SYSTEM-THEORETIC ANALYSIS OF NANO COMMUNICATION SIGNALS

Zeynep - Tandoğan zeyneptandogan@sabanciuniv.edu

Computer Science and Engineering & Electronics Engineering, Junior

Emirhan - Uçar emirhanucar@sabanciuniv.edu

Electronics Engineering, Junior

Esra Ceylin - Bormalı cbormali@sabanciuniv.edu

Electronics Engineering, Junior

Muhammed Orhun - Gale morhun@sabanciuniv.edu

Computer Science and Engineering, Sophomore

Suzan - Kanıgür suzankanigur@sabanciuniv.edu

Undeclared/FENS Sophomore

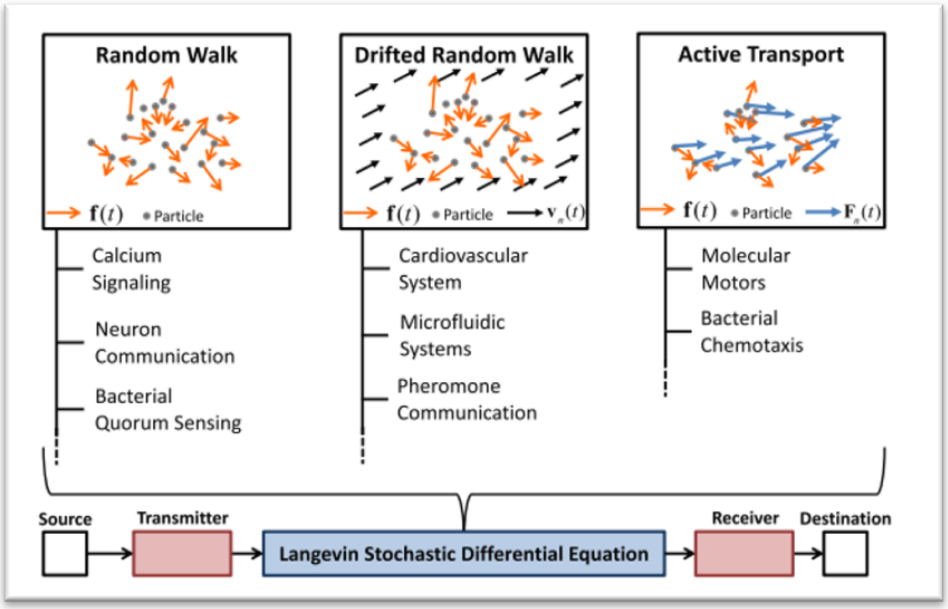
Ozan - Biçen ozan.bicen@sabanciuniv.edu

Electronics Engineering

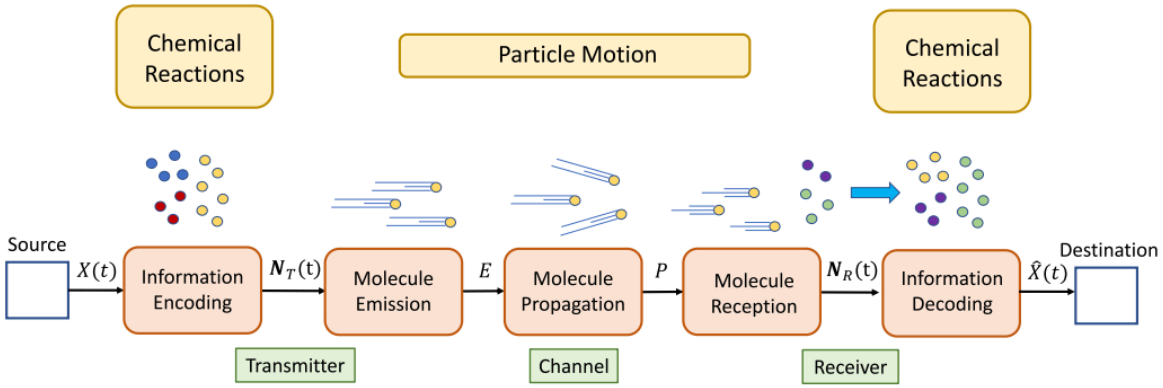
Section 1: Chemical Reaction Prediction and Molecular Communication

1 Introduction

Molecular communication has become a key transformative paradigm in the communication field especially in the last 10 years. The definition of a molecular communication system is that it is a system that uses absence or presence of the molecules that are involved in the process. ıts focus is on the modeling, characterization, and engineering of information transmission through molecule exchange, with immediate applications in biotechnology, medicine, ecology, and defense, among others. (Akyıldız, Pierobon, 2019)



Since the main focus of molecular communication is molecular propagation and reactions, randomness is definitely included in the calculations. The randomness types such as random walk and the events in which they are observed are included in the chart above. In short, molecular communication is supported by chemical kinetics and statistical mechanics and modeling.



The chart that is given above shows a diffusion based molecular communication system. In this project, our aim is to follow steps similar to the ones in the above and analyse the results. While doing this, we will take our reviews one step further by making prediction with machine learning algorithms.

2. Chemical Reaction Prediction and Molecular Communication

2.1. The analysis of the concepts with articles

During the project process, we read many articles about the project and its concepts, which require detailed information. The details that we will note here are the parts that we have learned throughout the project, and they have played an important role in developing the project in our mind.

- **[1] Modeling and Simulating Chemical Reactions - Desmond J. Higham**

This article forms the basis of our project in terms of information.

Let's say we have N different types of molecules (chemical species) and these molecules may take part in one or more of M types of chemical reactions.

Our first aim is to follow or observe the amount of their collision and the formation of their products. Since it is too difficult to keep track of the collision between molecules (molecular dynamics are too expensive!), the approach is to ignore some things like spatial information that will lead to some equations such as Chemical Langevin Equation and these will always have the randomness and probability with it.

There are several algorithms and equations for calculations:

1. Chemical Master Equation: Current state of the system, one ODE (ordinary differential eq.) for each possible state of the system.
2. Chemical Langevin Equation: One SDE (Stochastic diff. eq.) for each chemical species. Pushing approximation further than Tau Leaping.
3. Reaction Rate Equation: One ODE (ordinary diff. eq.) for each chemical species. Instantaneous rate of change is proportional to the product of concentrations of the reacting species. (without the stochastic part of CLE.)
4. Stochastic Simulation Algorithm: It gives one approach to computing indirectly with the CME, rather than solving the full set of ODEs we compute samples from it.
5. Tau Leaping: The approximation introduces errors that will be small as long as the state vector updates are relatively small. It is used with SSA.

The purpose by using MATLAB **is to get numerical simulations of the Michaelis-Menten system** using the SSA, CLE, RRE. In our project, we worked on the MATLAB code which is for Stochastic Simulation Algorithm.

The **Michaelis–Menten equation** is the rate **equation** for a one-substrate enzyme-catalyzed reaction. The changes in the substrates can be shown with a state vector.

In each implementation we have parameters and initial conditions. These are:

- Volume of system
- Molecules of substrate
- Molecules of enzyme
- Coefficient values

For calculations, there are some functions that already exist in MATLAB such as tau, rand, cumsum. These functions are used in SSA calculation.

In the simulation, the change in substrate and product is observed with respect to molecules vs time.

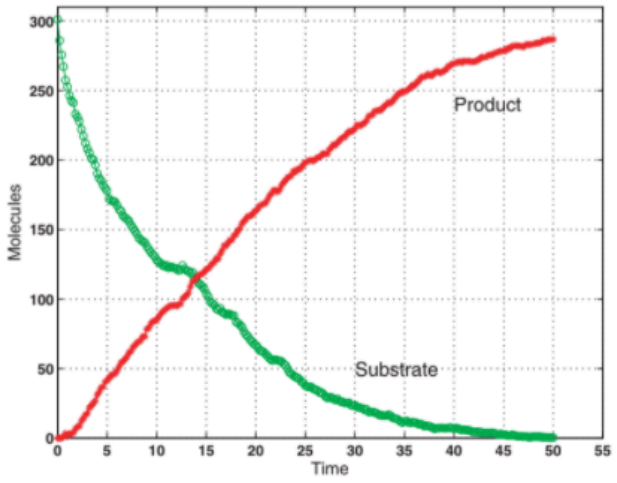


Fig. 3 *Molecules of substrate and product from an Euler–Maruyama simulation of the CLE for the Michaelis–Menten system.*

Stochastic Simulation Algorithm: The CME is too large to handle numerically. This fact moves us towards the stochastic simulation algorithm. SSA solves this problem by calculating single occurrences of the state vector rather than a complete probability distribution.

To derive SSA we need P0(τ |x, t), which is the probability that no reaction happens in the time interval [t, t + τ).

Time until Next Reaction

$$p(\tau, j|\mathbf{x}, t) = \frac{a_j(\mathbf{x})}{a_{\text{sum}}(\mathbf{x})} a_{\text{sum}}(\mathbf{x}) e^{-a_{\text{sum}}(\mathbf{x})\tau}.$$

Next Reaction Index

Next Reaction Index=pick one of the reactions with the rule that the chance of picking the jth reaction is proportional to aj (x).

Time until Next Reaction = The density function for a continuous random variable with an exponential distribution. These exponential random variables arise universally in descriptions of the time elapsing between unpredictable events.

- **[2] An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations - Desmond J. Higham**

This article aims to provide some mathematical models and algorithms for **stochastic differential equations**. At the same time, this article provides a serious background on the basics of how to simulate SDEs (Stochastic Differential Equations) numerically with just a background knowledge of Euler's method for deterministic ordinary differential equations and an intuitive understanding of random variables.

1.Brownian Motion: A *scalar standard Brownian Motion*, or *standard Wiener process*, over [0, T] is a random variable W(t) that depends continuously on $t \in [0, T]$ and satisfies the following three conditions.

a) W(0) = 0 (with probability 1).
b) For $0 \leq s<t \leq T$ the random variable given by the increment W(t)−W(s) is normally distributed with mean zero and variance t − s; equivalently, W(t) − W(s) ~ $\sqrt{t-s}$ N(0, 1), where N(0, 1) denotes a normally distributed random variable with zero mean and unit variance.
c) For $0 \leq s<t<u<v \leq T$ the increments W(t) − W(s) and W(v) − W(u) are independent.

This was the mathematical definition of Brownian motion. There is an analogy to understand the concept of Brownian Motion. This analogy includes a stadium full of spectators and one big huge ball. If this crowd tries to push the ball, it causes many forces on the ball which have different directions. So, the sum of these forces determines the direction of the ball which corresponds with collisions and forces implemented for each molecule.

```
%BPATH1  Brownian path simulation

randn('state',100)          % set the state of randn
T = 1; N = 500; dt = T/N;
dW = zeros(1,N);            % preallocate arrays ...
W = zeros(1,N);             % for efficiency

dW(1) = sqrt(dt)*randn;     % first approximation outside the loop ...
W(1) = dW(1);               % since W(0) = 0 is not allowed
for j = 2:N
   dW(j) = sqrt(dt)*randn;  % general increment
   W(j) = W(j-1) + dW(j);
end

plot([0:dt:T],[0,W],'r-')   % plot W against t
xlabel('t','FontSize',16)
ylabel('W(t)','FontSize',16,'Rotation',0)
```

This piece of code firstly preallocates arrays as a template then before the for loop obtains approximation outside loop. Then, increment process is occured for each index in for loop according to mathematical definition.

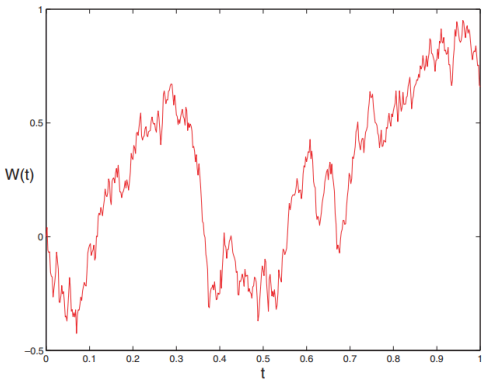


Fig. 1 *Discretized Brownian path from* `bpath1.m` *and* `bpath2.m`.

2. The Euler-Maruyama Method:

$$dX(t) = \lambda X(t)dt + \mu X(t)dW(t), \quad X(0) = X_0,$$

In Itô calculus, this is a method that can be applied stochastic differential equations on some interval of time [0, *T*], where λ and μ are real constants and W(t) stands for the **Wiener process**. This differential equation in above can be solved by this method recursively. Firstly, discretize the time for each index that desired to have a value for that index. Then, obtain the values recursively by solving the SDE.

$$Y_{n+1} = Y_n + a(Y_n)\,\Delta t + b(Y_n)\,\Delta W_n,$$

```
%EM  Euler-Maruyama method on linear SDE
%
% SDE is  dX = lambda*X dt + mu*X dW,   X(0) = Xzero,
%      where lambda = 2, mu = 1 and Xzero = 1.
%
% Discretized Brownian path over [0,1] has dt = 2^(-8).
% Euler-Maruyama uses timestep R*dt.

randn('state',100)
lambda = 2; mu = 1; Xzero = 1;    % problem parameters
T = 1; N = 2^8; dt = 1/N;
dW = sqrt(dt)*randn(1,N);         % Brownian increments
W = cumsum(dW);                   % discretized Brownian path

Xtrue = Xzero*exp((lambda-0.5*mu^2)*([dt:dt:T])+mu*W);
plot([0:dt:T],[Xzero,Xtrue],'m-'), hold on

R = 4; Dt = R*dt; L = N/R;        % L EM steps of size Dt = R*dt
Xem = zeros(1,L);                 % preallocate for efficiency
Xtemp = Xzero;
for j = 1:L
   Winc = sum(dW(R*(j-1)+1:R*j));
   Xtemp = Xtemp + Dt*lambda*Xtemp + mu*Xtemp*Winc;
   Xem(j) = Xtemp;
end

plot([0:Dt:T],[Xzero,Xem],'r--*'), hold off
xlabel('t','FontSize',12)
ylabel('X','FontSize',16,'Rotation',0,'HorizontalAlignment','right')

emerr = abs(Xem(end)-Xtrue(end))
```

In this piece of code, firstly it is defined the parameters for the problem and determined the step sizes for Brownian Motion and EM. Then it is created preallocated vector for efficiency. After that SDE is solved in the for loop for each index

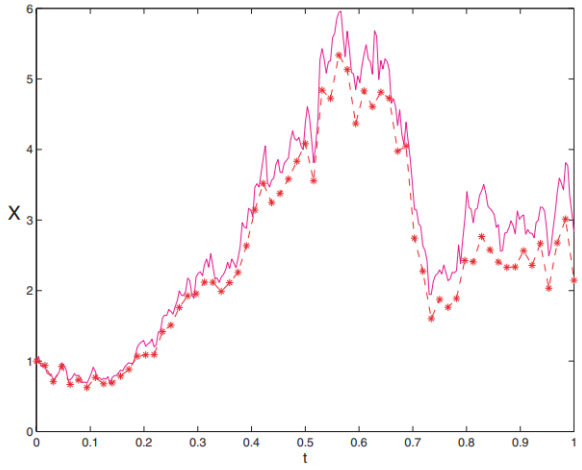


Fig. 3 *True solution and EM approximation, from* `em.m`.

- **[3] Extracting signal from noise: kinetic mechanisms from a Michaelis–Menten-like expression for enzymatic fluctuations - Jeffrey R. Moffitt and Carlos Bustamante**

Michaelis and Menten proposed that **the formation of an intermediate (i.e. the enzyme–substrate bound complex)** was a necessary step in the generation of the product. **Enzyme-catalyzed reactions are dominated by fluctuations**, which is related to stochastic quantities.

Statistical Kinetics: connection between statistical measures of enzymatic fluctuations and properties of the enzymatic mechanism. **Randomness parameter** is the simple quantification of fluctuations in an enzyme-catalyzed reaction.

Why and how do enzymes fluctuate?

The directionality comes from the **free energy released in the conversion of substrate to product.** It is thermal energy, the energy stored in the motion of the surrounding solvent molecules.

Why is the process stochastic?
Energy is transferred to the enzyme from **random collisions with the surrounding solvent**, these processes are naturally stochastic.

1. The number of kinetic intermediates
2. Their natural lifetimes
3. The order in which these intermediates are formed,
4. The number of times each intermediate is visited, all shape the fluctuations

Associated with the reaction of a single enzyme.

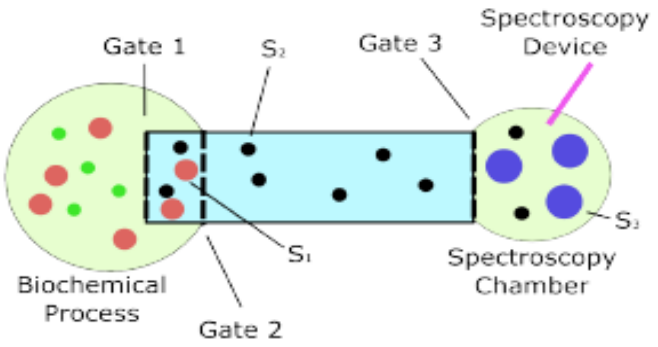
There are two types of fluctuations: fluctuations in the lifetime of a kinetic intermediate, fluctuations in the number of visits to each intermediate.

If an enzyme **has no memory** of how long it existed in a given state, its pdf is an exponential function. If an enzyme can produce multiple different products, then fluctuations can also occur in the choice of product.

This article has enabled me to obtain chemical and mathematical background information on the event we are examining.

- **[4] A molecular Communication Scheme to Estimate the State of Biochemical Processes on a Lab-on-a-Chip**

The purpose of this article is to reliably estimate the equilibrium state of the biochemical process and characterize this steady state regarding its parameters as enzyme, substrate, reaction rate and volume, etc. These estimations are based on **Gaussian Approximation**.



According to the adjacent figure, some environment is created which is divided into 2 chambers. One of them is the chamber that biochemical process occured and the other one is spectroscopy chamber, investigation chamber. First gate opens at **equilibrium state** allowing molecules S1 to diffuse inside. Second gate acts as a filter allowing only S2 further information carrier molecule.

There is a possibility that different species of molecules can react with each other. S1 and S2 can react, also in the spectroscopy chamber, S2 and S3 can react. Hence, it is necessary for one more gate to pass back to prevent S3 molecules. **Raman spectroscopy** can be used to analyze and track the biochemical process chamber. Actually, spectroscopy equipped with LoC can be used for both chambers but it requires special preparation.

System resets itself for each experiment in specific situations such as a biochemical process exposed to new environment conditions, change in temperature, change in input molecules, etc. If there does not exist S1,S2 and S3 molecules at the spectroscopy chamber at time t=0, after external stimulus, the biochemical process converges rapidly to an equilibrium state by an element {1,...,K}. At time δ, system shortly reach equilibrium, S1 is present at beginning of microchannel depending on the equilibrium state of the biochemical process.

Number of molecules S1 is defined as, $\Delta\ i = CiV\text{Tx}$ where Δi corresponds to the quantity of molecules of S1 in the beginning of the microchannel, as defined in (6). We note that $\delta > 0$ is chosen to be a sufficiently small period of time such that no reactions occur in the beginning of the microchannel.

The key idea behind our approach is that for sufficiently large Ts , the total number of molecules of species S2 and S3 in the detection chamber at the time of sampling will be approximately drawn from the stationary distribution of the **RDME** (Reaction-diffusion master equation).

To estimate the equilibrium state, **a Gaussian Mixture Model** is used which is based on expectation - maximization algorithm.

$fGM\ (x; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma})) = \boldsymbol{\Sigma}(x; \mu i\ , \sigma 2\ i\ )$ where the borders of sigma is from i = 1 to N.

where $g(x; \mu, \sigma 2\ ) = (1\ /\sqrt{}\ 2\pi\sigma 2) * \exp\ \ (-\ (x - \mu)\text{^}2\ /\ \ \text{^}(2\sigma)\ 2)$.

This method is basically, assigning each example to one cluster and finding an answer to what if there exists any overlapping between the clusters.

- **[5] Deep Learning for Deep Chemistry: Optimizing the Prediction of Chemical Patterns -Tânia F. G. G. Cova* and Alberto A. C. C. Pais**

This article focuses on the use and aims of deep learning and machine learning in chemical reaction in general.

The purpose: describing, solving and predicting chemical data and related phenomena with deep learning.

Reaction predictor: an application for **predicting chemical reactions and reaction pathways.** Deep learning is to predict and order elementary reactions.

- Identifying electron sources
- Pairing those sources and sinks to propose elementary reactions
- Ranking reactions by favorability

Fundamental problem in synthetic chemistry and its relation with ML: **Identification of unknown products observed via mass spectrometry** (a technique that lists ions with respect to their ion mass/load).

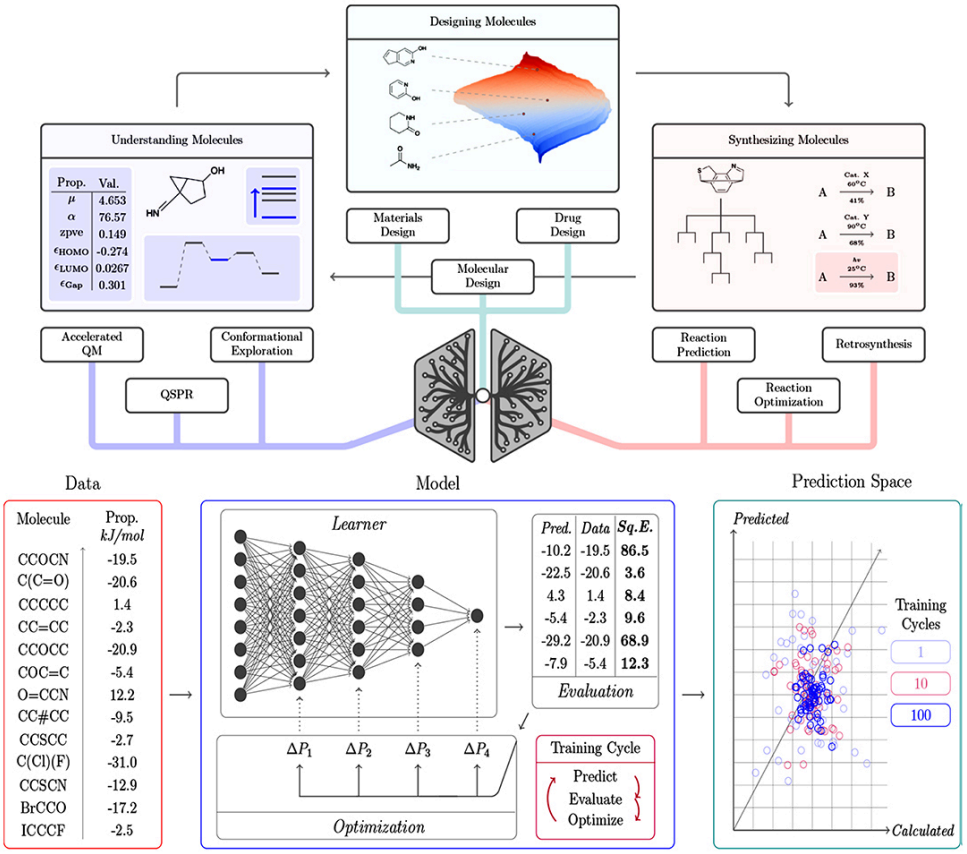
The areas that DL is used:

1. understanding and controlling chemical systems
2. calculating, optimizing, or predicting structure-property relationships
3. density functional theory (DFT) functionals, and interatomic potentials
4. driving generative models for inverse design
5. screening, synthesizing, and characterizing new compounds and materials etc.

Deep learning (DL) approaches can also be useful **to solving a variety of chemical problems, including compound identification and classification, and description of soft matter behavior.**

ML contributions have involved a variety of systems **including drugs, polymers, polypeptides, energetic materials, metal organic frameworks and organic solar cells.**

Besides the good effects of machine learning in chemical reactions, there are also some disadvantages. It is hard to clean data, produce an accurate chemical information that is free of bias and lack of standardization of chemical data.



This graph gives the big picture of the relation of the ML and chemistry.

Day by day, the role of Machine learning and deep learning is increasing thanks to the continuous growth of chemical data in public databases, **such as PubChem and Protein Data Bank**.

- **[6] A Novel Framework for Capacity Analysis of Diffusion-Based Molecular Communication Incorporating Chemical Reactions - Ching-Lun Tai and Ian F. Akyildiz**

The goal of this article is to evaluate the effects of the chemical reactions in the transmission process. After detailed calculations, the results are described by comparing capacity with other elements that have effect on the transmission. Due to the enhanced molecular noise with the increased distance, capacity decreases while the distance increases. They are inversely proportional.

If capacity and bandwidth are compared, it can be seen that they are proportional. Within a larger bandwidth, more information bits can be transferred; this shows that while bandwidth increases, capacity also increases. In the case of diffusion coefficient, when diffusion coefficient is large the transmission of molecules is increasing. It means that capacity and diffusion coefficient are proportional. In short, when distance is increasing and bandwidth is decreasing, the capacity is also decreasing. In addition to that, capacity is increasing if the bandwidth and diffusion coefficient are increasing.

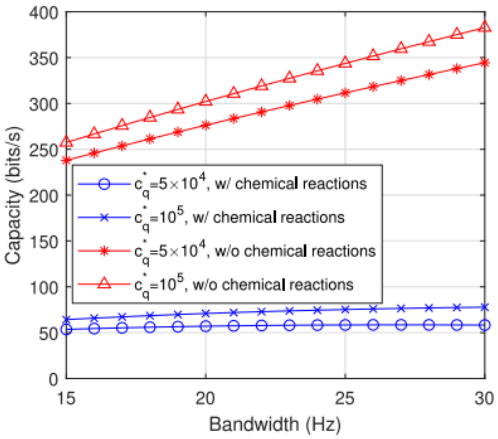


Fig. 4. Relationship between the capacity and the bandwidth under different c_q^* conditions with and without chemical reactions considered.

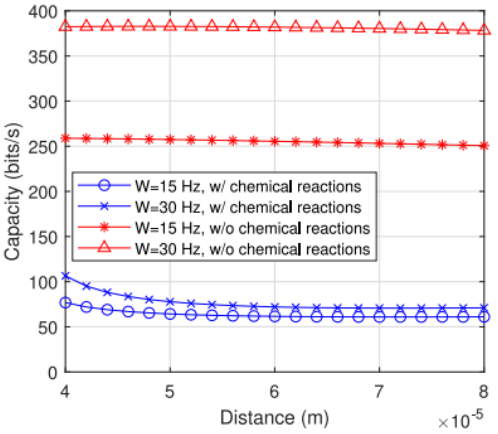


Fig. 6. Relationship between the capacity and the distance under different bandwidth conditions with and without chemical reactions considered.

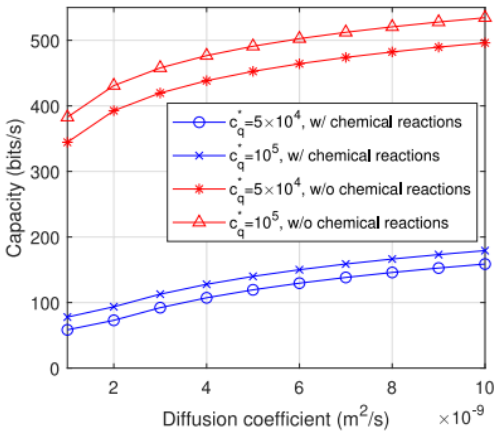


Fig. 5. Relationship between the capacity and the diffusion coefficient under different c_q^* conditions with and without chemical reactions considered.

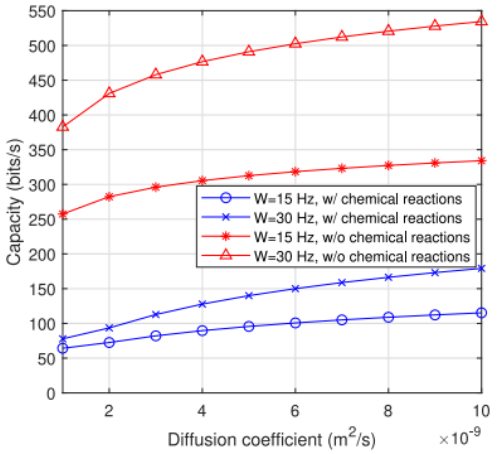


Fig. 7. Relationship between the capacity and the diffusion coefficient under different bandwidth conditions with and without chemical reactions considered.

2.2. Simulation and its analysis

trial_1.m

```
clear all;
clc;
rng('default')

r1 = randi(1000,1,5);

for z= 1:1:5
    rand('state',r1(z));

    %stoichiometric matrix
    V = [-1 1 0; -1 1 1; 1 -1 -1; 0 0 1];
    %%%%%%%%%% Parameters andInitial Conditions %%%%%%%%%
    nA = 6.023e23; % Avagadro's number
    vol = 1e-15; % volume of system
    X = zeros(4,1);
    X(1) = round(5e-7*nA*vol); % molecules of substrate
    X(2) = round(2e-7*nA*vol); % molecules of enzyme
    c(1) = 1e6/(nA*vol); c(2) = 1e-4; c(3) = 0.1;
    t = 0;
    tfinal = 50;
```

The code that is seen in the left shows the parameter values of the simulation and by using for loop 5 times we achieved to get random five different result with the use of randi function.

trial_1.m

```
22 -      i = 0;
23 -      Xsubt = zeros(tfinal - t-1,1);
24 -      Xenz = zeros(tfinal - t-1,1);
25 -      time = zeros(tfinal - t-1,1);
26 -      product = zeros(tfinal -t-1,1);
27 -      intermed = zeros(tfinal -t-1,1);
28
29 -      while t < tfinal
30 -      a(1) = c(1)*X(1)*X(2);
31 -      a(2) = c(2)*X(3);
32 -      a(3) = c(3)*X(3);
33 -      asum = sum(a);
34 -      j = min(find(rand<cumsum(a/asum)));
35 -      tau = log(1/rand)/asum
36 -      X = X + V(:,j);
37
38 -      i = i + 1;
39 -      Xsubt(i) = X(1);
40 -      Xenz(i) = X(2);
41 -      product(i) = X(4);
42 -      intermed(i) = X(3);
```

Before the calculations, in order to store the values we created vectors with the same size for each variable.

The calculations and reactions are done in the while loop, the equation for this calculation was given in part 2.1. Modeling and Simulating Chemical Reactions article.

trial_1.m

```
43 -      time(i) = t;
44
45 -      t = t + tau;
46
47 -      end
48
49 -      figure
50 -      plot(time,Xsubt);
51 -      hold on;
52 -      plot(time,Xenz);
53 -      plot(time,product);
54 -      plot(time,intermed);
55
56 -      legend('Substrate', 'Enzyme','Product','Intermediate Complex');
57 -      grid;
58 -      xlabel('Time','FontSize',10);
59 -      ylabel('Molecules','FontSize',10);
60 -      bw= bandwidth(intermed);
```

In this part, we plot the values that we get in the reactions.

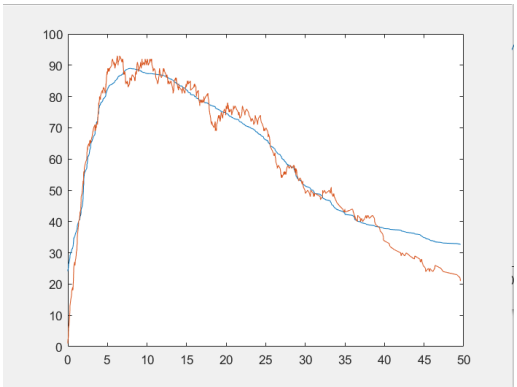
```
col_header={'time','Xsubt','Xenz','product','intermed'};
xlswrite('data.xlsx',[time(:), Xsubt(:),Xenz(:),product(:),intermed(:)],'Sheet1','A2');
xlswrite('data.xlsx',col_header,'Sheet1','A1');
```

	A	B	C	D	E
1	time	Xsubt	Xenz	product	intermed
2	0	300	119	0	1
3	0,006426	299	118	0	2
4	0,03547	298	117	0	3
5	0,035968	297	116	0	4
6	0,046661	296	115	0	5
7	0,05115	295	114	0	6
8	0,067569	294	113	0	7
9	0,08246	293	112	0	8
10	0,134683	292	111	0	9

We transferred the values to an excel file.

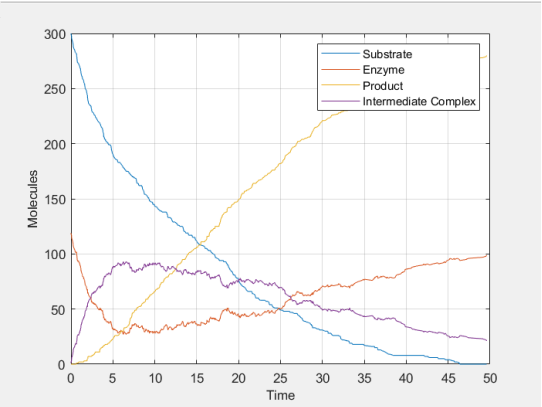
Mainly, we used the code that is given in the Modeling and Simulating Chemical Reactions - Desmond J. Higham. We make some manipulations in that code in order to be able to get random results, storing the results and plotting steps. In addition to that, we wanted to analyze the intermediate step which happens between the formation of product from substrate and enzyme.

The graph of the intermediate is found as:

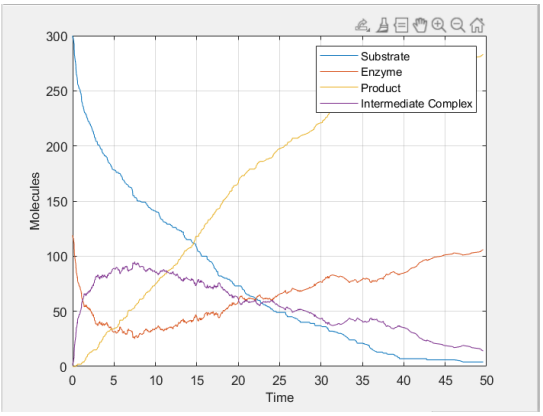


The orange line shows the original intermediate line, whereas the blue line shows its smoothed version.

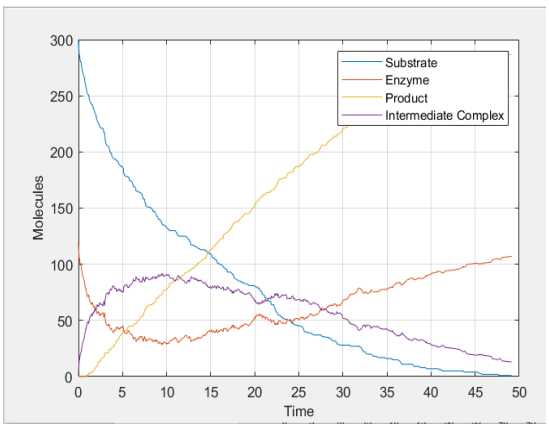
The plots of random reactions vary with respect to each runs:



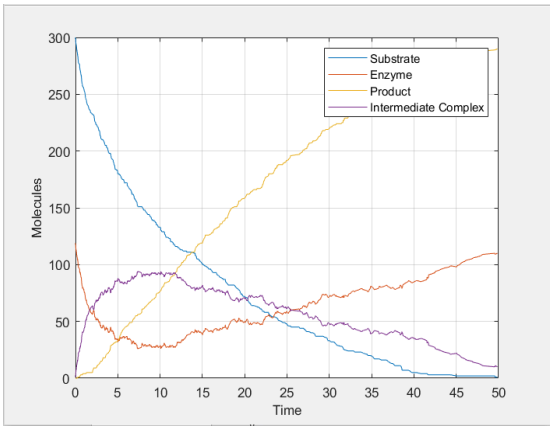
(i)



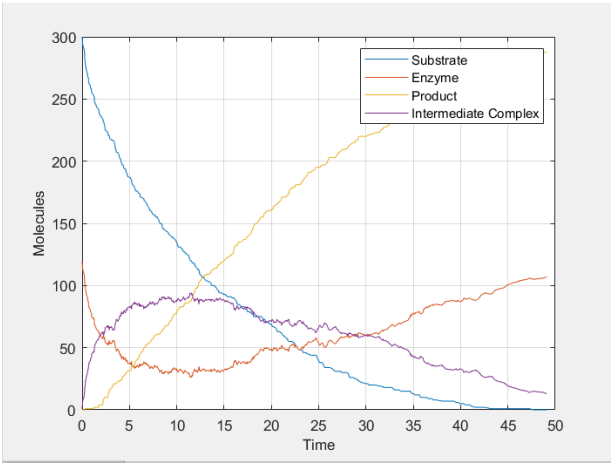
(ii)



(iii)



(iv)

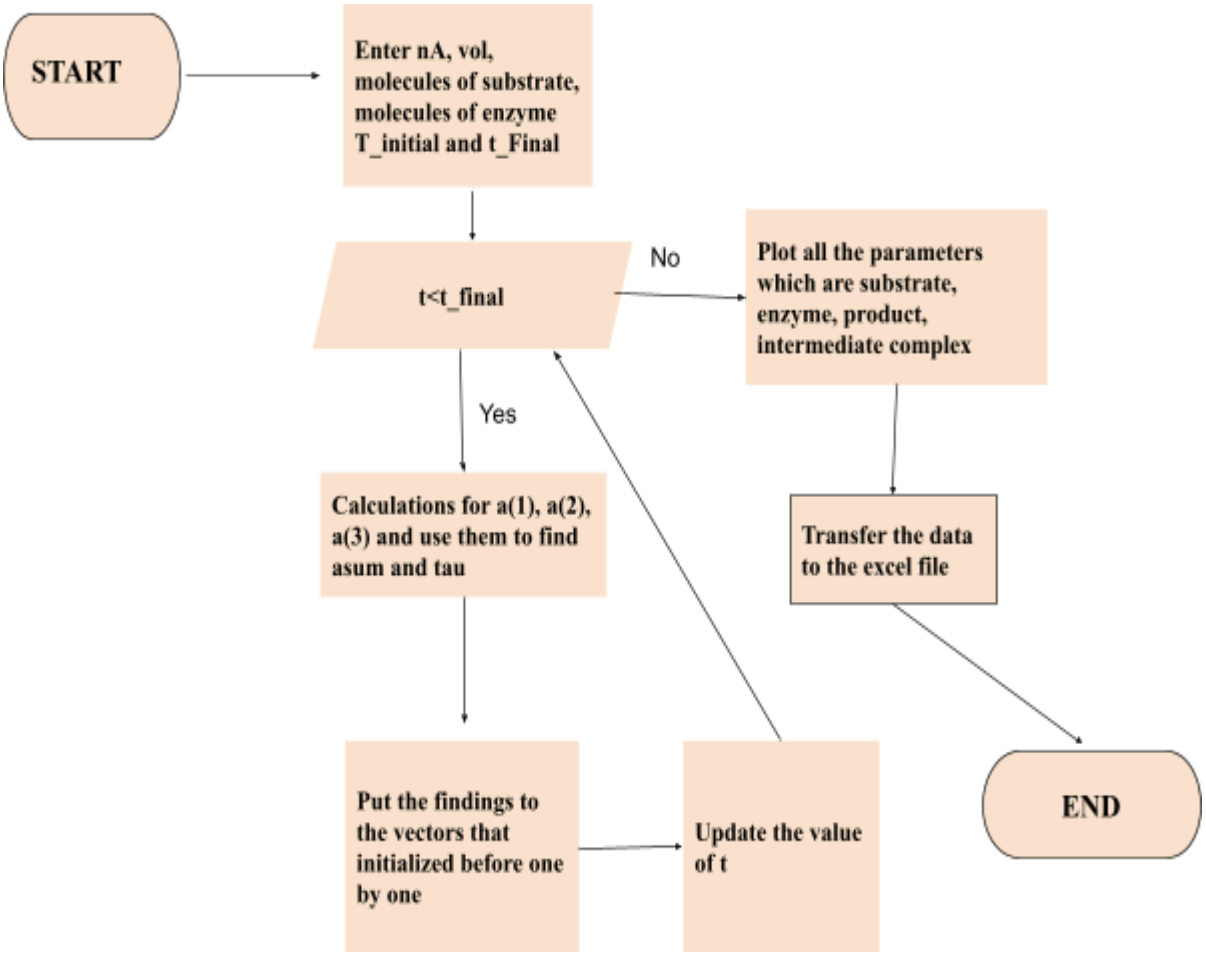


(v)

3. Discussion and Conclusion

Until this stage, we have read articles and expanded our knowledge and opinion on the subject in order to master the subject. At the same time, we developed the simulation algorithm that exists in one of the articles that we have read. Also, we extracted data from MATLAB simulation into Excel. In the second stage, it will be developed algorithms to estimate on a general basis the results by using reaction rates, substrate and enzyme components. In addition, we will obtain the noise covariance (convergence of sample covariance) for each sample. After that, we will decide statistical hypothesis, error probability and relative entropy based on digital communication.

BLOCK DIAGRAM OF THE SIMULATION



References

[1] Desmond Higham, 2008, Modeling and Simulating Chemical Reactions. Siam Review - SIAM REV. 50. 10.1137/060666457.

[2] D. J. Higham, *An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations*, SIAM Rev., 43 (2001), pp. 525-546.

[3] JR Moffitt and C Bustamante, "Extracting signal from noise: kinetic mechanisms from a Michaelis-Menten-like expression for enzymatic fluctuations." FEBS J. 2014 Jan;281(2):498-517. doi: 10.1111/febs.12545. Epub 2013 Oct 25. PMID: 24428386.

[4] B. C Akdeniz and M. Egan, ''A Molecular Communication Scheme to Estimate the State of Biochemical Processes on a Lab-on-a-Chip''. *In Proceedings of the 1st ACM International Workshop on Nanoscale Computing, Communication, and Applications Association for Computing Machinery*, New York, NY, USA, 8–13, 2020.

[5] Tania F.G.G. Cova and Alberto A. C. C. Pais, "Deep Learning for Deep Chemistry: Optimizing the Prediction of Chemical Patterns", 2020.

[6] C. -L. Tai and I. F. Akyildiz, "A Novel Framework for Capacity Analysis of Diffusion-Based Molecular Communication Incorporating Chemical Reactions," in *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 6, no. 3, pp. 233-243, Dec. 2020.

Section 2: Physical characteristics and mathematical modelling of Memristor

Memristor is a two terminal circuit element that has been first introduced by the article of Leon Chua in 1971 [1].The name, memristor, comes from the combination of memory and resistor. Its behavior can be described as a nonlinear resistor with memory. It is known to be the missing fourth basic circuit element as a part of the basic circuit element group consisting of capacitor, resistor and inductor. The potential of memristor is huge in terms of its memory applications. It is thought that the memristors will replace the transistors in the future computers [2].

Chua in his article [1] states that the memristor has not been yet discovered physically alone, without a power supply. In this section of the paper, we will examine the physical structure of this element and its behavior under some different power supplied conditions.

One of the fundamental device characteristics of a memristor is resistance of the memristor(memristance). Memristance relates the electrical charge and magnetic flux and it can be modelled as following [1]:

$$M(q) = d\varphi/dq \quad (1)$$

In order to understand the working principles of memristor it is important to construct its mathematical model. The model discussed in this section of the paper is taken from the work [3]. The physical and circuit modelling of a memristor is shown in figure1 [2]. The physical structure of a memristor is consisting of a two thin film of TiO_2 sandwiched between platinum contacts. One of them is doped with oxygen vacancies, acting as charge carriers and as a result showing a low resistance characteristic. The other layer is the undoped region which has an insulating property [2]. If a positive voltage is applied to the doped region we observe that the oxygen atoms start to diffuse within TiO_2.

The resistance of a memristor is defined by the length of its doped and undoped regions. If a memristor has all of its region as undoped than it has a very high resistance but, if TiO_2 is fully doped with oxygen ions than its resistance has the minimum value.

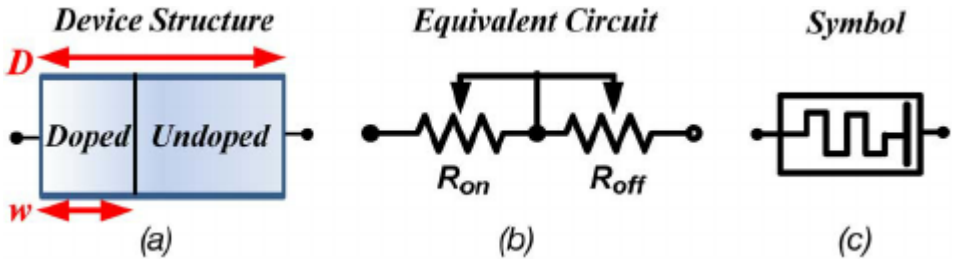


Figure 1

$$R_{MEM}(x) = R_{ON}(x) + R_{OFF}(1 - x) \quad (2)$$

$$\text{Where; } x = \frac{D}{W} \in (1,0) \quad (3)$$

Here, the R_{OFF} and R_{ON} are the limit values for the memristance and they are found by putting w=0 and w=D respectively. The ohms law can be also implemented to memristor as following:

$$V(t) = R_{MEM}(w) * i(t) \quad (4)$$

The speed of the border that separates the undoped and doped region can be found by the following:

$$\frac{dx}{dt} = k\, i(t) f(x) \;\; ; \;\; k = \frac{\mu_v R_{ON}}{D^2} \; where \; \mu_v \approx 10^{-14} m^2 s^{-1} V^{-1} \quad (5)$$

The nonlinearity character of a memristor comes form the fact that the amount of electric field when a small voltage is applied to a nanoscale devices being enormously high. Thus we can observe a nonlinear ion transportation in such devices [3]. This phenomenon is called as non-linear dopant drift and it is modeled by a window function as stated in (5). This form of function proves that the speed of the boundary is getting closer to zero when approaching either boundary(x=0 or x=D). Also noting that if p increases our overall drift function is becoming more similar to a linear model.

$$f(x) = 1 - (2x - 1)^p; where\ p\ being\ a\ positive\ integer \quad (6)$$

The above given mathematical models enable us to understand the fundamentals about the working principle of the device. For the purposes of the MATLAB modelling and simulation of the memristor given equations can be used. In this regard, we can now represent the I-V characteristics of the device in MATLAB environment and observe the behavior of memristor against the changing source voltage characteristics.

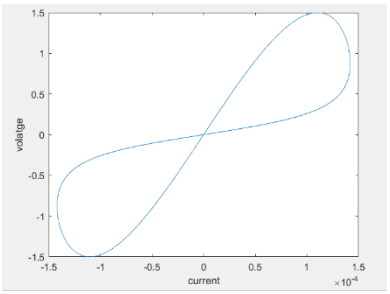
The following MATLAB code and simulation results has been acquired from the given work [4].

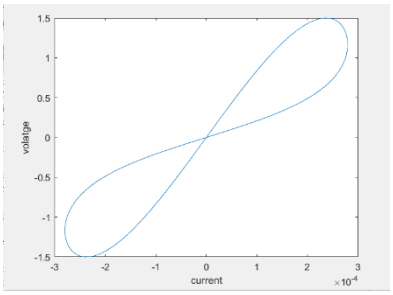
In this work the following features of the memristor has been considered and the results are taken according to these values.

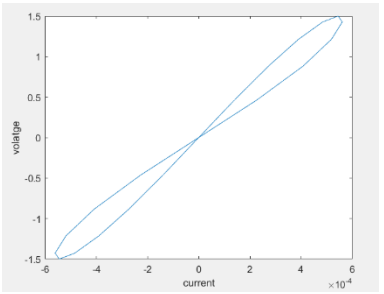
- R_i: intial resistance of the device (default: 1KΩ)

- R_{OFF}: maximum resistance of the device (default: 38KΩ)
- R_{ON}: minimum resistance of the device (default: 0.1Ω)
- D: device length in nm (default: 10nm)
- μ_v: dopant drift mobilty of the device material (default: 10^-14$m^2\ s^{-1}V^{-1}$)

In the following figures we will examine the behavior of the memristor against different source frequencies.

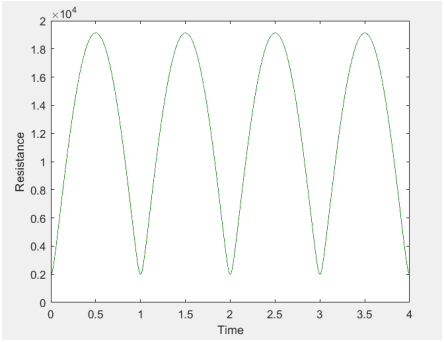


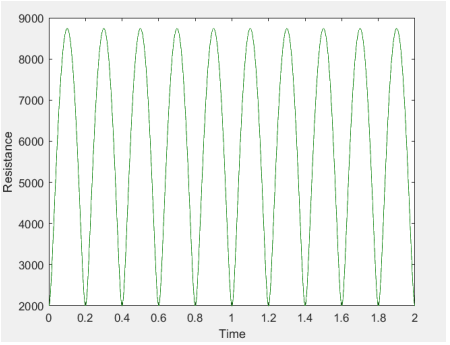




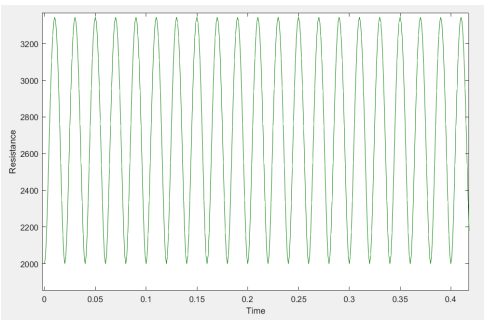
From figure 2 it is observed that the hysteresis loop of I-V plot is getting narrower as we increase the frequency of our voltage source. Looking at those figures one can suggest that at higher frequencies since the observed characteristic is getting closer to a linear form memristor is behaving as a resistor.

From figure 3 a,b and c the resistive switching property of the memristor is clearly seen. The response of our resistance values according to the changing frequency can be described as the frequency increases the switching of our resistance between its lowest and highest value is observed more sharply. Also, it is observed that the frequency change results in the change of resistance maximum and minimum value. Looking at figure3 one can see that at low frequency we have higher maximum resistance value.





(b) f=5Hz



(c) f=50Hz

The analysis of the change in the amplitude of the voltage and resulting I-V characteristic can be seen from figure 4. As expected, the increase in the amplitude of the voltage results in an increase in the current.

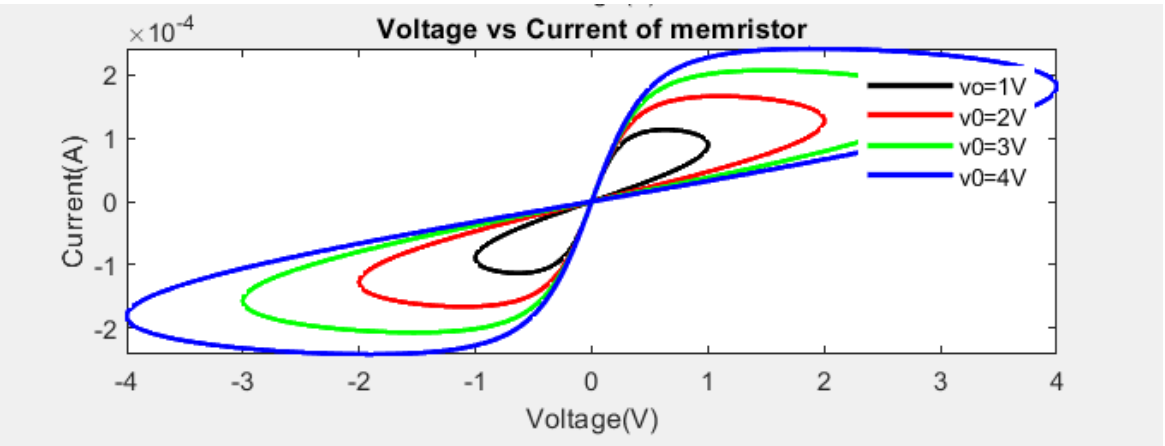


Figure 4

Discussion and Conclusion

In this section of the paper the physical characteristics and the mathematical modelling of memristor has been briefly discussed and the results of the implementation of the provided mathematical model has been shared. The potential of the memristor in technology applications is huge. In this section, the most useful property of this element that is the usage of it as a memory device component has not been discussed but the following work on this subject will be focused on the memory property and its potential advantages on the memory operations .

REFERENCES

[1] L. O. Chua, “Memristor—The missing circuit element,” IEEE Trans. Circuit Theory, vol. CT-18, pp. 507–519, Sep. 1971.

[2] Ho, Yenpo & Huang, Garng & Li, Peng. (2011). Dynamical Properties and Design Analysis for Nonvolatile Memristor Memories. Circuits and Systems I: Regular Papers, IEEE Transactions on. 58. 724 - 736. 10.1109/TCSI.2010.2078710

[3] Biolek, Zdenek & Biolek, Dalibor & V, Biolkova. (2009). SPICE Model of Memristor with Nonlinear Dopant Drift. Radioengineering. 18.

[4] A. G. Radwan, M. Affan Zidan, and K. N. Salama, On the mathematical modeling of Memristors, 22nd International Conference on Microelectronics (ICM 2010), pp. 284-287, Egyptime, December 2010

Section 3: COVID-19 Transmission Modelling with Computational Fluid Dynamics Tools

COVID-19 pandemic has aroused people's interest in transmission of respiratory infectious diseases recently. Modelling different transmission scenarios can be admitted as crucial in order to be able to halt spread of the virus. Since there are many factors that create these different scenarios such as spread on outdoors with/without wind, indoors that have/do not have ventilation, among people that wear/do not wear masks, a tool that can model them would be useful.

Even though the exact behavior of the droplets that carry the viruses is still debated, recently it has been observed that these droplets are carried by a turbulent cloud of gas [1]. Therefore, Computational Fluid Dynamics (CFD) is in the center of investigation of the transmission the airborne diseases. In order to model and generate data about the transmission OpenFOAM, which is a C++ oriented toolbox for numerical solving of continuum mechanic problems thus,

computational fluid dynamic problems, can be used [2]. In this research, a systematic modelling approach which can be used to model various virus transmission scenarios and opening a way to creation of easier ways to modelling them.

The modelling approach consists of two integrated parts which are coughing and droplet spreading. To model the cough of a person, human inspiratory capacity, shape of the solid surface (a tube), mouth width, initial velocity and other flow characteristics of the fluid are implemented as boundary conditions and Large Eddy Simulation (LES) is used. On the other hand, to model droplet spread; humidity, wind, volume of the space, temperature, droplet size are implemented and Lagrangian Droplet Simulation is used [3]. In order to make the model more realistic, effects of physical obstacles, multiple flow sources and masks will be implemented in the further steps.

Consequently, it is observed that virus transmission can be modelled by CFD tools and the generated data can be employed for mutual work with other statistical facilities to make the modelling easier for various conditions.

References:

[1] L. Bourouiba, “Turbulent Gas Clouds and Respiratory Pathogen Emissions: Potential Implications for Reducing Transmission of COVID-19”. *JAMA*, vol. 323, no. 18, pp. 1837–1838. Mar., 2020. doi:10.1001/jama.2020.4756

(https://jamanetwork.com/journals/jama/fullarticle/2763852) //Silinecek

[2] H. Jasak, “OpenFOAM: Open source CFD in research and industry”.

International Journal of Naval Architecture and Ocean Engineering, Vol. 1, No. 2, pp. 89-94, 2009. ISSN2092-6782. https://doi.org/10.2478/IJNAOE-2013-0011. (https://aip.scitation.org/doi/abs/10.1063/5.0011960)

[3] A.A. Aliabadi and S. Rogak and S. Green and K. Bartlett, “CFD Simulation of Human Coughs and Sneezes: A Study in Droplet Dispersion, Heat, and Mass Transfer”. *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 7. Pp 2-5, 2010.

(https://www.researchgate.net/publication/267592391_CFD_Simulation_of_Human_Coughs_and_Sneezes_A_Study_in_Droplet_Dispersion_Heat_and_Mass_Transfer)

Section 4 : *Chemical processes and reaction delay*

Abstarct:
Most of the chemical and biological reactions contain multiple steps, with either the same or different reaction rates. Even though these steps are hard to detect precisely, different approaches have been developed over time for this problem.
Here, we will mainly discuss how the experimental single-molecule kinetic data can be used to study multistep processes; how the developed chemical reaction delay models and simulations can be used in real life, such as telecommunication and queuing systems as well as the computer networks; and briefly give an introduction of the idea of machine learning and deep neural network (DNN) usage for deriving random variables and training the developed model for various occasions.

1 Introduction

Many reactions contain multiple steps with various rates. For example, taking an enzymatic reaction into account, there will be several sequential steps, including substrate binding, catalytic reaction, and product release. (4) Dealing processes with multiple steps and kinetics can be challenging to make

observations in many ways. Having said that, dealing with the kinetic measurements of single molecules are much easier and practial, due to their feature of not containing any of these limitations that we face with the multiple steps processes. This brings the idea that whether there is a way to use the experimental single-molecule kinetic data to wipe out the problems we come across in multistep process observations. Sensitive imaging and manipulation methods have been developed to analyse the number of biochemical and biophysical processes for this purpose. (1) Large ensembles will give us average times, which will contain many fluctuations, while single-molecules will give a more precise information about those changes in time.

The analysis of multi-step reaction kinetics is challenging in two aspects: determining the number of reaction steps from kinetic data with the "randomness parameter", r (6) and determining the distribution of kinetic rate constants, due to complications in inverse Laplace problem. (7) The second problem can be generally solved with different methods, either the Tikhonov regularization (8) and/or the maximum entropy method (9).

Variables of the chemical reactions that will differ from one to another are the reaction rate, k and the number of steps, N. The idea at first is to derive reaction rates from various rate equations and discuss how the use of randomness parameter, r can be encountered. Also, the expected reaction time will not be the same as the observed ones, due to the reaction delays. By understanding the intermediate processes, a linear signal model for reaction delays can be formed and a queuing system on top of it.

Delay analysis and its prediction are not that different in chemical reactions than it is in real life. To obtain B from A in chemical or biological terms, a time needs to be spent in the intermediate process. Lots of inter-products will be obtained and each time some particle takes another form, a time will be spent, which we will cause what we refer as reaction delay. This means that A->B process will take longer than expected due to many factors. This is the case in real life as well. For instance, let us think that we are waiting in a queue in front of a bank and there are five bankers dealing with the customers. The time we will spend to get our job done will contain waiting in the queue, time that both the individuals in front of you and the ones that are currently inside will spend. You will wait, enter, spend some time inside and then leave. These two processes are similar so if we form a linear signal model for the chemical reactions and understand their delay systems, the same thing can be implemented in real life, in the queuing systems, telecommunication systems and in the computer networks.

Another approach to these problems will be to apply a machine learning on the developed model and train it. In simpler terms, we will have to generate different numerical variables for k and N, which will be the parameters of the exponential distribution function that shows A->X->B, A representing the starting product, X representing the intermediate process and the products we get in between, B representing the output. The probability distribution of A -> B will be obtained by integrating over all possible times t (1), which will lead to an exponential function with parameters k and N. After getting random variables, we need to train a neural network with machine learning approaches until it begins to work for all different cases. For instance, if we trained the mechanism for one thousand different variable variations, the next step will be to check whether our mechanism works for ten thousand different combinations or not.

2 Probability Distribution Analysis

2.1 One intermediate step

Our chemical process will roughly look like this:

$$A \xrightarrow{k_1} X \xrightarrow{k_2} B,$$

p(t): their joint probability density.
If B is formed at T time, then the time for intermediate: T < t, then B was formed at t _ T.

The probability of A->B occurring at time t: joint probability of A -> X at T time and X -> B at t - T time.
İn order to find the probability distribution of A -> B, the one that we are interested in, we should integrate over T < t:

$$p_{A\to B}(\tau) = \int_0^{\tau} P_{A\to X}(t)P_{X\to B}(\tau - t)dt.$$

We must also take k, the reaction rates into account as it will affect the reaction time:

$$p_{A\to B}(\tau) = \begin{cases} \frac{k_1 k_2}{k_1 - K_2}\left(e^{-k_2\tau} - e^{-k_1\tau}\right), k_1 \neq k_2 \\ k^2\tau e^{-k\tau}, k_1 = k_2 \end{cases}.$$

2.2 Multiple intermediate steps, with the same rate constant

Single molecule measurements: Tracking the reaction course of individual molecules one by one.
Biological process: May consist of multiple steps. / May consist of 2 reactions but involve heterogeneous kinetics that complicates the process.

So far, we showed the logic behind the second case. However, the first one may be encountered as well. Leaving the having one intermediate step assumption behind, we now assume that the reaction rates will be equal to one another:

$$A \xrightarrow{k} X_1 \xrightarrow{k} X_2 \longrightarrow \xrightarrow{k} X_{N-1} \xrightarrow{k} B.$$

To avoid complications, we assume that reaction rates between each step is equal to k. Again, similar to what we did before, we find the probability distribution for this case, by taking the integral:

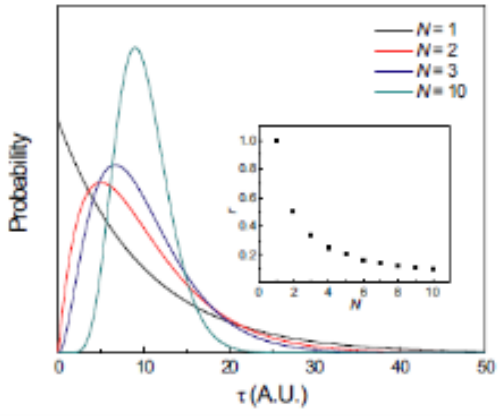
$$p_{A\to B}(\tau) = \frac{1}{2\pi i}\int_0^{i\infty} \tilde{P}_{A\to B}(s)e^{s\tau}ds = \frac{k^N\tau^{N-1}}{\Gamma(N)}e^{-k\tau}.$$

We obtain a well known gamma distribution function, therefore we can use it plot probabilities, for different N s to understand the relationship. (1)
It was mentioned that there were two challenges of kinetic analysis of multi-step chemical reactions:

1- Determining the number of steps.
2- Distribution of kinetic constants.

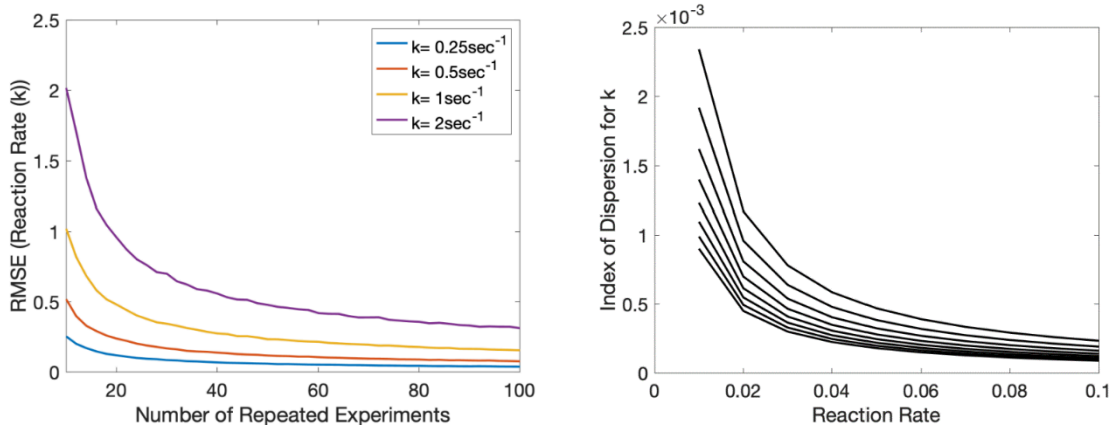
There are 2 stable integral transformations to solve these problems:

1. Decreasing number of rate-limiting steps.
2. Reconstruct the distribution of rate constants in the multistep reaction.



It can be seen that the more N increases, the narrower distribution becomes and the more symmetric it gets. It was also previously mentioned that the intermediate steps are hidden but they can be observed with the single kinetics through the final product (in this case B). By looking at this graph, one can easily observe intermediate steps as the waiting-time distributions.

We so far only looked into the N variables but k is the other random variable that we must generate, as we can see from the last equation, it will be the other parameter of the gamma and exponential distributions. In the previous example, we took k as constant, which is not always true. Below, there are 2 graphs given to help understand the relation between reactions and the changes in k.



2.3 Multiple intermediate steps, with different rate constants

As the last and the most realistic case, there is:

$$A_1 \xrightarrow[F_1(k)]{k_1} A_2 \xrightarrow[F_2(k)]{k_2} \cdots A_m \xrightarrow[F_m(k)]{k_m} B$$

where both multiple intermediate steps with different reaction rates exist. This is a challenging task, especially if the totally number of reaction steps are unknown or if some of the reaction steps cannot be described by a single rate constant. (2) There are two integral transforms to analyse the *m*-step reaction kinetics in general:

1. The "heat capacity transform" $\hat{C}$: Applicable to general *m*-step reactions; whether k s are different or not; it maps the overall kinetic data. However, it cannot resolve two reaction models. For example, while a single-molecule experiment with $N = 2000$ events may be sufficient to distinguish the 4-step and 5-step reaction models, experiments with $N = 200$ events could barely resolve 2-step versus 3-step reactions.
2. The "*m*-to-1 transform" $\hat{J}m$: It maps the overall kinetic data and effectively converts an *m*-step kinetic data set to a 1-step kinetic data set $\wp(t) = \int ke{-}ktd\mu(k)$, allowing the "effective" rate constant distribution $\mu(k)$ to be stably reconstructed using the "phase function approach", which stabilizes the inverse Laplace transform. (10) When combined with additional input of kinetic information, this approach offers a powerful and straight forward strategy to fully dissect the *m*-step reaction kinetics. (2)

3 Discussion and Conclusion

Firstly, different estimations about reactions and observation of the steps, moving from single molecule kinetics to multi molecule steps and explanation of the reasons behind this approach are

made. When there are multiple intermediate steps with different k s, different integral transforms called the "heat capacity transform" $\hat{C}$ and the "m-to-1 transform" $\hat{J}m$, must be used to analyse the steps and outcomes. We concluded that through studying chemical and biological reactions and their delay analysis, a linear system modelling and a queuing theory can be formed which also can be applied in real life, such as in the telecommunication and computer network systems. The importance of random variable generation as parameters to the gamma and exponential functions was also briefly mentioned.

This is a field, fully open for development and these will be among the aspects that the future research will contain: As mentioned before, without going into the details and its applications, introducing different machine learning approaches and improve it with DNN, would make this study applicable for many different scenarios in real life. In addition, we used autocorrelation as our Metric but CRLB would also be a good candidate. Developing a better queuing system model can also be done.

REFERENCES:

1. Daniel L. Floyd, Stephen C. Harrison and Antoine M. van Oijen, Analysis of Kinetic Intermediates in Single-Particle Dwell-Time Distributions, 10th ed. Boston: Jack and Eileen, 2010
2.Yajun Zhou and Xiaowei Zhuang, Kinetic Analysis of Sequential Multi-step Reactions, Massachusetts: Howard Hughes Medical Institute, 2007.
3. Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, Dmitry Kalenichenko, Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference
4. Nelson, DL.; Cox, MM. Lehninger Principles of Biochemistry. Vol. 3. Worth Publishers; Gordonsville, VA: 2000.
5. Seong Hyeon Hong, Haizhou Yang, Yi Wang, Inverse design of microfluidic concentration gradient generator using deep learning and physics-based component model, Germany: Springer Nature, 2020
6. Svoboda K, Mitra PP, Block SM. Proc Nat Acad Sci USA 1994;91:11782. [PubMed: 7991536]
7. McWhirter JG, Pike ER. J Phys A: Math Gen 1978;11:1729.
8. Tikhonov, AN.; Arsenin, VY. Solution of Ill-posed Problems. John Wiley and Sons; New York, NY: 1977.
9. Livesey AK, Brochon JC. Biophys J 1987;52:693.
10. Zhou Y, Zhuang X. Biophys J 2006;91:4045. [PubMed: 16980370]