Project no. 248992

Project acronym: NEUNEU

Project title: Artificial Wet Neuronal Networks from Compartmentalised Excitable Chemical Media

Small or medium-scale focused research project (STREP)

Deliverable 3.3 - Implementation of a computational model (software published on website)

Period covered: from 1.2.2010 to 15.2.2011 Date of preparation: 18.9.2009

Start date of project: 1.2.2010 Duration: 36 months

Project coordinator name: Dr. Peter Dittrich
Project coordinator organisation name: Friedrich Schiller University Jena

Technical Report on Simulation Software for Droplet Computers

Gerd Gruenert, Peter Dittrich

February 22, 2012

1 Introduction

An event-driven modelling approach for droplets of excitable media such as the Belousov-Zhabotinsky reaction is described in our report[1]. An implementation of this modelling framework is available as our droplet simulator `DropSim` on our website[2]. It is written in C++ and distributed as free software under the terms of the Gnu Public License[3] version 3.

The droplet kinetics are governed by defining a state transition distribution between the *excited*, the *refractory* and the *responsive* phases. Different implementations of the droplet kinetics can be plugged into the simulator, allowing deterministic and stochastic simulation modes. Currently, we have implemented Poisson and Normal Distributions to define the state transitions. By choosing zero variance values, deterministic system behaviour can be simulated.

In contrast to differential equations or cellular automaton simulation of the excitable medium, the discrete-event approach is chosen because it allows a simple *ad hoc* parameterisation of the reaction kinetics and fast simulations for large droplet systems. Computational complexity scales with $O(t \cdot n \cdot log(n))$ with the time t and the simulated number of droplets n.

Arbitrary network topologies can be analysed in `DropSim`, even though droplet computer realisations in the lab are currently mostly in two dimensions and though the simulation can graphically be visualised during the simulation on a two dimensional grid only. The core simulator, in contrast,

[1] G. Gruenert, J. Szymanski, J. Holley, G. Escuela, B. Ibrahim, J. Gorecki, and P. Dittrich. Multi-scale Modelling of Computers made from Excitable Chemical Droplets. *NEUNEU Technical Report*, 2012

[2] http://www.neu-n.eu

[3] http://www.gnu.org/licenses

does not use a positional representation of the droplet network but describes the network as a graph where each vertex is a droplet and each edge connects droplets that can trigger each other's excitations.

A number of droplets with non-standard behaviours are defined that correspond to the hypothetical droplet types with varying properties like differential excitability or oscillation periods to create for example *And* or *Diode* droplet as introduced in our report[1]. Furthermore, droplets can be marked as input or output droplets, such that certain droplets in the simulation are time-dependently stimulated and their state trajectories can be analysed by further programs, e.g. for optimisation experiments.

The simulator tool is mainly intended for command line use or for linking its library against other software to perform experiments. Below we give a short overview and explanation on the command line options.

2 Installation

To build the simulator, you will need installed `cmake`[4] and `xerces-c`[5] packages. To install the `DropSim` package, unzip the source archive to a new folder with subdirectories and enter the created folder `DropsimPackage_<date>` with a terminal program. Type `make` will create a subdirectory `build` that will contain the `DropSim` binary if no problems occur.

If building seems to succeed, you can test the executable by running some test simulations through the command `make test`.

3 User Interface

`DropSim` is called from the command line with the following options:

`-inp` <type> <file/string>
: Using the `-inp` option, different input filters can be applied to read a droplet network from a file or from a string at the command line into the simulator. This option is not to be confused with the `-stim` option: the `-inp` option does not allow a differential excitation of specific droplets at simulation time but is used to build the network topology for the simulation. Followingly, the possible <type> and <file/string> options are listed:

[4] http://www.cmake.org/
[5] http://xerces.apache.org/

type	Further Parameters	Meaning
`map`	<map file>	ascii "drawing" of the droplet network
`bnglgraph`	<string>	BNGL[6] graph strings can be interpreted as a network of droplets
`gplot`	<map file>	two dimensional data array as used by gnuplot. Moore Neighbourhood connectivity.
`gplot4`	<map file>	two dimensional data array as used by gnuplot. Von Neumann connectivity.

`-outp` <type> [<file>] [<time raster>]
Using the `-outp` option, different output filters can be applied to write the droplet activity to a file or render it on the screen.

type	Further Parameters	Meaning
`tui`	-	Text user interface. The network topology is drawn on an ASCII terminal
`rec`	<output file>	The activity of the output droplets is recorded to a file.
`rec_spikes`	<output file>	The activity of the output droplets is written in a different format.
`rec_gp_matrix`	<output file> <time raster>	The activities of the output droplets are written to a two dimensional data array as used by gnuplot.

Using the text user interface, the simulation speed is reduce to allow human tracking of the excitation waves. The keys `+` and `-` can then be used to increase and decrease the simulation speed. The timing scheme can be changed using the `r` key. The simulation screen can be left by typing `q`.

`-parKinFile` <file>

[6]BioNetGen is a software for rule-based biochemical models: `http://bionetgen.org/`

An XML file that describes the timing properties of the droplet types, their input/output behaviour as well as special characteristics that are used in the simulation. An example file is described in Section 3.1.

`-numSteps` <int>

This option gives the number of simulation steps until the program will terminate. Without either this option or the next option `-maxTime`, the simulation will not halt until the program is killed.

`-maxTime` <time>

This option gives the the length of the simulation in time units until the program will quit.

`-rand` <int>

The random seed that will be used by the simulator. When using the same droplet network and random seed, a simulation can be repeated identically. If no random seed is given it will be generated using the current time.

`-stim` <stimfile>

To test the droplet network using binary inputs and outputs, a stimulation script given in `stimfile` can be used. It defines different simulation phases, which input channels are stimulated during these phases and which outputs channels are expected to be stimulated. When the droplet network is expected to fulfil a binary input-output mapping, the quality of the mapping can be evaluated to a real number automatically. An example file for such an automatic evaluation is supplied as described in the next Section.

`-dynastim` <stimfile> <time-raster>
Dynamic stimulation patterns are fed into the droplet network at simulation time. In contrast to the phase-based stimulation when using the `-stim` option, the dynamic stimulation of each channel can be changed in tiny intervals. This interval size $\triangle t$ is given with the <time-raster> parameter. This option does not allow an automatic evaluation of computational success.

3.1 Example Files

All example files can be found in the folder `exampleFiles` of the `DropSim` distribution. A simulation can for example be run, visualised and recorded by the command:

```
dropsim -inp map dropnet.neu -parKinFile droptypes -outp tui \
-outp rec_spikes spikes.dat -maxTime 5000
```

`dropnet.neu` An example ASCII graphics droplet network that can be read using the parameters `-inp map dropnet.neu`.

`dropnet.gp` An example gnuplot matrix file, which defines a droplet network and can be read using the command line options `-inp gplot dropnet.gp`.

`droptypes` In this XML file, entries of the name `Subtype` and their `id` are listed for each type of droplet that can be used in the simulation. Further parameters to this tag are the droplet type `DropTp`, and the input and output channels `ioIn` and `ioOut`. The droplet type can be `DT_NORMAL`, `DT_AND`, `DT_DIRECT_A`, `DT_DIRECT_B` or `DT_INHIBIT`, representing normal, less excitable, diode or inhibitive droplets. Diode droplets are realised through two droplets, where any droplet can excite `DT_DIRECT_A` droplets but only `DT_DIRECT_A` droplets can excite `DT_DIRECT_B` droplets.

Child tags of `Subtype` with the name `Request` define the droplet timing behaviour for the transitions from *responsive* to *excited* (`RQ_LEAVE_RESPONSIVE`), from *excited* to *refractory* (`RQ_LEAVE_EXCITED`), from *refractory* to *responsive* (`RQ_LEAVE_REFRACTORY`) and finally for the signal transmission delay (`RQ_SIGNAL_PROP`).

`AndStim` Stimulation file for the option `-stim`, implementing the truth table of the *And* funtion in this case.

`dynastim` A gnuplot matrix file that can either be plotted by gnuplot or can be interpreted as a two dimensional matrix containing the temporal activation pattern for different input channels. Each row defines the states for one input channel where the values of $0, 1$ in row r_i define the stimulation in the time interval $[i \triangle t, (i+1) \triangle t[$ to be either active or inactive.

Acknowledgements

The research was supported by the NEUNEU project (248992) sponsored by the European Community within FP7-ICT-2009-4 ICT-4-8.3 - FET Proactive 3: Bio-chemistry-based Information Technology (CHEM-IT) program.