Biophysics Homework - 2

Annamaria Bupu and Kristian Pajanonot

January 5, 2022

1 Problem 1

Active Brownian particles. The following model has been extensively used to describe the motion of active self-propelled microswimmers (e.g. bacteria, Janus particles, etc.) in two dimensions [C Bechinger et al., *Rev. Mod. Phys.* **88**(4), 045006 (2016)].

$$\dot{\theta} = \omega + \sqrt{2D_r}\xi_\theta \tag{1}$$
$$\dot{x} = v\cos(\theta) + \sqrt{2D_t}\xi_x \tag{2}$$
$$\dot{y} = v\sin(\theta) + \sqrt{2D_t}\xi_y \tag{3}$$

Here, ω is the average angular velocity of the swimmer, v the self-propulsion speed, D_r the rotational diffusion coefficient, and D_t the translational diffusion coefficient. The three noises are zero-mean Gaussian, white, and independent to each other, each with autocorrelation function $\langle\xi_\theta(t)\xi_\theta(t')\rangle = \langle\xi_x(t)\xi_x(t')\rangle = \langle\xi_y(t)\xi_y(t')\rangle = \delta(t-t')$. For simplicity we may assume $\theta(0) = x(0) = y(0) = 0$.

(**a**) Derive an analytical formula for the probability density $P(\theta, t)$. Does θ reach a stationary state?

Answer
We can write Equation (1) in the following form

$$d\theta = \omega dt + \sqrt{2D_r}dB_\theta \tag{4}$$

where $f(\theta, t) = \omega$, $g(\theta, t) = \sqrt{2D_r}$, and $dB_\theta = \xi_\theta dt$ is a Gaussian random variable with mean equal to 0 and standard deviation equal to $\sqrt{dt}$ or $dB_\theta \sim \mathcal{N}(0, \sqrt{dt})$. (*We will use the notation $\mathcal{N}(\mu, \sigma)$ to refer the Gaussian distribution with mean = μ and standard deviation = σ throughout this homework*). The stochasticity in Equation (4) comes from dB_θ in the second term. Thus, the probability density $P(\theta, t)$ also follows a Gaussian distribution with $\mu = \omega t$ and $\sigma = \sqrt{2D_r t}$. Here, $\mu \neq 0$ because this is a drifted Brownian motion. Thus, the analytical formula for the probability density then is,

$$P(\theta, t) = \frac{1}{\sqrt{4\pi D_r t}} e^{-\frac{(\theta - \omega t)^2}{4D_r t}} \tag{5}$$

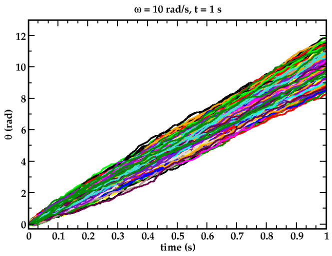
$\theta(t) = \int f(\theta, s)ds + \int g(\theta, s)dB(s)$. The first term $\int f(\theta, s)ds$ has solution in the form of ωt, so the deterministic part makes θ grow as t increases, it does not reach a stationary state.

(**b**) Compare the theoretical result obtained in (a) with the probability density of θ obtained from 10^3 simulations using the parameter values in (d) for observation times $t = 1$, 5, and 10 s.

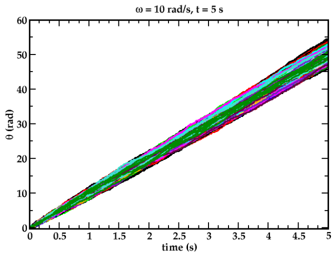
Answer
We can get the probability density for times t = 1, 5, and 10s from Equation (5)as follows:

$$P(\theta, t=1) = \frac{1}{\sqrt{4\pi D_r}} e^{-\frac{(\theta-\omega)^2}{4D_r}} \tag{6}$$
$$P(\theta, t=5) = \frac{1}{\sqrt{20\pi D_r}} e^{-\frac{(\theta-5\omega)^2}{20D_r}} \tag{7}$$
$$P(\theta, t=10) = \frac{1}{\sqrt{40\pi D_r}} e^{-\frac{(\theta-10\omega)^2}{40D_r}} \tag{8}$$

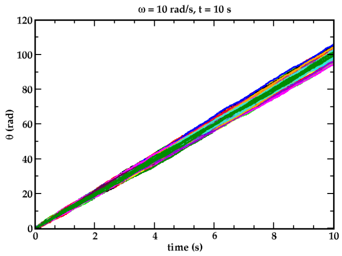
with $D_r = 0.17\text{rad}^2/\text{s}$ and $\omega = 10\text{rad/s}$ as given in (d). We plot Equation (6) to Equation (8) as seen in Figure 2. The probability density of θ is obtained from 10^3 simulations using the Euler's numerical simulation scheme (see Appendix B). First we simulate trajectories for time = 1, 5, and 10 s as shown in Figure 1. These figures also show that theta does not reach stationary state when $\mu = \omega t$.



(a) t = 1 s



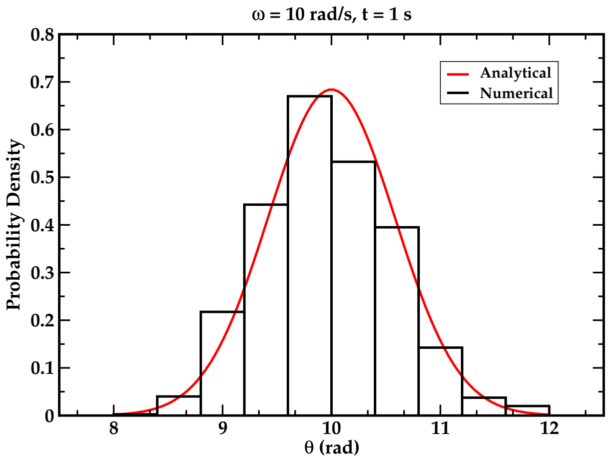
(b) t = 5 s



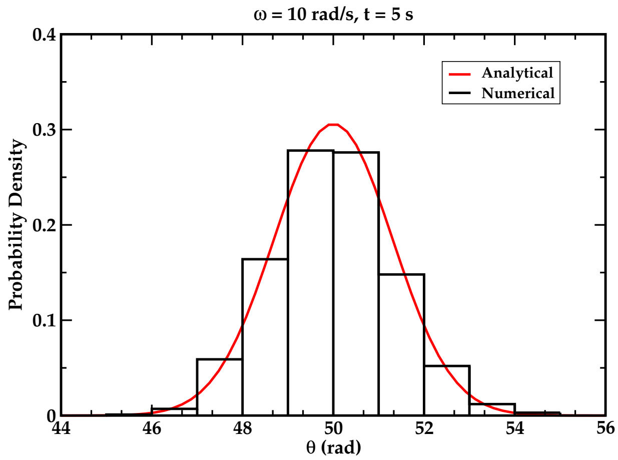
(c) t = 10 s

Figure 1. 10^3 stochastic trajectories of θ obtained from Euler's numerical simulation with $\omega = 10$ for observation times (a) $t = 1$, (b) $t = 5$, and (c) $t = 10$ s.

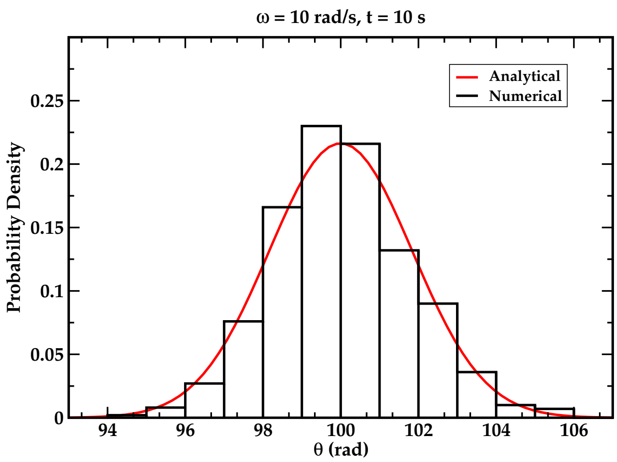
Afterwards, we collect the value of θ at $t = 1$, 5, and 10 s, then plot the histogram and compare these results with the theoretical results as shown in Figure 2.



(a) t = 1 s



(b) t = 5 s



(c) t = 10 s

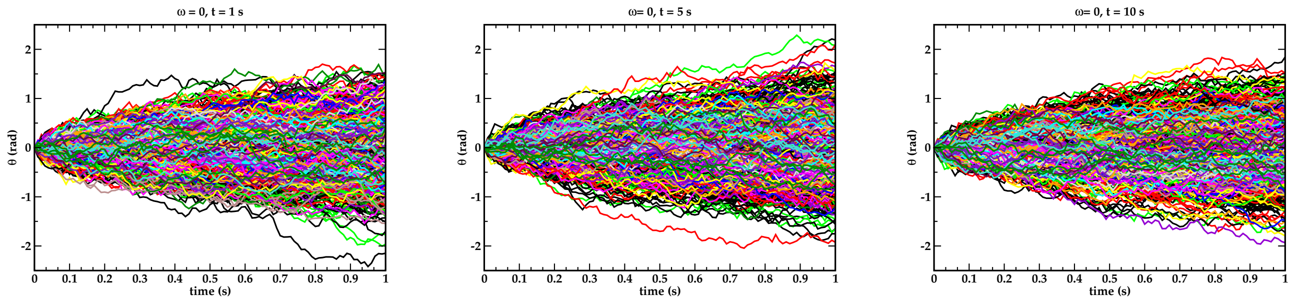
Figure 2. Probability density of θ with $\omega = 10$ for observation times (a) $t = 1$, (b) $t = 5$, and (c) $t = 10$ s. In red is the probability density obtained from the theoretical formula as given by Equation (6) to Equation (8), while the histogram (black) is the probability density obtained from numerical simulations.

From Figure 2, the numerical simulation gives us the following results

$$
\begin{aligned}
P(\theta, t=1) &\sim \mathcal{N}(10, 0.57) && (9)\\
P(\theta, t=5) &\sim \mathcal{N}(50, 1.27) && (10)\\
P(\theta, t=10) &\sim \mathcal{N}(100, 1.89) && (11)
\end{aligned}
$$

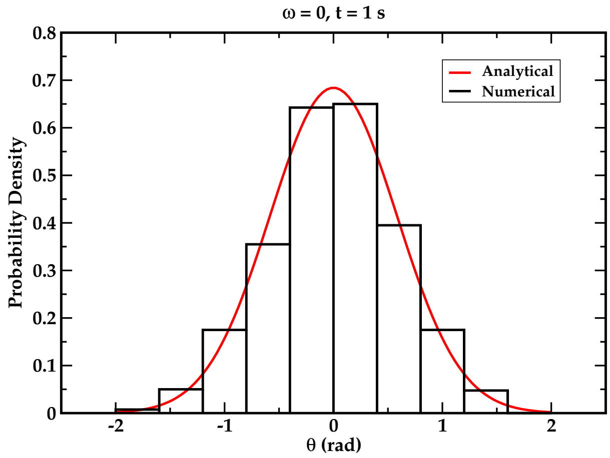
which is in agreement with the theoretical results in (a) where the mean value is equal to ωt and the standard deviation is given by $\sqrt{2D_r t}$.

A special case is when $\omega = 0$ as shown in Figure 3 and Figure 4. In this case, the differential of θ becomes a Brownian motion without the drift term, so the mean value of the probability density for both theoretical and numerical calculation is zero and variance (σ^2) grows as t increases.

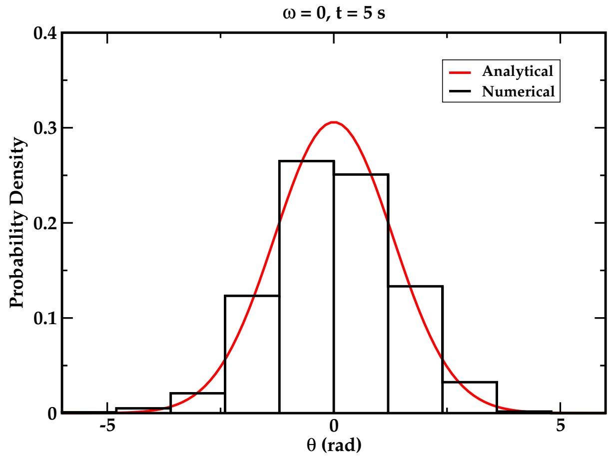


(a) t = 1 s **(b)** t = 5 s **(c)** t = 10 s

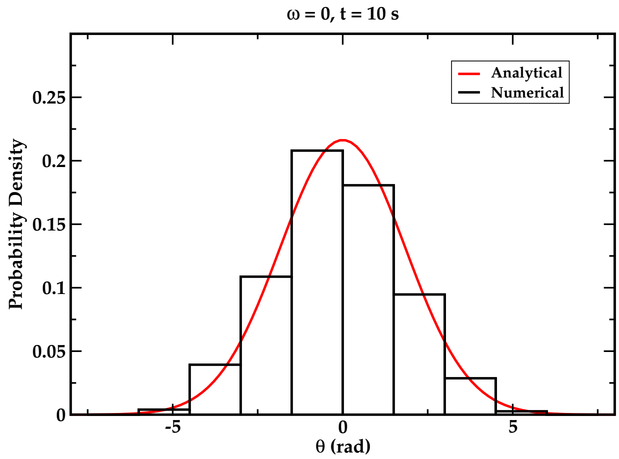
Figure 3. 10^3 stochastic trajectories of θ obtained from Euler's numerical simulation with $\omega = 0$ for observation times (a) $t = 1$, (b) $t = 5$, and (c) $t = 10$ s. Here, we observe that $\mu = 0$ for all observation times.



(a) t = 1 s



(b) t = 5 s



(c) t = 10 s

Figure 4. Probability density of θ with $\omega = 10$ for observation times (a) $t = 1$, (b) $t = 5$, and (c) $t = 10$ s. In red is the probability density obtained from the theoretical formula as given by Equation (6) to Equation (8), while the histogram (black) is the probability density obtained from numerical simulations.

(**c**) Using stochastic calculus derive a stochastic differential equation for the distance $r(t) = \sqrt{x^2(t) + y^2(t)}$ of the swimmer from the center (both in Itô and Stratonovich). What is the value of $\langle d_r(t) \rangle$ if $D_t = 0$?

Answer
We can rewrite Equations (1), (2), (3) as follows, with $dB_i = \xi_i dt$:

$$\begin{aligned} d\theta &= \omega dt + \sqrt{2D_r} dB_\theta && (12) \\ dx &= v\cos\theta dt + \sqrt{2D_t} dB_x && (13) \\ dy &= v\sin\theta dt + \sqrt{2D_t} dB_y && (14) \end{aligned}$$

We can write dr using multi-dimensional Itô formula:

$$dr = \left(\frac{\partial r}{\partial x}\right) dx + \left(\frac{\partial r}{\partial y}\right) dy + \frac{1}{2}\left[\left(\frac{\partial^2 r}{\partial x^2}\right)(dx)^2 + \left(\frac{\partial^2 r}{\partial x \partial y}\right) dxdy + \left(\frac{\partial^2 r}{\partial y \partial x}\right) dydx + \left(\frac{\partial^2 r}{\partial y^2}\right)(dy)^2\right] \quad (15)$$

where:

$$\begin{aligned} (dx)^2 &= (v\cos(\theta)dt + \sqrt{2D_t} dB_x)^2 \\ (dy)^2 &= (v\sin(\theta)dt + \sqrt{2D_t} dB_y)^2 \\ dxdy = dydx &= (v\sin(\theta)dt + \sqrt{2D_t} dB_y)(v\cos(\theta)dt + \sqrt{2D_t} dB_x) \end{aligned}$$

Since in Itô, $dB_i dB_j = \delta_{ij} dt$, $dB_i dt = dt dB_i = 0$, and $dB^2 = dt$, we get

$$
\begin{aligned}
(dx)^2 &= 2D_t(dB_x)^2 = 2D_t dt && (16)\\
(dy)^2 &= 2D_t(dB_y)^2 = 2D_t dt && (17)\\
dxdy &= dydx = 2D_t dB_x dB_y = 0 && (18)
\end{aligned}
$$

Substituting the values of dx and dy (Equation (13) and (14)), $(dx)^2$, $(dy)^2$, and $dxdy$ (Equation (16) to Equation (18)), then Equation (15) becomes:

$$
\begin{aligned}
dr &= \left(\frac{\partial r}{\partial x}\right) dx + \left(\frac{\partial r}{\partial y}\right) dy + \frac{1}{2}\left[\left(\frac{\partial^2 r}{\partial x^2}\right) 2D_t dt + \left(\frac{\partial^2 r}{\partial y^2}\right) 2D_t dt\right]\\
&= \left(\frac{\partial r}{\partial x}\right) dx + \left(\frac{\partial r}{\partial y}\right) dy + \left[\left(\frac{\partial^2 r}{\partial x^2}\right) + \left(\frac{\partial^2 r}{\partial y^2}\right)\right] D_t dt\\
&= \left(\frac{\partial r}{\partial x}\right) v\cos(\theta) dt + \left(\frac{\partial r}{\partial x}\right)\sqrt{2D_t} dB_x + \left(\frac{\partial r}{\partial y}\right) v\sin(\theta) dt + \left(\frac{\partial r}{\partial y}\right)\sqrt{2D_t} dB_y + \left[\left(\frac{\partial^2 r}{\partial x^2}\right) + \left(\frac{\partial^2 r}{\partial y^2}\right)\right] D_t dt\\
&= \left(\frac{\partial r}{\partial x}\right) v\cos(\theta) dt + \left(\frac{\partial r}{\partial y}\right) v\sin(\theta) dt + \left[\left(\frac{\partial^2 r}{\partial x^2}\right) + \left(\frac{\partial^2 r}{\partial y^2}\right) D_t\right] dt + \left[\left(\frac{\partial r}{\partial x}\right)\sqrt{2D_t} dB_x + \left(\frac{\partial r}{\partial y}\right)\sqrt{2D_t} dB_y\right]
\end{aligned}
$$

Solving for $\left(\frac{\partial r}{\partial x}\right)$, $\left(\frac{\partial r}{\partial y}\right)$, $\left(\frac{\partial^2 r}{\partial x^2}\right)$, and $\left(\frac{\partial^2 r}{\partial y^2}\right)$ using the given $r(t) = \sqrt{x^2(t) + y^2(t)} == (x^2 + y^2)^{1/2}$, we get

$$
\begin{aligned}
\left(\frac{\partial r}{\partial x}\right) &= \frac{x}{(x^2 + y^2)^{1/2}}\\
\left(\frac{\partial r}{\partial y}\right) &= \frac{y}{(x^2 + y^2)^{1/2}}\\
\left(\frac{\partial^2 r}{\partial x^2}\right) &= \frac{1}{(x^2 + y^2)^{1/2}} - \frac{x^2}{(x^2 + y^2)^{3/2}}\\
\left(\frac{\partial^2 r}{\partial y^2}\right) &= \frac{1}{(x^2 + y^2)^{1/2}} - \frac{y^2}{(x^2 + y^2)^{3/2}}
\end{aligned}
$$

Then dr becomes:

$$
\begin{aligned}
dr &= \left[\frac{xv\cos(\theta)}{(x^2 + y^2)^{1/2}} + \frac{yv\sin(\theta)}{(x^2 + y^2)^{1/2}} + \left[\frac{2}{(x^2 + y^2)^{1/2}} - \frac{x^2 + y^2}{(x^2 + y^2)^{3/2}}\right] D_t\right] dt\\
&+ \frac{x}{(x^2 + y^2)^{1/2}}\sqrt{2D_t} dB_x + \frac{y}{(x^2 + y^2)^{1/2}}\sqrt{2D_t} dB_y\\
&= \left[\frac{xv\cos(\theta) + yv\sin(\theta) + D_t}{(x^2 + y^2)^{1/2}}\right] dt + \frac{x}{(x^2 + y^2)^{1/2}}\sqrt{2D_t} dB_x + \frac{y}{(x^2 + y^2)^{1/2}}\sqrt{2D_t} dB_y
\end{aligned}
$$

Solving for dr in Stratonovich, we use the standard calculus:

$$
\begin{aligned}
dr &= \left(\frac{\partial r}{\partial x}\right) dx + \left(\frac{\partial r}{\partial y}\right) dy\\
&= \frac{x}{(x^2 + y^2)^{1/2}}[v\cos(\theta) dt + \sqrt{2D_t} \circ dB_x] + \frac{y}{(x^2 + y^2)^{1/2}}[v\sin(\theta) dt + \sqrt{2D_t} \circ dB_y]\\
&= \left[\frac{xv\cos(\theta) + yv\sin(\theta)}{(x^2 + y^2)^{1/2}}\right] dt + \frac{x}{(x^2 + y^2)^{1/2}}\sqrt{2D_t} \circ dB_x + \frac{y}{(x^2 + y^2)^{1/2}}\sqrt{2D_t} \circ dB_y
\end{aligned}
$$

When the value of $D_t = 0$, then dr becomes

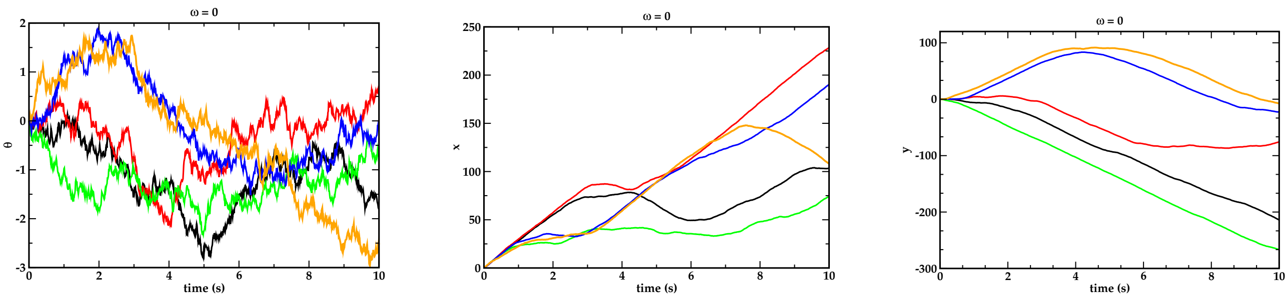
$$
dr = \left[\frac{xv\cos(\theta) + yv\sin(\theta)}{(x^2 + y^2)^{1/2}}\right] dt
$$

Then the value of $\langle dr\rangle$ is:

$$
\begin{aligned}
\langle dr\rangle &= \left\langle \frac{xv\cos(\theta)}{(x^2 + y^2)^{1/2}} dt\right\rangle + \left\langle \frac{yv\sin(\theta)}{(x^2 + y^2)^{1/2}} dt\right\rangle\\
&= v\left\langle \frac{x\cos(\theta)}{r} dt\right\rangle + v\left\langle \frac{y\sin(\theta)}{r} dt\right\rangle
\end{aligned}
$$

(**d**) Consider the parameter values $D_t = 0.2\mu\text{m}^2/\text{s}$, $D_r = 0.17\text{rad}^2/\text{s}$, $v = 30\text{m/s}$, and $\omega = 10\text{rad/s}$. Using Euler's numerical simulation scheme, plot five stochastic trajectories of duration 10s, using a simulation time step $\Delta t = 0.01\text{s}$. Repeat the same procedure setting $\omega = 0$ and discuss the obtained result.

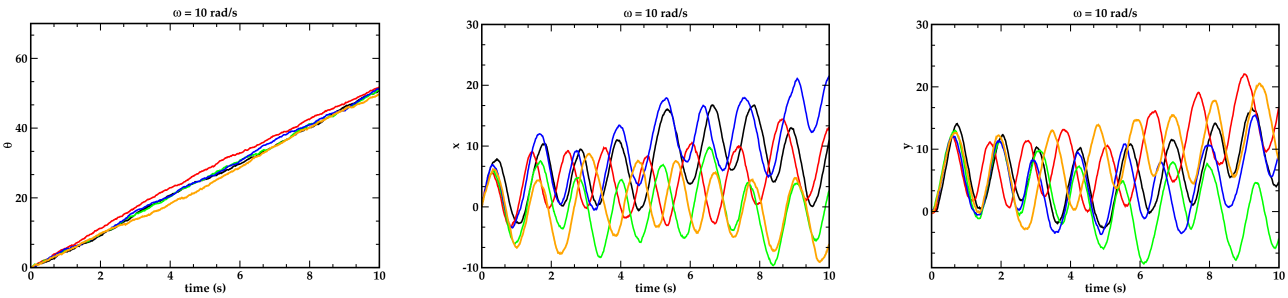
Answer
We start by plotting 5 trajectories of θ, x, and y with $\omega = 0$:



(a) θ (b) x (c) y

Figure 5. 5 stochastic trajectories of (a) θ, (b) x, and (c) y with $\omega = 0$ for observation time $t = 10s$. The graphs of x, and y are generated by feeding the value of θ to x and y following Equation (2) and Equation (3), respectively
.

The same procedure is repeated with $\omega = 10$ as shown in Figure 6:



(a) θ (b) x (c) y

Figure 6. 5 stochastic trajectories of (a) θ, (b) x, and (c) y with $\omega = 10$ for observation time $t = 10s$. The graphs of x, and y are generated by feeding the value of θ to x and y following Equation (2) and Equation (3), respectively.

When $\omega = 0$, the $\langle\theta\rangle = 0$, then there is no oscillation in x and y as the deterministic part of the equation is constant, as seen in Figure 5.

Meanwhile, when $\omega = 10$, the $\langle\theta\rangle = 10t$, then the deterministic part of the equations give the oscillations as seen in Figure 6.

This is clear when we plot x vs y of $\omega = 0$, and $\omega = 10$, where we see in $\omega = 10$ that there is a spiral-like behavior (Figure 7. This is referred to as the Chiral active Brownian motion[1]. This behavior is similar with Figure 2(c) to Figure 2(e) from the referenced paper.

[1] C. Bechinger, et al., Rev. Mod. Phys. **88**, 045006 (2016)

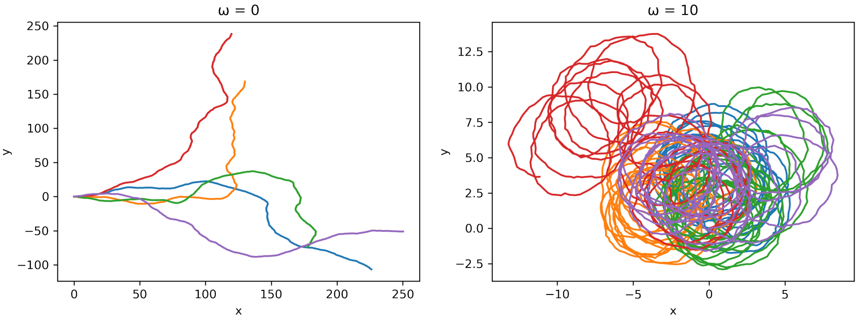


Figure 7. Trajectories of x vs. y. There is a spiral-like behavior when $\omega = 10$ (right).

In addition, when we change the noise (D_t) in dx (Equation (2)) and dy (Equation (3)), we see that the trajectory becomes more smooth or rough depending on the value. When the noise is larger, the trajectory become more rough, as seen in Figure 8.

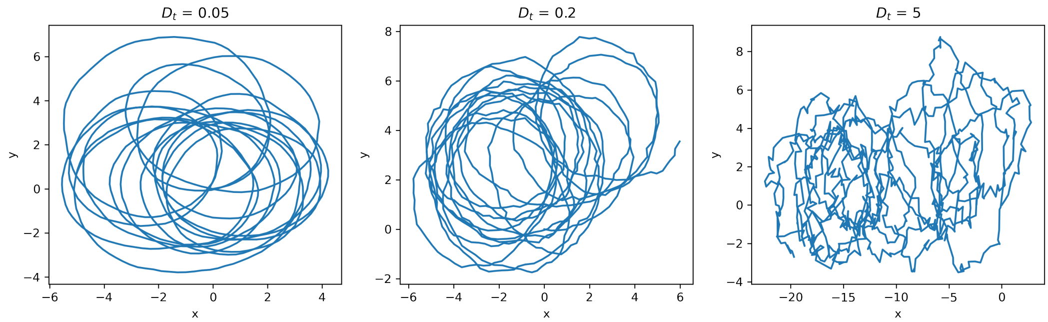
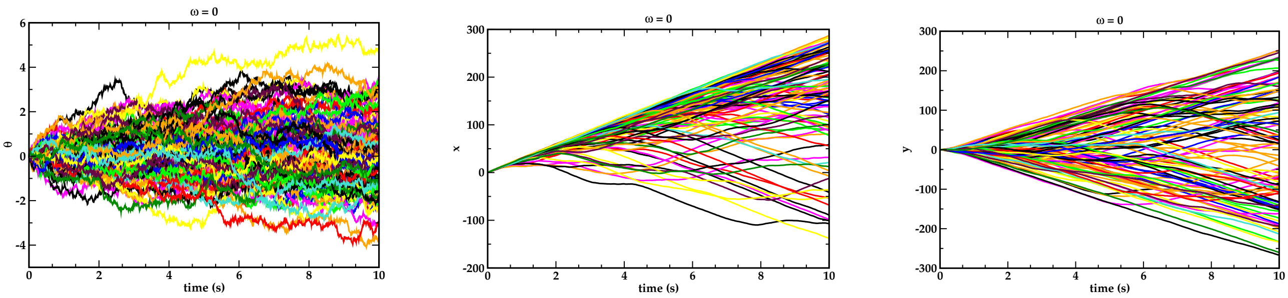


Figure 8. Trajectory of x vs. y with varying value of D_t (noise) at $\omega = 10$. As we increase the value of D_t (from left to right), the trajectory become more rough.

If we plot 100 trajectories for $\omega = 0$, we observed the same behavior as above. In addition, $\langle x \rangle = v$, whereas $\langle y \rangle = 0$, as seen in Figure 9.



(a) θ **(b)** x **(c)** y

Figure 9. 100 stochastic trajectories of (a) θ, (b) x, and (c) y with $\omega = 0$ for observation time $t = 10s$. The graphs of x, and y are generated by feeding the value of θ to x and y following Equation (2) and Equation (3), respectively.

2 Problem 2

Itô's lemma. Use Itô calculus rules to write the following stochastic processes on the standard form

$$dX = f(X,t)dt + g(X,t) \cdot dB$$

i.e. find the functions $f(X,t)$ and $g(X,t)$ for:
a. $X = B^2$
b. $X = A\cos(\omega t + B)$
c. $X = B/(1 + t/\tau)$
d. $X = e^{-B^2/2t}$

Solution:
Itô 's Lemma: Given $X(B,t)$:

$$dX(B,t) = \left(\frac{\partial X}{\partial t} + \frac{1}{2}\frac{\partial^2 X}{\partial B^2}\right)dt + \frac{\partial X}{\partial B}dB$$

a. $X = B^2$
Finding the first derivative of the function with respect to t and B and the second derivative with respect to B, we obtain

$$\begin{aligned} \frac{\partial X}{\partial t} &= 0 \\ \frac{\partial X}{\partial B} &= 2B \\ \frac{\partial^2 X}{\partial B^2} &= 2 \end{aligned}$$

Then dX becomes:

$$\begin{aligned} dX &= \left(0 + \frac{1}{2}(2)\right)dt + 2BdB \\ &= 1dt + 2BdB \end{aligned}$$

Since $X = B^2$, $B = \sqrt{X}$, so

$$\begin{aligned} dX &= 1dt + 2\sqrt{X}dB && (19) \\ f(X,t) &= 1 && (20) \\ g(X,t) &= 2\sqrt{X} && (21) \end{aligned}$$

b.$X = A\cos(\omega t + B)$
Finding the derivatives, we obtain

$$\begin{aligned} \frac{\partial X}{\partial t} &= -A\omega\sin(\omega t + B) \\ \frac{\partial X}{\partial B} &= -A\sin(\omega t + B) \\ \frac{\partial^2 X}{\partial B^2} &= -A\cos(\omega t + B) \end{aligned}$$

Then dX becomes:

$$dX = \left(-A\omega\sin(\omega t + B) - \frac{1}{2}A\cos(\omega t + B)\right)dt - A\sin(\omega t + B)dB \qquad (22)$$

Using $X = A\cos(\omega t + B)$ and

$$\begin{aligned} \cos^2(\omega t + B) + \sin^2(\omega t + B) &= 1 \\ \sin^2(\omega t + B) &= 1 - \cos^2(\omega t + B) \\ \sin^2(\omega t + B) &= 1 - \left(\frac{X}{A}\right)^2 \\ \sin^2(\omega t + B) &= \frac{A^2 - X^2}{A^2} \\ A^2\sin^2(\omega t + B) &= A^2 - X^2 \\ A\sin(\omega t + B) &= \sqrt{A^2 - X^2}, \end{aligned}$$

Equation (22) becomes

$$\begin{aligned}
dX &= \left(-\omega\sqrt{A^2 - X^2} - \frac{1}{2}X\right)dt - \sqrt{A^2 - X^2}dB && (23)\\
f(X,t) &= -\omega\sqrt{A^2 - X^2} - \frac{1}{2}X && (24)\\
g(X,t) &= -\sqrt{A^2 - X^2} && (25)
\end{aligned}$$

c. $X = B/(1 + t/\tau)$
Finding the derivatives, we obtain

$$\begin{aligned}
\frac{\partial X}{\partial t} &= \frac{-B}{\tau(1 + (t/\tau))^2}\\
\frac{\partial X}{\partial B} &= \frac{1}{1 + (t/\tau)}\\
\frac{\partial^2 X}{\partial B^2} &= 0
\end{aligned}$$

Then dX becomes:

$$dX = \left(\frac{-B}{\tau(1 + (t/\tau))^2}\right)dt + \frac{1}{1 + (t/\tau)}dB$$

Since $B = X(1 + (t/\tau)$, so

$$\begin{aligned}
dX &= \left(\frac{-X}{\tau(1 + (t/\tau))}\right)dt + \frac{1}{1 + (t/\tau)}dB && (26)\\
f(X,t) &= \frac{-X}{\tau(1 + (t/\tau))} && (27)\\
g(X,t) &= \frac{1}{1 + (t/\tau)} && (28)\\
& && (29)
\end{aligned}$$

d. $X = e^{-\frac{B^2}{2t}}$
Finding the derivatives, we obtain

$$\begin{aligned}
\frac{\partial X}{\partial t} &= \frac{B^2 e^{-\frac{B^2}{2t}}}{2t^2}\\
\frac{\partial X}{\partial B} &= \frac{-Be^{-\frac{B^2}{2t}}}{t}\\
\frac{\partial^2 X}{\partial B^2} &= e^{-\frac{B^2}{2t}}\left(\frac{-1}{t} + \frac{B^2}{t^2}\right)
\end{aligned}$$

Then dX becomes:

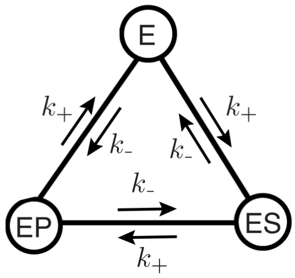
$$dX = \left[\frac{B^2 e^{-\frac{B^2}{2t}}}{2t^2} + \frac{1}{2}e^{-\frac{B^2}{2t}}\left(\frac{-1}{t} + \frac{B^2}{t^2}\right)\right]dt + \frac{-Be^{-\frac{B^2}{2t}}}{t}dB \qquad (30)$$

Since $B = \sqrt{-2t\ln X}$, so

$$\begin{aligned}
dX &= \left[-\frac{X\ln X}{t} + \frac{1}{2}X\left(-\frac{1}{t} - \frac{2\ln X}{t}\right)\right]dt - \frac{X\sqrt{-2t\ln X}}{t}dB && (31)\\
&= \left[\frac{-X\ln X}{t} - \frac{X}{2t} - \frac{X\ln X}{t}\right]dt - \frac{X\sqrt{-2t\ln X}}{t}dB && (32)\\
&= \left[-\frac{4X\ln X - X}{2t}\right]dt - \frac{X\sqrt{-2t\ln X}}{t}dB && (33)\\
f(X,t) &= -\frac{4X\ln X - X}{2t} && (34)\\
g(X,t) &= -\frac{X\sqrt{-2t\ln X}}{t} && (35)
\end{aligned}$$

3 Problem 3

Statistics of enzymatic reactions. Consider a three-state continuous-time Markov jump model of the cyclic enzymatic reaction illustrated in the figure



Here the circles denote the three different states of the enzyme (E: free enzyme; ES: enzyme bound to substrate; EP: enzyme bound to product), and k_+ and k_- are respectively the clockwise and counterclockwise transition rate in the state space.

(**a**) Write down the Master equation associated with the dynamics of the enzyme, both as a system of equations and also in its matrix form $d\vec{P}/dt = \mathbf{W}\vec{P}$, with $\vec{P}$ a probability column vector.

Answer
For Markov jump processes, the general master equation is as follows:

$$\frac{d}{dt}P_\mu(t) = \sum_\nu W_{\mu\nu}P_\nu(t) - W_{\nu\mu}P_\mu(t)$$

where $W_{\mu\nu}$ is the transition rate from state ν to state μ. In this problem, the master equation for each state is

$$\begin{aligned} \frac{dP_{\mathrm{EP}}}{dt} &= -(k_+ + k_-)P_{\mathrm{EP}} + k_- P_{\mathrm{E}} + k_+ P_{\mathrm{ES}} \\ \frac{dP_{\mathrm{E}}}{dt} &= k_+ P_{\mathrm{EP}} - (k_+ + k_-)P_{\mathrm{E}} + k_- P_{\mathrm{ES}} \\ \frac{dP_{\mathrm{ES}}}{dt} &= k_- P_{\mathrm{EP}} + k_+ P_{\mathrm{E}} - (k_+ + k_-)P_{\mathrm{ES}} \end{aligned} \tag{36}$$

The three equations above can be written in matrix form, $d\vec{P}/dt = \mathbf{W}\vec{P}$, as follows:

$$\frac{d}{dt}\underbrace{\begin{bmatrix} P_{\mathrm{EP}} \\ P_{\mathrm{E}} \\ P_{\mathrm{ES}} \end{bmatrix}}_{\vec{P}} = \underbrace{\begin{bmatrix} -(k_+ + k_-) & k_- & k_+ \\ k_+ & -(k_+ + k_-) & k_- \\ k_- & k_+ & -(k_+ + k_-) \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} P_{\mathrm{EP}} \\ P_{\mathrm{E}} \\ P_{\mathrm{ES}} \end{bmatrix}}_{\vec{P}} \tag{37}$$

Here, the columns of the transition matrix $\mathbf{W}$ sum to zero.

(**b**) Compute the eigenvalues and eigenvectors of the transition matrix $\mathbf{W}$.

Answer
The transition matrix $\mathbf{W}$ in Equation (37), is a circulant matrix. An $n \times n$ circulant matrix C is a matrix with elements as follows:

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_0 & \cdots & c_{n-3} & c_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_2 & c_3 & \cdots & c_0 & c_1 \\ c_1 & c_2 & \cdots & c_{n-1} & c_0 \end{pmatrix}$$

The eigenvalues of the circulant matrix C has the following form

$$\lambda_k = \sum_{j=0}^{n-1} c_j \omega_n^{jk} \tag{38}$$

where $k = 0, 1, \cdots, n-1$ and $\omega_n \equiv e^{2\pi i/n}$. The eigenvector for the k-th eigenvalue is

$$x^{(k)} = \begin{pmatrix} \omega_n^{0k} \\ \omega_n^{1k} \\ \omega_n^{2k} \\ \vdots \\ \omega_n^{(n-1)k} \end{pmatrix} \tag{39}$$

Thus, the transition matrix $\mathbf{W}$ is a circulant matrix with $n = 3$. Its eigenvalues can be calculated using Equation (38) with $k = 0, 1, 2$.
For $k = 0$

$$\lambda_0 = \sum_{j=0}^{2} W_j \omega_3^{0.j} = W_0 + W_1 + W_2 = -(k_+ + k_-) + k_- + k_+ = 0$$

For $k = 1$

$$\lambda_1 = \sum_{j=0}^{2} W_j \omega_3^{1.j} = W_0 + W_1\omega_3 + W_2\omega_3^2 = -(k_+ + k_-) + k_- e^{2\pi i/3} + k_+ e^{2(2\pi i/3)}$$

Since $e^{\alpha i} = \cos(\alpha) + i\sin(\alpha)$

$$\begin{aligned}
\lambda_1 &= -(k_+ + k_-) + k_-\left(\cos\left(\frac{2\pi}{3}\right) + i\sin\left(\frac{2\pi}{3}\right)\right) + k_+\left(\cos\left(\frac{4\pi}{3}\right) + i\sin\left(\frac{4\pi}{3}\right)\right) \\
&= -(k_+ + k_-) + k_-\left(-\frac{1}{2} + \frac{\sqrt{3}}{2}i\right) + k_+\left(-\frac{1}{2} - \frac{\sqrt{3}}{2}i\right) \\
&= -\frac{3}{2}(k_+ + k_-) - i\frac{\sqrt{3}}{2}(k_+ - k_-)
\end{aligned}$$

For $k = 2$

$$\lambda_2 = \sum_{j=0}^{2} W_j \omega_3^{2.j} = W_0 + W_1\omega_3^2 + W_2\omega_3^4 = -(k_+ + k_-) + k_- e^{2(2\pi i/3)} + k_+ e^{4(2\pi i/3)}$$

Using relation $e^{\alpha i} = \cos(\alpha) + i\sin(\alpha)$, then

$$\begin{aligned}
\lambda_2 &= -(k_+ + k_-) + k_-\left(\cos\left(\frac{4\pi}{3}\right) + i\sin\left(\frac{4\pi}{3}\right)\right) + k_+\left(\cos\left(\frac{8\pi}{3}\right) + i\sin\left(\frac{8\pi}{3}\right)\right) \\
&= -(k_+ + k_-) + k_-\left(-\frac{1}{2} - \frac{\sqrt{3}}{2}i\right) + k_+\left(-\frac{1}{2} + \frac{\sqrt{3}}{2}i\right) \\
&= -\frac{3}{2}(k_+ + k_-) + i\frac{\sqrt{3}}{2}(k_+ - k_-)
\end{aligned}$$

The corresponding eigenvectors for each eigenvalues (calculated using Equation (39)):
For $\lambda_0 = 0$, the eigenvector is:

$$x^{(0)} = \begin{pmatrix} \omega_3^0 \\ \omega_3^0 \\ \omega_3^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

For $\lambda_1 = -\frac{3}{2}(k_+ + k_-) - i\frac{\sqrt{3}}{2}(k_+ - k_-)$, the eigenvector is:

$$x^{(1)} = \begin{pmatrix} \omega_3^0 \\ \omega_3^1 \\ \omega_3^2 \end{pmatrix} = \begin{pmatrix} 1 \\ \omega_3 \\ \omega_3^2 \end{pmatrix} = \begin{pmatrix} 1 \\ e^{2\pi i/3} \\ e^{4\pi i/3} \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2} + \frac{\sqrt{3}}{2}i \\ -\frac{1}{2} - \frac{\sqrt{3}}{2}i \end{pmatrix}$$

For $\lambda_2 = -\frac{3}{2}(k_+ + k_-) + i\frac{\sqrt{3}}{2}(k_+ - k_-)$, the eigenvector is:

$$x^{(2)} = \begin{pmatrix} \omega_3^0 \\ \omega_3^2 \\ \omega_3^4 \end{pmatrix} = \begin{pmatrix} 1 \\ e^{4\pi i/3} \\ e^{8\pi i/3} \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2} - \frac{\sqrt{3}}{2}i \\ -\frac{1}{2} + \frac{\sqrt{3}}{2}i \end{pmatrix}$$

Notice that the real part of each eigenvalues, $\mathrm{Re}(\lambda_k) \leq 0$. The general solution for this problem is in the form of $P_\mu(t) = Ae^{\lambda_0 t} + Be^{\lambda_1 t} + Ce^{\lambda_2 t}$ with μ as the corresponding states: E, EP, ES, and A, B, C are constants. In the limit of $t \to \infty$, the probability can not go to ∞ because the value of the probability is between 0 and 1. Thus, the real part of eigenvalues that correspond to the decaying of the probability need to be ≤ 0.

(**c**) Calculate the stationary distribution and stationary current of the system, and the characteristic relaxation times of a system that is initially out of steady state.

Answer
The stationary distribution is $P^{\mathrm{st}} = \lim_{t\to\infty} P(t)$. In this limit, $\frac{dP(t)}{dt} = 0$. So, the master equation in Equation (36) becomes

$$0 = -(k_+ + k_-)P^{\mathrm{st}}_{\mathrm{EP}} + k_- P^{\mathrm{st}}_{\mathrm{E}} + k_+ P^{\mathrm{st}}_{\mathrm{ES}} \tag{40a}$$
$$0 = k_+ P^{\mathrm{st}}_{\mathrm{EP}} - (k_+ + k_-)P^{\mathrm{st}}_{\mathrm{E}} + k_- P^{\mathrm{st}}_{\mathrm{ES}} \tag{40b}$$
$$0 = k_- P^{\mathrm{st}}_{\mathrm{EP}} + k_+ P^{\mathrm{st}}_{\mathrm{E}} - (k_+ + k_-)P^{\mathrm{st}}_{\mathrm{ES}} \tag{40c}$$

Equation (40a) to Equation (40c) is system of equations with three equations and three unknown variables. So, the stationary distribution of each state can be calculated by solving the system of equations.
Using Equation (40a), we obtain:

$$\begin{aligned}
(k_+ + k_-)P^{\mathrm{st}}_{\mathrm{EP}} &= k_- P^{\mathrm{st}}_{\mathrm{E}} + k_+ P^{\mathrm{st}}_{\mathrm{ES}} \\
P^{\mathrm{st}}_{\mathrm{EP}} &= \frac{k_-}{k_+ + k_-} P^{\mathrm{st}}_{\mathrm{E}} + \frac{k_+}{k_+ + k_-} P^{\mathrm{st}}_{\mathrm{ES}}
\end{aligned}$$

Then, substituting $P^{\mathrm{st}}_{\mathrm{EP}}$ to Equation (40b), we get

$$\begin{aligned}
0 &= k_+ \left(\frac{k_-}{k_+ + k_-} P^{\mathrm{st}}_{\mathrm{E}} + \frac{k_+}{k_+ + k_-} P^{\mathrm{st}}_{\mathrm{ES}} \right) - (k_+ + k_-)P^{\mathrm{st}}_{\mathrm{E}} + k_- P^{\mathrm{st}}_{\mathrm{ES}} \\
0 &= \left(\frac{k_+ k_-}{k_+ + k_-} P^{\mathrm{st}}_{\mathrm{E}} + \frac{k_+ k_+}{k_+ + k_-} P^{\mathrm{st}}_{\mathrm{ES}} \right) - (k_+ + k_-)P^{\mathrm{st}}_{\mathrm{E}} + k_- P^{\mathrm{st}}_{\mathrm{ES}} \\
0 &= \left(\frac{k_+ k_-}{k_+ + k_-} - (k_+ + k_-) \right) P^{\mathrm{st}}_{\mathrm{E}} + \left(\frac{k_+ k_+}{k_+ + k_-} + k_- \right) P^{\mathrm{st}}_{\mathrm{ES}} \\
0 &= \left(\frac{k_+ k_- - k_+(k_+ + k_-) - k_-(k_+ + k_-)}{k_+ + k_-} \right) P^{\mathrm{st}}_{\mathrm{E}} + \left(\frac{k_+ k_+ + k_-(k_+ + k_-)}{k_+ + k_-} \right) P^{\mathrm{st}}_{\mathrm{ES}} \\
0 &= \left(\frac{-k_+ k_+ - k_-(k_+ + k_-)}{k_+ + k_-} \right) P^{\mathrm{st}}_{\mathrm{E}} + \left(\frac{k_+ k_+ + k_-(k_+ + k_-)}{k_+ + k_-} \right) P^{\mathrm{st}}_{\mathrm{ES}} \\
0 &= -\left(\frac{k_+ k_+ + k_-(k_+ + k_-)}{k_+ + k_-} \right) P^{\mathrm{st}}_{\mathrm{E}} + \left(\frac{k_+ k_+ + k_-(k_+ + k_-)}{k_+ + k_-} \right) P^{\mathrm{st}}_{\mathrm{ES}} \\
P^{\mathrm{st}}_{\mathrm{E}} &= P^{\mathrm{st}}_{\mathrm{ES}}
\end{aligned}$$

Thus,

$$P^{\mathrm{st}}_{\mathrm{EP}} = \frac{k_-}{k_+ + k_-} P^{\mathrm{st}}_{\mathrm{E}} + \frac{k_+}{k_+ + k_-} P^{\mathrm{st}}_{\mathrm{E}} = P^{\mathrm{st}}_{\mathrm{E}}$$

Because $P^{\mathrm{st}}_{\mathrm{EP}} = P^{\mathrm{st}}_{\mathrm{E}} = P^{\mathrm{st}}_{\mathrm{ES}}$ and from the normalization condition we know that $P^{\mathrm{st}}_{\mathrm{EP}} + P^{\mathrm{st}}_{\mathrm{E}} + P^{\mathrm{st}}_{\mathrm{ES}} = 1$, $P^{\mathrm{st}}_{\mathrm{EP}} = P^{\mathrm{st}}_{\mathrm{E}} = P^{\mathrm{st}}_{\mathrm{ES}} = \frac{1}{3}$.

The general form of the net probability current between the two states μ and ν is

$$J_{\mu\to\nu}(t) = W_{\nu\mu} P_\mu(t) - W_{\mu\nu} P_\nu(t)$$

In the stationary state, the net probability current above becomes

$$J^{\mathrm{st}}_{\mu\to\nu} = W_{\nu\mu} P^{\mathrm{st}}_\mu - W_{\mu\nu} P^{\mathrm{st}}_\nu$$

For this three-state Markov jump model, we can find the stationary currents between states: EP and E, E and ES, ES and EP as follows:

$$J^{\mathrm{st}}_{\mathrm{EP}\to\mathrm{E}} = -J^{\mathrm{st}}_{\mathrm{E}\to\mathrm{EP}} = k_+ P^{\mathrm{st}}_{\mathrm{EP}} - k_- P^{\mathrm{st}}_{\mathrm{E}} = \frac{1}{3}(k_+ - k_-)$$

$$J^{\text{st}}_{\text{E}\to\text{ES}} = -J^{\text{st}}_{\text{ES}\to\text{E}} = k_+ P^{\text{st}}_{\text{E}} - k_- P^{\text{st}}_{\text{ES}} = \frac{1}{3}(k_+ - k_-)$$

$$J^{\text{st}}_{\text{ES}\to\text{EP}} = -J^{\text{st}}_{\text{EP}\to\text{ES}} = k_+ P^{\text{st}}_{\text{ES}} - k_- P^{\text{st}}_{\text{EP}} = \frac{1}{3}(k_+ - k_-)$$

In the results above, if the net current is zero ($k_+ = k_-$) then it is an equilibrium stationary state, but if the net current is not zero ($k_+ \neq k_-$), then it is a non-equilibrium stationary state (NESS).

The relaxation time is given by

$$t_{\text{rel}} = \frac{1}{\lambda^*}$$

with λ^* is the maximum absolute value of the non-vanishing eigenvalue. Here we only take the real part of the eigenvalue. The non-zero eigenvalue in this problem is λ_1 and λ_2 and the real part for both eigenvalue is the same, that is $-\frac{3}{2}(k_+ + k_-)$. Thus,

$$\lambda^* = \left| -\frac{3}{2}(k_+ + k_-) \right| = \frac{3}{2}(k_+ + k_-)$$

and

$$t_{\text{rel}} = \frac{2}{3(k_+ + k_-)}$$

(**d**) Evaluate, with the help of a computer, the value of the three components of the probability vector $\vec{P}(t)$ describing the probability of an ensemble of enzymes to be in any of the three possible states. Assume that the enzymes are initially free, i.e. at time $t = 0$ the system is in state E with probability one. Compare the results of two different methods: (i) numerical integration of the system of ordinary differential equations given by the Master equation; (ii) performing the average over many (at least 10^3 stochastic trajectories obtained using the Gillespie algorithm. Plot few trajectories obtained from the Gillespie algorithm method. Values of the parameters $k_+ = 10$ Hz, $k_- = 1$ Hz, total integration time $t_{\max} = 3$ s.

Answer

For solving the system of ordinary differential equations given by Equation (36), we use the Runge-Kutta 4-th order method, implemented in Fortran 95 code (see Appendix A). We use $P_{\text{E}}(t=0) = 1$ and $P_{\text{ES}}(t=0) = P_{\text{ES}}(t=0) = 1$ as initial conditions. The result is shown in Figure 10. The enzymes are initially free then as time increases, it reaches the stationary state where the free enzyme (E), enzyme-substrate (ES), and enzyme-product state are equally probable. This result is in agreement with the stationary probability calculated previously.

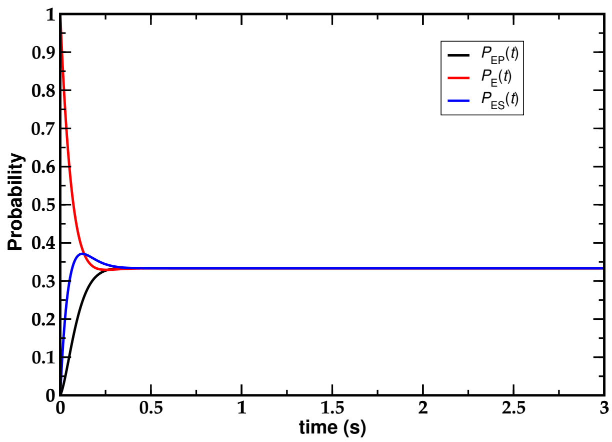


Figure 10. Probability distribution of the system of free enzyme (E), enzyme bound to substrate (ES) and enzyme bound to product (EP) obtained using Runge-Kutta 4-th order. Probability distribution of the free-enzyme state (red), enzyme-substrate state (black), enzyme-product state (blue)

Meanwhile, for the second method, we use the Gillespie algorithm in Python (see Appendix D). In the algorithm, the time spent on a particular state is determined by a random number drawn from an exponential distribution with a rate $\frac{1}{\Gamma(\mu)}$, where $\Gamma(\mu)$ is the escape rate from a particular state μ. For this problem,

$$\Gamma(\text{E}) = \Gamma(\text{ES}) = \Gamma(\text{EP}) = k_+ + k_-$$

the probability of a transition from state μ to state ν is given by:

$$P(\nu|\mu) = \frac{W_{\nu\mu}}{\sum_\nu W_{\nu\mu}} = \frac{W_{\nu\mu}}{\Gamma(\mu)}$$

For example, the probability to make a transition from state E to state EP is $P(\mathrm{EP}|\mathrm{E}) = \frac{k_-}{k_+ + k_-} = 0.0\bar{9}$ while the transition from state E to state ES is $P(\mathrm{ES}|\mathrm{E}) = \frac{k_+}{k_+ + k_-} = 0.\bar{9}0$. The event of the transition is given by a random number drawn from a uniform distribution between 0 and 1. If the random number is 0.5 and the current state is E, then the system will make the transition to state ES. Several trajectories of the stochastic process are shown in Figure 11.

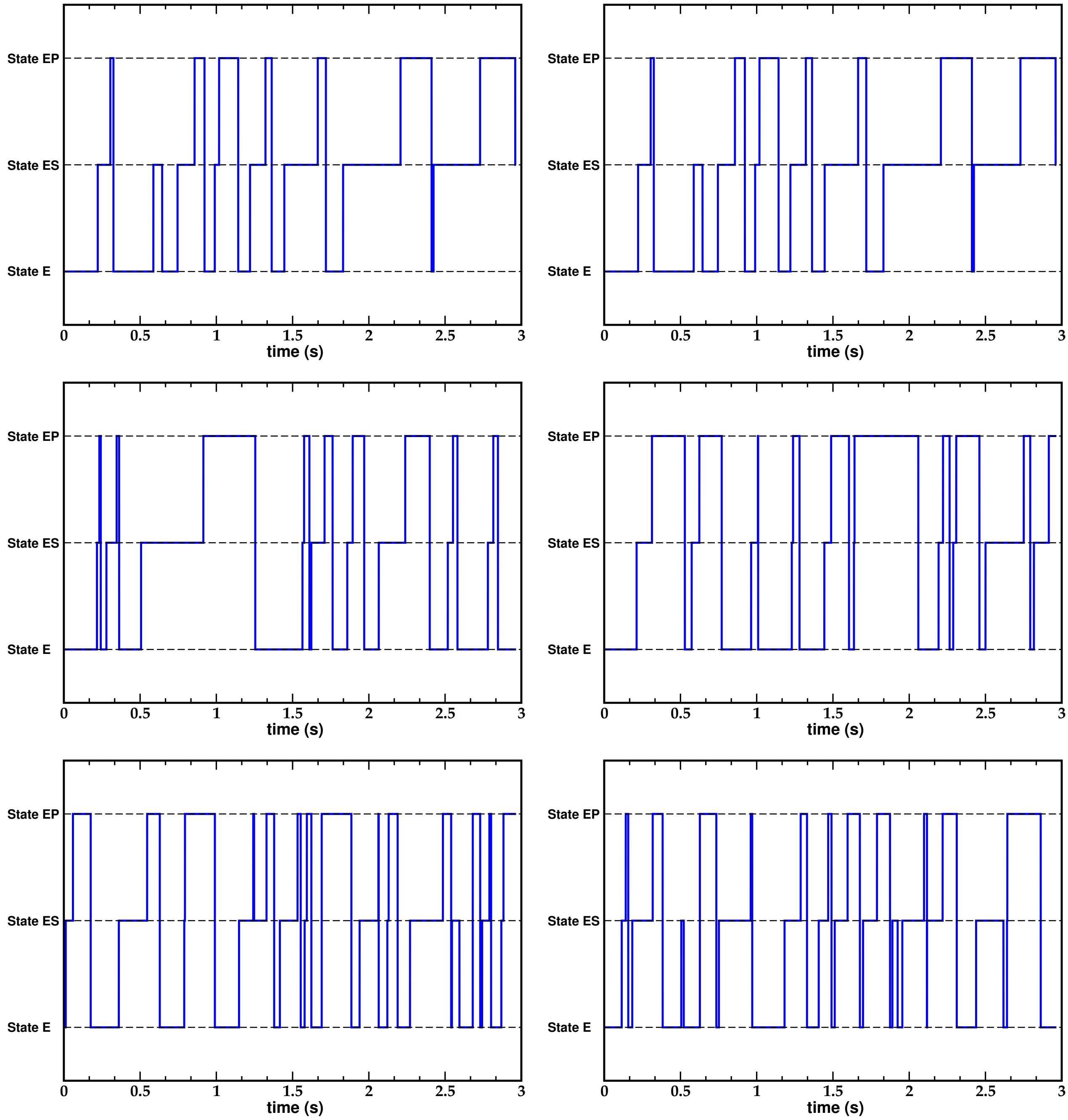


Figure 11. Trajectories of the stochastic processes in the ensemble of the enzyme. From the graphs, we observe the following: If it starts at State E, it is more likely to jump to State ES, compared to State EP. This is because the transition rate from State E to State ES is $k_+ = 10$, whereas the transition rate from State E to State EP is $k_- = 1$. Similarly, if it starts at State ES, it is more likely to jump to State EP compared to State E. And if it starts at State EP, it is more likely to jump to State E compared to State ES.

The probability distribution of each state is shown in Figure 12. Here we show two different numbers of stochastic trajectories (103 and 1000). The probability distribution is obtained by taking the average of the probability over the given trajectories.

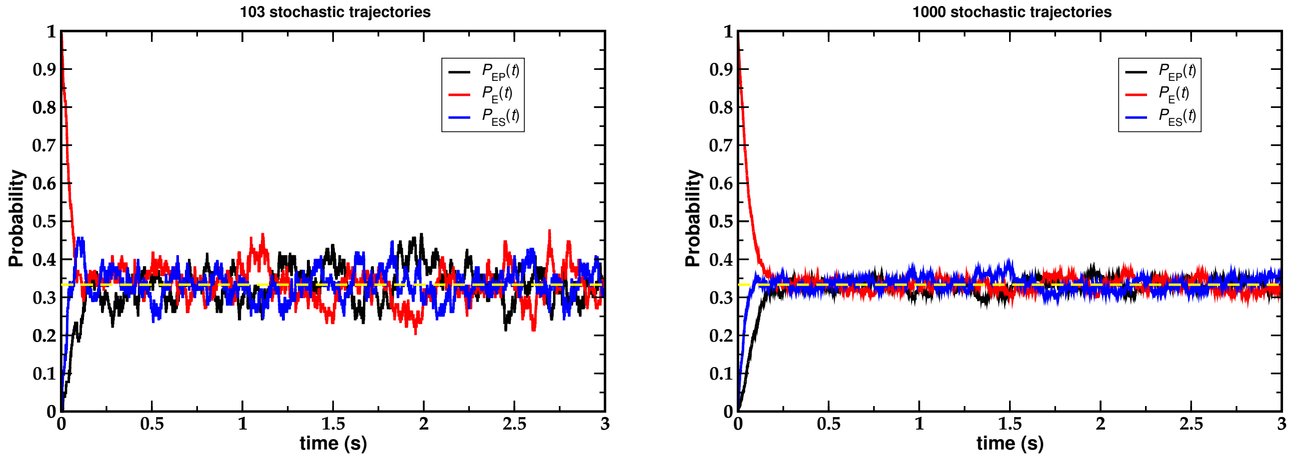


Figure 12. Average probability distribution from Gillespie algorithm using 103 stochastic trajectories (left) and 1000 stochastic trajectories (right). Probability distribution of the free-enzyme state (red), enzyme-substrate state (black), enzyme-product state (blue). The yellow-dashed line is the calculated stationary distribution from problem 3c.

The results in Figure 12 is in agreement with the results obtained by the Runge-Kutta method in Figure 10. It starts initially in the free-enzyme state then reaches the stationary state.

4 Problem 4

Four-state model of molecular motor: Experimentalists sketched a Markov jump model of molecular motor stepping:

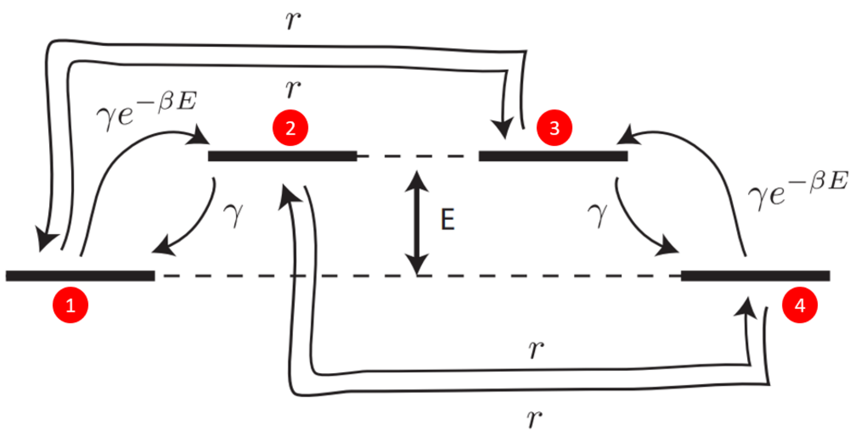


Figure 13. Sketch of the model

Here, r and γ are two rate parameters, $\beta = 1/k_B T$ and E is the energy difference between two energy levels, and T is the temperature of the environment.
(**a**) Write the Master equation of the model in matrix form.

Answer

The Master equation of the model written in matrix form is

$$\frac{\partial}{\partial_t}\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} -\gamma e^{-\beta E} - r & \gamma & r & 0 \\ \gamma e^{-\beta E} & -\gamma - r & 0 & r \\ r & 0 & -\gamma - r & \gamma e^{-\beta E} \\ 0 & r & \gamma & -\gamma e^{-\beta E} - r \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} \tag{41}$$

As a sanity check, we observed that the columns sum to 0, which follows from the conservation of probability ($\frac{\partial}{\partial_t}(P_1 + P_2 + P_3 + P_4) = \frac{\partial}{\partial_t}(1) = 0$). This shows that the Master equation of the model is correct.

(**b**) Calculate the probability to be in any of the state in the long time limit, i.e. its stationary value. Compare this result with the associated Boltzmann distribution.

Answer

The probability to be in any state in the long time limit, i.e. its stationary value, is calculated by equating Equation (41) to $\vec{0}$, wherein there will be no rate of change in the probabilities $\frac{\partial \vec{P}}{\partial_t} = \vec{0}$. This gives us 5 equations with 4 unknowns when we include the normalization equation in Equation (42).

$$P_1 + P_2 + P_3 + P_4 = 1 \tag{42}$$

The stationary distribution then are the following:

$$P_1^{st} = \frac{\gamma + r}{2(2r + \gamma + \gamma e^{-\beta E})} = P_4^{st} \tag{43}$$

$$P_2^{st} = \frac{\gamma e^{-\beta E} + r}{2(2r + \gamma + \gamma e^{-\beta E})} = P_3^{st} \tag{44}$$

To compare this result with the associated Boltzmann distribution, we have to solve the associated Boltzmann distribution first. The associated Boltzmann distribution of the system is when there is detailed balance - where energy is required for a state to transition to another state of higher energy level. This happens when we set $r = 0$. Thus, the associated Boltzmann distribution are the following:

$$P_1^{eq} = \frac{e^{-\beta E_1}}{e^{-\beta E_1} + e^{-\beta E_2} + e^{-\beta E_3} + e^{-\beta E_4}} = \frac{1}{2(1 + e^{-\beta E})} = P_4^{eq} \tag{45}$$

Since $E_1 = E_4 = 0$, and $E_2 = E_3 = E$
Similarly, we can solve for P_2^{eq} and P_4^{eq} as:

$$P_2^{eq} = \frac{e^{-\beta E}}{2(1+e^{-\beta E})} = P_3^{eq} \tag{46}$$

where the partition function is $Z(E) = 2(1+e^{-\beta E})$.

However, the ratio of the stationary distribution of the molecular motor (from Equations 45, 46) does not only depend on E, but also depends on r and γ as shown in Equation (47), compared to its associated Boltzmann distribution which only depends on E, since $r = 0$.

$$\frac{P_{1,4}^{st}}{P_{2,3}^{st}} = \frac{\gamma + r}{\gamma e^{-\beta E} + r} \tag{47}$$

(**c**) What is the stationary current between each of the states? Sketch the direction of the current between all the states. For which value of E is the current equal to zero?

Answer
We refer to J_{ab} as the current from state a to b, whereas $J_{ba} = -J_{ab}$ (having the same magnitude but opposite direction). The stationary current between each states are the following:

$$\begin{aligned}
J_{12}^{st} = -J_{21}^{st} &= P_1^{st}(\gamma e^{-\beta E}) - P_2^{st}\gamma \\
&= \frac{\gamma + r}{2(2r+\gamma+\gamma e^{-\beta E})}(\gamma e^{-\beta E}) - \frac{\gamma e^{-\beta E} + r}{2(2r+\gamma+\gamma e^{-\beta E})}(\gamma) \\
&= \frac{r\gamma(e^{-\beta E} - 1)}{2(2r+\gamma+\gamma e^{-\beta E})}
\end{aligned} \tag{48}$$

$$\begin{aligned}
J_{13}^{st} = -J_{31}^{st} &= P_1^{st} r - P_3^{st} r \\
&= \frac{\gamma + r}{2(2r+\gamma+\gamma e^{-\beta E})}(r) - \frac{\gamma e^{-\beta E} + r}{2(2r+\gamma+\gamma e^{-\beta E})}(r) \\
&= \frac{r\gamma(1 - e^{-\beta E})}{2(2r+\gamma+\gamma e^{-\beta E})}
\end{aligned} \tag{49}$$

$$\begin{aligned}
J_{24}^{st} = -J_{42}^{st} &= P_2^{st} r - P_4^{st} r \\
&= \frac{\gamma e^{-\beta E} + r}{2(2r+\gamma+\gamma e^{-\beta E})}(r) - \frac{\gamma + r}{2(2r+\gamma+\gamma e^{-\beta E})}(r) \\
&= \frac{r\gamma(e^{-\beta E} - 1)}{2(2r+\gamma+\gamma e^{-\beta E})}
\end{aligned} \tag{50}$$

$$\begin{aligned}
J_{34}^{st} = -J_{43}^{st} &= P_3^{st}\gamma - P_4^{st}(\gamma e^{-\beta E}) \\
&= \frac{\gamma e^{-\beta E} + r}{2(2r+\gamma+\gamma e^{-\beta E})}(\gamma) - \frac{\gamma + r}{2(2r+\gamma+\gamma e^{-\beta E})}(\gamma e^{-\beta E}) \\
&= \frac{r\gamma(1 - e^{-\beta E})}{2(2r+\gamma+\gamma e^{-\beta E})}
\end{aligned} \tag{51}$$

$$J_{23}^{st} = -J_{32}^{st} = 0 \tag{52}$$

$$J_{14}^{st} = -J_{41}^{st} = 0 \tag{53}$$

Because we have no transition rates between States 1 and 4, and States 2 and 3.

To sketch the direction of the current between all states, we assume that $e^{-\beta E} < 1$. In this case, the direction of the current flow is shown in Figure 14. This stepping motor can move forward by jumping from State $2 \to 1 \to 3 \to 4 \to 2$.

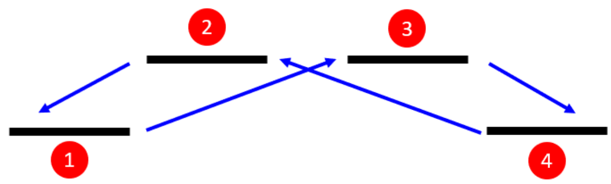


Figure 14. Stationary current flow between the states of a Markov jump model of molecular motor stepping. This stepping motor can move from State 1 to State 3 (as an example), and not from State 3 to State 1.

The value of E for which the current is equal to zero, is when $E = 0$ from Equations 48, 49, 50, 51. In general, $current = 0$ when the product of the $rates$ is equal to the product of $rates$, i.e., when

$$\gamma(r)(\gamma)(r) = \gamma(r)(\gamma)(r) \tag{54}$$

as depicted in Figure 15, when $E = 0$.

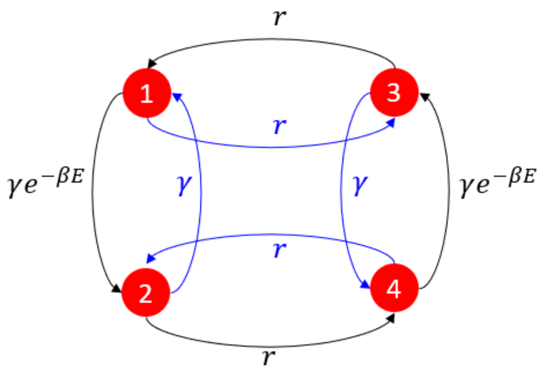


Figure 15. Rates of a four-state model of molecular motor. Current is equal to 0 when $E = 0$

.

5 Problem 5

The dynamics of a photoreceptor neuron subject to a light source of frequency ω can be modelled as a two-state continuous-time Markov process with time-dependent transition rates

$$\omega_{21}(t) = \mu \cos^2(\omega t) \tag{55}$$

$$\omega_{12}(t) = \mu \sin^2(\omega t) \tag{56}$$

where $\mu > 0$ is a characteristic rate parameter.
(**a**) Plot the transition rates as a function of time, for a given value of ω. What happens if the frequency of the incident light is doubled?

Answer
Plotting the transition rates as a function of time, for a given value of ω:

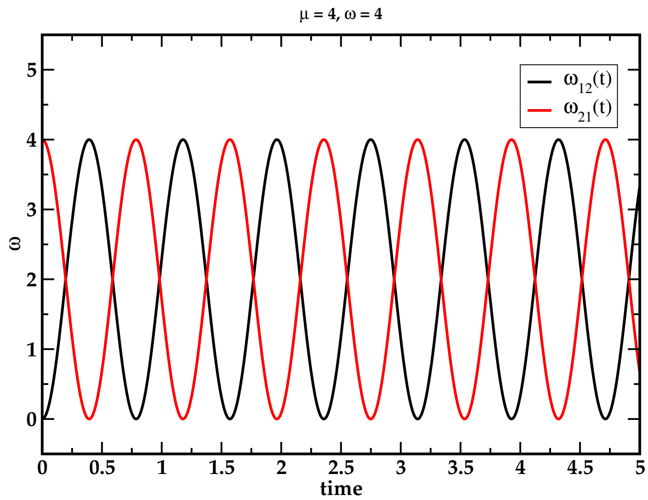


Figure 16. Transition rates as a function of time

Figure 16 shows the plot of transition rates as a function of time. The transition rates are oscillating since the rates are described by sine and cosine functions. When the frequency of the incident light is doubled, we observed that the period is halved, as shown in Figure 17.

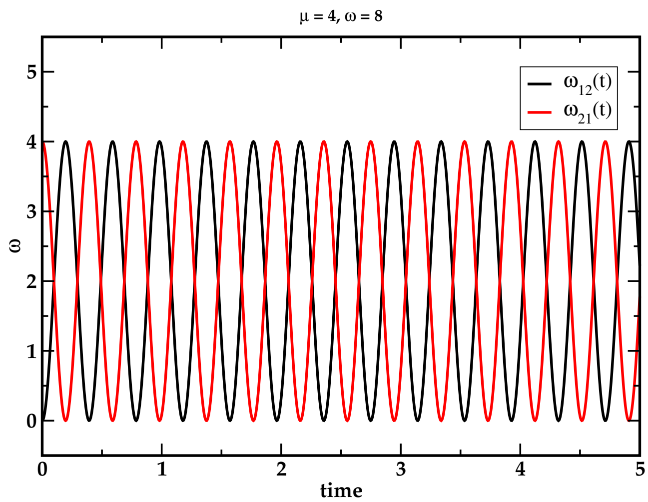


Figure 17. Transition rates as a function of time when frequency is doubled

(**b**) Write the Master equation describing the dynamics of the probabilities $P_1(t)$ and $P_2(t)$ for the neuron to be at state 1 and 2 at time t, respectively.

Answer
Writing the Master equation describing the dynamics of the probabilities $P_1(t)$ and $P_2(t)$ for the neuron to be at state 1 and 2 at time t, respectively. We have:

$$\frac{\partial P_1}{\partial t} = -\omega_{21}(P_1) + \omega_{12}(P_2) = -\mu\cos^2(\omega t)(P_1) + \mu\sin^2(\omega t)(P_2) \tag{57}$$

$$\frac{\partial P_2}{\partial t} = \omega_{21}(P_1) - \omega_{12}(P_2) = \mu\cos^2(\omega t)(P_1) - \mu\sin^2(\omega t)(P_2) \tag{58}$$

(**c**) Derive analytical expressions for $P_1(t)$ and $P_2(t)$.
Answer

For deriving the analytical expressions for $P_1(t)$ and $P_2(t)$, we solve the system of 2 ordinary differential equations:

$$\frac{\partial P_1}{\partial t} = -\mu\cos^2(\omega t)(P_1) + \mu\sin^2(\omega t)(P_2) \tag{59}$$

$$\frac{\partial P_2}{\partial t} = \mu\cos^2(\omega t)(P_1) - \mu\sin^2(\omega t)(P_2) \tag{60}$$

Since $P_1(t) + P_2(t) = 1$ then we can express $P_2(t) = 1 - P_1(t)$
Then we can rewrite Equation (59) as

$$\begin{aligned}
\frac{\partial P_1}{\partial t} &= -\mu\cos^2(\omega t)(P_1) + \mu\sin^2(\omega t)(P_2)\\
\frac{\partial P_1}{\partial t} &= -\mu\cos^2(\omega t)(P_1) + \mu\sin^2(\omega t)(1 - P_1)\\
\frac{\partial P_1}{\partial t} &= -\mu(\cos^2(\omega t) + \sin^2(\omega t))P_1 + \mu\sin^2(\omega t)\\
\frac{\partial P_1}{\partial t} &= -\mu P_1 + \mu\sin^2(\omega t)\\
\frac{\partial P_1}{\partial t} &= -\mu P_1 + \mu\left[\frac{1-\cos(2\omega t)}{2}\right]
\end{aligned}$$

We now solve the the ordinary differential equation:

$$\frac{\partial P_1}{\partial t} + \mu P_1 - \frac{\mu}{2} + \frac{\mu\cos(2\omega t)}{2} = 0 \tag{61}$$

where the solution is of the form $P_1(t) = P_{1h}(t) + P_{1p}(t)$ with h and p as homogeneous and particular solutions, respectively.
Solving for $P_{1h}(t)$, we find:

$$\begin{aligned}
\frac{\partial P_{1h}}{\partial t} + \mu P_{1h} - \frac{\mu}{2} &= 0\\
P_{1h}(t) &= Ae^{-\mu t} + \frac{1}{2} \qquad (62)
\end{aligned}$$

Solving for $P_{1p}(t)$, we propose that the particular solution has the form

$$P_{1p}(t) = C\cos(2\omega t) + B\sin(2\omega t) \tag{63}$$

Substituting Equation (63) to the original ordinary differential equation in Equation (61):

$$\begin{aligned}
-2\omega C\sin(2\omega t) + 2\omega B\cos(2\omega t) + \mu C\cos(2\omega t) + \mu B\sin(2\omega t) - \frac{\mu}{2} + \frac{\mu\cos(2\omega t)}{2} &= 0\\
(-2\omega C + \mu B)\sin(2\omega t) + \left(2\omega B + \mu C + \frac{\mu}{2}\right)\cos(2\omega t) - \frac{\mu}{2} &= 0
\end{aligned}$$

When $t = 0$, then $\sin(2\omega t) = 0$, and $\cos(2\omega t) = 1$

$$\begin{aligned}
2\omega B + \mu C + \frac{\mu}{2} - \frac{\mu}{2} &= 0\\
C &= -\frac{2\omega B}{\mu} \qquad (64)
\end{aligned}$$

Then we have $P_{1p}(t)$ as:

$$P_{1p}(t) = \frac{-2\omega B}{\mu}\cos(2\omega t) + B\sin(2\omega t) \tag{65}$$

Finally, $P_1(t) = P_{1h}(t) + P_{1p}(t)$ from Equations (62) and (65):

$$P_1(t) = Ae^{-\mu t} + \frac{1}{2} + B\sin(2\omega t) - \frac{2\omega B}{\mu}\cos(2\omega t) \tag{66}$$

Since $P_2 = 1 - P_1$,

$$P_2(t) = \frac{1}{2} - Ae^{-\mu t} - B\sin(2\omega t) + \frac{2\omega B}{\mu}\cos(2\omega t) \tag{67}$$

where A and B are constants that depend on the initial conditions.
(**d**) Evaluate $P_1(t)$ and $P_2(t)$ in the limit of t large, i.e. when $t >> \mu^{-1}$

Answer
In the limit of t large, i.e. when $t >> \mu^{-1}$,

$$P_1^{\mathrm{st}} = \frac{1}{2} + B\sin(2\omega t) - \frac{2\omega B}{\mu}\cos(2\omega t) \tag{68}$$

$$P_2^{\mathrm{st}} = \frac{1}{2} - B\sin(2\omega t) + \frac{2\omega B}{\mu}\cos(2\omega t) \tag{69}$$

Only the oscillation remains in the long time limit. Here, $P_1^{\mathrm{st}} + P_2^{\mathrm{st}} = 1$.

Appendices

A Runge-Kutta 4th Order

This is the Runge-Kutta 4-th Order that is used to obtained the results in Figure 10

```
PROGRAM rungekutta

IMPLICIT NONE
INTEGER,PARAMETER  :: dp = KIND(1.D0)
INTEGER,PARAMETER :: N = 3001
INTEGER :: i
REAL(KIND=dp),PARAMETER:: t0 = 0._dp, tmax = 3._dp
REAL(KIND=dp),PARAMETER:: x0 = 0._dp, y0 = 1._dp, z0 = 0._dp
REAL(KIND=dp) :: f0,f1,f2,f3, g0,g1,g2,g3, h0,h1,h2,h3, delta
REAL(KIND=dp), ALLOCATABLE :: t(:), x(:), y(:), z(:)
REAL(KIND=dp), EXTERNAL :: f,g,h

ALLOCATE(t(0:N))
ALLOCATE(x(0:N))
ALLOCATE(y(0:N))
ALLOCATE(z(0:N))

OPEN(UNIT=96, FILE='rungekutta_x.dat', STATUS='UNKNOWN')
OPEN(UNIT=97, FILE='rungekutta_y.dat', STATUS='UNKNOWN')
OPEN(UNIT=98, FILE='rungekutta_z.dat', STATUS='UNKNOWN')

!interval
delta = (tmax-tmin)/(N-1)

t(0) = t0
x(0) = x0
y(0) = y0
z(0) = z0

WRITE(96,*) t(0), x(0)
WRITE(97,*) t(0), y(0)
WRITE(98,*) t(0), z(0)

DO i=1,N
t(i) = t0+(i)*delta

f0 = f(t(i-1),x(i-1),y(i-1),z(i-1))
g0 = g(t(i-1),x(i-1),y(i-1),z(i-1))
h0 = h(t(i-1),x(i-1),y(i-1),z(i-1))

f1 = f(t(i-1)+0.5_dp*delta, x(i-1)+0.5*delta*f0, y(i-1)+0.5*delta*f0, z(i-1)+0.5*delta*f0)
g1 = g(t(i-1)+0.5_dp*delta, x(i-1)+0.5*delta*g0, y(i-1)+0.5*delta*g0, z(i-1)+0.5*delta*g0)
h1 = h(t(i-1)+0.5_dp*delta, x(i-1)+0.5*delta*h0, y(i-1)+0.5*delta*h0, z(i-1)+0.5*delta*h0)

f2 = f(t(i-1)+0.5_dp*delta, x(i-1)+0.5*delta*f1, y(i-1)+0.5*delta*f1, z(i-1)+0.5*delta*f1)
g2 = g(t(i-1)+0.5_dp*delta, x(i-1)+0.5*delta*g1, y(i-1)+0.5*delta*g1, z(i-1)+0.5*delta*g1)
h2 = h(t(i-1)+0.5_dp*delta, x(i-1)+0.5*delta*h1, y(i-1)+0.5*delta*h1, z(i-1)+0.5*delta*h1)

f3 = f(t(i-1)+delta, x(i-1)+delta*f2, y(i-1)+delta*f2, z(i-1)+delta*f2)
g3 = g(t(i-1)+delta, x(i-1)+delta*g2, y(i-1)+delta*g2, z(i-1)+delta*g2)
h3 = h(t(i-1)+delta, x(i-1)+delta*h2, y(i-1)+delta*h2, z(i-1)+delta*h2)

x(i) = x(i-1) + delta*(f0+2*f1+2*f2+f3)/6._dp
y(i) = y(i-1) + delta*(g0+2*g1+2*g2+g3)/6._dp
```

```
z(i) = z(i-1) + delta*(h0+2*h1+2*h2+h3)/6._dp

WRITE(96,*) t(i), x(i)
WRITE(97,*) t(i), y(i)
WRITE(98,*) t(i), z(i)

END DO

CLOSE(96)
CLOSE(97)
CLOSE(98)

DEALLOCATE(t,x,y,z)

END PROGRAM rungekutta

FUNCTION f(t,x,y,z) RESULT (dx)!EP
IMPLICIT NONE
INTEGER,PARAMETER :: dp = KIND(1.D0)
REAL(KIND=dp),PARAMETER :: kp = 10._dp, km = 1._dp
REAL(KIND=dp), INTENT(IN) :: t,x,y,z
REAL(KIND=dp)  :: dx

dx = -(kp+km)*x+km*y+kp*z

RETURN
END FUNCTION

FUNCTION g(t,x,y,z) RESULT (dy)!E
IMPLICIT NONE
INTEGER,PARAMETER :: dp = KIND(1.D0)
REAL(KIND=dp),PARAMETER :: kp = 10._dp, km = 1._dp
REAL(KIND=dp), INTENT(IN) :: t,x,y,z
REAL(KIND=dp)  :: dy

dy = kp*x-(kp+km)*y+km*z

RETURN
END FUNCTION

FUNCTION h(t,x,y,z) RESULT (dz)!ES
IMPLICIT NONE
INTEGER,PARAMETER :: dp = KIND(1.D0)
REAL(KIND=dp),PARAMETER :: kp = 10._dp, km = 1._dp
REAL(KIND=dp), INTENT(IN) :: t,x,y,z
REAL(KIND=dp)  :: dz


dz = km*x+kp*y-(kp+km)*z

RETURN
END FUNCTION
```

B Euler's Numerical Simulation Scheme: Probability density of θ

This is the code used to calculate the probability density of θ when $\omega = 10$ as shown in Figures 1, and 2. The same code is used to generate Figures 3 and 4, only changing $\omega = 0$.

breakatwhitespace

```
# -*- coding: utf-8 -*-
"""prob_1b_omega10.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1AdPi_0CXxhk1NUirwjUex9JTQG0H_8r8
"""

import numpy as np
import matplotlib.pyplot as plt

#parameter
Dr = 0.17
Dt = 0.2
omega = 10
v = 30
delta_t = 0.01
n_realizations = 1000

#parameter for gaussian
mu = 0
sigma = np.sqrt(delta_t)

#function to perform Euler scheme
def euler_one_traj(N_time, mu, sigma, delta_t, Dr):
    #initial condition
    theta_init = 0
    theta_traj = np.array([],dtype=int)

    theta = theta_init
    theta_traj = np.append(theta_traj, theta)

    for i in range(N_time-1):
        s = np.random.normal(mu, sigma)
        theta = theta + omega*delta_t + np.sqrt(2*Dr)*s
        theta_traj = np.append(theta_traj,theta)
    return theta_traj

#function to calculate n_realizations for different t
def traj_n_realizations(t_max):
    t_min = 0.0
    N_time = int((t_max-t_min)/delta_t +1)
    time = np.linspace(t_min, t_max, N_time)

    traj_list = []
    for n in range(n_realizations):
        traj = euler_one_traj(N_time, mu, sigma, delta_t, Dr)
        traj_list.append(traj)
    return traj_list, time

#Plotting the trajectories at different t

traj_list_1, time1 =traj_n_realizations(1)
traj_list_5, time5 =traj_n_realizations(5)
traj_list_10, time10 =traj_n_realizations(10)

fig, ax = plt.subplots(1,3, figsize=(15,5))
for i in traj_list_1:
    ax[0].plot(time1, i)

for j in traj_list_5:
    ax[1].plot(time5, j)

for k in traj_list_10:
    ax[2].plot(time10, k)

ax[0].set_title('t = 1s')
ax[1].set_title('t = 5s')
```

```
ax[2].set_title('t = 10s')
ax[0].set_xlabel('t')
ax[0].set_ylabel('  ')
ax[1].set_xlabel('t')
ax[1].set_ylabel('  ')
ax[2].set_xlabel('t')
ax[2].set_ylabel('  ')

plt.tight_layout()
plt.savefig('prob_1b_omega10.png', dpi=300)
plt.show()

#Plotting the probability density of theta from the last element of the trajectory list
element_list1 = []
for i in traj_list_1:
    last_element = i[-1]
    element_list1.append(last_element)

element_list5 = []
for j in traj_list_5:
    last_element = j[-1]
    element_list5.append(last_element)

element_list10 = []
for k in traj_list_10:
    last_element = k[-1]
    element_list10.append(last_element)


fig, ax = plt.subplots(1,3, figsize=(15,5))
ax[0].hist(element_list1, density=True)
ax[0].set_title('t = 1s')

ax[1].hist(element_list5, density=True)
ax[1].set_title('t = 5s')

ax[2].hist(element_list10, density=True)
ax[2].set_title('t = 10s')

plt.show()

#function to calculate the probability density from the analytical solution
def analytical(t, start, end):
    prob_list = []
    theta = np.linspace(start,end,100)
    mean_ = omega*t
    for th in theta:
        prob = np.exp(-(th-mean_)**2/(4*Dr*t))/np.sqrt(4*np.pi*Dr*t)
        prob_list.append(prob)
    return prob_list, theta

analytical_1s, theta_1s = analytical(1, 8, 12)
analytical_5s, theta_5s = analytical(5, 40, 60)
analytical_10s, theta_10s = analytical(10, 90, 110)

#Plotting together the numerical and analytical solution of the probability density
fig, ax = plt.subplots(1,3, figsize=(15,5))
ax[0].hist(element_list1, density=True, label = 'numerical')
ax[0].plot(theta_1s, analytical_1s, label = 'theoretical')
ax[0].set_title('t = 1s')

ax[1].hist(element_list5, density=True, label = 'numerical')
ax[1].plot(theta_5s, analytical_5s, label = 'theoretical')
ax[1].set_title('t = 5s')

ax[2].hist(element_list10, density=True, label = 'numerical')
ax[2].plot(theta_10s, analytical_10s, label = 'theoretical')
ax[2].set_title('t = 10s')

plt.suptitle('Probability density of   ')
ax[0].set_ylabel('Probability density')
ax[1].set_ylabel('Probability density')
ax[2].set_ylabel('Probability density')
ax[0].set_xlabel('  ')
ax[1].set_xlabel('  ')
ax[2].set_xlabel('  ')
```

```
ax[0].legend()
ax[1].legend()
ax[2].legend()

plt.tight_layout()
plt.savefig('prob_1b_omega10_2.png', dpi=300)
plt.show()
```

C Euler's Numerical Simulation Scheme: Stochastic trajectories of θ, x, y

This is the code used to generate the stochastic trajectories of θ, x, and y with the following parameter values: $D_t = 0.2\mu\text{m}^2/\text{s}$, $D_r = 0.17\text{rad}^2/\text{s}$, $v = 30\text{m/s}$, and $\omega = 10\text{rad/s}$, as shown in Figures 6, and **??**. The same code is used to generate Figures 5 and **??**, only changing $\omega = 0$.

breakatwhitespace

```
# -*- coding: utf-8 -*-
"""prob_1d_omega10.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1IN9Y2jmQ3n6vyVEfHq5rkyioiedt5lUF
"""

import numpy as np
import matplotlib.pyplot as plt

#parameter
Dr = 0.17
Dt = 0.2
omega = 10
v = 30
delta_t = 0.01
n_realizations = 5

#parameter for gaussian
mu = 0
sigma = np.sqrt(delta_t)

#function to calculate theta, x, y using Euler scheme
def euler_one_traj(N_time, mu, sigma, delta_t, Dr, Dt):
    #initial condition
    theta_init = 0
    x_init = 0
    y_init = 0
    theta_traj = np.array([],dtype=int)
    x_traj = np.array([],dtype=int)
    y_traj = np.array([],dtype=int)

    theta = theta_init
    x = x_init
    y = y_init
    theta_traj = np.append(theta_traj, theta)
    x_traj = np.append(x_traj, x)
    y_traj = np.append(y_traj, y)

    for i in range(N_time-1):
        s1 = np.random.normal(mu, sigma)
        s2 = np.random.normal(mu, sigma)
        s3 = np.random.normal(mu, sigma)
        theta = theta + omega*delta_t + np.sqrt(2*Dr)*s1
        x = x + v*np.cos(theta)*delta_t + np.sqrt(2*Dt)*s2
        y = y + v*np.sin(theta)*delta_t + np.sqrt(2*Dt)*s3

        theta_traj = np.append(theta_traj,theta)
        x_traj = np.append(x_traj, x)
        y_traj = np.append(y_traj, y)
    return theta_traj, x_traj, y_traj

#function to generate multiple(n_realizations) trajectories
def traj_n_realizations(t_max):
    t_min = 0.0
    N_time = int((t_max-t_min)/delta_t +1)
    time = np.linspace(t_min, t_max, N_time)

    traj_theta_list = []
    traj_x_list = []
    traj_y_list = []
    for n in range(n_realizations):
        theta_traj, x_traj, y_traj = euler_one_traj(N_time, mu, sigma, delta_t, Dr, Dt)
        traj_theta_list.append(theta_traj)
```

```
        traj_x_list.append(x_traj)
        traj_y_list.append(y_traj)
    return traj_theta_list, traj_x_list, traj_y_list, time

#plotting the trajectories with t = 10s

traj_theta_list, traj_x_list, traj_y_list, time = traj_n_realizations(10)

fig, ax = plt.subplots(1,3, figsize=(15,5))
for i in traj_theta_list:
    ax[0].plot(time, i)

for j in traj_x_list:
    ax[1].plot(time, j)

for k in traj_y_list:
    ax[2].plot(time, k)

ax[0].set_xlabel('t')
ax[0].set_ylabel('  ')
ax[1].set_xlabel('t')
ax[1].set_ylabel('x')
ax[2].set_xlabel('t')
ax[2].set_ylabel('y')

plt.tight_layout()
plt.savefig('prob_1d_omega10.png', dpi=300)
plt.show()
```

D Gillespie Algorithm

This is the code of the Gillespie Algorithm that is used to calculate the average probability distribution in Figure 12.

breakatwhitespace

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm
import time


# In[2]:


#state E = 1, state ES = 2, state EP = 3


# In[3]:


k_plus = 10
k_min = 1

W21 = W32 = W13 = k_plus
W12 = W31 = W23 = k_min

gamma_1 = W21+W31
gamma_2 = W12+W32
gamma_3 = W13+W23


# In[4]:


def tau_generator(state):
    if (state == 1):
        tau = np.random.exponential(1/gamma_1, 1)
    elif (state ==2):
        tau = np.random.exponential(1/gamma_2, 1)
    else:
        tau = np.random.exponential(1/gamma_3, 1)
    return tau[0]


# In[5]:


def change_state(state):
    if (state==1):
        if np.random.uniform()<P21:
            state = 2
        else:
            state = 3
    elif (state==2):
        if np.random.uniform()<P12:
            state = 1
        else:
            state = 3
    else:
        if np.random.uniform()<P13:
            state = 1
        else:
            state = 2
    return state


# In[6]:
```

```
# where to jump
#from 1:
P21 = W21/gamma_1
P31 = W31/gamma_1
#from 2:
P12 = W12/gamma_2
P32 = W32/gamma_2
#from 3:
P13 = W13/gamma_3
P23 = W23/gamma_3


# In[7]:


def trajectory_generator(init_state,delta_t,t_max,N_realizations,N_time):
    state_traj = np.array([],dtype=int)
    P1_ave = np.array([])
    P2_ave = np.array([])
    P3_ave = np.array([])

    #initial condition, t = 0, state initial = init_traj
    t = 0
    state = init_state
    state_traj = np.append(state_traj, state)

    #initial condition of probability distribution
    P1_ave = np.append(P1_ave,1/N_realizations)
    P2_ave = np.append(P2_ave,0)
    P3_ave = np.append(P3_ave,0)

    while t<=(N_time-1):
        t0 = 0
        tau = tau_generator(state)
        while abs(t0-tau)>delta_t:
            t0 = t0+delta_t
            t = t+1
            state_traj = np.append(state_traj, state)
            if (state==1):
                P1 = 1/N_realizations
                P2 = 0
                P3 = 0
                #print('state=1','P1',P1,'P2',P2,'P3',P3)
            elif (state==2):
                P1 = 0
                P2 = 1/N_realizations
                P3 = 0
                #print('state=2','P1',P1,'P2',P2,'P3',P3)
            else:
                P1 = 0
                P2 = 0
                P3 = 1/N_realizations
                #print('state=3','P1',P1,'P2',P2,'P3',P3)
            P1_ave = np.append(P1_ave,P1)
            P2_ave = np.append(P2_ave,P2)
            P3_ave = np.append(P3_ave,P3)
            #print(t)
            if(t==(N_time-1)):
                #print(t,'break')
                break
        state = change_state(state)
        if(t==(N_time-1)):
            #print(t,'break')
            break
    return state_traj,P1_ave,P2_ave,P3_ave


# In[8]:


init_state = 1
N_realizations = 103

t_min=0.0
t_max=3.0
```

```
delta_t = 0.0001
N_time = int(t_max/delta_t +1)

P1_traj = 0
P2_traj = 0
P3_traj = 0
all_state_traj = np.array([])
for j in tqdm(range(N_realizations)):
    state,P1,P2,P3 = trajectory_generator(init_state,delta_t,t_max,N_realizations,N_time)
    P1_traj += P1
    P2_traj += P2
    P3_traj += P3
    all_state_traj = np.append(all_state_traj,state)

#timelist
time = np.array([])
t = t_min
for i in range(N_time):
    time = np.append(time,t)
    t += delta_t


# In[11]:


f = open("P_E_103.txt", "a")
g = open("P_ES_103.txt", "a")
h = open("P_EP_103.txt", "a")
for i in range(N_time):
    print(time[i],P1_traj[i],file=f)
    print(time[i],P2_traj[i],file=g)
    print(time[i],P3_traj[i],file=h)
f.close()
g.close()
h.close()
```