version of the Saint-Venant equations for the inverse modeling where as the underlying system dynamics are nonlinear.

Reference [16] describes development of a fuzzy logic based dynamic wave model inversion algorithm for canal regulation, which was capable of overcoming the aforementioned drawbacks of existing algorithms. Tuning of the algorithm was easy and computations were easy to implement. No linearizations of the governing equations were involved and the recovery time was rightly taken as the downstream wave traveling time. The fuzzy control algorithm was transparent and intuitive and can be expected to be more acceptable for canal operators as compared to "black box" type controllers. However, the algorithm was developed using the fuzzy logic tool box of MATLAB, which is proprietary software not available freely and hence cannot be adopted for general use. Present study describes development of a canal automation software based on this algorithm, for a single pool canal, where in freely available components are used.

II. THEORETICAL ASPECTS

The concept of fuzzy logic was first introduced by Zadeh [17]. Details of fuzzy sets, fuzzy numbers, membership functions and fuzzy rules can be found in literature [18, 19]. Fuzzy rule based models are suitable for nonlinear input-output mapping [18]. In fuzzy rule based modeling, a given input data undergoes the processes of fuzzification, application of fuzzy operators, implication, aggregation and defuzzification, before finally forming into output from the model [20]. Various steps involved in the design of fuzzy logic controllers have been discussed in literature [18].

As mentioned previously, present study describes development of a canal automation software based on the fuzzy control algorithm proposed in [16]. The fuzzy algorithm involves replacing (2) by a fuzzy rule based model while retaining (1) in its complete form. The fuzzy rule based model has been developed on fuzzification of a new mathematical model for dynamic wave velocity, obtained as:

$$(V_w)_m^t=\frac{(Discontinuity\ in\ gradient\ of\ Q)_m^t}{(Discontinuity\ in\ gradient\ of\ A)_m^t}=\frac{(\Delta \mathrm{Q})_m^t}{(\Delta \mathrm{A})_m^t} \qquad (3)$$

where, $(V_w)_m^t$ is the wave velocity at point m along the canal, at time t.

Discontinuity in gradient of a flow parameter (Q or A), at point m and at time t, has been determined as the difference between its gradient corresponding to the final steady condition (which the flow will attain under the effect of the perturbation) and its gradient corresponding to the condition at time t, both gradients being evaluated at point m. The fuzzy rule based model is developed on fuzzification of (3). While computing the dynamic wave velocity (V_w) at any node (i,j+1) in the computational domain (Fig. 2), the corresponding values of ΔA and ΔQ are determined approximately by adopting the following discretization:

$$\Delta \mathrm{A}_{\mathrm{i}}^{\mathrm{j+1}}=[A_{\mathrm{i+1}}^{\mathrm{F.S}}-A_{\mathrm{i}}^{\mathrm{F.S}}]-[A_{\mathrm{i+1}}^{\mathrm{j+1}}-A_{\mathrm{i}}^{\mathrm{j+1}}] \qquad (4)$$

$$\Delta \mathrm{Q}_{\mathrm{i}}^{\mathrm{j+1}}=[Q_{\mathrm{i+1}}^{\mathrm{F.S}}-Q_{\mathrm{i}}^{\mathrm{F.S}}]-[Q_{\mathrm{i+1}}^{\mathrm{j+1}}-Q_{\mathrm{i}}^{\mathrm{j+1}}] \qquad (5)$$

In (4) and (5), the notations with superscript "F.S" indicate the values of the flow parameters corresponding to the final steady state.

Equation (1) has been discretized at the node (i,j), in the x-t computational domain (Fig. 2), using the Vasiliev scheme [21], which is an unconditionally stable implicit finite difference scheme, as follows:

$$\frac{1}{\Delta t}\left[A_{\mathrm{i}}^{\mathrm{j+1}}-A_{\mathrm{i}}^{\mathrm{j}}\right]+\frac{1}{2\Delta x}\left[Q_{\mathrm{i+1}}^{\mathrm{j+1}}-Q_{\mathrm{i-1}}^{\mathrm{j+1}}\right]=0 \qquad (6)$$

where i = node number in the x – direction and j = node number in the t – direction.

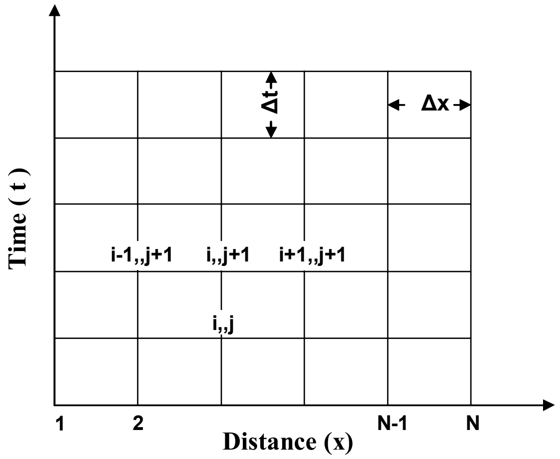


Figure 2. Computational domain for the fuzzy control algorithm: N = number of nodes in the x – direction, i = node number in the x – direction, j = node number in the t – direction.

Here, it is assumed that the control is achieved by rapid movement of check gates, which will result in the formation of gravity waves. If the wave velocity (V_w) corresponding to a certain inflow discharge is assumed to remain constant over the entire pool under consideration then a set of crisp rules can be written, based on (3) and applicable everywhere in the pool, as IF ΔQ is low THEN ΔA is low, IF ΔQ is medium THEN ΔA is medium, IF ΔQ is high THEN ΔA is high etc. where "low", "medium", "high" etc. are quantitative variables and ΔQ and ΔA indicate crisp numerical values. If the above assumption on V_w holds good only approximately then the rules become fuzzy, "low", "medium", "high" etc. become linguistic variables and ΔQ and ΔA indicate fuzzy subsets. The fuzzy rule based model is developed on this principle. A major advantage of the above fuzzy formulation is that it is developed on fuzzification of a mathematical model (3) relating the input and output variables. Hence the rules linking these variables can be accurately specified. Such a fuzzy model works in the best way and also the results are significantly better than the case where the model is developed using sample data [22]. The error, associated with the assumption on V_w remaining approximately constant in a pool, reduces as the wave amplitude reduces because as the amplitude reduces the wave resembles more closely a linear gravity wave which propagates downstream with minimum distortion, attenuation and change in velocity [23, 24, 25]. Hence it can be expected that the algorithmic error will be a minimum and consequently stable control at the downstream end will be achieved early in those cases where the final steady profile conform to uniform flow. Such a case is very difficult to control by other existing algorithms [26]. Software in the present context has been developed for this particular case.

III. DEVELOPMENT OF THE SOFTWARE

The software consists of three modules; the first one for tuning, the second one for fuzzy inference and the third one for computing the required inflow rate, at the upstream end, using the first and second modules. The Graphical User Interface of the software is developed using 'Qt' and the computational modules are developed using 'C' language. For the applications specified in the algorithm, only one to one correspondence is used. Therefore the fuzzy libraries and tool boxes given in [27, 28, 29] are not required as they are designed for a wide variety of applications.

A. *Tuning*

Tuning of the fuzzy control algorithm involves determination of the membership function parameters (core, support and boundary) of each fuzzy subset and construction of the fuzzy rule base. It is done by considering a fictitious case where the flow corresponding to maximum design discharge in the canal is uniform at the upstream end. For a base flow corresponding to the maximum design discharge and for various inflow discharges, respective values of ΔQ and ΔA are obtained by solving (3) along with the basic equation for gravity wave velocity [30] given below, which has been derived from momentum equation:

$$V_w = V_2 + \sqrt{\frac{gA_1}{A_2(A_1 - A_2)}(A_1\overline{Y_1} - A_2\overline{Y_2})} \qquad (7)$$

In (7), the subscript 1 indicates the upstream end from where the wave is produced and the subscript 2 indicates a neighboring section where the effect of the wave has not yet reached, V is flow velocity and $\overline{Y}$ is depth of centroid of the section below the free surface. The solution procedure is as follows. For a certain inflow discharge, ΔQ = $Q_1 - Q_2$ is a known value. Also, as flow is uniform at the upstream end, ΔA can be replaced by A_1-A_2 in (3). Thus in (3) and (7) the only unknowns are A_1 and $\overline{Y_1}$. But $\overline{Y_1}$ is a function of A_1. So for a trial value of A_1 corresponding value of $\overline{Y_1}$ is determined and applied in (7) to determine V_w. Same value of V_w should be obtained when the above value of A_1 is used in (3). If not, the above procedure is repeated with other trial values of A_1 until V_w obtained from the two equations are equal. An optimization technique has been adopted to reduce the number of trials required in tuning. It involves comparing the difference between V_w values, resulting from (3) and (7), during successive trials and the computation will proceed in that direction where the above difference will keep reducing. This is continued until the difference lies within some permitted tolerance. Once the correct value of A_1 is obtained, the value of ΔA= A_1-A_2 is calculated. Thus values of ΔQ and ΔA are obtained for the specified inflow discharge. This procedure is repeated for other inflow discharges also and respective values of ΔQ and ΔA are determined. These values are stored in two arrays. Each value in the input array (ΔQ) has a corresponding value in the output array (ΔA) also, both having degree of membership equal to one (core values of the fuzzy sets). Then by adopting the method of partitioning [18], a set of triangular membership functions and fuzzy rules are derived relating ΔQ and ΔA.

B. Fuzzy Inference

The software introduces a new procedure to compute the output (ΔA) for a given input (ΔQ), based on the Mamdani Implication method of inference [31] and the Centroid Method of defuzzification [18], as described below.

Suppose x_n is the given input data (ΔQ), with m_1 and m_2 as the corresponding degrees of membership in the neighboring membership functions shown in Fig. 3. Since these membership functions have been developed by method of partitioning, they will form isosceles triangles.

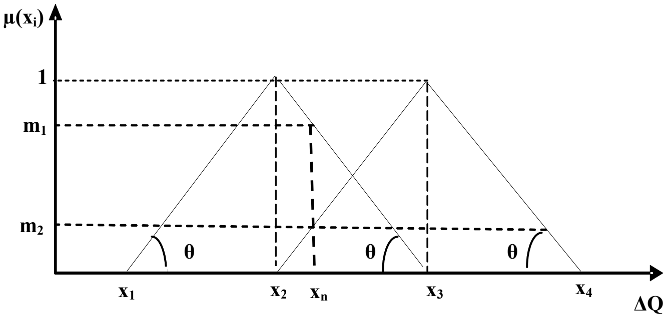


Figure 3. Fuzzification of input data

The value of x_2 nearest to x_n is found from the array of ΔQ. The values x_1, x_3 and x_4 are also obtained from the same array. Then the values of m_1 and m_2 are computed as follows:

$\tan\theta = 1/(x_2 - x_1)$. $m_1 = (x_3 - x_n)\tan\theta$. $m_2 = (x_n - x_2)\tan\theta$

There is one to one correspondence between the input and output fuzzy sets stored in the two arrays. Hence m_1 and m_2 can be easily mapped to the corresponding output membership functions, as shown in Fig. 4.

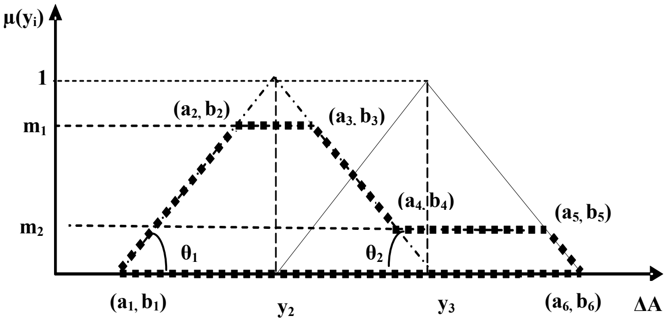


Figure 4. Computing the output membership function by implication.

The values of a_2, a_3, a_4 and a_5 shall be obtained as follows:

Calculation of a_2

$\tan\theta_1 = 1/(y_2 - a_1)$. Suppose $o_1 = a_2 - a_1$.
Then, $\tan\theta_1 = b_2/o1$. But $b_2 = m_1$.
Therefore, $o_1 = m_1/\tan\theta_1$. Hence, $a_2 = a_1 + o_1$.

Calculation of a_3

$\tan\theta_2 = 1/(y_3 - y_2)$. Suppose $o_2 = y_3 - a_3$.
Then, $\tan\theta_2 = b_3/o_2$. But $b_3 = m_1$
Therefore, $o_2 = m_1/\tan\theta_2$. Hence, $a_3 = y_3 - o_2$.

Calculation of a_4

Suppose $o_3 = y_3 - a_4$. Then, $\tan\theta_2 = b_4/o_3$.
But $b_4 = m_2$
Therefore, $o_3 = m_2/\tan\theta_2$. Hence, $a_4 = y_3 - o_3$
Calculation of a_5 is similar to that of a_4
The above calculations are for the case when $m_1 > m_2$. Calculations for the case where $m_1 < m_2$ is similar.

For the resulting polygon with coordinates as shown in the Fig. 5, the x-coordinate of the centroid , ΔA, which is the required output, is calculated using following equations [32].

$$P = 1/2\sum_{i=1}^{6}(a_i b_{i+1}) - (a_{i+1} b_i) \tag{8}$$

$$\Delta A = 1/(6P)\sum_{i=1}^{6}(a_i + a_{i+1}) \times (a_i b_{i+1} - a_{i+1} b_i) \tag{9}$$

This module performs only mathematical calculations. The computational cost is independent of any characteristics of the canal and hence the time and space complexity is constant.

C. Steps in Computation

1. Recovery time (Δt) is obtained as the downstream wave traveling time.
2. The canal reach is discretized into (N-1) sub-reaches using N nodes, as shown in Fig. 2; node 1 being at the upstream end and node N at the downstream end.
3. Computation starts at the last node N, assuming that *Q* and *h* at this node will be equal to their target values at the end of the time step (Δt).
4. *Q* and *h* at the nodes N-1, N-2 …,2, 1 are obtained successively, by application of (6) and the fuzzy rule based model. Note that *A* is a function of *h* depending on geometry of the canal.
5. The upstream gate is adjusted to release the discharge computed at node 1 for the recovery time period (Δt).
6. If the flow depth (*h*) at node N at the end of Δt deviate from its target value then repeat the above steps for the next Δt.
7. Otherwise stop.

The computation module performs a set of operations for each node in the pool. Hence the time complexity is linear with the number of nodes in the pool. But the number of nodes is not a parameter that takes very high values. Therefore the computational time of this module will not vary much with the number of nodes.

IV. SOFTWARE TESTING

The software is tested by comparing its outputs with that obtained using MATLAB fuzzy logic tool box [16]. The software has been applied to water level control problem in a fictitious canal with a single pool [15]. Schematic diagram of the canal is shown in Fig. 5. The canal length (L) is 5000 m, bottom width 5.0 m and side slope 1.0. Its bottom slope is

0.0003. Width of upstream and downstream rectangular gates is 5.0 m with discharge coefficient 0.75. There is a constant head reservoir at upstream end and another one at downstream end. Their respective water levels are 6.3 m and 2.8 m. Also bottom elevation of the pool at the upstream end is 1.5 m and at the downstream end is 0.0 m. An offtake canal is provided just upstream of the downstream gate. Target water depth at downstream end of the canal is 3.46 m, which corresponds to normal depth at a peak flow rate of 40 m^3/s.

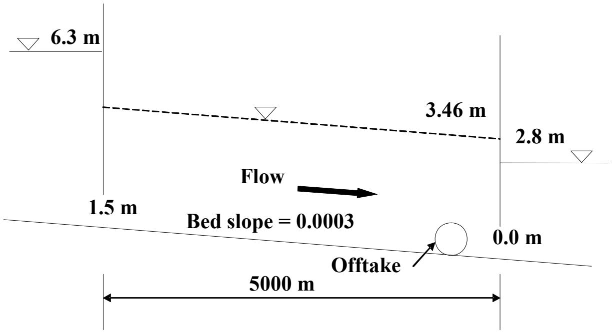


Figure 5. Schematic diagram of the single-pool canal used for testing the canal automation software.

For a base flow corresponding to the peak flow rate of 40 m^3/s, and for various inflow discharges ranging from 35 m^3/s to 45 m^3/s, corresponding values of ΔQ and ΔA are obtained and membership functions are developed by following the procedure explained under the section "Tuning". They are shown in Fig. 6. In the figure "L" implies "low" values and "H" implies "high" values. Higher the coefficient of "L" and "H" higher will be their gradation towards "low" and "high" values respectively. The fuzzy rules, connecting the antecedent ΔQ and the consequent ΔA, are derived in the form IF ΔQ is "a" THEN ΔA is "a" where "a" take values in ascending order from 9L to 9H (Fig. 6). Hence a total of eighteen rules are included in the fuzzy rule base.

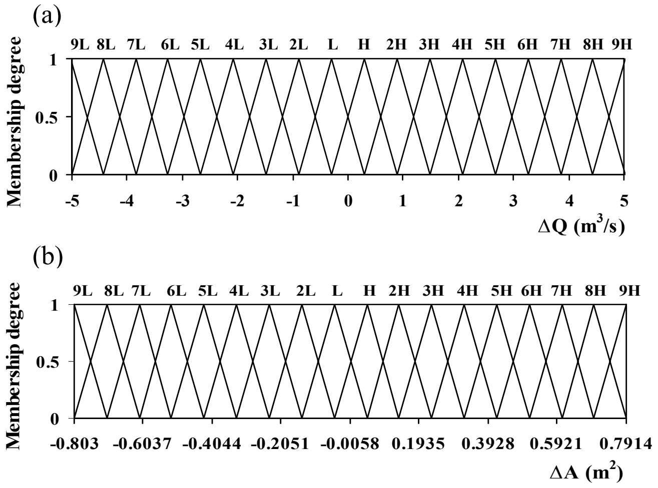


Figure 6. Membership functions of the fuzzy model for the single-pool canal: (a) input membership functions, (b) output membership functions.

Initially the flow is steady in the canal with a discharge of 20 m^3/s. There is no flow in the offtake canal and the entire flow is diverted to the downstream reservoir. Flow in the offtake canal is then increased from 0 to 20 m^3/s in 2 minutes. With such a disturbance, the response of the system is studied. The computational nodes are placed at a distance interval (Δx) of 100 m each. The recovery time (Δt) is taken as downstream wave traveling time of 797 s [15]. At every recovery time step, the upstream gate opening is adjusted, in accordance with the gate equation given in [15] so as to maintain the constant gate discharge output from the fuzzy control algorithm.

The simulation results include variation of the upstream gate opening with time (Fig. 7), development of water surface profile along the canal with time (Fig. 8), variation of the discharge with time at both upstream and downstream ends (Fig. 9) and variation of the change in water level with time at both these locations (Fig. 10). These results are comparable to those obtained using MATLAB fuzzy logic tool box [16] and hence it can be concluded that the software is giving accurate outputs.

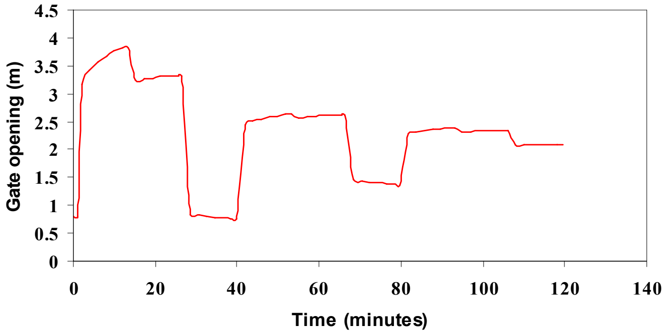


Figure 7. Variation of the upstream gate opening of the single-pool canal with time during control

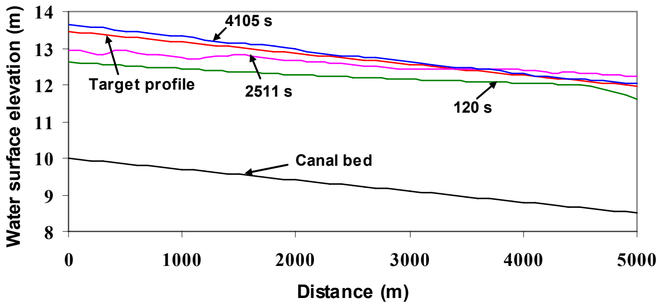


Figure 8. Development of water surface profiles with time in the single-pool canal during control

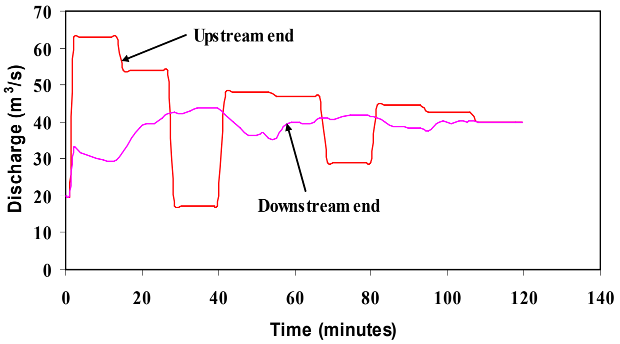


Figure 9. Variation of discharge with time, at upstream and downstream ends of the single-pool canal during control

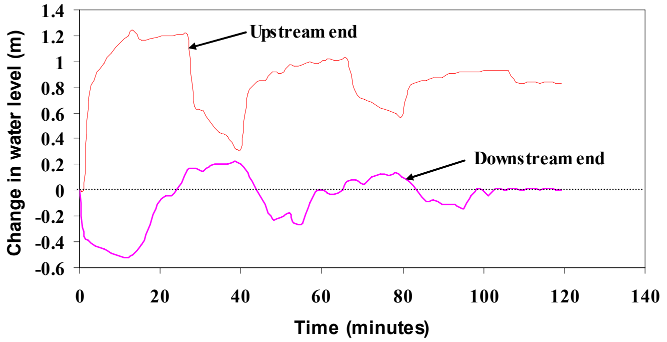


Figure 10. Variation of change in water level with time, at upstream and downstream ends of the single-pool canal during control

V. CONCLUSIONS

A software for automation of a single-pool canal, based on a fuzzy control algorithm, is presented in this study. The fuzzy algorithm was originally developed using MATLAB fuzzy logic tool box, which is not freely available. Thus the original work was unsuitable for common field applications. This problem has been overcome in the present study using free and open source packages. Additionally, existing canal control software are based on either control systems theory or dynamic wave model inversion, both require strong theoretical background for application. This difficulty does not arise for the present software. It is transparent and intuitive and can be easily applied by field engineers. Number of trials required for tuning the fuzzy model has been effectively reduced by including an optimization technique. Also, a new procedure has been introduced for fuzzy inference based on the Mamdani Implication method. The technical details of computation are abstracted from the end user and therefore the software is extremely user friendly. As the front end was designed using Qt and the computational modules uses 'C' language, only a minimum number of freely available packages are required to install the software. The software has been tested by applying it to a canal control problem reported in literature. The results obtained are comparable to those obtained using MATLAB fuzzy logic tool box, which proves that the software is giving accurate outputs.

ACKNOWLEDGMENT

The funding provided by CERD (Centre for Engineering Research and Development), Kerala, in carrying out this research is gratefully acknowledged.

REFERENCES

[1] A.J. Clemmens. "Editorial", Journal of Irrigation and Drainage Engineering, ASCE, vol. 124(1), 1998, pp. 1-2.

[2] P-O. Malaterre, D.C. Rogers and J. Schuurmans. "Classification of canal control algorithms", Journal of Irrigation and Drainage Engineering, ASCE, vol. 124(1), 1998, pp. 3-10.

[3] B.T. Wahlin and A.J. Clemmens. "Performance of historic downstream canal control algorithms on ASCE Test Canal 1", Journal of Irrigation and Drainage Engineering, ASCE, vol. 128(6), 2002, pp. 365-375.

[4] B.T. Wahlin. "Performance of model predictive control on ASCE Test Canal 1", Journal of Irrigation and Drainage Engineering, ASCE, vol. 130(3), 2004, pp. 227-238.

[5] A.J. Clemmens and J. Schuurmans. "Simple optimal downstream feedback canal controllers: theory", Journal of Irrigation and Drainage Engineering, ASCE, vol. 130(1), 2004, pp. 26-34.

[6] V.M. Ruiz-Carmona, A.J. Clemmens and J. Schuurmans. "Canal control algorithm formulations", Journal of Irrigation and Drainage Engineering, ASCE, vol. 124(1), 1998, pp. 31-39.

[7] V.T. Chow, D.R. Maidment and L.W.Mays. Applied Hydrology, McGraw-Hill International Editions. 1988.

[8] E.B. Wylie. "Control of transient free-surface flow", Journal of Hydraulic Division, ASCE, vol. 95(HY1), 1969, pp. 347-361.

[9] F. Liu, J. Feyen, P-O. Malaterre, J-P Baume and P. Kosuth. "Development and evaluation of canal automation algorithm CLIS", Journal of Irrigation and Drainage Engineering, ASCE, vol. 124(1), 1998, pp. 40-46.

[10] E. Bautista, T.S. Strelkoff, and A.J. Clemmens. "General characteristics of solutions to the open-channel flow, feed forward control problem", Journal of Irrigation and Drainage Engineering, ASCE, vol. 129(2), 2003, pp. 129-137.

[11] E. Bautista and A.J. Clemmens. "Volume compensation method for routing irrigation canal demand changes", Journal of Irrigation and Drainage Engineering, ASCE, vol. 131(6), 2005, pp. 494-503.

[12] G.Chevereau. Contribution a l'étude de la régulation dans les systèmes hydrauliques a surface libre, Ph.D. thesis, Institut National Polytechnique de Grenoble, Grenoble, France. 1991.

[13] F. Liu, J. Feyen and J. Berlamont. "Computation method for regulating unsteady flow in open channels", Journal of Irrigation and Drainage Engineering, ASCE, vol. 118(10), 1992, pp. 674-689.

[14] J.A. Cunge, F.M. Holly Jr and A.Verwey. Practical Aspects of Computational River Hydraulics, London: Pitman. 1980.

[15] F. Liu, J. Feyen and J. Berlamont. “Downstream control algorithm for Irrigation canals”, Journal of Irrigation and Drainage Engineering, ASCE, vol. 120(3), 1994, pp. 468-483.

[16] R. Gopakumar and P.P Mujumdar. “A fuzzy logic based dynamic wave model inversion algorithm for canal regulation”, Hydrological Processes, Vol. 23, No. 12, 2009. pp. 1739-1752.

[17] L.A. Zadeh. “Fuzzy Sets”, Information and Control, 8, 1965, 338-353.

[18] T.J. Ross. Fuzzy Logic with Engineering Applications, McGraw-Hill International Editions. 1995

[19] H.J. Zimmermann. Fuzzy Set Theory and Its Applications, Kluwer Academic Publishers. 1991.

[20] MATLAB. Fuzzy Logic Tool Box, The Math Works. 1995.

[21] J.A. Liggett and J.A Cunge. “Numerical methods of solution of the unsteady flow equations”, Unsteady flow in open channels, K Mahmood and V Yevjevich eds. Volume 1. Water Resources Publications. 1975.

[22] S.O. Russell and P.F. Campbell. “Reservoir operating rules with fuzzy programming”, Journal of Water Resources Planning and Management, ASCE, vol. 122(3), 1996, pp.165-170.

[23] M.G. Ferrick. “Simple wave and monoclinal wave models: River flow surge applications and implications”, Water Resources Research, vol. 41(W11402), 2005, pp. 1-19.

[24] V.M. Ponce and D.B. Simons. “Shallow wave propagation in open channel flow”, Journal of Hydraulic Division, ASCE, vol. 103(HY12), 1977, pp. 1461-1476.

[25] C.W-S. Tsai and B.C. Yen. “Linear analysis of shallow water wave propagation in open channels”, Journal of Engineering Mechanics, ASCE, vol. 127(5), 2001, pp. 459-472.

[26] A.J. Clemmens, T.F. Kacerek, B. Grawitz, W. Schuurmans.. Test cases for canal control algorithms. *Journal of Irrigation and Drainage Engineering*, ASCE, 124(1), 1998, pp. 23-30.

[27] FISLAB: the Fuzzy Inference tool-box for SCILAB. Ann. Univ. Tibiscus Comp. Sci. Series V, 2007, pp. 105-114

[28] Fuzzy Lite – A fuzzy logic library written in C++. http://code.google.com/p/fuzzy-lite, accessed on 02/02/2011

[29] E.L. Posselt , L.F.D.R.E Silva, R. Frozza and R. Fredimolz. “Qualitative Study of Software for Fuzzy Systems Simulation and Development”, Proceedings of the 11th WSEAS (World Scientific and Engineering Academy and Society) international conference on neural networks, evolutionary computing and Fuzzy systems, Stevens Point, Wisconsin, USA. 2010

[30] M.H. Chaudhry. Open-Channel Flow, Englewood Cliffs, N.J: Prentice-Hall. 1993.

[31] E.H. Mamdani and S. Assilian. “An experiment in linguistic synthesis with a fuzzy logic controller”, International Journal of Man-Machine Studies, vol. 7(1), 1975, pp.1-13.

[32] http://en.wikipedia.org/wiki/centroid, accessed on 30/5/2011