CAUSAL OR ACAUSAL MODELLING: LABOUR FOR HUMANS OR LABOUR FOR MACHINES

J. Kofránek, M. Mateják, P. Privitzer, M. Tribula

Laboratory of Biocybernetics, Institute of Pathological Physiology, First Faculty of Medicine, Charles University in Prague

Abstract

Models designed using classical Simulink networks provide a clear graphic visualization of individual mathematical relationships. Signals flow in connections between individual blocks, transmitting values of individual variables from the output of one block to inputs of other blocks. Processing of input information to output information takes place in the blocks. Interconnection of the blocks in Simulink thus reflects rather the calculation procedure than the very structure of the modelled reality. This is the so called causal modelling. However, it is important in designing and especially presenting and describing the model that the very structure of the model, rather than the very algorithm of the simulation calculation, captures well especially the physical essence of the modelled reality. Therefore, declarative (acausal) notation of models is starting to be used in a growing extent in modern simulation environments; this means that individual components of the model describe the equations directly and not the algorithm of their solution. By interconnecting individual components, the systems of equations become connected with each other. Interconnection of the components does not define the calculation procedure but the modelled reality. The way of solving the equations is then "left up to the machines". Application of acausal approach has been made possible by new Simulink libraries Simscape, and linked domain libraries SimElectronics, SimHydraulics, SimMechanics etc. Modelica is a modern simulation language built directly on acausal notation of models. Implementation of this language of the company Dynasim is interesting for Mathworks products users as it allows for direct connection with Simulink and Matlab (Modelica is implemented under the name Dymola in this case).

1 Introduction

An article was published in the journal Annual Review of Physiology [3] 36 years ago, whose form surpassed at the very first sight the accustomed form of the then physiological articles. It was introduced by a large diagram pasted in as an attachment (Fig. 1). At the first sight, the diagram full of lines and mutually connected elements remotely resembled a drawing of some electrotechnical device. However, instead of tubes or other electrotechnical parts, computational blocks were shown in the diagram (multiplication, division, summation, integration, functional blocks), symbolizing mathematical operations performed with physiological quantities. Bunches of connecting conductors between the blocks seemed at the first sight to express complex feedback interconnection of physiological quantities. The blocks were grouped in 18 groups, representing individual interconnected physiological subsystems.

The very article gave a quantitative description of physiological regulation of the circulation system in a manner absolutely new in the then medical professional literature, as well as its connexion with and linkage to other subsystems of the organism – the kidneys, regulation of volumetric and electrolyte equilibrium etc. Instead of writing a set of mathematical equations, the article made use of graphic depiction of mathematical relationships. This syntax made it possible to provide a graphic depiction of connexions among individual physiological quantities in the form of interconnected blocks representing mathematical operations. These elements are very similar to Simulink blocks; there is a difference only in their graphic shape.

nothing but copy the diagram mindlessly had bad luck. After all, in those times, the authors also mailed out source program codes of their model in the Fortran programming language – thus if somebody wanted only to test the model behaviour, there was no need to program anything (perhaps only to perform a routine conversion from Fortran to another programming language at the maximum).

Our Simulink implementation of the (corrected) Guyton's model (Fig. 3) is available to those interested at www.physiome.cz/guyton. On this website, our Simulink implementation of a much more complex model version of Guyton et al from subsequent years can be found, as well. At the same time, very detailed description of all the mathematical relationships applied is presented together with the reasons.

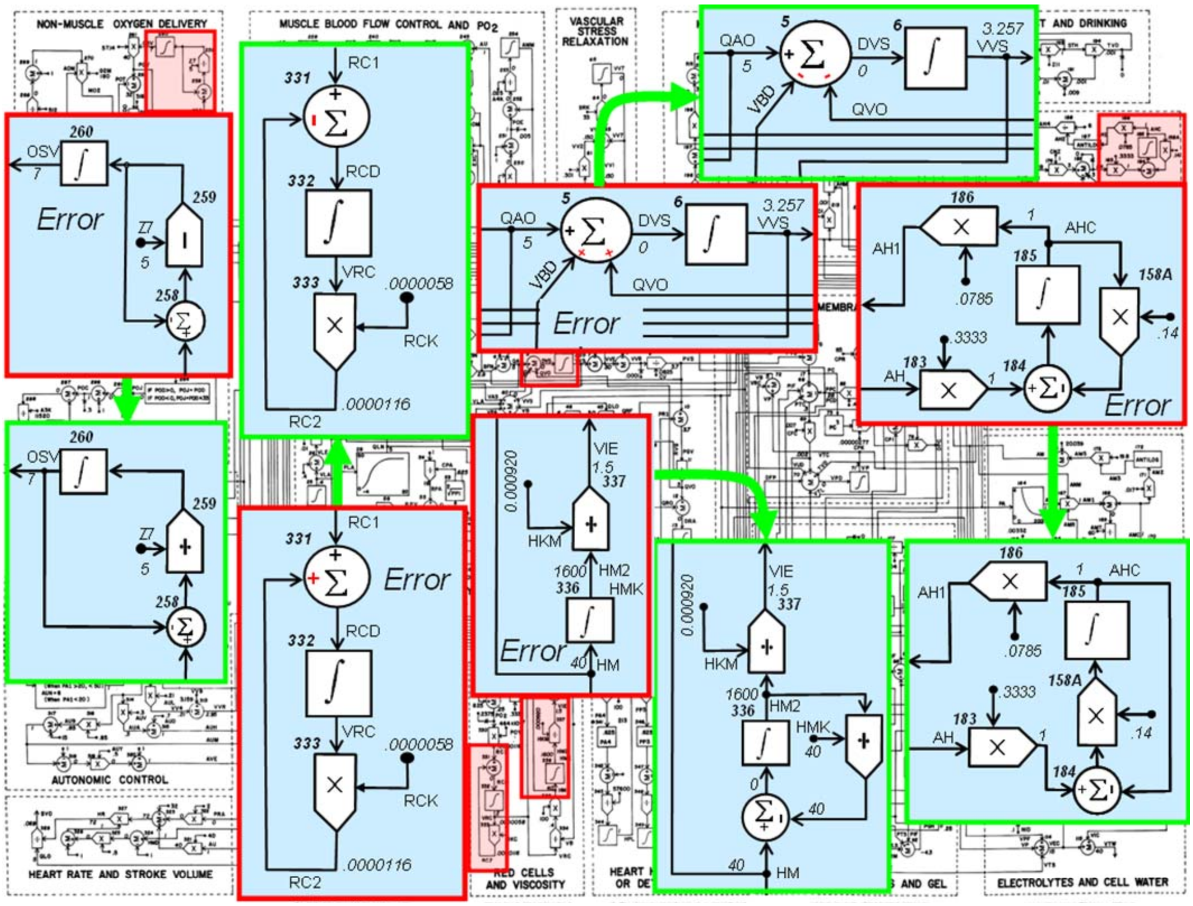

Figure 2: Correction of the most significant errors in the model of A. C. Guyton et al.

3 Block-Oriented Simulation Networks

The Guyton's block diagram augured occurrence of block-oriented simulation languages. However, in the year 1972, Guyton himself and his collaborators implemented the model in Fortran – Simulink version 1 was not marketed until 18 years later (in 1990). Block-oriented simulation languages, Simulink being one of their typical representatives, make it possible to set up computer models using individual hierarchically arranged blocks with defined inputs and outputs. It is apparent from such a hierarchically arranged block-oriented description how the values of individual variables are calculated in the model – i.e. the computational algorithm. The blocks can be grouped into individual subsystems, which communicate with the surrounding environment through defined input and output "pins", thus representing something like "simulation chips". A simulation chip hides the simulation network structure from the user, similarly as an electronic chip hides the interconnection of individual transistors and other electronic elements. The user can thus be concerned only with the chip behaviour, without having to take care of its inner structure and the computational algorithm. Behaviour of a simulation chip can be thus tested by observing the outputs on connected virtual displays or virtual oscilloscopes. This is very advantageous especially in testing the model behaviour and to express mutual dependencies among variables. The entire complex model can be then displayed as interconnected simulation chips, while it is clear from the structure of their mutual connections what effects are taken into account in the model and how.

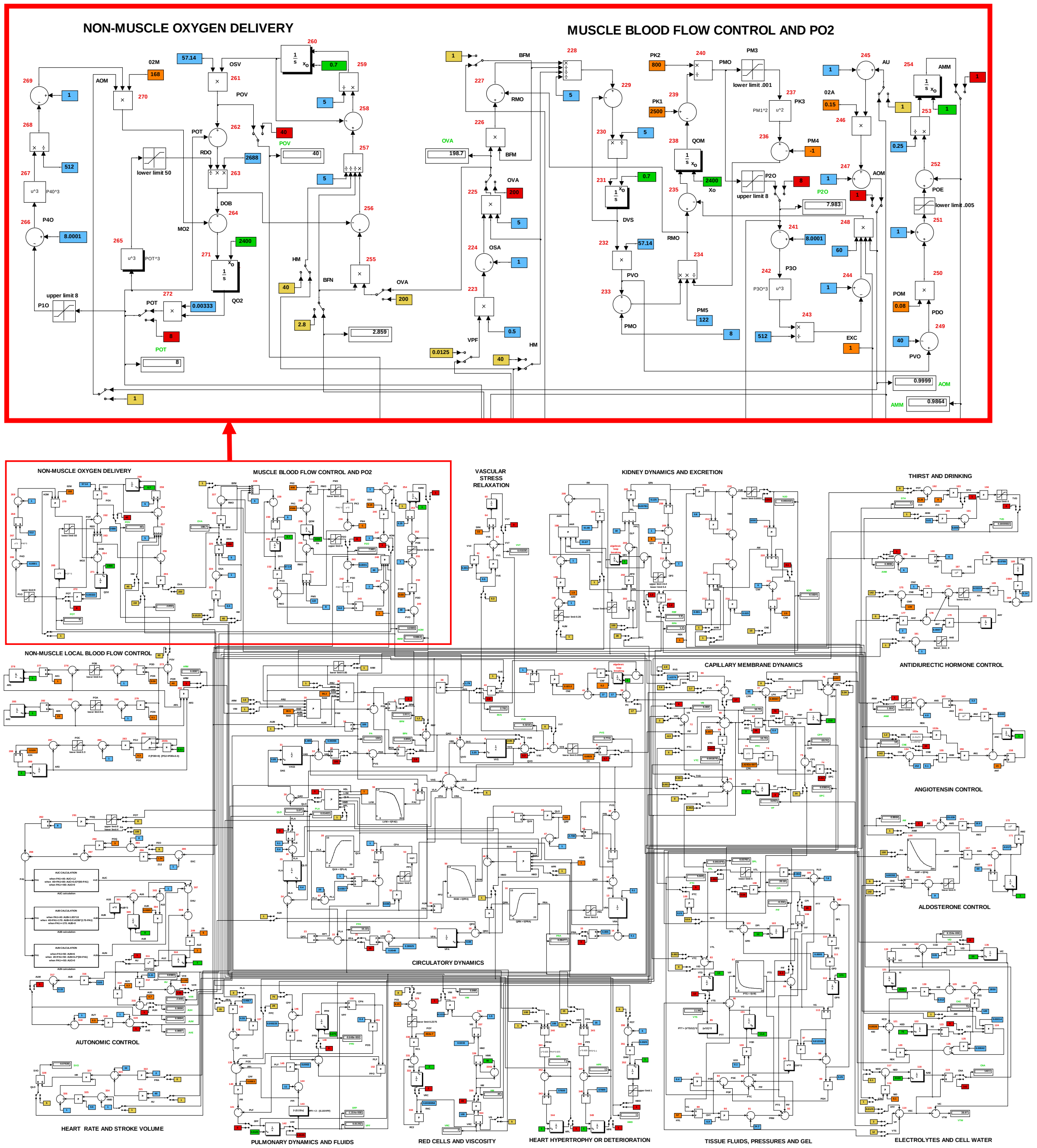

Figure 3: Implementation of the model of A. C. Guyton et al in Simulink

This means a high advantage for multidisciplinary cooperation – especially in borderline areas such as, for example, biomedical systems modelling [9]. An experimental physiologist is not forced to examine in detail what mathematical relationships are hidden “inside” the simulation chip; however, based on interconnection of individual simulation chips with each other he or she can understand the model structure and can verify its behaviour in an appropriate simulation visualization environment.

Hierarchical block-oriented simulation tools have therefore found their significant application in description of complex regulation systems, which can be seen in physiology. The international project PHYSIOME is devoted to formalized description of physiological systems; it is the successor of

GENOME project. GENOME project output was represented by detailed description of the human genome; PHYSIOME is aimed at formalized description of physiological functions. Computer models are used as the methodological tool [2, 7].

Several block-oriented simulation tools were designed as part of PHYSIOME project, serving as a reference database for formalized description of the structure of complex physiological models. They include JSIM [11] (http://www.physiome.org/model/doku.php) and also the CEllML language (http://www.cellml.org/). Disciples and followers of Prof. Guyton expanded the original extensive simulator of the circulatory system (Quantitative Circulatory Physiology [1]) by an integrated connection of all important physiological systems. The simulator Quantitative Human Physiology is the most recent result, which describes the currently most complex and most extensive model of physiological functions. The model can be downloaded from http://physiology.umc.edu/themodelingworkshop/. In order to express the complex structure of the model, its authors created a special, block-oriented simulation system, QHP.

4 Causal and Acausal Approach

Block-oriented tools use hierarchical connected blocks. Signals are transmitted through links between individual blocks; the signals serve to transfer values of individual variables from the output of one block to inputs of other blocks. Input information is processed in the blocks to output information. Interconnection of blocks therefore reflects rather the calculation procedure than the very structure of the modelled reality.

In complex systems, ***physical reality of the modelled system slowly disappears under the computation structure*** thanks to this approach.

That is why in recent times, such tools are starting to be used in modelling complex systems, in which individual parts of the model are directly described as equations and not as an algorithm of the solution of such equations. This is the so called ***declarative (acausal) notation*** of models, unlike ***causal notation*** in block-oriented languages where the (causal) description of the way of calculating individual model variables must be (for example, also visually using graphic connection of individual computational elements) expressed.

Acausal approach is made possible by the new Simulink libraries ***Simscape*** and linked domain libraries SimElectronics, SimHydraulics, SimMechanics etc.

A modern simulation language built directly on acausal notation of models is the programming language **Modelica** [4]. Originally, it was developed in Sweden and now is available both as an open-source version (developed under the auspices of the international organization Modelica Association, http://www.modelica.org/), and in two commercial implementations. One of them is the commercial implementation of the company Dynasim AB – purchased recently by the supranational concern Dassault Systemes (marketed under the name ***Dymola***, currently version 7.1) and the other one is offered by MathCore (marketed as ***MathModelica***). Dynasim Modelica provides good linkage to Matlab and Simulink while MathModelica can connect with the Mathematica environment supplied by the company Wolfram.

Modelica utilizes mutually connected components, which represent instances of individual classes. Unlike implementation of classes in other object-oriented languages (such as jw C#, Java etc.), classes in Modelica have a special section in addition, in which ***equations*** are defined. The equations do not express assignment (i.e. saving of the calculation result of an assigned statement into a given variable) but a definition of relationships among variables (as is the custom in mathematics and physics).

In Modelica, components (instances of classes) can be connected through exactly defined interfaces – ***connectors***. Connectors are instances of connector classes in which variables used for connection are defined. Connectors falling in the same connector classes can be connected (while in these classes, variables of equivalent types can be connected). In other words – in the connection, the type of plugs must correspond exactly to the socket types.

The important thing is that by connecting the components, ***connection of sets of equations in individual components*** with each other occurs actually. in cases an identical variable is defined in the connector of connected components (and if it does not represent the flow – see further), by establishing the connection it is defined that the value of this variable should be the same in all components. When the variable is a part of equations in the connected components, by means of connection we also define interconnection of the equations in which the given variable is found. At the point where multiple components are connected, values of the variables connected through the common point must be the same (similarly to voltage on the common terminal in electric circuits, which must correspond to Kirchhoff's first law). And not only that. In the connector, it is possible to define that some connected variables shall represent the flow – in such a case, this means that values of all variables marked in this way shall be set in all components interconnected at the same point to such a value so that their algebraic sum is equal to zero (similarly to the sum of currents on the common terminal in electric circuits, which must correspond to Kirchhoff's second law).

If a variable marked in this way is a part of equations in interconnected components, another equation is thus added to the connected set of equations, which defines the requirement of the zero value of the algebraic sum of this variable's values).

By means of connecting Modelica components, we thus do not define the computation procedure but the modelled reality. The way of solving the equations is then "left up to the machines".

5 Generalized System Properties

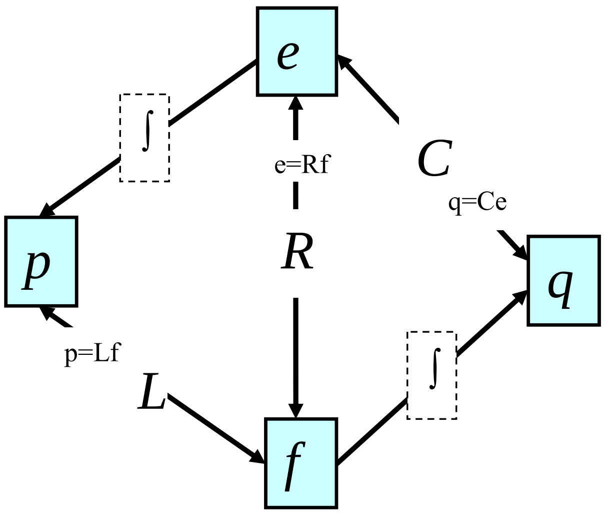

Figure 4: Relationships among generalized system properties:

- ***e*** *means generalized effort – matched by force in mechanics, voltage in electric diagrams, pressure in hydraulics etc.*
- ***f*** *is generalized flow – matched by velocity in mechanics, current in electric diagrams, flow in hydraulics, temperature flow in thermodynamics etc.*
- ***q*** *is generalized accumulation or deflection, represented by the generalized flow integral. It is matched, for example, by spring stretching in mechanics, liquid volume in hydraulics, charge in electric diagram, accumulated heat in thermodynamics etc.*
- ***p*** *represents generalized inertance – generalized effort integral, representing kinetic energy; matched by the flow velocity change proportional to pressure difference (inertance of flow) in hydraulics, potential needed to change electric current (induction) in electric circuits etc.*
- ***R****,* ***C*** *and* ***L*** *represent constants of proportionality among individual generalized system properties. Matched, for example, by resistance, capacitance or mass.*

Depiction of the model in the acausal simulation environment thus resembles the physical reality of the modelled world more than classical interconnected block schemata in Simulink or QHP. This is related to generalized system properties of the real world (Fig. 4) in which ***generalized efforts*** (to which the force, pressure, voltage etc. correspond in real world) and ***generalized flow*** (to which the current, flow etc. correspond in real world) play an important role.

Provided that the reality is depicted in Modelica by means of connected components, then as for flow variables, the value at the connection point must correspond to Kirchhoff's second law (the current cannot accumulate or be lost at the connection point), and equality must hold true for other variables at the common connecting point (according to Kirchhoff's first law).

In Modelica, there is a standard library of varied classes to model electric, mechanical, hydraulic objects of the real world. Mathworks, too, has responded to development trends of acausal modelling by creating Simscape toolboxes and linked application libraries to model electric circuits, mechanical and hydraulic systems.

We shall illustrate the difference between modelling in block-oriented simulation tools and in Modelica on two examples of modelling physiological reality: On the model of simple mechanics of pulmonary ventilation and on implementation of a classical model of cellular membrane of the neurone pursuant to Hodgkin-Huxley [6].

6 Pulmonary Ventilation Mechanics Model

Let us consider a ***simple model of pulmonary mechanics***, shown schematically in Fig. 5. Upon applying significant simplification, the lungs can be viewed as three bags connected using two tubes. The lungs are connected to the ventilator of artificial pulmonary ventilation, which blows air into the lungs periodically under the pressure **PAO**. **P0** is the pressure of ambient atmosphere. The air flow **Q** flows through upper respiratory tract whose resistance is ***RC***. From the upper respiratory tract, air struggles through the lower respiratory tract into alveoli. The resistance of the lower respiratory tract is ***RP***, the pressure in central parts of the respiratory tract (at the borderline of the upper and lower respiratory tract) is ***PAW***, pressure in alveoli is ***PA***.

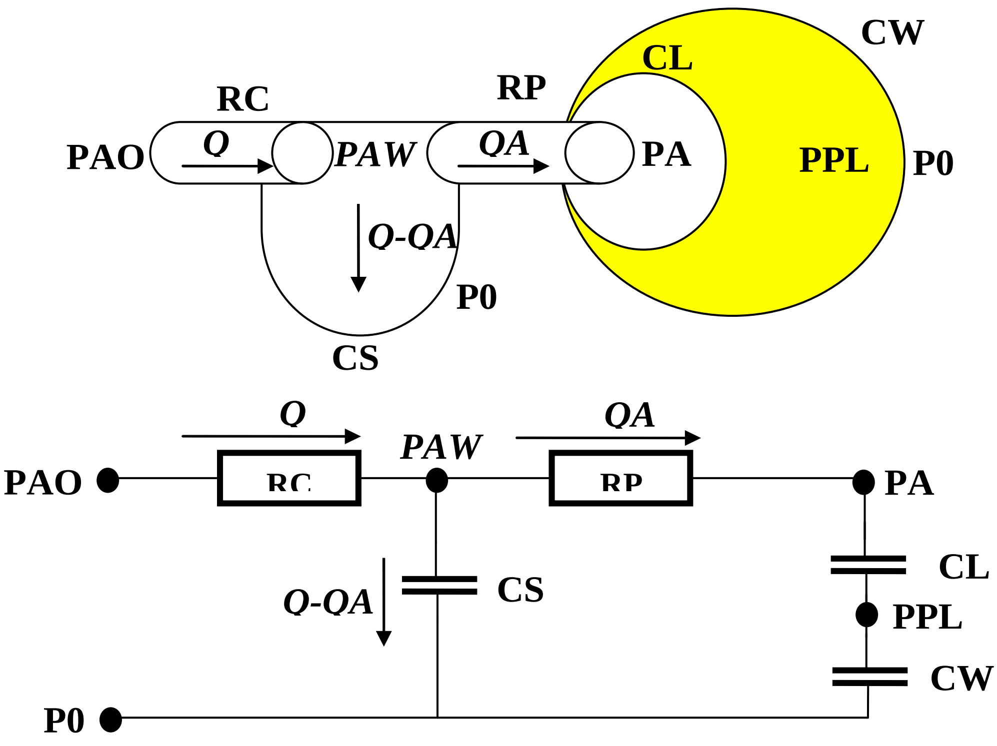

Figure 5: Simple model of pulmonary mechanics (hydraulic and electric similarity)

The air expands pulmonary alveoli whose elasticity is ***CL*** (as total pulmonary elasticity). The interpleural cavity is found between the lungs and the chest. The pressure in the chest is ***PPL***. In artificial pulmonary ventilation in which case the air is blown into the lungs under pressure, the chest must expand in addition – chest elasticity is **CW**. A small part of air, which does not reach all the way to the alveoli only expands the respiratory tract – its elasticity is ***CS*** (the so called dead space breathing).

Now the equations can be set up. According to Ohm's law the following must hold true:

$$\begin{aligned} PAW - PA &= RP\ QA \\ PAO - PAW &= RC\ Q \end{aligned} \tag{1}$$

The relationship between elasticity, pressure gradient and volume (expressed as the flow integral) is expressed by the equations:

$$\begin{aligned} PA - PPL &= \frac{1}{CL}\int QA\ dt \\ PPL - P0 &= \frac{1}{CW}\int QA\ dt \\ PAW - P0 &= \frac{1}{CS}\int \left(Q - QA\right) dt \end{aligned} \tag{2}$$

According to the generalized Kirchhoff's law, the sum of all pressures (voltages) along a closed loop must be equal to zero, i.e. the following must hold true in the loop along the PAW node and along the PA0 node:

$$(PAW - PA) + (PA - PPL) + (PPL - P0) + (P0 - PAW) = 0$$

$$(PAO - PAW) + (PAW - P0) + (PO - PAO) = 0$$

And upon substitution using Ohm's law and elasticity equations:

$$\begin{cases} RP\,QA + \left(\dfrac{1}{CL} + \dfrac{1}{CW}\right)\displaystyle\int QA\,dt - \dfrac{1}{CS}\int (Q - QA)\,dt = 0 \\ Q\,RC + \dfrac{1}{CS}\displaystyle\int (Q - QA)\,dt + (P0 - PAO) = 0 \end{cases} \tag{3}$$

6.1 Implementation of the Pulmonary Ventilation Mechanics Model in Simulink

When setting up the model in Simulink, the computation procedure from input variables to the output ones must be exactly defined. If we want to calculate the reaction of air flow to/from the lungs (***Q***) to the input – i.e. pressure changes at the initial point of the respiratory tract (***PAO***) caused by the artificial pulmonary ventilation apparatus – the Simulink model shall be similar to that in Fig. 6.

The Simulink model can also be expressed in a simpler form. At first, from Equation (3) we shall derive the differential equation (input variable ***PAO***, output variable ***Q***):

$$\frac{d^2PAO}{dt^2} + \frac{1}{RP\ CT}\frac{dPAO}{dt} = RC\frac{d^2Q}{dt^2} + \left(\frac{1}{CS} + \frac{RC}{RP\,CT}\right)\frac{dQ}{dt} + \frac{1}{RP\ CS}\left(\frac{1}{CL} + \frac{1}{CW}\right)Q \tag{4}$$

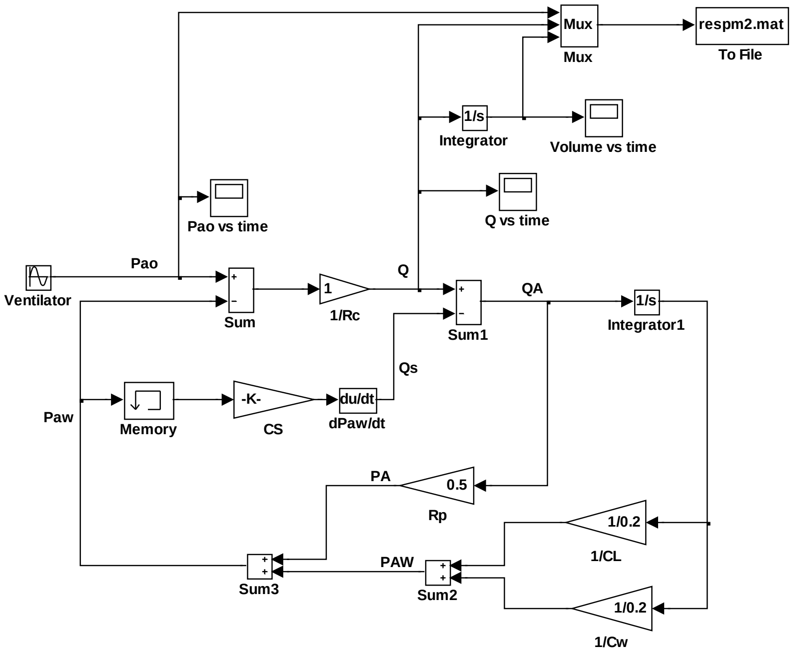

Figure 6: Implementation of the model in Simulink according to Equations (3)

Upon entering the following numerical parameters of resistances (in the units: cm H_2O/L/sec) and elasticities (in the units: L/cmH_2O) [8]:

$$RC = 1;\ RP = 0.5;\ CL = 0.2;\ CW = 0.2;\ CS = 0.005$$

Equation (4) becomes simplified:

$$\frac{d^2PAO}{dt^2} + 420\frac{dPAO}{dt} = \frac{d^2Q}{dt^2} + 620\frac{dQ}{dt} + 4000Q \tag{5}$$

Upon Laplace transform of Equation (5) we obtain:

$$\frac{Q(s)}{PAO(s)} = \frac{s^2 + 420s}{s^2 + 620s + 4000} \tag{6}$$

This gives us the possibility to simplify the Simulink model (Fig. 7):

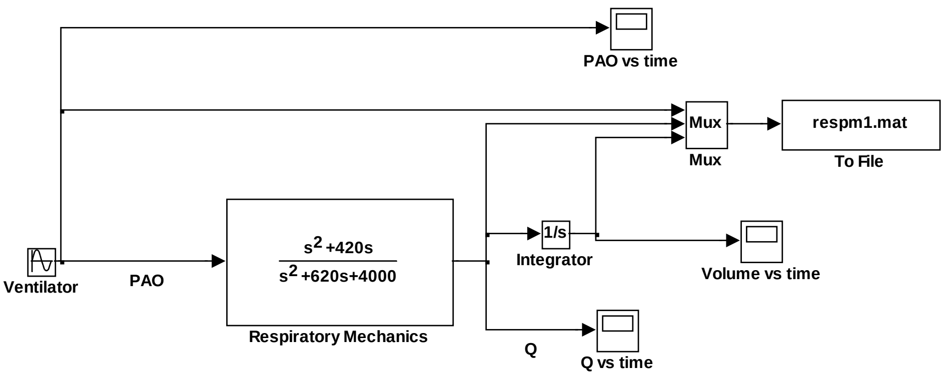

Figure 7: Implementation of the model in Simulink using Laplace transform according to Equation (6)

However, when the parameter values change, the transformation function (6) must be recalculated and the Simulink model changes. Now let us make the model slightly more complicated by taking into account ***inertia of air*** in the upper respiratory tract (Fig. 8).

Now we shall moreover include the inertial element **LC**=0.01 cm H_2O s^2 L^{-1}:

$$LC = \frac{\Delta P}{\frac{dQ}{dt}}$$

where ΔP is the pressure gradient and $\frac{dQ}{dt}$ is flow acceleration, or:

$$\Delta P = LC\frac{dQ}{dt} \tag{7}$$

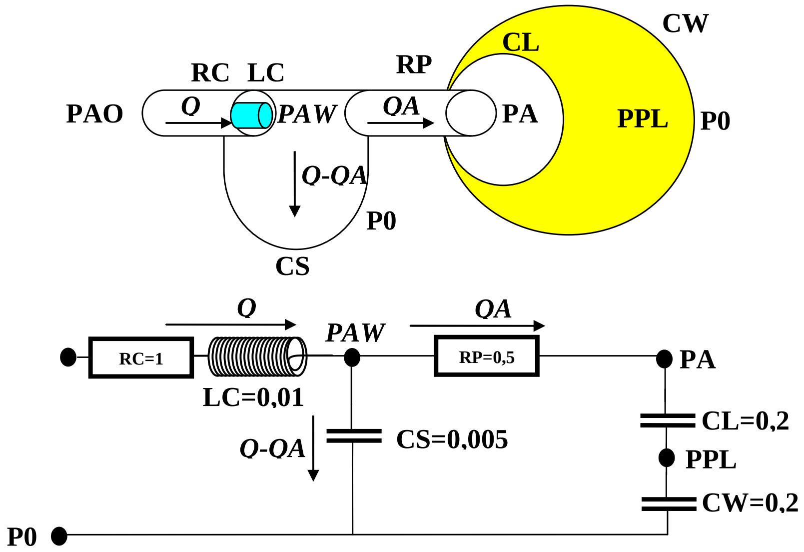

Figure 8: Simple model of pulmonary mechanics taking inertia into account (hydraulic and electric similarity)

Then instead of the Set of Equations (3) we shall obtain:

$$\begin{cases} RP\dfrac{dQA}{dt} + \left(\dfrac{1}{CL} + \dfrac{1}{CW}\right)QA - \dfrac{1}{CS}(Q - QA) = 0 \\ RC\dfrac{dQ}{dt} + LC\dfrac{d^2Q}{dt^2} + \dfrac{1}{CS}(Q - QA) + \dfrac{dP0}{dt} - \dfrac{dPAO}{dt} = 0 \end{cases} \tag{8}$$

And instead of Equation (5):

$$\frac{d^2PAO}{dt^2} + 420\frac{dPAO}{dt} = 0{,}01\frac{d^3Q}{dt^3} + 5{,}2\frac{d^2Q}{dt^2} + 620\frac{dQ}{dt} + 4000\,Q \tag{9}$$

And finally, upon Laplace transform:

$$\frac{Q(s)}{PAO(s)} = \frac{s^2 + 420s}{0{,}01s^3 + 5{,}2s^2 + 620s + 4000} \tag{10}$$

The Simulink model shall change (Fig. 9):

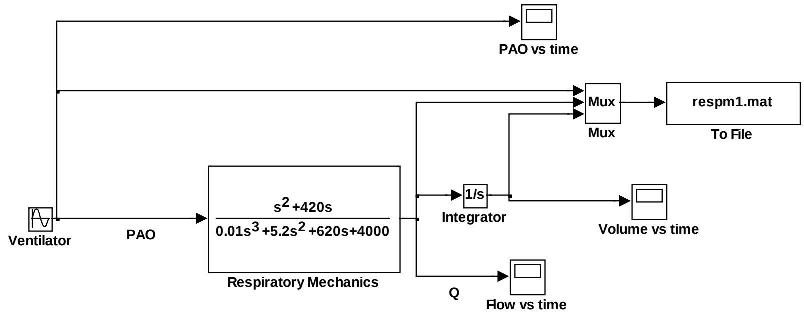

Figure 9: Implementation of the model in Simulink using Laplace transform according to Equation (10)

Thanks to the fact that the computational direction must be always taken into account in Simulink, the very Simulink diagram is quite distant from the actual physical reality of the system described. Even a small change in the model, such as addition of the inertial element, causes the need of careful calculation and change of the model structure. An essential change of the model occurs also in the case that spontaneous breathing is considered instead of artificial pulmonary ventilation. The model input shall not be represented by the pressure ***PAO*** created by the artificial pulmonary ventilation respirator but, for example, the thoracic wall elasticity ***CW*** (function of the breathing muscles can be modelled by cyclic change of elasticity).

Connection of blocks in Simulink reflects rather the ***calculation procedure*** and not the modelled reality structure.

6.2 Implementation of the Pulmonary Ventilation Mechanics Model in Modelica

In Modelica (or even in the Simulink library Simscape), the situation is different (source codes of the example can be downloaded from http://patf-biokyb.lf1.cuni.cz/wiki/objekty_2008). Instead of blocks, Modelica operates with connected components, which represent instances of individual classes, in Modelica moreover equipped with a special section in which ***equations*** are defined. ***By connecting Modelica components, not the calculation procedure but the modelled reality is defined. The way of solving the equations is thus "left up to the machines".***

The model representation in Modelica thus resembles rather the physical reality of the modelled world than connected block diagrams in Simulink. The simple ventilation mechanics model according to Fig. 3 can be expressed in a very straightforward way in Modelica. We shall make use of the fact that Modelica includes libraries of various classes to model electric, mechanical and hydraulic objects of the real world. Representation of the relationships of resistance, pressure gradient and flow according to Equation (1) and relationships of elasticity, pressure and flow according to Equation (2) thus takes the following form in Modelica:

Making the model more complex by adding the inertial element - pursuant to Equation (7) and Fig. 7 is easy (see Figure 11).

In the given case, we have used visual components of electric circuits for quick setting up of the model; however, nothing prevents us from creating a different shape of icons representing individual resistance, capacity and inertial elements in the lungs. These are far more than picture icons. Modelica is an object language and there is no obstruction in creating a special class using which it is possible, for example, to model oxygen and carbon dioxide flows (considering the oxygen bond to hemoglobin, CO2 conversion to bicarbonate, acidobasic equilibrium effect on blood gases transport etc.).

Formation of the physiological relationships library is one of our future aims. We are planning to transfer (and extend further in Modelica) our Simulink library of physiological relationships, Physiology Blockset (www.physiome.cz/simchips).

In respect of the fact that there are numerous relationships in physiological models (leading to solution of implicit equations in Simulink), ***acausal description of modelled relationships*** applied in Modelica provides a great advantage. Acausal description describes much better the essence of modelled reality and the simulation models are much more legible, and thus much less prone to mistakes, as well. Modelica is therefore a very suitable environment to model physiological systems.

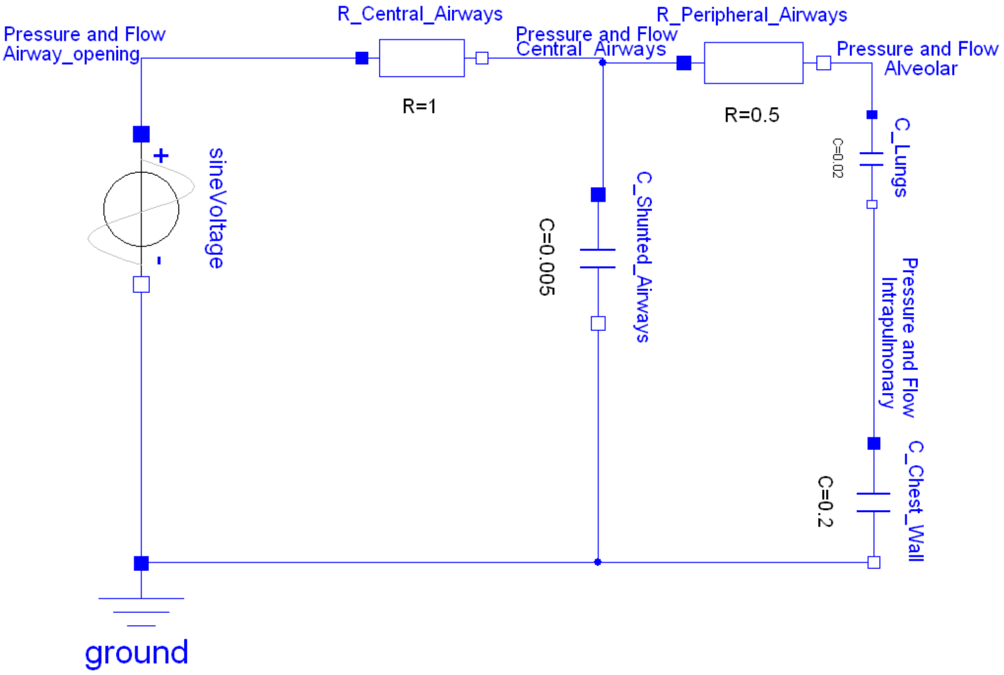

Figure 10: Simple model of pulmonary mechanics according to Fig. 5 in Modelica

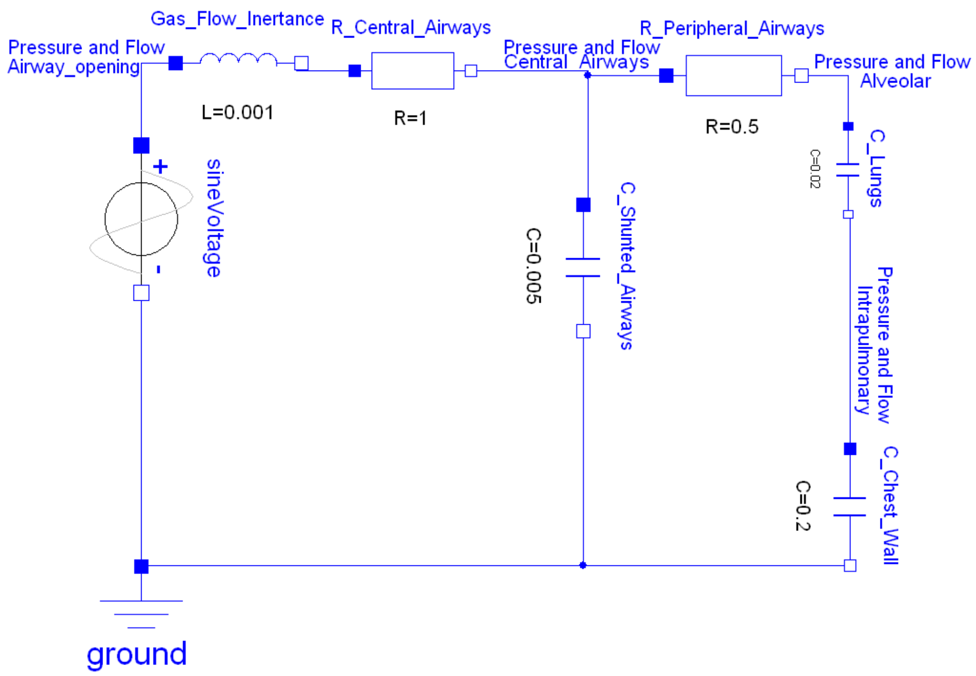

Figure 11: Modified model of pulmonary mechanics upon adding the inertia element in Modelica

7 Excitable Membrane of the Neuron according to Hodgkin and Huxley

In 1963, Alan Lloyd Hodgkin and Andrew Huxley were awarded the Nobel Prize for physiology and the mathematical model of the action membrane potential. Precisely their article [6] has become the foundation of many models describing the membrane potentials behaviour in varied cell types.

Primarily, the model provides explanation of the nervous impulse spreading by depolarization of the cellular membrane using the connection of two physical domains – the electric and chemical one. Thanks to the conductivity study of membrane channels, in dependence on time and current membrane voltage, the model describes the course of electric current that is formed by the flow of sodium and potassium ions through the membrane.

This electric current subsequently affects the current electric voltage. At rest, the inside of the cellular membrane has a negative charge. Proteins with the negative charge cannot pass through the cellular membrane and remain on the inside of the membrane. Potassium and sodium ions are (thanks to the sodium-potassium pump) distributed non-uniformly between the cell and its surroundings – inside the cell, there is a high potassium concentration and low sodium concentration in respect of the surroundings of the cell. Transfer of the ions can take place only through ionic channels. The concentration and electric gradient has an impact on the ions movement – at rest, the cellular membrane has a negative charge and sodium concentration outside the cell is much higher than inside. Sodium is pushed into the cell both by the electric and concentration gradient. As for potassium, both gradients act against each other; however, the concentration gradient prevails (and potassium thus shows the tendency to leave the cell).

Nernst equation describes how the chemical and concentration gradients can be compared. Differences in concentrations are transformed using this equation to the membrane voltage needed to maintain their different concentrations on both sides of the membrane. These concentration differences in the model form a kind of power supplies for specific ions. When the difference of the Nernst minus current voltage for the cation is greater than zero, the cation is pushed into the cell. The sodium-potassium pump (Na-K-ATPasis) takes care, from the long-term point of view, of maintaining the concentration differences of potassium and sodium in the cell and outside the cell; this pump keeps pumping sodium from the cell and potassium into the cell constantly. However, its functionality and change of sodium and potassium concentrations is disregarded in this model and the concentrations are considered constant, and the concentration differences (thanks to the Nernst equation) form a power supply of +40 mV for sodium and -87 mV for potassium.

Charge accumulation on the membrane is a typical example of a capacitor, where the non-conductive cellular membrane acts as a medium, which separates two charged surfaces.

However, sodium and potassium flows through appropriate channels depend on the channels permeability (much lower for sodium than for potassium at rest). The channels permeability depends on the membrane voltage. If the membrane voltage rises from the rest (negative) voltage to a certain boundary value, sodium channels start to open for a very short moment and the current of sodium ions breaks into the cell and "discharges the negatively charged membrane capacitor" – and even charges it to the opposite (positive) value – we speak of depolarization and occurrence of the action potential. At the same time, sodium channels start to close again due to the change of the voltage (and thus make sure that the concentration of ions in the cell virtually does not change). At the same time, permeability for potassium channels changes, potassium starts to leave the cell more rapidly (and the current of positive potassium ions flowing out of the cell recharges the "membrane capacitor" again to the rest negative value).

7.1 Implementation of the Model according to Hodgkin and Huxley in Modelica

The model implementation in Modelica corresponds to an electric diagram (Fig. 12). Source codes can be downloaded from http://patf-biokyb.lf1.cuni.cz/wiki/objekty_2008. Special components are represented by membrane channels for sodium (or potassium, respectively). These behave as a power supply with changing inner resistance. Its constant voltage is given by the chemical gradient of sodium (or potassium, respectively) according to the Nernst equation, while their inner resistance corresponds to the open state (permeability) of the channels. Only potassium cations can pass through potassium channels and only sodium ones through sodium channels. Other electrically charged atoms that can pass through the membrane are implemented using a constant power supply and constant resistance, not changing in dependence on time and voltage.

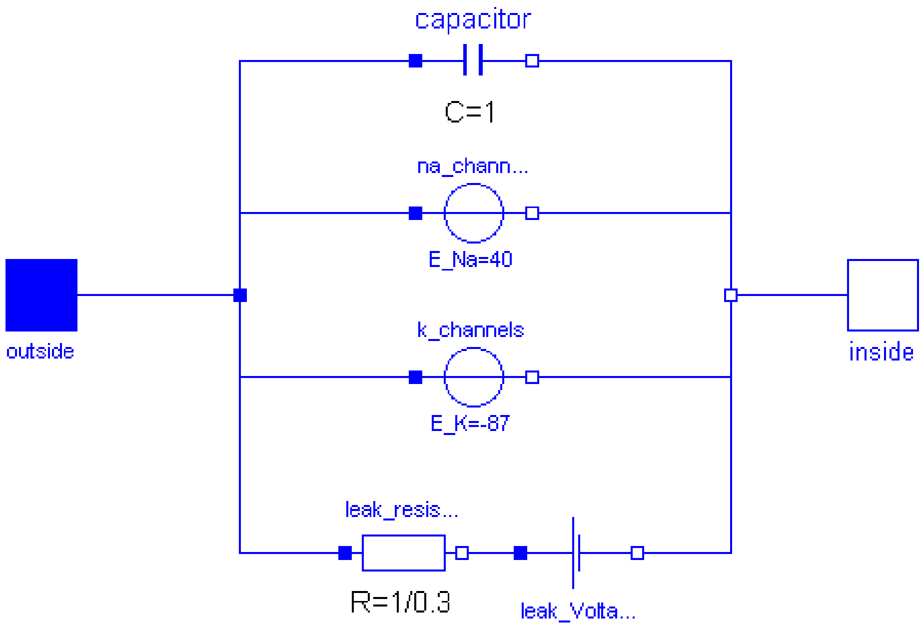

Figure 12: Diagram of the Hodgkin-Huxley's model in Modelica (in the Dymola environment)

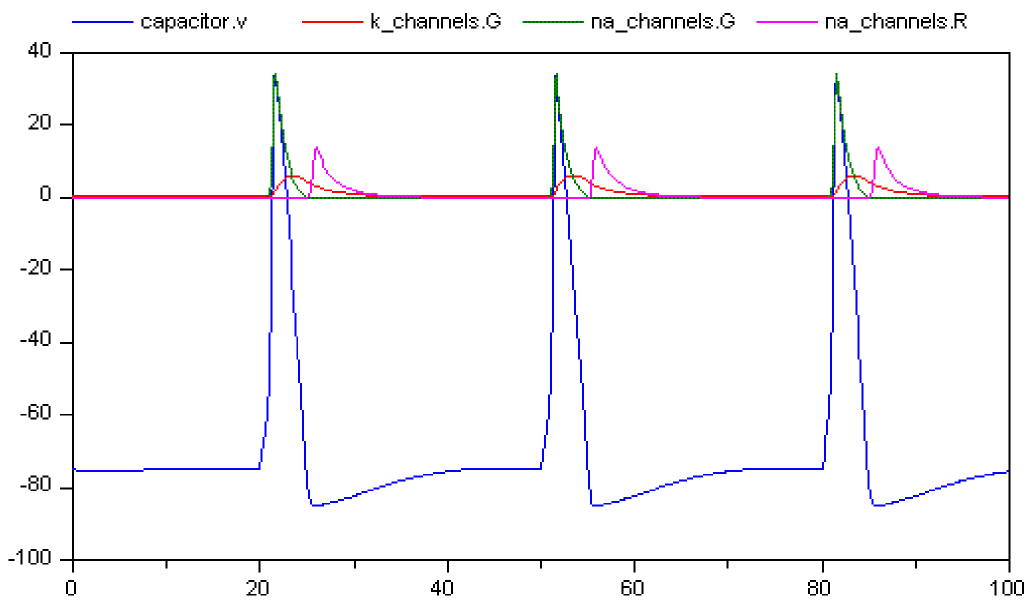

Figure 13: Model simulation outputs. Time units are given in ms.

- ***capacitor.v** is current voltage of the cellular membrane in mV*
- ***k_channels.G** is electric conductivity for potassium channels*
- ***na_channels.G** is electric conductivity for sodium channels*
- ***na_channels.R** is electric resistance for sodium channels = 1/(1000*G)*

When implementing the model in Modelica, basic electric components from the standard library can be used: We shall thus use models of a constant voltage supply, a capacitor. Afterwards, it is only necessary to define the membrane channel classes and connect all elements visually.

Membrane channels can be modelled as special components. Using the empirically confirmed relationships (equations) from the Hodgkin's and Huxley's article [6], we shall create a new type of an electric component describing the behaviour of membrane channels as constant voltage sources (pursuant to the Nernst equation) with changing inner resistance dependent on the membrane voltage. And finally, it only remains to connect all the elements visually.

Upon running the simulation of the entire circuit actuated by electric current pulses, the required course of the action potential and conductivity of the channels during an impulse is obtained (Fig. 13).

7.2 Implementation of the Model according to Hodgkin and Huxley in Simulink

In implementation of the model in Simulink, the model structure corresponds more to the calculation process than the physical structure of the modelled system. Comparison of the model implementation in Simulink (Fig. 14) and Modelica (Figs. 12 and 13) is very eloquent.

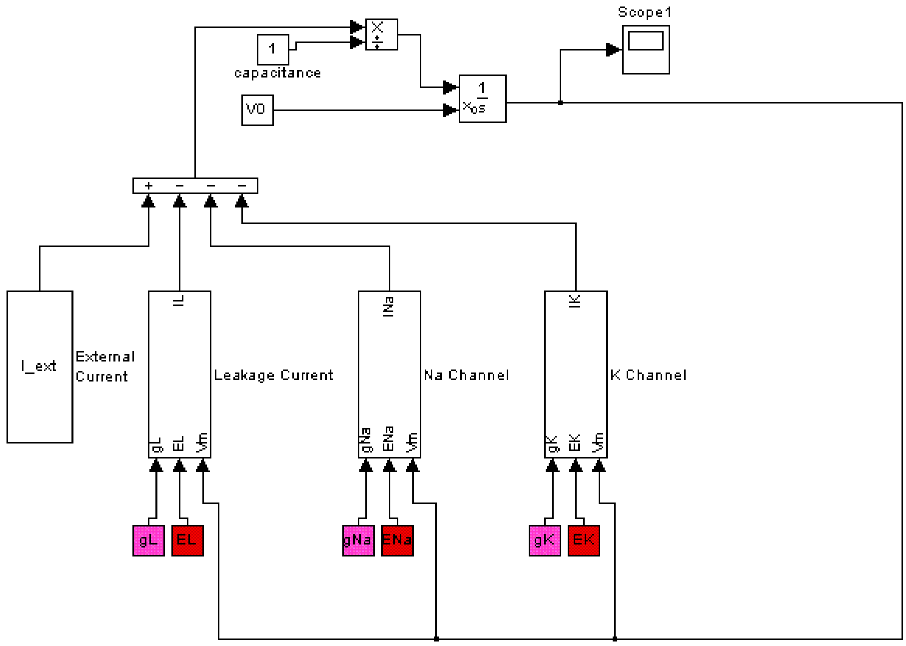

Figure 14: The Hodgkin-Huxley's model [6] can also be implemented using the block-oriented Matlab/Simulink environment; however, this environment remains tied to the calculation procedure.

8 Conclusion

New technologies provide new opportunities and challenges in simulation models creation. One of such technologies is represented by acausal simulation environments, such as the library Simscape in Simulink and especially the new object simulation language ***Modelica***, which shall make modelling of complicated and complex systems such as physiological systems considerably easier.

In respect of the fact that numerous relationships are found in physiological models (leading to solution of implicit equations in block-oriented languages), the ***acausal description of modelled relationships*** used in Modelica (but also in Simscape) provides a great advantage. Acausal description captures the essence of the modelled reality much better, and the simulation models are much more legible and thus also less prone to mistakes. Acausal simulation tools therefore represent a very suitable environment in physiological systems modelling.

References

[1] Abram, S.R., Hodnett, B.L., Summers, R.L., Coleman, T.G., Hester R.L.: Quantitative Circulatory Physiology: An Integrative Mathematical Model of Human Physiology for medical education. *Advanced Physiology Education*, 31 (2), 2007, 202 - 210.

[2] Bassingthwaighte J. B. Strategies for the Physiome Project. *Annals of Biomedical Engineering* 28, 2000, 1043-1058.

[3] Guyton AC, Coleman TA, and Grander HJ. (1972): Circulation: Overall Regulation. *Ann. Rev. Physiol.*, 41, p. 13-41.

[4] Fritzson P. (2003). *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1,* Wiley-IEEE Press.

[5] Hall J.E. (2004): The pioneering use of system analysis to study cardiac output regulation. *Am.J.Physiol.Regul.Integr.Comp.Physiol.* 287:R1009-R10011,2004,287: p. 1009-1001.

[6] Hodgkin, A.L., Huxley A.F.: The components of membrane conductance in the giant axon of Loligo. *Journal of Physiology,* 116, 1952, p. 473-496.

[7] Hunter P.J., Robins, P., Noble D. (2002) The IUPS Physiome Project. *Pflugers Archive-European Journal of Physiology*, 445, p.1–9

[8] Kho M.C.K. (2000) *Physiological control systems*. IEE Press, New York 2000, ISBN 0-7808-3408-6

[9] Kofránek J., Andrlík M., Kripner T, and Mašek J.: From Simulation chips to biomedical simulator. Amborski, K. and Meuth, H. 2002. Darmstadt, SCS Publishing House. *Modelling and Simulation 2002 Proc. of 16th European Simulation Multiconference*, p. 431-436 (full text version: http://patf-biokyb.lf1.cuni.cz/wiki/technical_computing_2007).

[10] Kofránek, J, Rusz, J., Matoušek S., (2007): *Guytons Diagram Brought to Life - from Graphic Chart to Simulation Model for Teaching Physiology.* In Technical Computing Prague 2007. Full paper CD-ROM proceedings. (P. Byron Ed.), Humusoft s.r.o. & Institute of Chemical Technology, Prague, ISBN 978-80-78-658-6, 1-13, 2007. The article including source program codes is available at http://www.humusoft.cz/akce/matlab07/sbor07.htm#k

[11] Raymond, G. M., Butterworth E, Bassingthwaighte J. B.: JSIM: Free Software Package for Teaching Physiological Modeling and Research. *Experimental Biology* 280, 2003, p. 102-107

[12] Van Vliet, B.N., Montani J.P. (2005):,Circulation and fluid volume control. In: *Integrative Physiology in the Proteomica and Post Genomics Age.* Humana Press, 2005, ISBN 918-1-58829-315-2, p. 43-66

Acknowledgments

This work has been supported by research MSM 0021620806 and by the aid grant MŠMT 2C06031, and by Creative Connections s.r.o.

Jiří Kofránek, M.D., Ph.D.
Laboratory of Biocybernetics,
Institute of Pathological Physiology, First Faculty of Medicine,
Charles University in Prague
U Nemocnice 5
128 53 Praha 2
e-mail: kofranek@gmail.com
Tel.: 777 68 68 68