# Component-based in-silico patch-clamp and whole-cell simulator using CellML

Marcellinus Aroli[1*], Ki Moo Lim[1]
IT Convergence Engineering, Kumoh Institute of Technology, South Korea
*armarcell@kumoh.ac.kr

**Abstract**

The initiation of Physiome Project leads to several development of biological framework. One of the result is CellML, an XML-based markup language for storing the information of cellular model. Several simulation program has been develop based on CellML However, those program have limitation to conduct a research for individual channel. Conducting a research from individual channel, commonly by patch clamping, is vital for several research application, mainly for developing a new cell model. This paper will explain about the development of *in-silico* simulator program that can provide patch-clamping simulations and combining different ion channel from various cell models to create a new cell models.

## 1. Introduction

During its early establishment, the Physiome Project aims to make a framework of biological structure and physiological functions at multiple scale of time and space [1]. From this idea, several biological framework have been created, one of them is CellML [CEL]. CellML enable researchers for storing and sharing the existing cell models so that they can reproduce the result and customize the parameters of the cell models based on their research interest. Additionally, Physiome Project provides a repository of CellML files to facilitate the researcher for making the CellML cell model files more accessible to other researchers [3].

Several simulation programs have been developed using CellML as their key component. One of the common simulation program is OpenCOR [4]. OpenCOR has the capabilities to simulate the CellML file and also converting the file into a programming language of user choices, such as C or Python. Overall, OpenCOR can help researcher for conducting the simulation study using existing cell model. However, OpenCOR and most of the simulation program using CellML don't have any capabilities to conduct a simulation for making a new cell model. In order to develop a new cell model, researchers need to conduct experiments to the individual ion channel by using patch-clamping. With the current version of CellML, such things cannot be done.

This study attempts to develop a simulator using CellML but with extended features for conducting *in-silico* simulation for single or multiple ion channels. Additionally, this study aims to make a simulator for creating a new cell models by combining different ion channels from various cell models.

## 2.System Overview

The simulator program consists of several component represented by files of code snippet that contains the mathematical formula of cell models. Each component describes one particular ion channel of a certain cell model. Component files are created from the CellML file of particular cell model [2]. CellML file is downloaded from the CellML Model Repository, then converted into C language source code using C Code Generation Service (CCGS) module provided by CellML API [5]. Then, each individual components are generated from the C source code based on ion channels and concentrations that existed in the particular cell model. All component files have name that represent the component that belong to a certain cell model. For instance, component file IKr$tn2004.dat describes the IKr channel from ten Tusscher et al. 2004 cell model [6].

The contents of each component are shown in the Figure 1. There are 5 sections inside the file, namely Constants, States, Algebraic, Rates, and Connection. The section classification is based on how CellML API generate the source code from the CellML file via C Code Generation Service (CCGS) module [5]. Constants section contains information about conductance and several experimental constant values, such as temperature, cell geometry, external ion concentration, etc. Algebraic section consists of mathematical equation of ionic currents and properties of ionic gate, e. g. steady-state values and time constants. Rates section defines the ordinary differential equations (ODEs) of the ion channel, such as the states of the gates, ion concentration, and membrane potential, while States section holds the solution of the ODEs and their initial values. Lastly, the Connection section is similar with the States section, albeit it has different purpose. It consists of initial values of states belonging to other components, but without their respective ODEs. In the Figure 1, the initial value of membrane potential V, potassium concentration K_i, and sodium concentration Na_i belong to other components, but all of the formula inside Algebraic section need those states. This section will ensure that each of the component can function properly without depending on other data files.

The simulation program provides two types of simulations, those are patch-clamp simulation and whole-cell simulation. Patch-clamp simulation uses a single component for simulating one individual channel to observe the characteristics of the channel under several voltage value. The input of the patch-clamp simulation is the step

```
[CONSTANTS]
CONSTANTS[g_Ks] = 0.062;
CONSTANTS[K_o] = 5.4;
CONSTANTS[Na_o] = 140;
CONSTANTS[P_kna] = 0.03;

[STATES]
STATES[Xs] = 0;

[ALGEBRAIC]
ALGEBRAIC[E_Ks] =  (( CONSTANTS[R]*CONSTANTS[T])/CONSTANTS[F])*
log((CONSTANTS[K_o]+ CONSTANTS[P_kna]*CONSTANTS[Na_o])/(STATES[K_i]+
CONSTANTS[P_kna]*STATES[Na_i]));
ALGEBRAIC[i_Ks] =  CONSTANTS[g_Ks]*pow(STATES[Xs], 2.00000)*(STATES[V] - ALGEBRAIC[E_Ks]);
ALGEBRAIC[alpha_xs] = 1100.00/ pow((1.00000+exp((- 10.0000 - STATES[V])/6.00000)), 1.0 / 2);
ALGEBRAIC[beta_xs] = 1.00000/(1.00000+exp((STATES[V] - 60.0000)/20.0000));
ALGEBRAIC[tau_xs] =  1.00000*ALGEBRAIC[alpha_xs]*ALGEBRAIC[beta_xs];
ALGEBRAIC[xs_inf] = 1.00000/(1.00000+exp((- 5.00000 - STATES[V])/14.0000));

[RATES]
RATES[Xs] = (ALGEBRAIC[xs_inf] - STATES[Xs])/ALGEBRAIC[tau_xs];

[CONNECTION]
STATES[V] = -86.2;
STATES[K_i] = 138.3;
STATES[Na_i] = 11.6;
```

Figure 1. The content of the data file of IKs from ten Tusscher et al. [6].

voltage. Step voltage has adjustable low and high voltages and step start time and end time. Using various step voltage configurations, researcher can observe the characteristics of the particular ion channel and measure the steady-state value and time constant of each gate in the channel within various conditions.

As opposed to the patch-clamp simulation, the whole-cell simulation uses multiple components combined into a single source code of cell model which has unique ion channels. The cell model can be consists of either components from the same cell model or different cell model, but with unique ion channels. For instance, researchers can create their own cell models by combining sodium current from ten Tusscher et al. [6], potassium current from Courtemanche et al. [7], L-type Calcium current from O'Hara Rudy et al. [8], and transient outward current from Tomek et al. [9]. For the whole-cell simulation, stimulus current is used as the input within interval of time. By varying the current input properties, researcher can observe the characteristics of their own cell model by checking the generated action potential and each individual currents from those components.

The main module of the simulator is written using C++ with Qt library, and CVode is used for solving the differential equations inside the components [10].

## 3. Simulation Result

We conducted the patch-clamp simulation using the IKr channel from ten Tusscher et al., as shown in the formula in Figure 2 [6]. There are two gates in IKr channel, namely Xr1 and Xr2. The patch-clamp simulator can produce the time-series result of the gates within various voltages. Based on that result, we can get the steady-state value and time constant result from *in-silico* experiments, as shown in the Figure 3, Figure 4 and Figure 5, respectively [6]. The results from our patch-clamp simulator (coloured markers) follow the line from the reference paper, which is showing that our patch-clamp simulator works as the experimental result from the reference.

$$I_{\mathrm{Kr}} = G_{\mathrm{Kr}} \sqrt{\frac{\mathrm{K_o}}{5.4}}\, x_{r1} x_{r2} (V - E_{\mathrm{K}})$$

Figure 2. The formulation of IKr from ten Tusscher et al. [6].

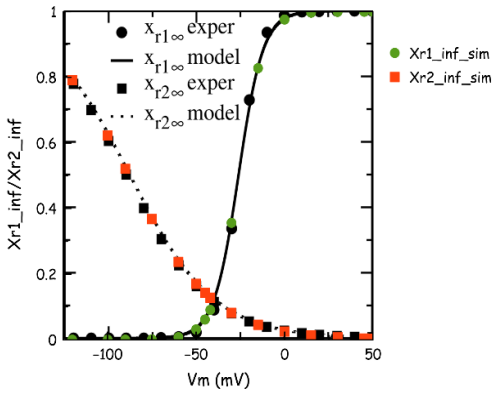


Figure 3. The steady-state graph within various voltages. The colored marker is the result from simulator.

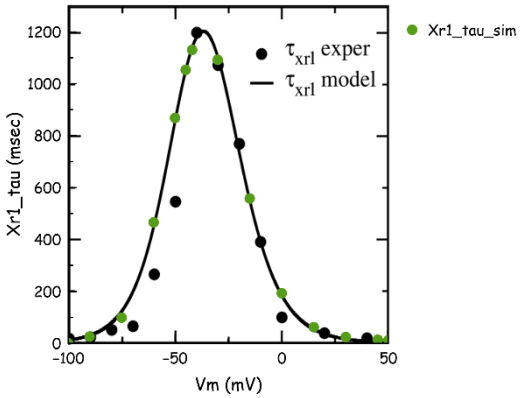


Figure 4. The time constants graph of Xr1. The colored marker is the result from simulator.

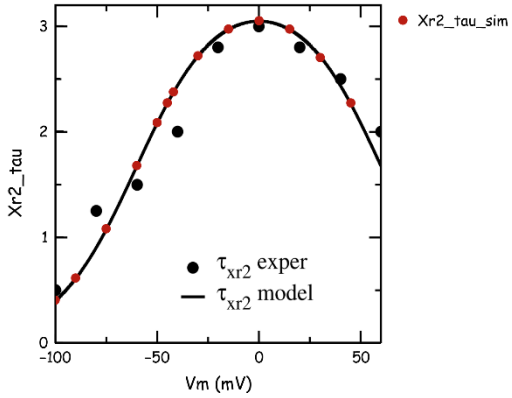


Figure 5. The time constants graph of Xr2. The colored marker is the result from simulator.

## 4. Acknowledgements


이 연구는 National Research Foundation of Korea (NRF) 과제의 지원을 받아 수행하였음. (NRF-2011-0031866 and NRF-2018M3A9D7079485)


## 5.References


[1] Hunter, P. J., & Borg, T. K. (2003). Integration from proteins to organs: the Physiome Project. Nature Reviews Molecular Cell Biology, 4(3), 237–243.

[2] Garny, A., Nickerson, D. P., Cooper, J., Santos, R. W. dos, Miller, A. K., McKeever, S., Nielsen, P. M. ., & Hunter, P. J. (2008). CellML and associated tools and techniques. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 366(1878), 3017–3043.

[3] Lloyd, C. M., Lawson, J. R., Hunter, P. J., & Nielsen, P. F. (2008). The CellML Model Repository. Bioinformatics, 24(18), 2122–2123.

[4] Garny, A., & Hunter, P. J. (2015). OpenCOR: a modular and interoperable approach to computational biology. Frontiers in Physiology, 6.

[5] Miller, A. K., Marsh, J., Reeve, A., Garny, A., Britten, R., Halstead, M., Cooper, J., Nickerson, D. P., & Nielsen, P. F. (2010). An overview of the CellML API and its implementation. BMC Bioinformatics. 11, 178 (2010).

[6] ten Tusscher, K. H. W. J., Noble, D., Noble, P. J., & Panfilov, A. V. (2004). A model for human ventricular tissue. American Journal of

Physiology-Heart and Circulatory Physiology, 286(4), H1573–H1589.
[7] Courtemanche, M., Ramirez, R. J., & Nattel, S. (1998). Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. American Journal of Physiology-Heart and Circulatory Physiology, 275(1), H301–H321.
[8] O'Hara, T., Virág, L., Varró, A., & Rudy, Y. (2011). Simulation of the Undiseased Human Cardiac Ventricular Action Potential: Model Formulation and Experimental Validation. PLoS Computational Biology, 7(5), e1002061.
[9] Tomek, J., Bueno-Orovio, A., Passini, E., Zhou, X., Minchole, A., Britton, O., Bartolucci, C., Severi, S., Shrier, A., Virag, L., Varro, A., & Rodriguez, B. (2019). Development, calibration, and validation of a novel human ventricular myocyte model in health, disease, and drug block. ELife, 8.
[10] S. D. Cohen and A. C. Hindmarsh, "CVODE, A Stiff/Nonstiff ODE Solver in C," Computers in Physics, 10(2), 1996, pp. 138-143. Also available as LLNL technical report UCRL-JC-121014 Rev. 1, August 1995.