**Molecular Watchdog Timers for Heartbeat Monitoring SOPs by Blake Gerold**

This documentation is for future students who are interested in working on chemical reactions networks (CRNs) and using Kaemika as a modeling tool. The majority of my work is based on Ellis’s molecular watchdog timers CRNS and modeling them using the Kaemika programming language. [1][2] Future work may involve modifying these watchdog timer (WDT) CRNs for interrupting electrocardiography (ECG) data. I was able to simulate an ECG output in Kaemika using M. A. Quiroz-Juárez *et* *al* research paper but ran out of time to modify the WDT for this purpose.[3] I would recommend reading my abstract along with the code that should be bundled with this document. I will make sure to properly label code (they are saved as .txt files) to allow future users to easily find the code that they are looking for.

**Hints and Tricks for Kaemika**

Using Kaemika has been an enjoyable but at times a frustrating experience mostly because of the lack of documentation. If you are ever stuck with an issue in Kaemika, I would recommend looking through all of the example code for the Kaemika app. This can be found in the top left hand corner of the app and appears as 5 horizontal lines. The ones that found most useful for this project are “StartHere” (simple programing example with syntax), “Reactions” (this is where I learned about “trigger” keyword) and “LorenzAttractor” (I found here here the how to directly add ordinary differential equations in the Kaemika equation. Very useful for ECG research.)

**Trigger:** This was a very useful aspect of Kaemika. It allows for an “Event” to occur. I used this keyword to spike my system with a high amount of molecules to replicate a heartbeat. For example, I would use “trigger” every second to simulate a heartbeat rate of 60 beats per minute (bpm). I varied this from every 0.5 seconds to 2 seconds to get heart beat rates of 120 bpm to 30 bpm respectably. To replicate 120 bpm over the course of 15 seconds requires the word trigger to be repeated 30 times at the *exact time* (example 0.5 s, 1.0 t, 1.5 t) I wanted so the code gets very long with this method. Once I established the time frame that I wanted for the experiments, it was simple as copying the and pasting the rows of trigger codes. I went with 120, 90, 60 and 30 bpm over the course of 60 seconds due to the ease of calculating when the trigger needed to occur over those time frames. I was not able to figure out how to code this as a function in Kaemika so it might be worth your time to figure this out for ease of use and a quicker way to modify the code. This also may not be possible given Kaemika’s design so do not be too worried if you cannot figure out a way to replicate the heartbeat rates. You might be able to forgo the multiple trigger method of replicating heartbeats similarly to how I was able to replicate ECG signals but you will need to research on your own how to accomplish this.

NOTE: Although you can use triggers multiple times, whenever I tried to graph the trigger spikes in Kaemika, after the first visual spike in concentration on the graph other spikes have not been seen on the graph. The concentration spikes still occur because otherwise the heart beats would not be able to be simulated, they just do not visually appear on the graph if you try to graph them. I do not know if this user error or an error with Kaemika. Those took me a while to figure out on my own and I could not find documentation on why this is the case.

**ECG Simulation**: I have the ecg simulation programmed by inserting the four ODE directly into the program. I was then able to add the four ODE’s together in the report section and labeling them as ECG(t) with the following code (**report (α₁\*x₁ + α₂\*x₂ + α₃\*x₃ + α₄\*x₄) as "ECG(t)"**). One other nice aspect of this ECG code that I made is that you can change the C, H, β and α variables to replicate different ECG outputs in order to simulate heart defects. I never got around to changing these variables so at the moment the only ECG I plotted was the one for a standard health heart. Refer to the “Generation of ECG signals from a reaction-diffusion model spatially discretized” paper to further research on what these variables need to be to simulate various heart defects.

[1] S. J. Ellis, “Designing a molecular watchdog timer for Safety Critical Systems - core,” *Iowa State University*, 2014. [Online]. Available:https://dr.lib.iastate.edu/server/api/core/bitstreams/7bdaa0c8-d693-42a8-aee0-b166d918871f/content. [Accessed: 01-Feb-2023].

[2] L. Cardelli, “Kaemika app, integrating protocols and Chemical Simulation,” *arXiv.org*, 16-May-2020. [Online]. Available: https://arxiv.org/abs/2005.08097. [Accessed: 01-Feb-2023].

[3] M. A. Quiroz-Juárez, O. Jiménez-Ramírez, R. Vázquez-Medina, V. Breña-Medina, J. L. Aragón, and R. A. Barrio, “Generation of ECG signals from a reaction-diffusion model spatially discretized,” *Nature News*, 12-Dec-2019. [Online]. Available: https://www.nature.com/articles/s41598-019-55448-5. [Accessed: 24-Apr-2023