MODELING, SIMULATION AND VALDATION OF THE HYDROPOWER UNITS DYNAMICS BY CO-SIMULATION

Radu PUHALSCHI[1], Nicolae VASILIU[2], Constantin CĂLINOIU[1]
[1]University „Politehnica" of Bucharest
[2]Correspondent member of the Academy of Technical Sciences in Romania

Rezumat. Lucrarea prezintă cercetările teoretice și experimentale întreprinse de autori în domeniul fenomenelor tranzitorii din conductele centralelor hidroelectrice de mare cădere, utilizate pentru reglarea frecvenței în sistemele energetice, îndeosebi în contextul integrării noilor surse de energie (eoliană și fotovoltaică), greu predictibile. Fenomenul de șoc hidraulic are o influență considerabilă asupra stabilității și securității oricărui sistem hidraulic ce include conducte prin care circulă fluid sub presiune. Înțelegerea efectelor acestui fenomen prin modelare și simulare constituie o etapă vitală în procesul de concepție și exploatare eficientă și sigură a oricărui sistem hidraulic. Dezvoltarea rapidă a calculatoarelor industriale, a sistemelor de operare de timp real și a limbajelor de simulare numerică au permis o predicție mai bună a propagării undelor de presiune generate de sistemele de reglare a debitului, prin utilizarea unor metode precise, dar care necesită un volum mare de calcule. Autorii au elaborat o analiza comparativă de eficiență a limbajelor de simulare moderne: Matlab, LabView şi AMESim. Simularea unor fenomene tranzitorii electrohidraulice complexe a dovedit că simularea combinată cu Matlab şi LabView este cea mai flexibilă. Co-simularea cu aceste medii necesită utilizarea modulului MathScript din LabView care permite introducerea unui cod Matlab în diagrama de blocuri LabView. Aceasta abordare oferă multiple avantaje procedurale: ca orice limbaj de tip *text*, Matlab simplifică esențial implementarea ecuațiilor complexe specifice modelării fenomenelor tranzitorii, în timp ce LabVIEW, împreună cu soluțiile hardware oferite de National Instruments prin familiile PXI şi CompactRIO permit o dezvoltare liniară a interfeței grafice și facilitează integrarea modelului șocului hidraulic în modele complexe de simulare în timp real (RTS) sau cu Hardware-in-the-Loop ale turbinelor hidraulice cu dublu reglaj.

Cuvinte cheie: şoc hidraulic, cosimulare, Matlab, LabVIEW.

Abstract. The paper presents the authors' theoretical and experimental research in the field of numerical simulation of fluid transients from the pipelines of the high head hydropower units, used for frequency control in large power systems including wind and solar unpredictable energy sources. Water hammer is a dangerous phenomenon that strongly influences the stability and the safe operation of any hydraulic system that includes pipelines containing fluid under pressure. Therefore, understanding its effects through proper modeling and simulation is a vital step in the process of designing effective and safe hydraulic systems. Recent advances in the fields of both hardware performance and simulation languages have allowed more accurate simulation of the pressure waves occurring in the pipes, by employing methods of great precision like the characteristic method., but with a very high volume of calculus. The authors have performed comparative analysis of multiple simulation software solutions for modeling water hammer, including Matlab, LabView and AMESim. The optimal solution has been found to be the co-simulation between Matlab and LabView, by the aid of MathScript LabView module, which allows implementation of Matlab code directly into the LabView block diagram. This approach offers multiple advantages: as a text-based language, Matlab deals very well with the complex equations involved in fluid transients simulation, while LabView, together with the hardware solutions offered by National Instruments by the PXI and CompactRIO families allows easy development of graphical interfaces and facilitates the integration of the water hammer model into more complex real-time and HIL hydropower unit simulation models.

Keywords: waterhammer, co-simulation, Matlab, LabVIEW.

1. INTRODUCTION

The phenomenon known as water hammer represents the sudden pressure rise in a pipeline in the form of a pressure wave caused by a sudden change in the rate of flow in the line (Fig. 1). Overpressures generated by water hammer are often quite high, and failing to account for such effects in the design process of a hydraulic system may lead to significant performance drops during transient working conditions and in some case even significant damage to the system.

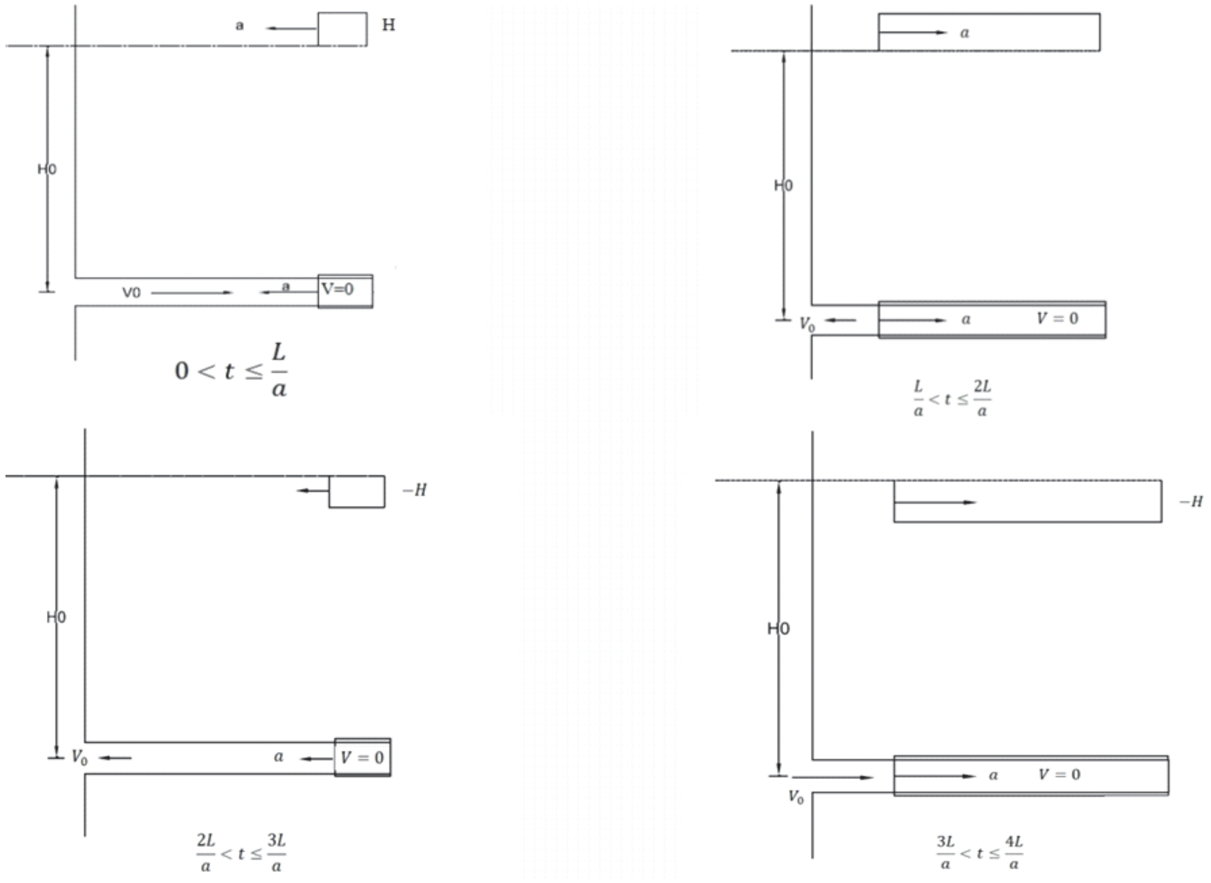

a
H
H0
V0
a
V=0
$0 < t \le \frac{L}{a}$
a
H0
V_0
a
$V = 0$
$\frac{L}{a} < t \le \frac{2L}{a}$
$-H$
H0
V_0
a
$V = 0$
$\frac{2L}{a} < t \le \frac{3L}{a}$
$-H$
H0
a
$V = 0$
V_0
$\frac{3L}{a} < t \le \frac{4L}{a}$

Fig. 1. Waterhammer.

Therefore, the authors present in this paper several solutions that they have developed for simulating waterhammer in the pipes of high-head hydropower units, with the final goal of exploring new methods, especially new control algorithms for the by-pass valve, for reducing the effects of this phenomenon.

2. MATHEMATICAL MODEL AND NUMERICAL SOLUTION

At the foundation of every correct simulation stands an accurate mathematical model. The waterhammer phenomenon has been studied theoretically for a long time, and the mathematical model is well-known, detailed in numerous specialty papers. As such, the authors consider the deduction of waterhammer equations beyond the scope of this paper. For Matlab and LabVIEW implementation, the starting point has been one of the common simplified forms in which the two waterhammer equations (the unsteady momentum equation applied to a section of the pipe and the continuity equation) can be found in literature (Wiley and Streeter 1983):

As it can be seen, both equations are partial differential equations and therefore cannot be solved algebraically in a general case. There are however several numerical methods that allow calculating solutions for particular cases of these equations with enough accuracy for most engineering problems.

The authors have considered two numerical methods for solving the waterhammer equations: the characteristic method and the quadripole method (T. J. Viersma). The first was chosen for implementation because the quadripole method has proven inadequate for modelling large disturbances that occur in case of hydropower unit unload.

The characteristic method turns the two partial differential equations into two pairs of ordinal differential equations (usually called and), that are solved numerically with ease, allowing the calculation of the two defining parameters of the waterhammer (pressure and flow) across an evenly spaced grid in the *x*-*t* (distance-time) plane (Fig. 2).

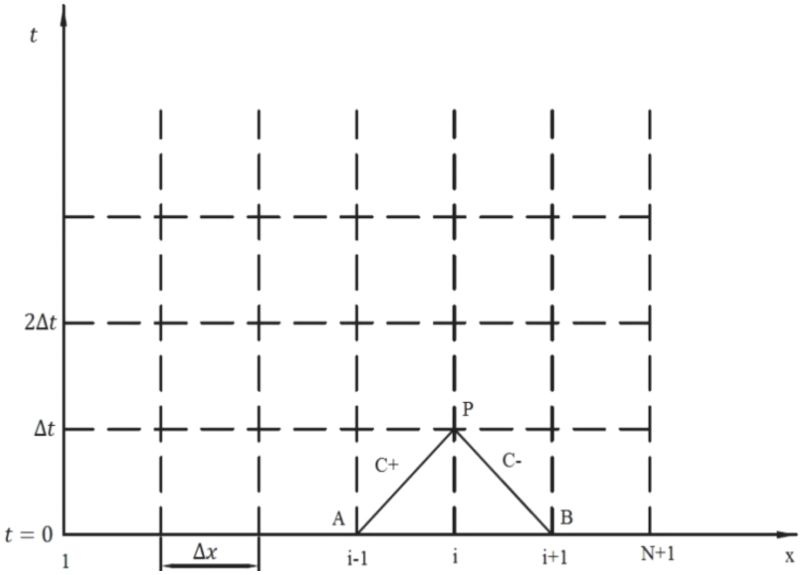


Fig. 2. The characteristics grid.

The first stage of implementing this numerical solution in the chosen simulation software has used some of the simplest boundary conditions available: tank modeled as constant pressure source at the upstream end and valve closing after a given law at the downstream end.

3. MATLAB IMPLEMENTATION

Matlab was the authors' first choice for implementing the characteristics method. As a text-based simulation and numerical computing environment, Matlab is well suited for handling large equations.

Graphical environments, like Simulink or LabVIEW, despite being easier to use in general, have difficulties in handling tasks such as the one at hand (numerical solving of large amounts of equations). In a graphical programming environment, even an equation as simple as '*a* + *b* = *c*' requires at least four blocks (the three variables and the summing operation) and the respective connection lines. It is easy to extrapolate how large and complex would a program built to implement the method presented above be.

With Matlab, the authors obtained a compact (68 lines) as well as accurate application for computing waterhammer in a pipe. The results (Fig. 3) have been validated using the vast amounts of experimental data a research collective lead by Prof. N. Vasiliu collected during the design of a new generation of speed governors for the Râmnicu-Vâlcea hydropower plant.

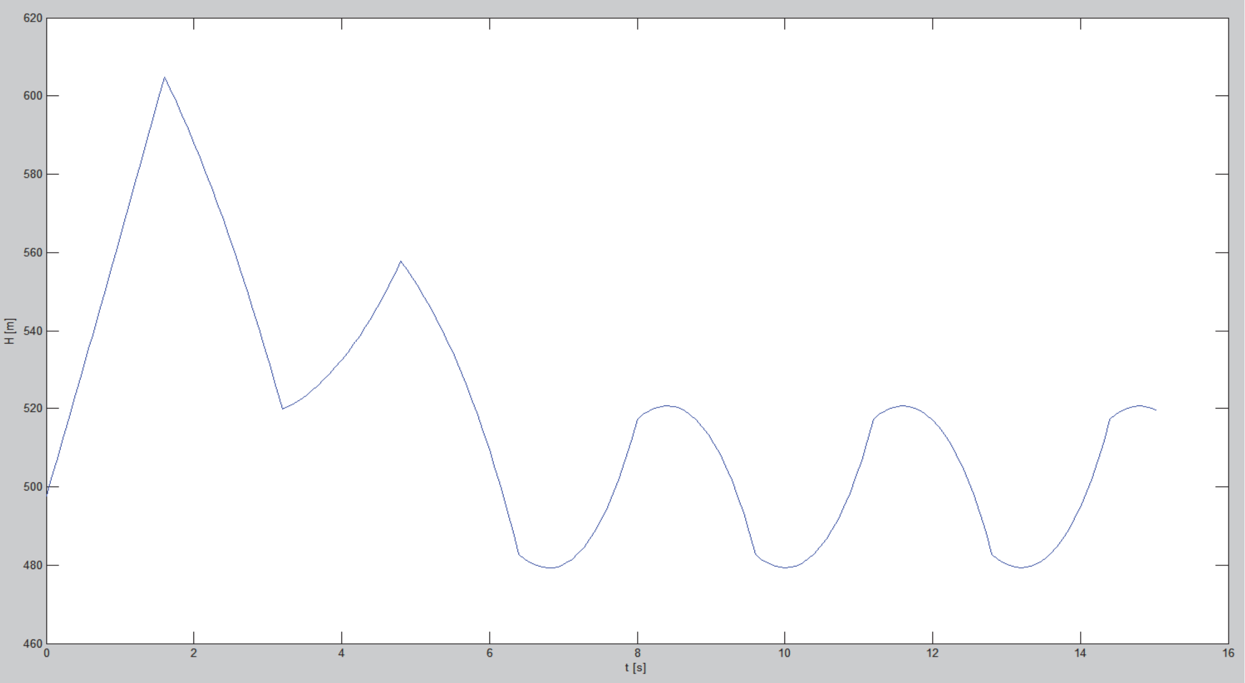


Fig. 3. Typical pressure variation in a hydropower unit pipe during unload.

Although the Matlab program provides accurate and reliable results, several shortcomings in regard to planned future development.

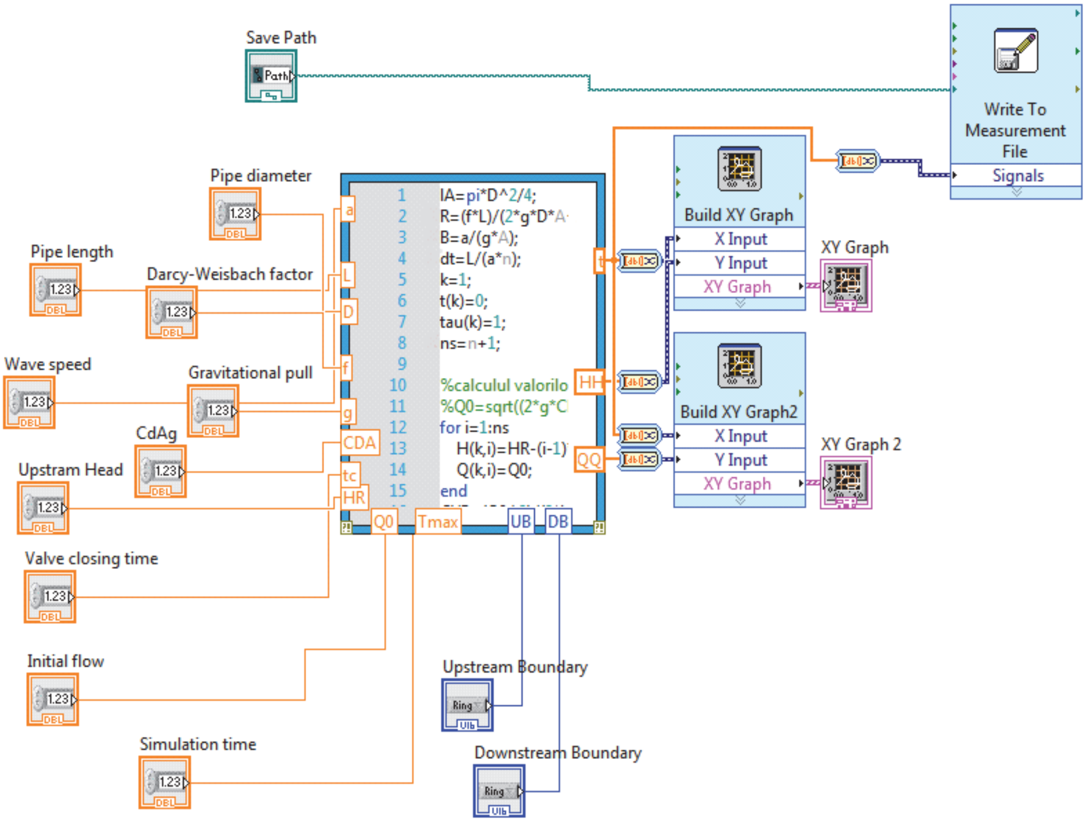
First of all, as a simple, text-based program, any modification that needs to be made, even something as simple as changing boundary conditions or simulation parameters has to be done directly inside the code. This means that any user that needs to do such changes needs to have access to the program source code (it’s not possible to distribute the program to students as an executable file for example) and to have the required Matlab knowledge to identify the sections of code that need to be modified. The authors have looked into developing a graphical interface for the program but this direction has been abandoned due to the volume of work involved (by the authors’ estimate, the code for a graphical interface would be at least twice as long as the program itself) as well as the advanced graphical programming knowledge required.

Secondly, integrating a Matlab program inside a larger simulation platform and later on into a Hardware-in-the-Loop test stand has proven also problematic. The only available solution (importing the Matlab code into Simulink as a function, then importing the Simulink model into LabVIEW for deployment on a National Instruments Real Time platform) presented numerous disadvantages, main ones being the difficulty of any eventual troubleshooting (due to so many code transformations) and the difficulty in changing the parameters of the Matlab program.

Ultimately, the authors have decided to use co-simulation with LabVIEW, which solves both issues mentioned above.

4. CO-SIMULATION WITH LABVIEW

The core of the Matlab-LabVIEW co-simulation solution (fig. 4) is the LabVIEW MathScript module, which allows integrating portions of Matlab code inside a LabVIEW program (just like a regular LabVIEW subVI).



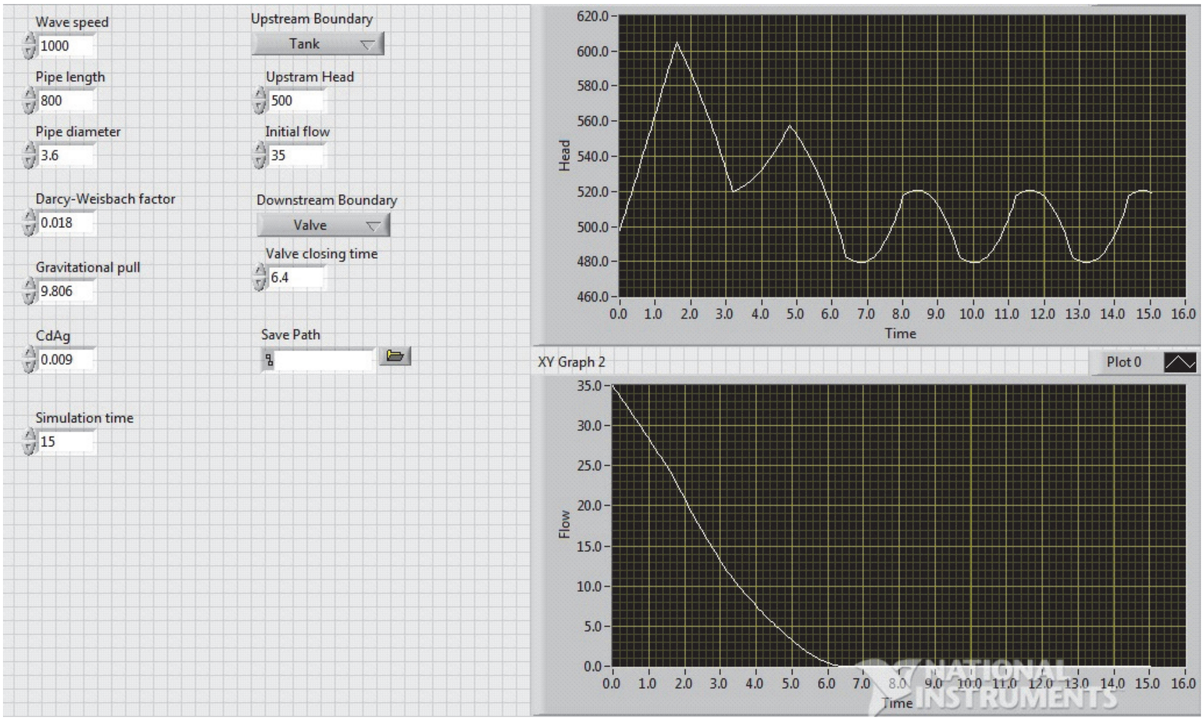
```
IA=pi*D^2/4;
R=(f*L)/(2*g*D*A
B=a/(g*A);
dt=L/(a*n);
k=1;
t(k)=0;
tau(k)=1;
ns=n+1;

%calculul valorilo
%Q0=sqrt((2*g*C
for i=1:ns
    H(k,i)=HR-(i-1)
    Q(k,i)=Q0;
end
```

Fig. 4. Matlab-LabVIEW co-simulation.

This means the Matlab code already developed could be used almost entirely. Only thing that needed changed was the part handling the program inputs and outputs, which communicate with LabVIEW now instead of being self-contained in the Matlab code. Further alteration to the Matlab code was caused by the addition of further upstream and downstream boundary conditions.

As it can be seen in fig. 4, the inputs and outputs of the MathScript node, which encapsulates the whole waterhammer simulations come from and go to data ports in the graphical interface. They are however simple LabVIEW data inputs/outputs, which means the program requires absolutely no changes to wire it to a bigger simulation model (like a completely hydropower until model or HIL-stand) and subsequently deploy it on a National Instruments Real-time machine.

Like any LabVIEW program, the graphical interface is integrated, being automatically built together with the block diagram. The end result can be seen in fig. 5. From the interface, the user can set all the flow parameters, pick the boundary conditions and then visualize the flow and pressure on a pair of on-screen graphs. If further analysis is needed, then the results can be saved in a file for further analysis.



Wave speed 1000

Pipe length 800

Pipe diameter 3.6

Darcy-Weisbach factor 0.018

Gravitational pull 9.806

CdAg 0.009

Simulation time 15

Upstream Boundary Tank

Upstram Head 500

Initial flow 35

Downstream Boundary Valve

Valve closing time 6.4

Save Path

Fig. 4. LabVIEW waterhammer graphical interface.

5. RESEARCH DIRECTION

As it has been stated earlier in the paper, the end goal of the authors' waterhammer research is desired to be the development of new control algorithms for hydropower units by-pass valves. The LabVIEW program presented above provides adequate means for the numerical simulation stage of this research. The downstream boundary condition can be set to two valves, one simulating the turbine wicket gate closing after a set of laws determined based on a turbine speed governor model and the other one simulating the by-pass valve, closing after a law determined by a control system. This setup can be used to tune and test the performance of various control algorithms for the by-pass valve. The optimal solution will then be validated by experiments on a Hardware-in-the-Loop test stand.

The stand (Fig. 5) which is currently in development consists of a hose simulating the hydropower unit pipeline, with a pressure source (pump and accumulator) at the upstream end, and a pair of servovalves at the downstream end. The first servovalve acts as a turbine wicket gate and is controlled by a National Instruments PXI real time machine which contains a turbine, generator and speed governor model. The second servovalve simulates the by-pass valve, and is controlled by a National Instruments compactRIO embedded controller, which contains the by-pass valve control algorithm, or 'pressure governor'. It was named so because it monitors the pressures at both end of the pipeline and calculates the required servovalve position to minimize the pressure waves.

6. EDUCATIONAL DIRECTION

The understanding of the waterhammer phenomenon at a theoretical level requires advanced fluid mechanics and mathematics knowledge. Understanding its numerical simulation representations often requires understanding numerical methods as well as programming knowledge of the software being used. Experimental study requires specialized laboratory test stands, which are not always available. As such, presenting the phenomenon to students is often difficult.

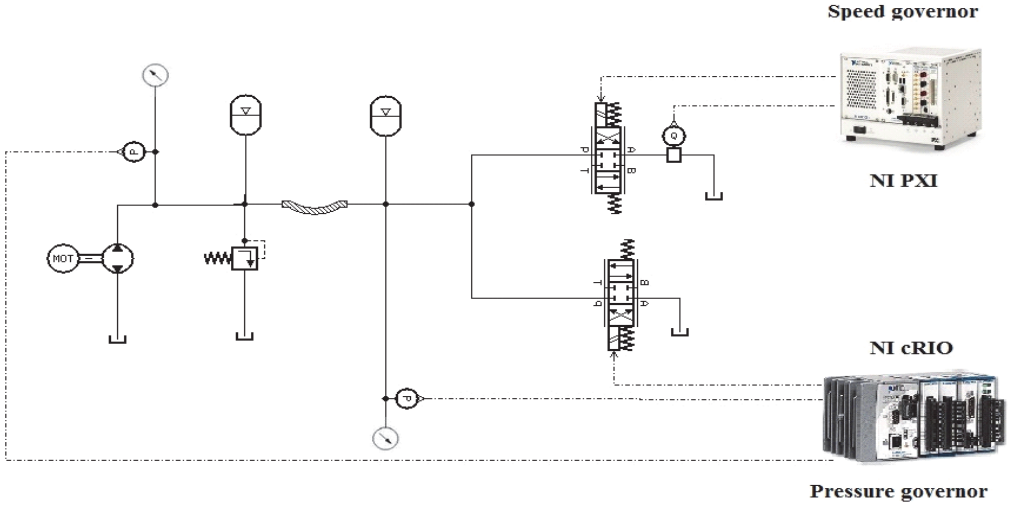
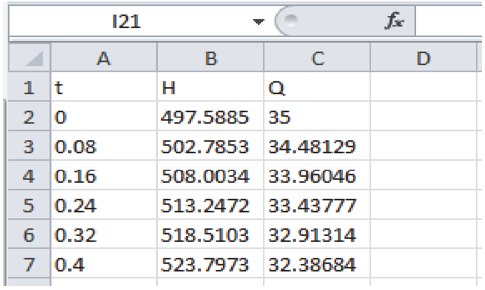


Fig. 5. Waterhammer HIL test stand.

The authors feel that the LabVIEW application they have developed can be a very useful tool in such a process. The graphical interface is streamlined and easy to use, simulating a given case of waterhammer requiring next to no LabVIEW knowledge. Additionally, being developed in LabVIEW, it means it benefits of the integrated Web Publishing tool. The program can be run on a local machine and published on the Internet, meaning that anyone could access this waterhammer calculator over the web. A further improvement in this direction has been made with the release of LabVIEW 2013, which now supports saving measurement or simulation results in Excel files. Previously, the results could be saved only in LabVIEW (.lvm) or binary formats, limiting the access to data for the average user. Currently however, the user has the facility to save on his local drive an Excel file which clearly presents all the simulation data (time *t*, head *H* and flow *Q*) for further analysis and storage (Fig. 6).



	A	B	C	D
1	t	H	Q	
2	0	497.5885	35	
3	0.08	502.7853	34.48129	
4	0.16	508.0034	33.96046	
5	0.24	513.2472	33.43777	
6	0.32	518.5103	32.91314	
7	0.4	523.7973	32.38684	

Fig. 5. Head of the .xlsx file.

7. CONCLUSIONS

By incorporating the original MATLAB software into LabVIEW, the authors have managed to greatly improve both the interface as well as the structure of the program itself, making it suitable for a variety of uses.

As a standalone application, available over Internet or simply distributed as an executable file, the program can be of use in the teaching process as well as for anyone that is interested in the study of waterhammer and pipelines. The intuitive and easy-to-use interface makes sure no LabVIEW (or other programming language) knowledge is needed, while the Excel export provides all the simulation data in an accessible and almost universal format. As a module, this program can be incorporated in order to model waterhammer in a variety of larger models of hydraulic systems, for example in the authors' current ongoing work (the Hardware-in-the-Loop test stand).

References

[1] N. Vasiliu, C. Calumnious, „Digital Electro Hydraulic Speed Governor for KAPLAN Turbines“. Romanian Patent No. 120101/2003.

[2] D. Vasiliu, C. Călinoiu and N. Vasiliu, *Governing Hydraulic Turbines By Servo Solenoid Valves – Numerical Simulation And Test.* 2000 ESS Conference, The Society for Computer Simulation, Hamburg, 2000.

[3] N. Vasiliu, D. Vasiliu, C. Călinoiu, D. Ofrim, *Digital Control Systems for Synchronizing Hydraulic Servo Cylinders.* Scientific Bulletin of the „Politehnica“ University of Timişoara, Transactions on Mechanics Special Issue. The 6th International Conference on Hydraulic Machinery and Hydrodynamics, Timişoara, Romania, October 21-22, 2004.

[4] N. Vasiliu, D. Vasiliu, *Fluid Control Systems*, Vol. I., Technical Press House Bucharest, 2005 (in Romanian).

[5] N. Vasiliu, D. Vasiliu, C. Calumnious, *Modeling, Simulation and Identification of the Electrohydraulic Speed Governors for Kaplan Turbines*, PTMC 2007, Bath, 2007.

[6] N. Vasiliu, C. Călinoiu, D. Vasiliu, *Electrohydraulic Servomechanisms with Two Stages DDV for Heavy Load Simulators Controlled by ADwin*. Recent Advances in Aerospace Actuation Systems and Components, Toulouse, France, November 24-26, 2004.

[7] T. J. Viersma, Analysis, *Synthesis and Design of Hydraulic Servosystems and Pipelines*. Elsevier Scientific Publishing Company, Amsterdam, 1980.

[8] R. Puhalschi, S. Feher, D. Vasiliu, N. Vasiliu, C. Irimia, *Simulation Languages Facilities for Innovation.* EUROSIS 2013 SIMEX Conference, Bruxelles, 2013.

[9] P. Kundur, *Power System Stability and Control,* McGraw-Hill, 1994.

[10] E. B. Wiley and V. L. Streeter, *Fluid Transients*, McGraw-Hill, 1983.

[11] J. Watton, *Fundamentals of Fluid Power Control*, Cambridge University Press, 2009.

WEB REFERENCES

[1] www.mathworks.com

[2] www.ni.com

[3] www.lmsintl.com