

APPLICATIONS DEVELOPMENT IN GREENHOUSE ENVIRONMENT

Alexandru Dumitrascu

Automatic Control and Computers Faculty, University „Politehnica” of Bucharest

Abstract: The actual climate in the world has an important impact in the entire agriculture field, especially in the sector of the culture of plants. This reason together with the global heating are determine the people to choose the greenhouse solution and, also, their specific applications. These applications are developed to monitoring and control the energy balance of the greenhouse climate. In the first part, this study focuses on the comparison between a classical method of the numeric algorithm PI controller and the intelligent control techniques represented by a fuzzy logic controller (FLC) with an efficient realization. Another objective of this paper is to describe the principles, methods and results of data compression employed when operating data transfers from environmental processes, which are the reasons why the data compression methods acquire a high importance for an efficient remote information transfer.

The practical part of this paper proposes some applications of greenhouse climate, developed into the LabVIEW environment that offers user a good graphical interface easy to use and understandable.

Keywords: control systems, intelligent techniques, greenhouse environment, data transfer, data compression.

1. INTRODUCTION

The requirements of the greenhouse environment are the following capabilities that need to be incorporated into a greenhouse facility in order to produce a healthy crop: 1) modification of greenhouse environment parameters of light, temperature, relative humidity, sun radiation and carbon dioxide, to within acceptable limits; 2) air movement through all foliage e.g. air speed not less than 0.3 meters per second; 3) ability to vent all greenhouse air in a controlled manner; 4) uniform crop conditions throughout the greenhouse space.

Tall greenhouses generally have more stable environments, lower maximum temperatures and less variability in climate (such as temperature and relative humidity). The advantages of increased height include more stable environmental conditions due to greater air mass, lower maximum air temperatures and less fluctuation in relative humidity - healthier crop environment. Some of the disadvantages of increased height include increased cost, increased side loads on structure due to wind, resulting heavier structural components and possibly increased shading of crop and increased heat loss due to large greenhouse covering surface area.

2. ENERGY BALANCE OF THE GREENHOUSE CLIMATE

High levels of relative humidity present serious problems in the management of greenhouse environments. In some cases, it is the high relative humidity in the atmosphere that creates the risk and in other cases; it is the actual wetness of the leaves and other plant tissue that is a consequence of the high relative humidity that creates the problem. The level of relative humidity can change quite rapidly in the greenhouse as a result of water being evaporated into the greenhouse atmosphere by plants (transpiration) or evaporation of irrigation and drainage water. Maintaining acceptable levels of relative humidity is therefore important. The most common requirement in many greenhouses is to reduce relative humidity and the most effective method is to remove or dump the moist air from the greenhouse. If natural ventilation is not adequate, then fans can be used. Control of the fans can be through a relative humidity sensor or a simpler approach is to operate the fans at regular intervals, for short periods, just long enough to exhaust all the moist air from within the building. Another approach is to use heaters to increase the air temperature sufficiently to reduce relative humidity to acceptable levels. This method can however be expensive in heating fuel costs.

3. GREENHOUSE THERMAL MODEL

To construct any control system, an appropriate building thermal model is needed. In [3], Gates reviewed some recent models for greenhouses which take the following dynamic equation (1) for interior air temperature:

$$\frac{dT}{dt} = \frac{1}{\rho C_p V} [q_{heater} + aSA_f] - \frac{\dot{V}}{V}(T - T_{out}) - \frac{UA_s}{\rho C_p V}(T - T_{out}) \quad (1)$$

where:

Af = floor area, m²

As = surface area, m²

A = building's solar heating efficiency (0.28)

Cp = specific heat of air, J (kg °C)⁻¹

q_{heater} = heater output, W

q_{internal} = net sensible heat from birds/animals and other sources, W

D = air density, kg m⁻³

S = solar irradiance, W m⁻²

T = interior air temperature, °C

T_{out} = outside air temperature, °C

U = heat transfer coefficient, W (m² °C)⁻¹

V = building volume, m³

V' = volumetric ventilation rate, m³s⁻¹

The equivalent block diagram of dynamic equation is given in the following figure that is represented in Simulink:

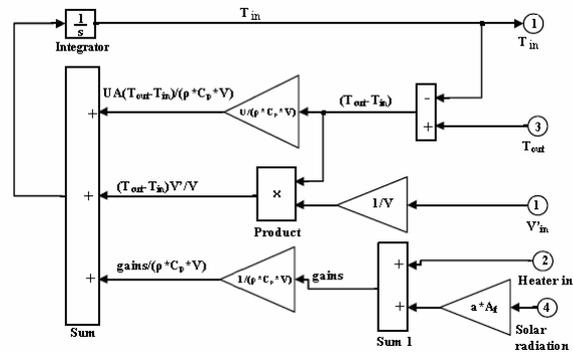


Fig. 1. Dynamic model for a greenhouse

The block diagram has 4 inputs (ventilation rate, V'; supplemental heat; outside temperature, T_{out} and solar irradiance. S) and a single output (inside temperature, T), with input-output ports labeled numerically.

4. THE CONVENTIONAL CONTROLLER

4.1. Numeric algorithm of the conventional controller

The conventional controller consists of temperature control system represented by a PI controller. The idea is to obtain the discrete forms of the process and controller.

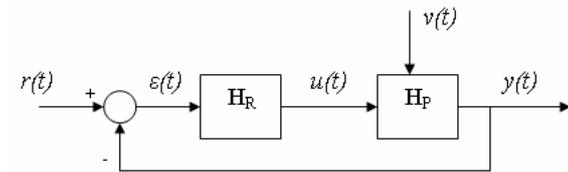


Fig. 2. The blocks of transfer functions

There is a 1st order process: $H_p(s) = \frac{k_p}{T_p s + 1}$, and

PI type controller: $\frac{T_i s + 1}{s}$, where $T_i = T_p \Rightarrow H_R(s) = \frac{T_p s + 1}{s}$.

The stair perturbations are rejected because one pole in origin is. The time constant of the process, T_p , is computed according to the process parameters.

Using the Tustin method $s = \frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}}$ with

$T=1$, we obtain:

$$H_p(z^{-1}) = \frac{k_p}{2T_p \frac{1-z^{-1}}{1+z^{-1}} + 1} = \frac{k_p(1+z^{-1})}{2T_p(1-z^{-1}) + 1 + z^{-1}}$$

$$\frac{k_p + k_p z^{-1}}{2T_p - 2T_p z^{-1} + 1 + z^{-1}} = \frac{k_p + k_p z^{-1}}{2T_p + 1 + z^{-1}(1 - 2T_p)} \quad (2)$$

$$H_R(z^{-1}) = \frac{2T_p \frac{1-z^{-1}}{1+z^{-1}} + 1}{2 \frac{1-z^{-1}}{1+z^{-1}}} = \frac{2T_p(1-z^{-1}) + 1 + z^{-1}}{2 - 2z^{-1}}$$

$$\frac{2T_p + 1 + z^{-1}(1 - 2T_p)}{2 - 2z^{-1}} \quad (3)$$

The transfer function of the process in the close loop is:

$$H_0 = \frac{H_R H_p}{H_R H_p + 1} \quad (4)$$

$$H_R H_p = \frac{2(T_p + 1) + z^{-1}(1 - 2T_p)}{2 - 2z^{-1}} * \frac{k_p + k_p z^{-1}}{2T_p + 1 + z^{-1}(1 - 2T_p)} =$$

$$\frac{k_p(2T_p + 1) + k_p z^{-1}(1 - 2T_p + 2T_p + 1) + k_p(1 - 2T_p)z^{-2}}{2(2T_p + 1) + 2z^{-1}(1 - 2T_p - 2T_p - 1) - 2(1 - 2T_p)z^{-2}} =$$

$$\frac{k_p(2T_p + 1) + 2k_p z^{-1} + k_p(1 - 2T_p)z^{-2}}{2(2T_p + 1) - 2z^{-1}(4T_p) - 2(1 - 2T_p)z^{-2}} \quad (5)$$

$$H_0(z^{-1}) = \frac{k_p(2T_p + 1) + 2k_p z^{-1} + k_p(1 - 2T_p)z^{-2}}{k_p(2T_p + 1) + 2k_p z^{-1} + k_p(1 - 2T_p)z^{-2} + 2(2T_p + 1) - 8T_p - 2(1 - 2T_p)z^{-2}} =$$

$$\frac{k_p(2T_p + 1) + 2k_p z^{-1} + k_p(1 - 2T_p)z^{-2}}{(k_p + 2)(2T_p + 1) - 2z^{-1}(4T_p - k_p) + z^{-2}(k_p - 2)(1 - 2T_p)} \quad (6)$$

$$H_0(z^{-1}) = \frac{Y(z^{-1})}{R(z^{-1})} \quad (7)$$

The output (8) is:

$$y(k) = \frac{1}{(k_p + 2)(2T_p + 1)} \left(k_p(2T_p + 1)r(k) + 2k_p(1 - 2T_p)r(k-2) + (2(4T_p - k_p))r(k-1) - (k_p - 2)(1 - 2T_p)r(k-2) \right) \quad (8)$$

4.2. The application of the conventional controller

The process was simulated by the implementatin in LabVIEW environment of the $y(k)$ recurrence, using a *while* loop and *shift registers* to keep the previous values of $r(k)$ si $y(k)$.

The block diagram of the implemented algorithm is represented in the following figure.

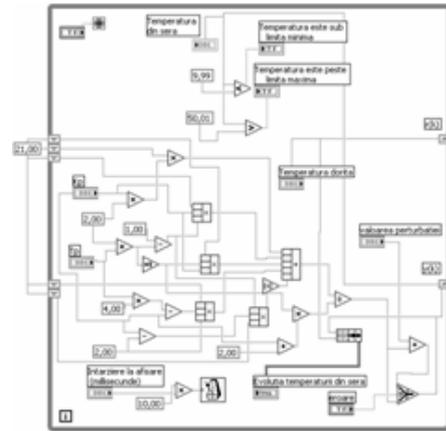


Fig. 3. The block diagram of PI controller

At every step, the current temperature value inside the greenhouse is compared with the admissible maximum and minimum values. In the case which the current value isn't inside the interval (min. val. ... max. val.), a warning element (LED) is switch on.

Using a digital control, the user can change the reference value (the value of the desirable temperature). The inside temperature is showed by another digital thermometer, near to the graphic. Also, the graphical interface contains the process parameters (K_p , T_p) and a delay of vizualization.

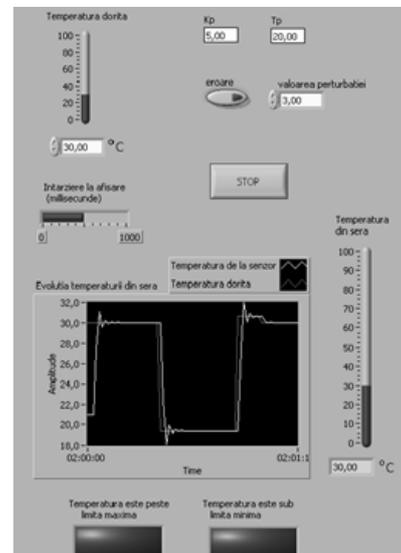


Fig. 4. The graphical interface of PI controller

The graphic shows the reference value with red line, and output value with white line. The reference value must be inside of interval (min. val. ... max. val.). The user can introduce an error by pressing a button and using a digital control to type his value. The application is switched off by pressing the STOP button.

The traditional methods to modeling and control a process are used when the user may determine the relations between the system's variables. The intelligent techniques are an alternative way to control the complex systems.

5. FUZZY LOGIC CONTROLLER (FLC) MODEL

The main complexity of implementing a FLC involves incorporation of the discrete nature of the staged heating/ventilating equipment. Is not necessary a mathematic model in the process of the temperature control using FLC, but must know only the relations between input and output variables of the process.

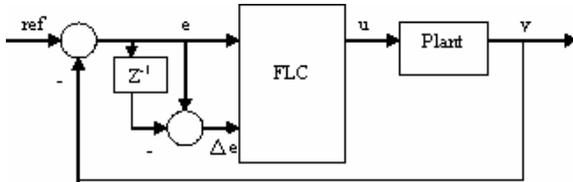


Fig. 5. Control loop of the FLC controller

The fuzzy variables not describe a numeric data but use a function which is tuning dial between 0 and 1. The fuzzy variable operations and fuzzy rules not begin from the precise models of the process, but begin from understanding of the physical phenomenon, like:

IF (high temperature) THEN (small command)

5.1. The steps in the design of the FLC

The fuzzy logic process is implemented with 5 operations:

1. fuzzification of numerical inputs (control error) using input membership functions;
2. apply fuzzy operators to the linguistic variables of the rule base;
3. inference - perform implication;
4. aggregate each rule's output into a common fuzzy set;
5. defuzzification of the aggregate fuzzy set to obtain control output using a center of gravity output rounded to the nearest integer.

Fuzzification and defuzzification – involve mapping the fuzzy variables of interest to “crisp” values used by the control system. Fuzzification translates a numeric value for the error (e) or error rate (Δe) in linguistic values. Defuzzification takes the fuzzy output of the

rules base and generates a “crisp” values used as input to control the plant.

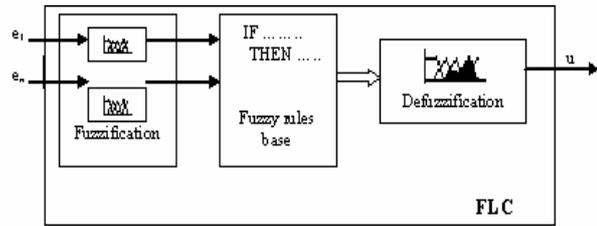


Fig. 6. The FLC block diagram

The common method of defuzzification is center of gravity method, which determines that value which divides the surface S in two well-balanced areas, S₁ and S₂, like the follows equation (9) and figure 7:

$$x_r = C_0 A(x) = \frac{\int x \cdot \mu(x) \cdot dx}{\int \mu(x) \cdot dx} \approx \frac{\sum_i x_i \mu(x_i)}{\sum_i \mu(x_i)} \quad (9)$$

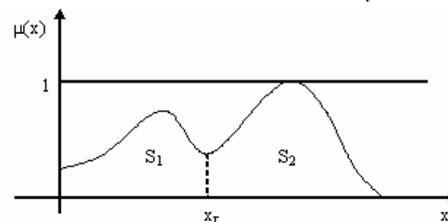


Fig. 7. The defuzzification method

A rule base (Table 1) maps linguistic inputs to outputs and the fuzzy process quantifies these actions.

The connection between inputs and outputs, both of which are "crisp" values, is made via the linguistic transformation of inputs using input membership functions, implication and aggregation using the rule base, and defuzzification of the linguistic output to a numerical value representing stage of ventilation.

Table 1. The rules base of FLC

Control error	Consequence
NB	heat2
NS	heat1
Z	no-change
PS	low cooling
PM	medium cooling
PB	high cooling

Input membership functions (Fig. 8) for the control error use 6 linguistic variables to apportion over the range of the error (NB, NS, Z, PS, PM and PB). For example, NB refers to a Negative Big difference between inside temperature and reference value, i.e. it is much colder than desired in the building. Each possible linguistic value of input control error is assigned a consequential action; for example if control error is PB then control action is "cooling-high".

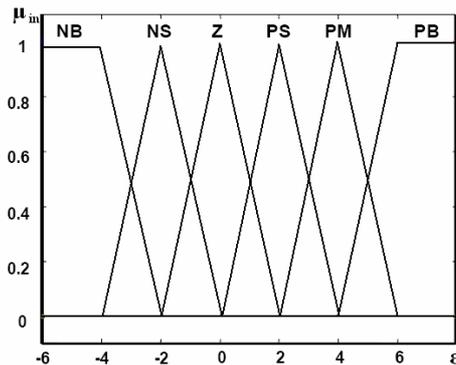


Fig. 8. Input membership function

The output membership functions (Fig. 9) perform the process of processing rules, and aggregating a response to provide a crisp output command. The output membership functions chosen in this application consist of two levels of heat (heat 2, heat 1) that may represent two different supplemental heating levels, with heat 2 being greater than heat 1. Working from cold to hot, the next four membership grades are: "no-change", "low cooling", "medium cooling" and "high cooling". These levels cover a range of possible heating/ventilation stages.

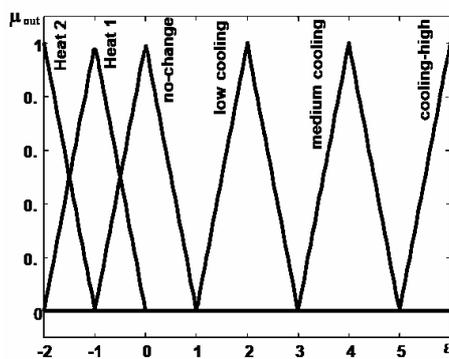


Fig. 9. Output membership function

5.2. The application of FLC controller

The fuzzy systems are more flexible than conventional systems because the change of the

command rules is realized by adding of the fuzzy variables. The control system with the fuzzy logic controller (FLC) is robust, with good performance in large interval of variables. But the disadvantage is the hard design of the FLC for many inputs of the controller (case of the multivariable processes).

The following application designed using LabVIEW environment is a control system of temperature and humidity values inside of greenhouse climate. The application is implemented with a fuzzy logic controller (FLC). Supposedly, the values are received from ten sensors installed in greenhouse. The actuator is represented by air cooler device that works with six levels. The FLC is implemented using a While loop with a Script Matlab, where the fuzzy logic part is written.

The Front Panel, which represents the user interface, in the follow figure is:

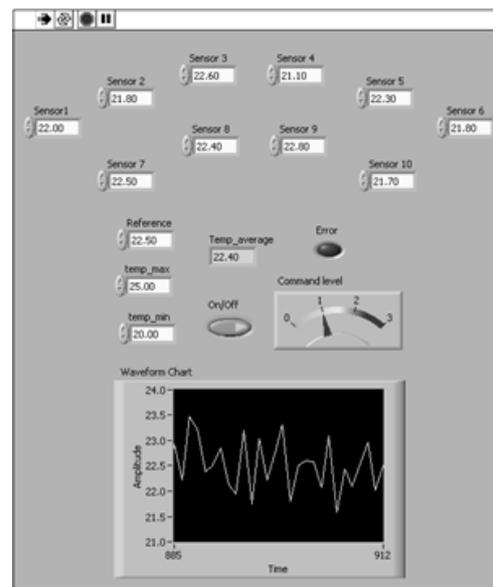


Fig. 10. Front Panel of the temperature and humidity control system

The program computes the average value of the greenhouse climate, and then the FLC receives the follow inputs: the average of the temperatures, the reference (eligible value), and the maximum and minimum temperature values. If the reference value isn't situated between the limits, than an error LED is switch on. The FLC computes the error represented by the difference between reference and current temperature and gives the command to the actuator. Therefore, the output of FLC is represented by the following command levels: heat2, heat1, no-change, low cooling, medium cooling, and high

cooling. On the graphic the user may observe the current temperature values, which are inside of the desirable interval.

The Block Diagram of this application the follow figure is:

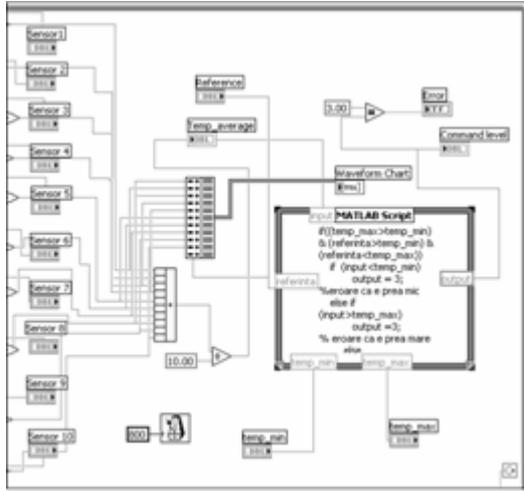


Fig. 11. Block Diagram of the temperature and humidity control system

The interface between the Matlab script and the LabVIEW application is very simple because the user has the possibility to choose the Matlab script structure from the LabVIEW tools palette, put the text code inside the structure and fix the inputs and outputs on the border of structure. When the user run the LabVIEW application, the Matlab program also run automatically and begins to compute the code of Matlab script.

6. THE APPLICATION OF DATA ACQUISITION AND TRANSFER

6.1. The application of data acquisition

The other application realizes the acquisition of following values: temperature, humidity, fluid level, and CO₂ density values from sensors to process computer. With other small LabVIEW application the data transfer is realized.

The acquisition application is a Matlab application. In this interface the user has the possibility to switch between temperature, humidity, fluid level and CO₂ density graphs using a pop-up button (Fig. 12).

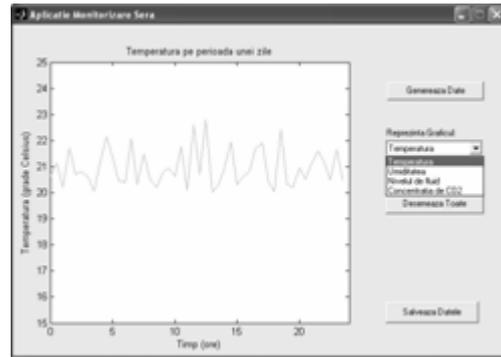


Fig. 12. The acquisition application

Another option is to have all four graphics in the same time, pressing to the *Design all* button (Fig. 13). With *Save data* button a data file is created and prepared to be send.

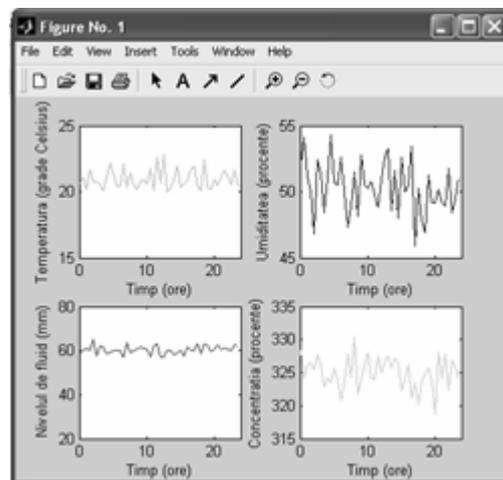


Fig. 13. Design all four graphics

6.2. The application of data transfer

The development of Internet applications and the possibility to use data acquisition instruments connected to a PC allows the users to make a central system of measurements near to the process and use it to transfer data towards every place where users need.

Using a TCP/IP connection between the computers (that includes a Server part and a Client part) [4], it is possible to develop a system of subVIs using LabVIEW environment. This works like a Server-Client structure, allowing the users to perform data acquisition and transfer. Usually, data are transferred to a different place to be processed by a monitoring system [5].

The software application described hereafter integrates two parts. The first part is represented by the LabVIEW environment that helps to perform data acquisition, saving and transfer.

The TCP/IP connection uses a server small application which is the following diagram.

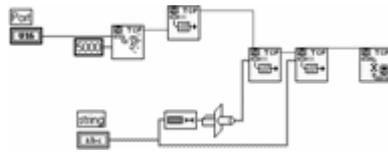


Fig. 14. Data server structure

The server gives the result of a request on its ports and then a client that reads the port takes the result.

The server blocks are as follows:

- *String* is the numeric field to send.
- *Port* is the communication port where the data are received.
- *TCP Listen* checks for client requests.
- *TCP Write* writes the data on the communication port.
- *TCP Close* closes the communication.

The client diagram is drawn in Figure 15.

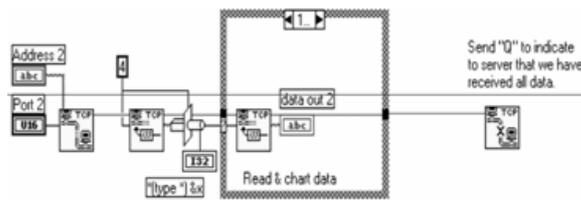


Fig. 15. Data Client structure

The blocks are here as follows:

- *Address* is the address of server, in numeric format.
- *Port2* is the port of communication where the data are stored (if the data are available) ;
- *TCP Read* reads the data if they arrived to the port of communication.

In Figure 16 it is illustrated how the capture of data in the *.txt files type is realized. The file resides in the same place where as the instrumentation for data acquisition.

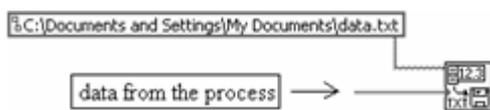


Fig. 16. The structure of Write block charged to capture the data

The next figure (Fig. 17) shows an example of *.txt file, created with the *Write To Spreadsheet File.vi* block. The numerical values are encoded by digits, using the ASCII table.

8.338	35.543	39.171	9.523	12.513	37.709	30.473	9.651
27.111	10.239	18.220	45.291	35.048	30.973	43.922	16.298
16.301	30.335	46.735	26.623	32.470	13.091	44.750	33.549
43.960	39.263	43.110	15.507	49.270	28.228	6.590	14.605
40.272	3.673	2.575	42.655	1.455	44.692	39.910	11.249
48.473	30.925	10.717	33.708	31.531	17.735	37.987	23.973
27.108	46.358	13.912	11.015	37.940	35.435	14.062	12.547
29.426	23.415	34.495	39.110	49.269	35.517	0.669	36.409
23.858	48.801	25.295	10.317	13.660	47.488	7.149	45.596
27.269	8.791	35.946	32.139	48.678	42.032	4.480	14.644
30.795	19.192	47.958	43.585	10.999	15.765	16.725	33.129
11.354	35.516	23.230	17.834	5.878	12.900	37.943	12.066
39.374	41.797	37.358	43.891	11.550	7.352	5.675	1.932
20.817	9.331	24.756	22.920	21.848	47.719	42.665	37.785
29.713	35.055	26.567	33.479	5.960	12.584	26.238	16.759
7.363	19.204	24.855	49.481	16.155	11.572	1.641	34.503
44.255	26.213	19.522	0.811	11.851	16.240	17.026	10.210
22.274	27.540	11.132	27.796	2.401	6.799	48.437	21.824

Fig. 17. An example of data files in ASCII format

6.3. The application of data compression

After the data set has been saved into the *data.txt* file, the user may use the second application that makes the data compression. This application is programmed within C-Builder programming environment and has a user friendly interface. The main window of the interface is illustrated by the snapshot in Figure 18.

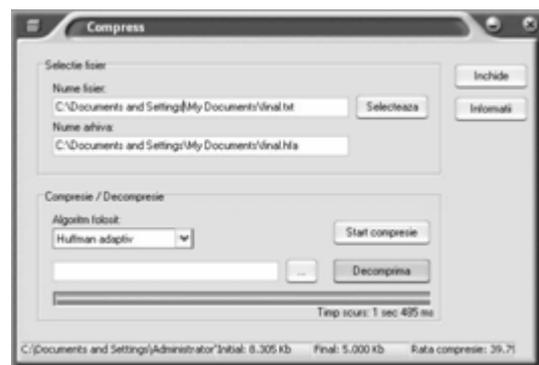
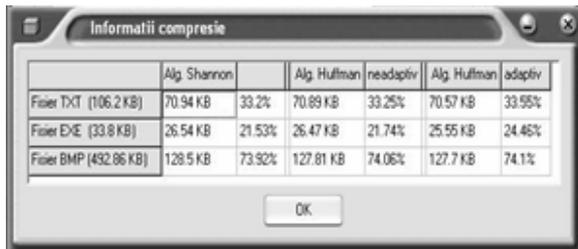


Fig. 18. The main window of data compress utility interface

The interface offers the user the following options:

- select the path of file to be compressed by pressing the button “*Selecteaza*” (*Browse*); the selected path is shown in the first bar; the second bar shows the path and the file name of the compressed data; the path is the same as for the original (uncompressed) file; the file extension is *.shf* for Shannon-Fano method, *.hfn* for classical Huffman method and *.hfa* for adaptive Huffman method;
- select one of the 3 compression methods by pressing the arrow on the menu entitled “*Algoritm folosit*” (Algorithm to use);
- the button “*Start compresie*” starts the compression procedure;
- the button “*Decomprima*” starts the decompression procedure;

- display the elapsed time for compression/decompression process;
- display the initial and final dimensions of files [Kb] and the compression rate [%] on the bottom bar;
- the button “*Informatii*” (*Information*) allows the user to see the results of compression process and to perform a comparison between the three algorithms for three types of files: *.txt*, *.exe* and *.bmp*, as illustrated in Figure 19;
- the button “*Inchide*” (*Close*) quits the program.



	Alg Shannon		Alg Huffman	neadaptiv	Alg Huffman	adaptiv
Fișier TXT (106.2 KB)	70.94 KB	33.2%	70.69 KB	33.25%	70.57 KB	33.95%
Fișier EXE (33.6 KB)	26.54 KB	21.53%	26.47 KB	21.74%	25.55 KB	24.46%
Fișier BMP (492.86 KB)	128.5 KB	73.92%	127.81 KB	74.06%	127.7 KB	74.1%

Fig. 19. Comparison on compression methods window

7. CONCLUSION

A principal advantage of FLC is to provide the building operator with a preferential input to balance energy use and control precision. Conventional controllers provide this to a limited extent only, by using variable differential temperatures between stages, and cannot approach a zero steady-state control error because there is no means for implementing an integrator into the conventional controller. Conventional PI controllers are capable of removing steady-state error, but require tuning for each type of facility, and re-tuning if equipment selection changes in a given facility. Further, PI controllers cannot be readily adjusted to balance energy use. In contrast, the FLC design presented in this paper can be readily incorporated into existing facilities, and the logic is scaleable over a variety of building and equipment sizes. The FLC appears to retain the robustness and flexibility of the conventional controller, with enhanced control features

including recovery from step and diurnal disturbances.

Secondly, the signals and data processing methods are more and more embraced by specialists that develop applications in various fields. For example, condition monitoring of an environmental process is often based on a large amount of hybrid data regarding the overall behavior of the observed part of nature. The last application intended to show that communication between the monitoring system and the environmental system can more efficiently be performed with the help of data acquisition, transfer, monitoring and control methods.

REFERENCES

- [1] Jeffrey Y. B. LabVIEW Programming, Data Acquisition and Analysis. Prentice Hall PTR. 2000.
- [2] Maier, V. and Maier, C.D. LabVIEW in calitatea energiei electrice. Editura Albastra. Cluj-Napoca. 2000.
- [3] Gates, R.S., K. Chao and N. Sigrimis. 1998. Design parameters for fuzzy-based control of agricultural ventilation systems. In: Control Applications and Ergonomics in Agriculture, First Workshop on Intelligent Control for Agricultural Application. International Federation of Automatic Control, IFAC Publications, Elsevier Science Ltd., Oxford, UK. pp 137-142.
- [4] Pallapa V., Sunilkumar S. M. Communication Protocol Engineering. Prentice-Hall. 2004.
- [5] Radescu R. Internet communication equipment and protocols. Editura Matrix Rom. 2003.
- [6] Alexandru Dumitrascu, Hangiu Mihai, Dan Stefanoiu. Data compression and transfer within environmental processes. CSCS15 International Conference, Vol. 1, Bucharest 2005.
- [7] Stefanoiu, D. (2003). *Compresia datelor*. Editura Printech, Bucuresti.