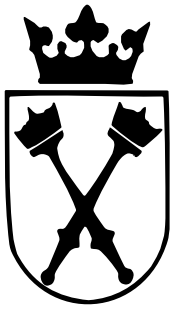
Transport and signal processing in noisy biophysical systems

Michał A. Żabicki

Jagiellonian University

PhD Thesis

Supervisor: Prof. Ewa Gudowska-Nowak



Marian Smoluchowski Institute of Physics

Faculty of Physics, Astronomy and Applied Computer Science

Kraków, 2011-2012

Oświadczenie

Ja niżej podpisany Michał Andrzej Żabicki (nr indeksu: 1096789) doktorant Wydziału Fizyki, Astronomii i Informatyki Stosowanej Uniwersytetu Jagiellońskiego oświadczam, że przedłożona przeze mnie rozprawa doktorska pt. "Transport and signal processing in noisy biophysical systems" jest oryginalna i przedstawia wyniki badań wykonanych przeze mnie osobiście, pod kierunkiem prof. Ewy Gudowskiej-Nowak. Pracę napisałem samodzielnie.

Oświadczam, że moja rozprawa doktorska została opracowana zgodnie z Ustawą o prawie autorskim i prawach pokrewnych z dnia 4 lutego 1994 r. (Dziennik Ustaw 1994 nr 24 poz. 83 wraz z późniejszymi zmianami).

Jestem świadom, że niezgodność niniejszego oświadczenia z prawdą ujawniona w dowolnym czasie, niezależnie od skutków prawnych wynikających z ww. ustawy, może spowodować unieważnienie stopnia nabytego na podstawie tej rozprawy.

Kraków, dnia ....

Contents

List of Figures

List of Tables

Acknowledgements

In the first place I would like to thank prof. Werner Ebeling from Humboldt-Universität zu Berlin, for the original idea behind the research presented in this thesis.

Secondly, for the never-ending patience I have to thank my supervisor, prof. Ewa Gudowska - Nowak, who have always tried to find time for my numerous iterations of this manuscript.

Financial support for the research has been partially provided by the ESF program Exploring Physics of Small Devices (EPSD) and has operated within the Foundation for Polish Science co-financed by the European Regional Development Fund covering, under the Agreement No. MPD/2009/6, the Jagiellonian University International Ph.D. Studies in Physics of Complex Systems.

I would also like to thank Bartosz Lisowski from my department for hours of stimulating discussions on the subject of molecular motors (and more).

For the remarks on molecular motors efficiency and inspirations for the future research I would like to specially thank Martin Bier from East Carolina University.

I would like to thank my mother for the math education that brought me to finally study science and for all her "when-will-you-finish" questions...

Last but not least - for patience, support and giving up a lot of time we could spend together - I have to thank my wife, Marta.

In the memory of my grandfather,
Andrzej Żabicki (1929–2006),
physicist.

Chapter 1

Introduction

"Molecular motor is NOT a steam engine!"

The most popular phrase
of the 2010 Molecular Motors Conference in Santa Fe, NM

Long, long time ago when I still had been in the primary school, there was this strategy computer game called Settlers. While I hardly remember the plot, characters or the game mechanics, the flashback of serfs transporting goods along paths came right in the moment when I close my eyes and started thinking about transport phenomenon.

Day after, I have searched for some more information and again, right in the moment I have looked at the game screenshot (Fig. 1.1), an analogy with the topic of my scientific work got to be even more remarkable. Picture was showing crowded scene of Barracks, Slaughterhouses, Pig Farms and Gold Mines - all connected by signposted paths with tiny workers carrying timber, pork heads or gold between the aforementioned buildings.

Even a short episode of the strategy game shows importance of transport issues. The easiest way to conquer opponent's empire is to destroy his carriers and to blur his paths, cutting down supplies distribution and subsequently bringing it to a standstill.

The situation in not so different in the case of a biological cell (see Fig. 1.2). Crowded and busy body of the cell can be forcibly crippled by mutations that induce reduced transport and, as a result, can cause number of neurological defects [1] as Parkinson's disease [2], Alzheimer's disease [3] or Huntington's disease [4]. But neurology is not an only division of medicine that has to deal with problems caused by disfunction of cell transport abilities [5]. Disabilites connected with mo-

Figure 1.1: Screenshot from "Settlers II" game (1996)

tor protein failures include hypertrophic cardiomyopathy [6], Usher syndrome [7] or Griscelli syndrome [8].

What constitutes the cell transport infrastructure? Along with microfilaments, intermediate filaments, the microtubules, formed as a polymers of α- and β-tubulin dimers, are the signposted paths.

On the other hand, molecular proteins as myosin, dynein or kinesin are the cell serfs. While varying in size, complexity and function they all in common convert chemical energy from the hydrolysis of the biological main source of chemical energy — adenosine triphosphate (ATP) into the mechanical work.

1.1 Reynolds number

There are two main features that distinguish motors that can be observed around, like engine in the car, from the *molecular* ones. First of all, as the name itself indicates, their size is on the molecular scale. As an example, kinesin head which can be loosely associated with a car wheel is approximately a ball of a 3nm radius [9]. To compare, a radius of a midsize car wheel is about 300mm — 8 orders of magnitude larger.

The other crucial aspect is the environment in which a molecular motor op-

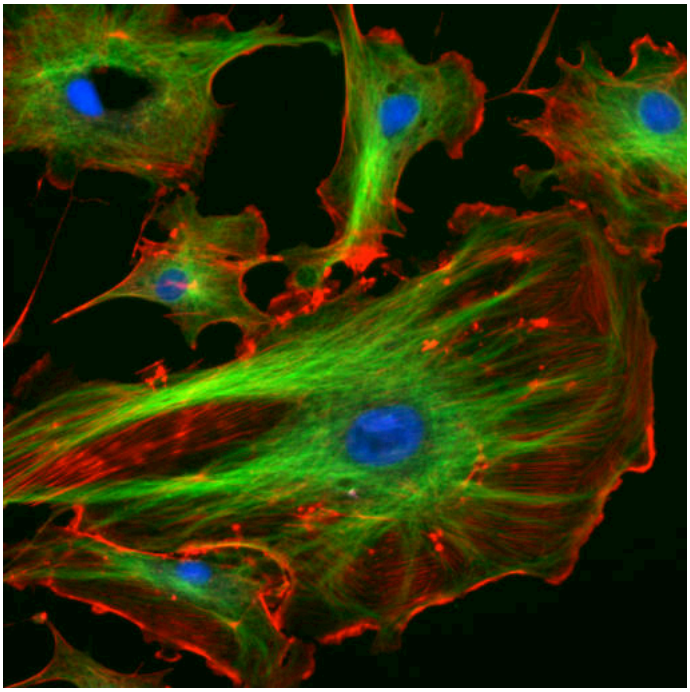


Figure 1.2: Endothelial cells under the microscope. Nuclei are stained blue with DAPI, microtubles are marked green by an antibody bound to FITC and actin filaments are labelled red with phalloidin bound to TRITC. Bovine pulmonary artery endothelial cells. From http://rsb.info.nih.gov/ij/images/ (public domain)

erates. Instead of sparse air in which most cars cruise, motor proteins as kinesin have to overcome viscous forces of dense, crowded cell surroundings. It is more like trying to travel by a car across the Amazonian jungle.

Although it is intuitively known that walking in a pool filled with water is more struggling than walking in the empty one, it is always convenient to have an objective measure for certain phenomenon. The basic equation for the fluid dynamics is the Navier-Stokes equation, i.e. the Newton's law for a fluid:

$$\rho\left[\frac{\partial\vec{u}}{\partial t}+\left(\vec{u}\cdot\vec{\nabla}\right)\vec{u}\right]=-\vec{\nabla}p+\eta\nabla^2 u, \tag{1.1}$$

where p is pressure and η stands for fluid viscosity and ρ represents fluid density. By introducing the dimensionless quotient

$$\mathrm{Re}=\frac{inertial forces}{viscous forces}=\rho\frac{\frac{u}{t}+\frac{u^2}{L}}{\eta\frac{u}{L^2}} \tag{1.2}$$

and then assuming that $t\cdot u\approx L$, with a characteristic lenghth scale L, we can specify conditions for laminar and turblent fluid flow. The so called Reynolds number

$$\mathrm{Re}=\rho\frac{\frac{L^2}{u}\left(\frac{u}{t}+\frac{u^2}{L}\right)}{\eta}=\rho\frac{\frac{L^2}{u}\left(\frac{u^2}{L}+\frac{u^2}{L}\right)}{\eta}=\rho\frac{uL}{\eta}, \tag{1.3}$$

is small for laminar flows, when viscosity η is dominant and diffusion term on the right hand side of (1.1) governs the fluid dynamics. In contrast, when inertial forces (LHS of (1.1)) dominate the flow Re is a big number indicating disorganized, turbulent flow.

For a man (whose spatial dimensions is of order of $L=1$ m) swimming (moderately fast about $v=1m/s$) in water (kinematic viscosity at $20^{\circ}C$ is $\nu=\frac{\eta}{\rho}=10^{-6}m^2/s$,) Reynolds number is about $\mathrm{Re}=10^6$. For a Kinesin-1 traveling on an average velocity of $0.78\mu m/s$ [10] along microtubule surrounded by a cytoplasm it is approximately $\mathrm{Re}=10^{-13}$. As an other example, a duck flying at 20 m/s experiences Reynolds number of approximately 300000, a dragon-fly at 7 m/s about an magnitude less. In the context of studying molecular motors it has to be understood what are the practical implications of the kind of environment they operate in.

As nicely expressed by E. Purcell in his very interesting article:

> *If you are at very low Reynolds number, what you are doing at the moment is entirely determined by the forces that are exerted on you at that moment, and by nothing in the past. [11]*

Those observations will be very important through the thesis and the mathematical aspect of it will be discussed in Section 1.2.

1.2 Langevin equation for a Brownian Motion

The most obvious starting point to model any dynamics is the Newton's Second Law:

$$\vec{F} = m\ddot{\vec{x}} \tag{1.4}$$

That is: a body with a mass m experiences acceleration $\ddot{x}$ when force F is acting on that body. The problem is that, as mentioned in previous paragraphs, environment in which molecular motor operates varies significantly from lab vacuum, for which eq. 1.4 can be easily applied. Another, viscous force $\gamma\dot{x}$ exerted on an object should be added to take account of dense cell surroundings:

$$\vec{F} = m\ddot{\vec{x}} + \gamma\dot{\vec{x}} \tag{1.5}$$

But viscous force is still not enough. In 1827 Robert Brown has been examining pollen grains floating on a water surface. From what he had observed under microscope, he developed the idea that leaping of particles in fluid should not be associated only with organic matter, as it was believed at his time. Instead of that, irrespectively of a kind of particle he put onto the water, their chaotic movement was the same. From that, he claimed that this motion is not life-related, but rather is induced by some external conditions.

It was almost one century until in 1908 Paul Langevin proposed [12] an equation which described movement of Brown's grains in a language of, what we call now, stochastic differential equations (SDE). Langevin's idea was to account for action of different degrees of freedom by introducing single random variable $\bar{\eta}$. The resulting one-dimensional Newton-like equation takes the form:

$$\bar{\eta}\left(t\right) = m\ddot{x} + \gamma\dot{x}, \tag{1.6}$$

with $\bar{\eta}\left(t\right)$ called usually a *noise term* and having a physical meaning of a *stochastic force* — a fluctuating, uncorrelated in time force of a zero mean which is independent of x and $\langle\bar{\eta}(t)\bar{\eta}(t')\rangle = 2D\delta(t-t')$. Usually, instead using $\bar{\eta}\left(t\right)$ one defines $\bar{\eta}\left(t\right) = \sqrt{2D}\eta\left(t\right)$. Now, $\eta\left(t\right)$ is a white Gaussian noise with a zero mean and a correlation function $< \eta\left(t\right)\eta\left(t'\right) >= \delta(t-t')$ and D characterizes the noise intensity, related to the friction coefficient by the fluctuation-dissipation theorem:

$$D = \gamma k_B T/m \tag{1.7}$$

where k_B is the Boltzmann constant and T is temperature.

The actual value of noise term will be discussed later in Chapter 2, devoted to models concepts.

1.3 Smoluchowski experiment and the ratchet potential

In 1912 polish physicist Marian Smoluchowski [13] analyzed a thought experiment, which has been later popularized by Richard Feynmann in his famous "Lectures on Physics" [14]. The idea is based on a pawl and ratchet mechanism. Let's take a round gear with asymmetric teeth which can freely rotate around axis given by a rod. Then we put a pivoting finger called pawl that allows movement of a gear in one direction but blocks it whenever it tries to operate in the opposite direction. This mechanism now should be put into a heat bath of temperature T_2 (see Fig. 1.3). By random collisions of the gas molecules (Brownian motion in fluids) with gear's teeth, it will rotate to one side or the other. But because of the ratchet mechanism, the actual rotation can only be achieved in the one direction, determined by the asymmetry of teeth shape. Having rotating rod we can use its motion for some useful work, the same way as it used in e.g. watermills.

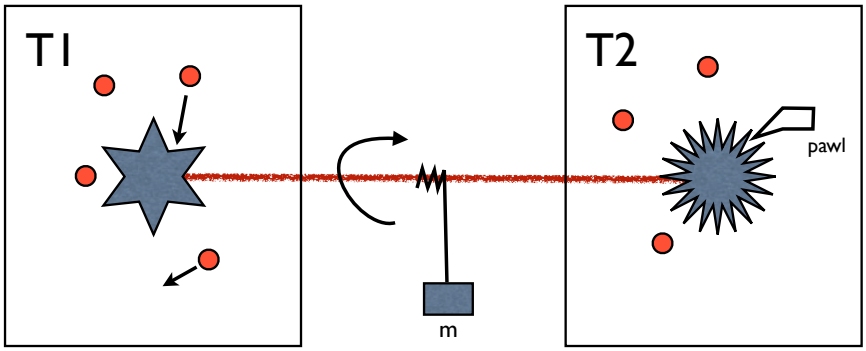


Figure 1.3: Feynman ratchet

The problem is that this experiment would violate the second law of thermodynamics — no matter how cleverly designed, structural features alone cannot bias Brownian motion — as formulated by Astumian in [15]. Paradox has been solved by Feynmann along with his lecture on the topic and the mathematical proof can be found in the Magnasco's article [16]. Despite that, apparatus proposed by Smolu-

chowski could work. There must be however, a temperature gradient $\Delta T = T_1 - T_2 \neq 0$ to obey thermodynamics second law. Experimental realization of a Smoluchowski-Feynamnn ratchet, yet outside of thermal equilibrium, has been recently constructed by Eshuis *et al.* [17].

1.4 Active Brownian motion or Brownian Motors

Motor is a machine that converts some kind of energy into useful mechanical work. Key features of an engine can be named as follows:

1. Fuel — the kind of energy that is used by engine;
2. Power — which gives information what kind of work can by done by an engine in a given period of time;
3. Efficiency — the measure of how effectively provided energy is converted into mechanical power.

For an average car, engine is a gas driven, 120 kW strong device of an efficiency of about 20%. In this work I focus however on much smaller motors, the ones that operate inside the living cells.

Kinesin, dynein and other motor proteins work in a dense cell environment, "Brownian domain" as it is called by Magnasco [16]. That statement has been the foundation of using Smoluchowski's idea, as an explanatory model for molecular motors. As mentioned in Section 1.3, the original device violates the second law of thermodynamics and it must get something extra to operate.

In biological reality it is hard to achieve thermal gradients large enough to drive directed motion [15]. There are however other ways of providing energy that will result in net movement of a particle.

1.4.1 Fluctuation driven ratchets

The first kind of models are based on the external fluctuations of the ratchet-shaped potentials. This includes cyclicly turning potential on and off ("flashing ratchets", see Fig. 1.4) or applying a fluctuating force which appears as rocking potential ("rocking" or "tilting" ratchets) [18]. This group of models can be jointly described by a Langevin equation in the following form: ε

$$m\ddot{x} + \gamma\dot{x} + \frac{dV(x,t)}{dx} = \eta(t), \tag{1.8}$$

where $V(x,t)$ is a periodic, asymmetric potential that **fluctuates** in time.

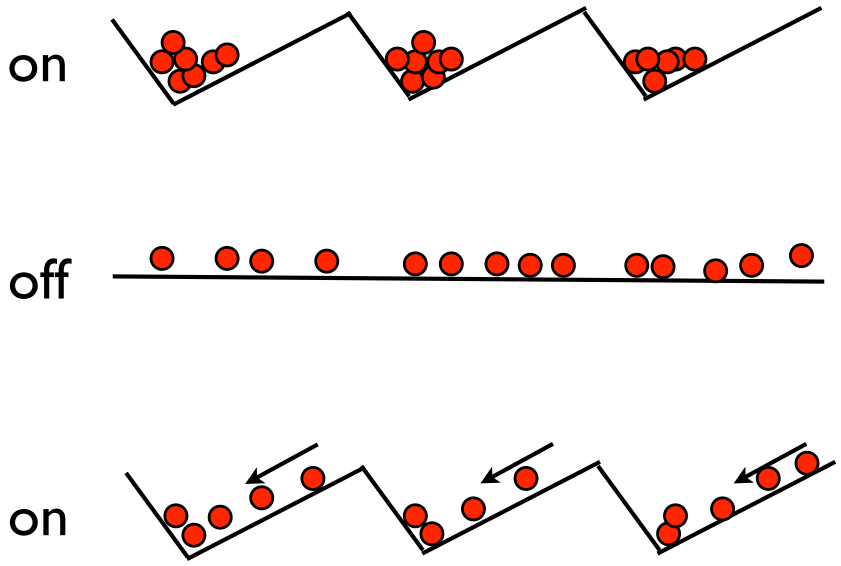


Figure 1.4: Flashing ratchet over the time. Asymmetric potential is cyclically turned on and off allowing net movement in designed direction, without breaking the second law of thermodynamics. Figure adopted from the Heiner Linke website.

1.4.2 From Rayleigh friction function to depot models

Models in a previous Section are not in a sense "active" brownian motors. They are still passive particles, yet experiencing fluctuations of a potential field in which they are embedded. Here, I introduce after Lindner [19] two kinds of models in which particles uptake energy, what is realized by a variable friction coefficient. In this class of models friction may become negative at low body's velocity. Langevin equation under those circumstances reads:

$$m\ddot{x} + \gamma\left(\dot{x}\right)\dot{x} = \eta\left(t\right), \tag{1.9}$$

One should notice a non-constant friction coefficient here. Depending on an approach two different velocity-dependent friction functions are postulated. The first one proposed by Schweitzer *et al.* [20] reads:

$$\gamma\left(\dot{x}\right)_{SET} = \gamma_0\left(1 - \frac{\beta}{1+\dot{x}^2}\right), \tag{1.10}$$

where β is an arbitrary coefficient.

This model implicates negative friction for low velocities within a region $|\dot{x}| < \sqrt{\beta - 1}$ and a "standard" positive value outside of this region. A simpler so called Rayleigh-Helmholtz friction model has been motivated by studies on propagation of sound [21]:

$$\gamma\left(\dot{x}\right)_{RH} = \gamma_0\left(\dot{x}^2 - \alpha\right), \tag{1.11}$$

with α being an arbitrary constant.

Here friction is negative within region of $|\dot{x}| < \sqrt{\alpha}$. Both models (eqs. (1.10)-(1.11)) exhibit similar behavior for low velocities.

In Schweitzer's *et al.* "negative friction" scheme one can include internal energy depot [20], which acquires energy from the environment with a rate $q\left(r\right)$, stores it as an internal energy $e\left(t\right)$ and then provides it for conversion into kinetic energy with a rate $d\left(v\right)$:

$$\frac{d}{dt}e(t) = q(x) - ce(t) - d(\dot{x})e(t) \tag{1.12}$$

After taking into account mechanical energy balance, Langevin equation for the depot-based active Brownian particle reads:

$$m\ddot{x} + \gamma\dot{x} + \nabla V\left(x\right) = d_2 e(t)\dot{x} + \eta\left(t\right), \tag{1.13}$$

where $d_2 = v^2/d(\dot{x})$. Detailed discussion is available in [20].

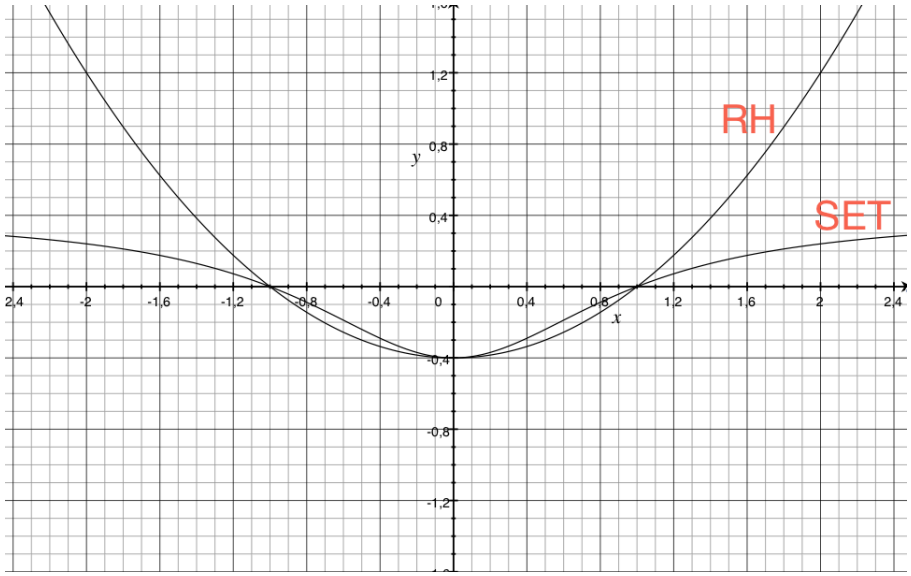


Figure 1.5: Author's comparison of Rayleigh-Helmholtz friction model (RH) with Schweitzer *et al.* model (SET). For values of $|x| < 1$ both models reproduce similar values of friction. Outside of this range, in the Rayleigh-Helmholtz model friction rises much quicker comparing with the Schweitzer model.

Comparing to eq. (1.8), the most evident difference is the new term $d_2 e(t)\dot{x}$, which is responsible for coupling the energy depot with the particle motion and *vice versa*.

In this work, all of the models are based on the depot concept.

1.5 Chaotic behavior

Majority of works on the subject of molecular motors relate to the overdamped case, i.e. skip the inertia term in Langevin equation. This is a perfectly legitmate practice as a single Langevin equation in all those works models behavior of a whole Brownian motor. Having said that, there are models that incorporate more than one equation of motion [22]. In case of motor proteins like kinesin, it is a wise choice to model each protein's head with a distinct Langevin equation. For a reason which will be presented in the following Chapter, it is convenient to move into center—of—mass formalism (CM). As a result, one ends up again with two equations. The one for the CM can be associated with the one and only single equation considered in other models; its friction coefficient and a consequent low Reynolds number is a reason for skipping an inertia term. On the other hand, second equa-

tion is responsible for relative motion of protein's heads. Motor domains, "engine" of a kinesin or dynein "car", do not have to necessary operate in the exact same low Reynolds number. It can be somehow shielded from a crowded cell environment by other, encircling protein's structures. Because of that, to stay on a safe side and to be able to carefully estimate the effect of inertial forces, $m\ddot{x}$ term is kept in the overall analysis.

There are however certain consequences. Including inertia $m\ddot{x}$ term in Langevin equation implicates possible occurrence of chaotic behavior which has been documented in former studies [23–26].

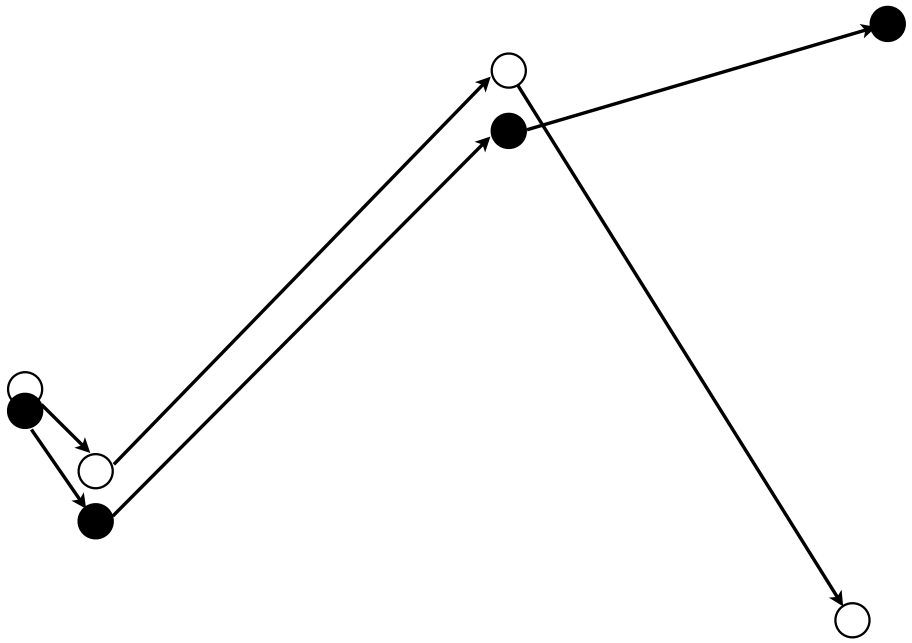


Figure 1.6: Sensitivity to initial conditions - one of a required features to call dynamic system chaotic

In dynamic system theory chaos is explained as a feature of a system where small change in initial conditions results in a significant change of a time trajectory (see Fig. 1.6). The formal definition of a chaotic system [27] includes also topological mixing and that occuring periodic orbits should be dense. There are certain conditions under which model behaves in a chaotic manner and they are defined by the Poincaré–Bendixson theorem, which implication is that chaos cannot occur in 2-dimensional phase space.

The models studied in this work include inertia, along with different variables. As a result, they are all subject of chaotic behavior and in all cases both initial

conditions and other parameters should be carefully adjusted. In the following Chapter, integral part of models presentation will be defining limits of parameters for which they work.

1.6 Outline

In Chapter 2 different models of inertial ratchets will be introduced and commented. They all are based on a concept, that every motor protein should be modeled with at least two Langevin equations to emphasize molecular structure information that are now available. Number of variables and initial conditions result in rich dynamics. Consequently, stability and work regimes of the motion will be studied.

In Chapter 3 performance aspects of models presented in Chapter 2 will be studied and compared with other, both theoretical and experimental works.

Appendix Chapter A will try to compare experimental data with the results obtained from the presented models, while in appendix B the idea of rectifying motor protein movement by allowing backsteps in the context of studied models will be discussed. Finally, in appendix C I present my technical method of solving stochastic differential equations on the graphic cards using the novel OpenCL framework.

Chapter 2

Models of the kinesin motors

2.1 Origin

In 1994, in the spirit similar to work of Magnasco [16], Martin Bier and Dean Astumian proposed a model [28] which explained then-recent experimental data obtained by Svoboda *et al.* [29] for a kinesin motor taking regular steps on a microtubule. The model discussed a Brownian particle in a periodic asymmetric potential with fluctuating barriers. The fluctuating potential $V(x,t)$ was shown to induce net transport even for a net zero additive stochastic force.

Subsequently other models have been proposed [20, 22] and are briefly presented in following sections.

2.1.1 Energy depots models

In the work of Schweitzer *et al.* [20] a following ratchet model has been proposed:

$$\frac{dv(t)}{dt} + \gamma v(t) + U'(x) = F_0 + de(t)v(t) + \sqrt{2D}\xi(t), \tag{2.1}$$

with a energy dissipation term in a form:

$$\frac{de(t)}{dt} = q(t) - ce(t) - dv(t)^2 e(t). \tag{2.2}$$

Here mass $m = 1$, γ stands for friction coefficient, U is a periodic, asymmetric potential — a function of the position x moving with a velocity v; particle is also under influence of constant force F_0 and a stochastic force $\xi(t)$ with the instensity $\sqrt{2D}$. Energy equation 2.2 is coupled with a mechanical Langevin equation 2.2 by $de(t)v(t)$ term, where d denotes energy transmission rate. Energy depot itself is supplied with the energy inflow $q(t)$ and energy from it is dissipated with a rate c and used for mechanical work with a rate d.

2.1.2 Coupled Brownian heads

Another extension of models aimed to discuss transport by motor proteins was the work by Imre Derényi and Tamás Vicsek. The authors proposed a model in which kinesin motor has been composed with two "**elastically coupled** Brownian heads". In the terms of Langevin equation their model takes the form:

$$\begin{aligned} \gamma\dot{x}_1(t) &= -U'(x_1) - F_0 + K\left(x_2 - x_1 - l(t)\right) + \xi_1(t), \\ \gamma\dot{x}_2(t) &= -U'(x_2) - F_0 + K\left(x_1 - x_2 - l(t)\right) + \xi_2(t), \end{aligned} \tag{2.3}$$

where mass has been again assigned to $m = 1$, x_1,x_2 and $\dot{x}_1$,$\dot{x}_2$ denote kinesin heads positions and velocities respectively and K stands for the stiffness of the spring, which has a length $l(t)$=$l_0 + \delta l(t)$[1]. Here the spring obeys classical Hooke's law — the elastic force is directly proportional to the spring extension .

2.2 A dynamic model of the kinesin motor with two heads elastically coupled by a rubber band — Model 1

Taking into account the two aforementioned approaches, we have formulated the new model [30], which incorporated linear dynamics and coupling to the external energy depot. We have assumed that for **every motor head the separate equation of motion** is needed — in that Derényi's work has been followed. On the other hand, we have coupled both heads but in opposite to Derényi and Vicsek, elastomer that binds the heads together has a nonlinear form. This stays in agreement with a recent work of Gräter *et al.* [31].

As discussed in the introductory Chapter, **instead of generally used overdamped approach** in which inertia term is omitted, in a presented model **a more general method has been proposed**. This leads to much more complicated motor dynamics. Careful studies of initial conditions and model parameters are substantial part of this Chapter.

The other key feature of the model, shared with the one proposed by Schweitzer *et al.* [20]) is existence of an energy depot. In this concept chemical energy (e.g ATP) is uptaken form the environment, stored in the energy depot and then released and used for performing mechanical work. This reasoning leads to a third, **additional differential equation for the energy in the depot**.

[1] As one can notice, l(t) is a time variable, not a constant. The authors of the model explain this choice, as a possible alternation of a rest length due to conformational changes

2.2.1 Dimension equations

For motor heads $x_{1,2}$ of a mass m each, we define a set of twin equations of motion:

$$m\frac{dv_1(t)}{dt} + m\gamma_0 v_1(t) + U'(x_1) - F_S(x_1 - x_2) \\ = mde(t)v_1 - M\Gamma_0\frac{dx_0(t)}{dt} + m\sqrt{2D_v}\xi_1(t) \quad (2.4)$$

$$m\frac{dv_2(t)}{dt} + m\gamma_0 v_2(t) + U'(x_2) - F_S(x_2 - x_1) \\ = mde(t)v_2 - M\Gamma_0\frac{dx_0(t)}{dt} + m\sqrt{2D_v}\xi_2(t) \quad (2.5)$$

$$\frac{de(t)}{dt} = q_0 - ce(t) - md(v_1^2 + v_2^2)e(t), \quad (2.6)$$

where $v_i = \frac{dx_i}{dt}$ and $U(x)$ is a ratchet potential (originating from microtubule's periodic structure) given by [32]:

$$U(x)/E_0 = -F_0 x + U_1(x) \\ U_1(x) = h[0.499 - 0.453(sin(2\pi(x + 0.1903)) \\ + \frac{1}{4}(sin(2\pi(x + 0.1903)))]. \quad (2.7)$$

Here F_0 is an external constant force acting on the motor and h controls the barrier height. Further, M stands for the mass of a cargo carried by the molecular motor. Due to the presence of a cargo, the motor experiences an additional friction Γ. γ stands for a friction experienced by motor heads. As presented model is in a Langevin equation form, white Gaussian noise ξ_i of an intensity $m\sqrt{2D_v}$ is present as well and its level is governed by Einstein-Smoluchowski relation $D_v = \frac{\gamma}{m}k_B T$.

The final element of the "head equations" requires some more attention. F_s is an entropic force from an elastomer, which in this case acts like a rubber connecting two motor heads. As already mentioned above, in contrary to Derényi and Vicsek in our model, following recent findings on elastomers behavior [31], force between two heads is modeled by a nonlinear spring:

$$F_S = 2ax - 4bx^3. \quad (2.8)$$

It should be also noted, that there is a good reason to call the force *entropic*. For a fully stretched polymer there is only one possible state (here: molecular conformation) that can be associated with it. In contrary, for a looser stretching, a higher number of conformations is expected. When a polymer is slightly less stretched there are however more options, for a given length there are more than one suitable conformation. It means that this state is less ordered and its entropy is therefore higher.

In the third equation of the model (Eq. 2.6) chemical energy is acquired by the depot with the constant rate q_0, some part is dissipated with a rate c and the other is used for mechanical work with the rate d.

As a simplification, load carried by the molecular motor, is assumed to be constant distance from the two heads center of mass and fixed as dragged (load is always on the opposite side relatively to the direction of movement):

$$x_0(t) = \frac{1}{2}(x_1 + x_2) - sgn(v_1 + v_2)s. \tag{2.9}$$

2.2.2 Dimensionless equations

For the purpose of computer simulations and also for better understanding of existing relations, it is wise to introduce dimensionless equations. By that, one can relates physical variables one to other instead of operating on absolute values. It is especially important when model tries to explain something far from the macroscopic world.

Following Machura *et al.* [33], mass of a Brownian particle (here: kinesin head) is normalized to one. The unit length of the ratchet l_0 is a distance between two neighboring docking locations on the microtubule and E_0 is an energy equal to value of biological activation energy ($E_0 \approx 0.1$ eV $\approx 1.602 \times 10^{-20} J \approx 2.3$ kcal/mol). Those units can be bind together by the characteristic time t_0 as:

$$t_0^2 = m l_0^2 / E_0 \tag{2.10}$$

Given all those units, rescaled equations are of a form:

$$\begin{aligned} &\frac{dv_1(t)}{dt} + \gamma v_1(t) + \hat{U}'(x_1) - \hat{F}_S(x_1 - x_2) \\ &\qquad = \hat{d}e(t)v_1 - \Gamma\frac{dx_0(t)}{dt} + \sqrt{2D}\xi_1(t), \end{aligned} \tag{2.11}$$

$$\begin{aligned} &\frac{dv_2(t)}{dt} + \gamma v_2(t) + \hat{U}'(x_2) - \hat{F}_S(x_2 - x_1) \\ &\qquad = \hat{d}e(t)v_2 - \Gamma\frac{dx_0(t)}{dt} + \sqrt{2D}\xi_2(t), \end{aligned} \tag{2.12}$$

$$\frac{de(t)}{dt} = \hat{q} - \hat{c}e(t) - \hat{d}(v_1^2 + v_2^2)e(t), \tag{2.13}$$

where $\hat{x} = \frac{x}{l_0}$ and $\hat{t} = \frac{t}{t_0}$. Consequently $\gamma = \gamma_0 t_0, \hat{U'} = U'/E_0, \hat{F_S} = F_S/E_0, D = \frac{mt_0}{E_0} D_v, \hat{d} = dt_0, \Gamma = M\Gamma_0 t_0/m, \hat{q} = q_0 t_0$ and $\hat{c} = ct_0$.

To keep notation simple we will omit "hats" in the subsequent equations. The form of Rayleigh oscillator as in [20] is kept in equations (2.11) and (2.12).

2.2.3 Working regime

In presented model there are 5 initial condition variables and 8 more constants. It sums up to 13D space of parameters. It would be very long and unnecessary to test the model for all of them at once. On the other hand, if all but one variable would be fixed, it is possible that the actual working regime would miss some important areas of the parameter space. Here, a hybrid approach to that problem is presented. We start with just mentioned simple evaluation for one variable at a time, even neglecting the stochastic behavior of the process - for each set of parameters only one simulation is made. Having that preliminary results we choose only those variables, which behavior seemed to be nonmonotonous in course of simulation. At that point it is decided which of the parameters should be varied simultaneously and the results of that operation are presented on the 2-dimensional maps of the motor performance. This approach, while compromising, let us to conclude about working regime. In this Section we neglect external force F_0 and focus on finding a parameter region for which motor moves in one defined direction. Further description of the "working regime" of the motor defined in Eqs. (2.11)-(2.13) can be found in Chapter 3.

In the very first plot (see Fig. 2.1 on page 20) we present the model trajectory for a set of parameters that are in the molecular motion regime. It is also a starting point for further parameters evaluation. For a given set of variables, one can observe monotonous motion of the carried load with both heads inexchanging as a leading ones. It stays with an agreement to a hypothesis, that kinesin performs hand-over-hand kind of motion along microtubule [34].

Following a "zero" step, in which example set of parameters have been found, the next task was to perform simple variation of all of the conditions, one at a time. Here, we have divided them into two groups. In Fig. 2.2 one can find set of plots dedicated to **initial conditions**. On the other hand, in Figs. 2.3 and 2.4 there are plots for the model parameters.

While all of the plots found in Fig. 2.2 display some kind of dependency of initial conditions, the weight of this dependency varies. In the case of the initial depot energy e, mean velocity changes not more that just a few percent. Comparing

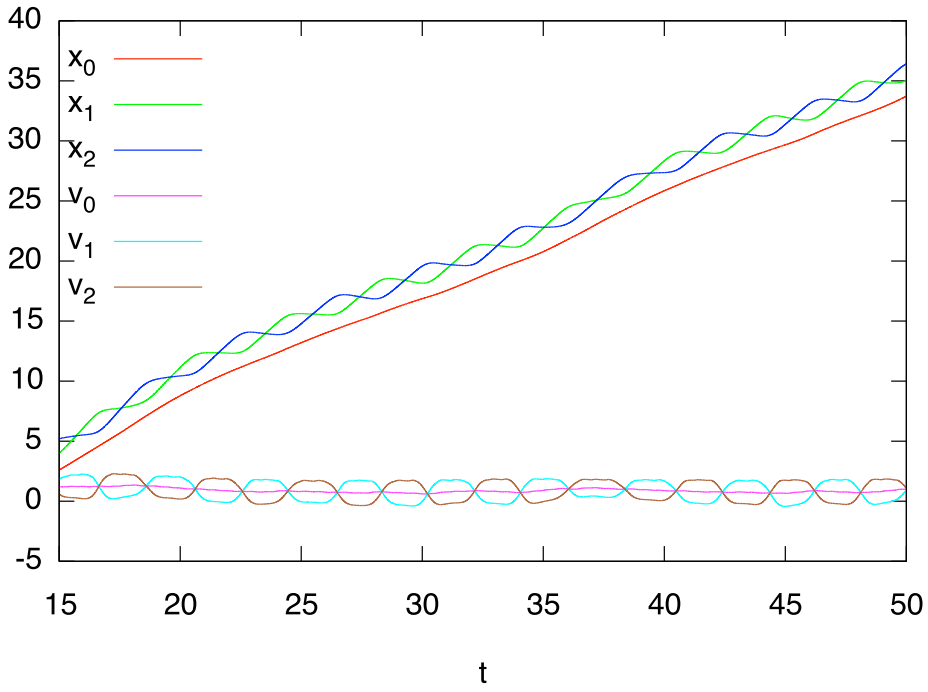


Figure 2.1: Result of a simulation for the parameter set: $s_0 = 8$, $a = 0.5$, $b = 1$, $e = 0.1$, $h = 0.1$, $\gamma_0 = 0.02$, $q = 1.0$, $c = 0.1$,$d = 1$,$\Gamma = 0.5$, $D = 0.25$. We show the positions of the two heads x_1 (green), x_2 (blue), the velocities of the heads $v_1(t)$ (light blue), $v_2(t)$ (brown). The position of the load x_0 is marked in red and its corresponding velocity v_0 is displayed in magenta. The time step of simulations, $\Delta t = 10^{-3}$.

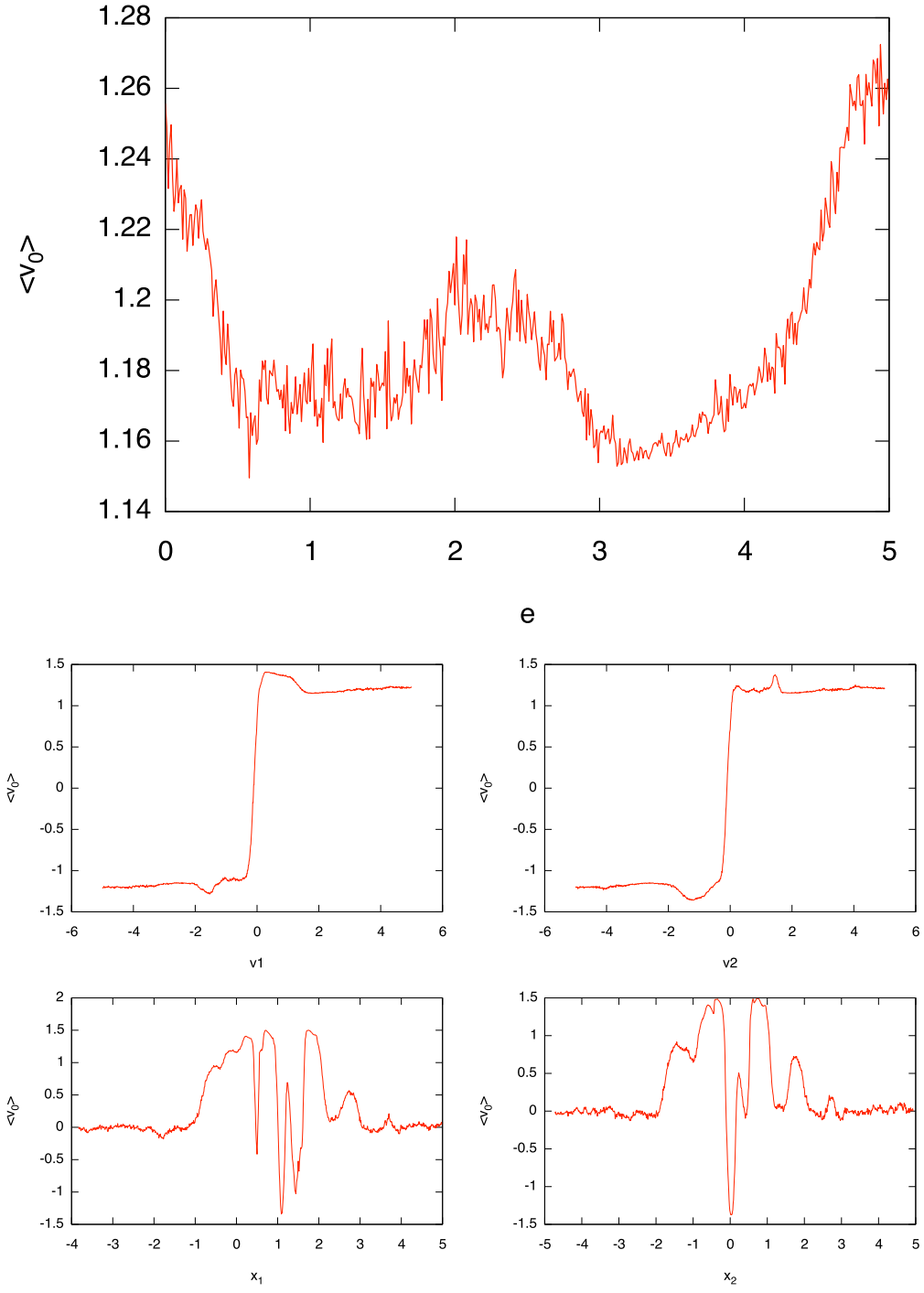


Figure 2.2: Time-averaged velocity $\langle v_0 \rangle$ *versus* variation of various *initial* conditions. Result of a simulation for the parameter set: $s_0 = 8, a = 0.5, b = 1, e = 0.1, h = 0.1, \gamma_0 = 0.02, q = 1.0, c = 0.1, d = 1, \Gamma = 0.5, D = 0.25$. The time step of simulations is $\Delta t = 10^{-2}$.

with other parameter changes, it makes e variations insignificant.

Contrary to that, both initial velocity and position seem to be very important parameters. In general, velocity sign (direction of movement) at the beginning of motion determines very strongly the mean velocity. Interestingly, the actual value of initial velocity does not have any significant influence on the mean velocity.

The most interesting behavior can be observed when probing initial position of the kinesin heads. On average, a sign of head's position does not influence the sign of the mean velocity. There are some plot spikes (one or two, depending on whether one considers x_1 or x_2), where some values of initial position drives mean velocity to the minus sign region.

Those observations led us to study in more detail initial values of x_1 vs x_2 and x_1 vs v_1. Results of those calculations are presented in the end of this section.

Other 9 plots presented in Fig. 2.3 and 2.4, are dedicated to the analysis of model parameters. It should be noted that in further study initial conditions have been chosen in a way that motor would operate in the working regime.

The parameters a and b of the elastomer expansion function play the most significant part in the Langevin equation, as the analysis in appendix A reveals. This term determines how entropic force exerts on kinesin, based on a relative head position. In turn, variations of a parameter a behavior, as depicted in Fig. 2.3 is complex and even slight change of this parameter can cause changing regime from productive, to-the-left direction to nonproductive, to-the-right movement. For a small $a < 0.5$ average velocity has almost constant value off 1. For $0.5 < a < 2.5$ a motor ceases to operate in either way. In a domain of a variability $0.5 < a < 2.5$ two maxima of the velocity can be detected ($a = 1.2$ and $a = 2$) with an additional minimal value of the velocity observed at $a = 1.4$.

With b situation is much more simple. Under 0.5, motor is in the negative regime, with a minimum at $b = 0.2$. By crossing the zero, the mean velocity is quickly rising, reaching the maximum value at $b = 1$ and slowly decaying afterwards.

Moreover, the lower energy dissipation rate c is, the higher the overall mean velocity of the motor.

For every but very little value of d, the motor proceeds in the right direction. It reaches a maximum of $\langle v_0 \rangle$ for $d = 0.3$, after which the average velocity decreases slowly.

Maximal value of mean velocity can be achieved for a tiny noise intensity D. However, nonzero, mean velocity of directed motion can be registered in the absence of noise. It should be anyhow noted that a weak noise is leading to better motor performance than the total absence of it.

Both frictions γ_0 and additional Γ should be kept as low as possible to sustain motion possibility. It applies especially to Γ, for which $\langle v_0 \rangle$ experiences exponential-like decay, cf. Fig. 2.3 and 2.3. However, while high frictional constants can slow

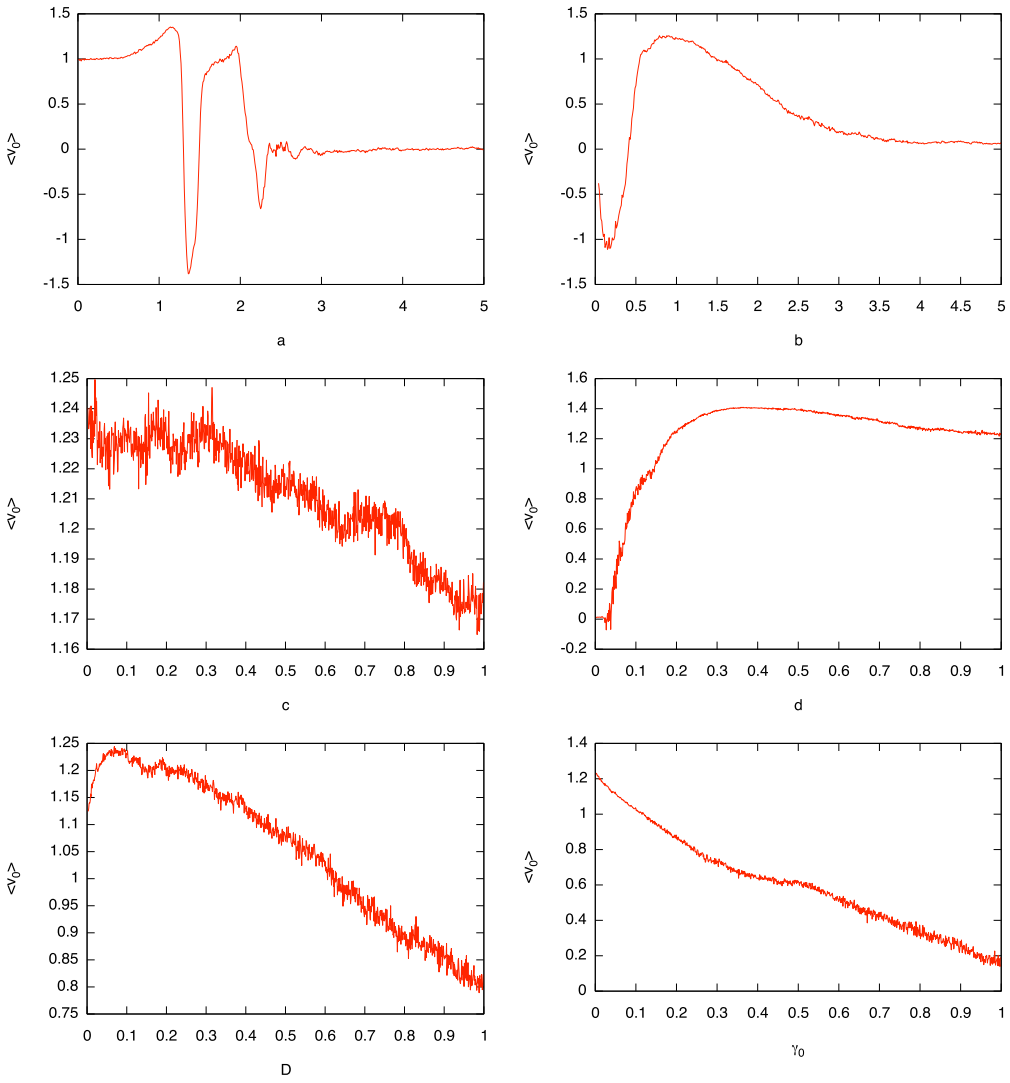


Figure 2.3: Time-averaged velocity $\langle v_0 \rangle$ *versus* variation of *model* parameters. Results for simulations for the parameter set: $s_0 = 8$, $a = 0.5$, $b = 1$, $e = 0.1$, $h = 0.1$, $\gamma_0 = 0.02$, $q = 1.0$, $c = 0.1$,$d = 1$,$\Gamma = 0.5$, $D = 0.25$. The time step of simulations is $\Delta t = 10^{-2}$.

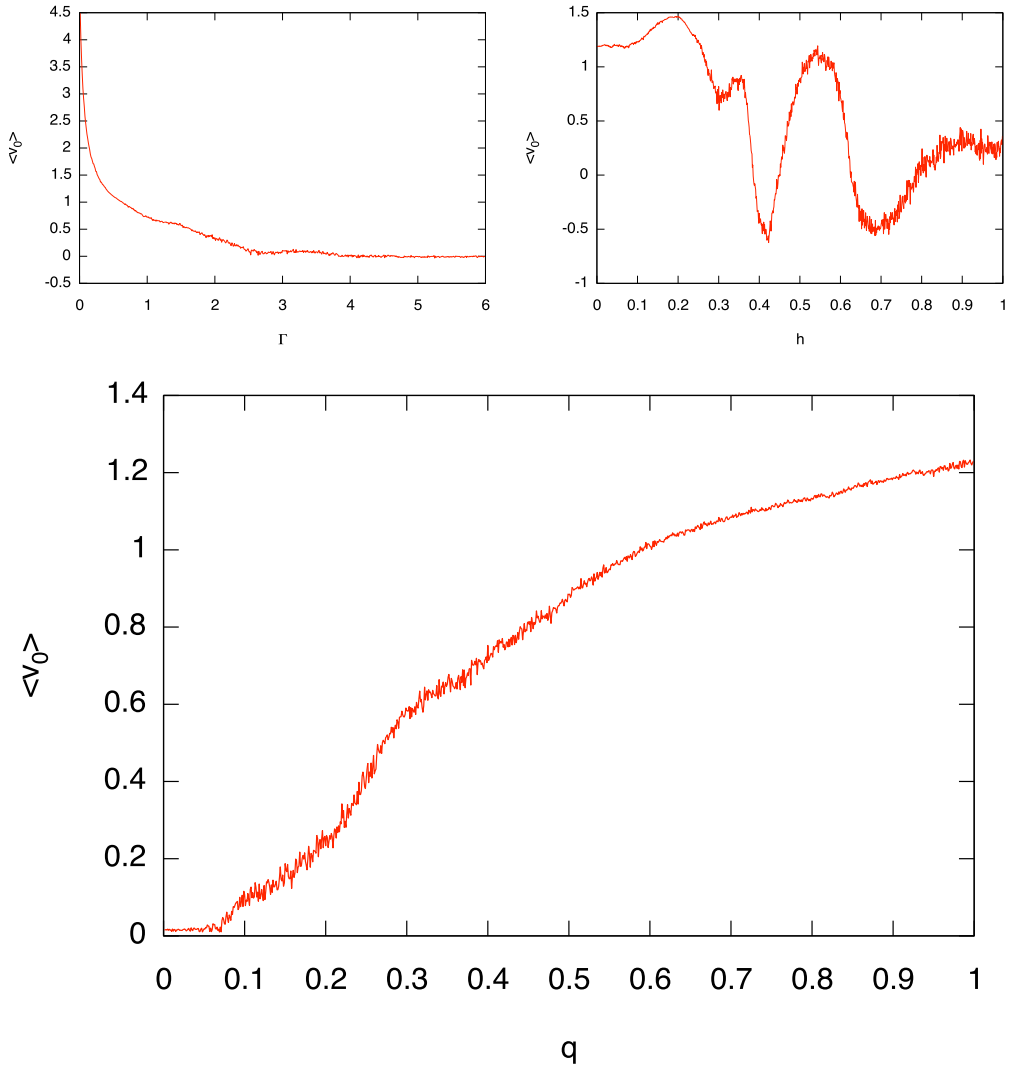


Figure 2.4: Time-averaged velocity $\langle v_0 \rangle$ *versus* variation of *model* parameters. Results of simulations for the parameter set: $s_0 = 8, a = 0.5, b = 1, e = 0.1, h = 0.1, \gamma_0 = 0.02, q = 1.0, c = 0.1, d = 1, \Gamma = 0.5, D = 0.25$. The time step of simulations is $\Delta t = 10^{-2}$.

Table 2.1: Model 1 Working Regime Summary

parameter	description	working regime	negative motion
$\mathbf{x_1(0)}$	**head 1 initial position**	$4 > x_1 > -1$	fluct. in-between
$\mathbf{x_2(0)}$	**head 2 initial position**	$3 > x_2 > -2$	fluct. in-between
$\mathbf{v_1(0)}$	**head 1 initial velocity**	$v_1 > 0$	$v_1 < 0$
$\mathbf{v_2(0)}$	**head 2 initial velocity**	$v_2 > 0$	$v_2 < 0$
$e(0)$	depot initial energy	—	—
a	**elastomer parameter**	$a < 3$	two negative regions
b	**elastomer parameter**	$3 > b > 0.5$	negative for $b < 0.5$
c	energy dissipation rate	—	—
d	energy transmission rate	$d > 0.03$	—
D	**noise intensity**	—	—
γ	friction frequency	—	constant drop
Γ	additional load friction	$\Gamma < 3$	—
h	**ratchet height**	$h < 0.8$	two negative regions
q	energy inflow rate	$q > 0.08$	—

down or even stop the motor, they would not reverse the motion. That conclusion is in agreement with a physical intuition.

The barrier height h influences on mean velocity, appears to be one of the most complex ones. Apart from the elastomer coefficients a and b, this is the only constant parameter that can reverse the direction of motion. For a given set of other parameters, motor operates for low $h < 0.37$ after which it alternately goes to the left or to the right.

The higher energy inflow rate q is, the faster motor operates. There exists however a minimal value of q, under which kinesin would not move — by inspection of Fig. 2.4 we conclude that q should be grater than 0.08 for motor to operate.

As in this Section we study actual working regime, summarized version of findings reported in the previous paragraphs is presented in table 2.1.

2.2.4 Further analysis of the working regime

In the table 2.1 the parameters that behave in a unpredictable manner have been bolded and will be studied in this section.

In figure 2.6 we present a vs b plot. For a given range of parameters ($0 < a, b < 2$) "safe" region can be established for points above linear function $b(a) = 2a - 1$.

Plot of $x_1(0)$ vs $v_1(0)$ is present in Fig. 2.7 . While plot in Fig. 2.2 show complicated behavior for different initial positions, observation of this plot reveal that the

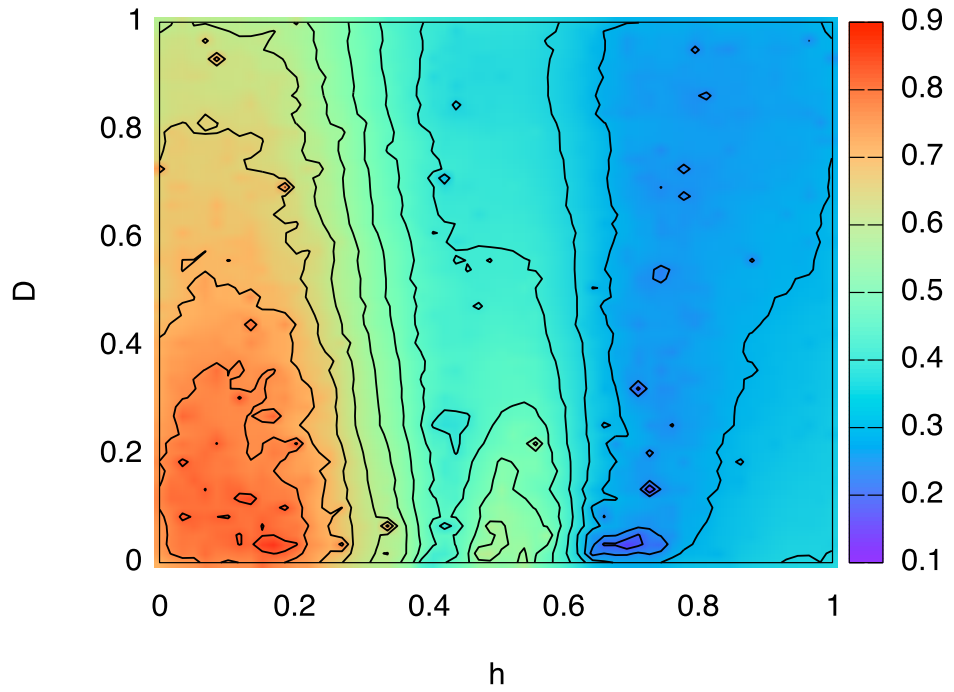


Figure 2.5: Time-averaged velocity $\langle v_0 \rangle$ *versus* variation of h and D. Result of a simulation for the parameter set: $s_0 = 8$, $a = 0.5$, $b = 1$, $e = 0.1$, $h = 0.1$, $\gamma_0 = 0.02$, $q = 1.0$, $c = 0.1$,$d = 1$,$\Gamma = 0.5$, $D = 0.25$. The time step of simulations is $\Delta t = 10^{-2}$.

initial velocity is really important. Variation of $x_1(t = 0)$ has some slight impact on the resulting mean velocity but is absolutely insignificant in comparison to the initial velocity. This suggests that system without a load is in a state of metastable equilibrium and initial direction of one of the heads sets the direction for the whole system.

In Fig. 2.5 we present h vs D plot. Somehow similarly like in the case of x and v there is a dominating actor here - ratchet height h. For small values, as presented before, mean velocity reaches its maximum and decreases afterwards. Noise level only influences pace given by the ratchet potential.

2.2.5 Criticism

As mentioned at the beginning, in contrary to other works, inertia term has been kept in our original Langevin equations. While the internal motion of the motor could be underdamped (that assertion lead to constructing model **with** inertia), as

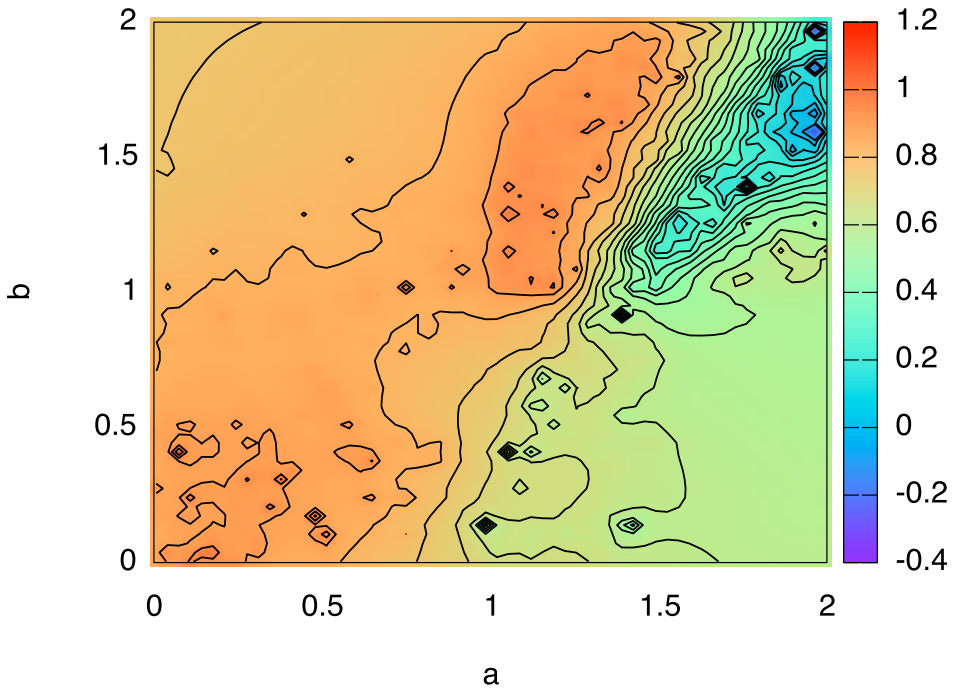


Figure 2.6: Time-averaged velocity $\langle v_0 \rangle$ ***versus*** variation of a and b. Result of the simulation for the parameter set: $s_0 = 8$, $a = 0.5$, $b = 1$, $e = 0.1$, $h = 0.1$, $\gamma_0 = 0.02$, $q = 1.0$, $c = 0.1$,$d = 1$,$\Gamma = 0.5$, $D = 0.25$. The time step of simulations is $\Delta t = 10^{-2}$. Working regime can be observed for parameters above linear function $b(a) = 2a - 1$.

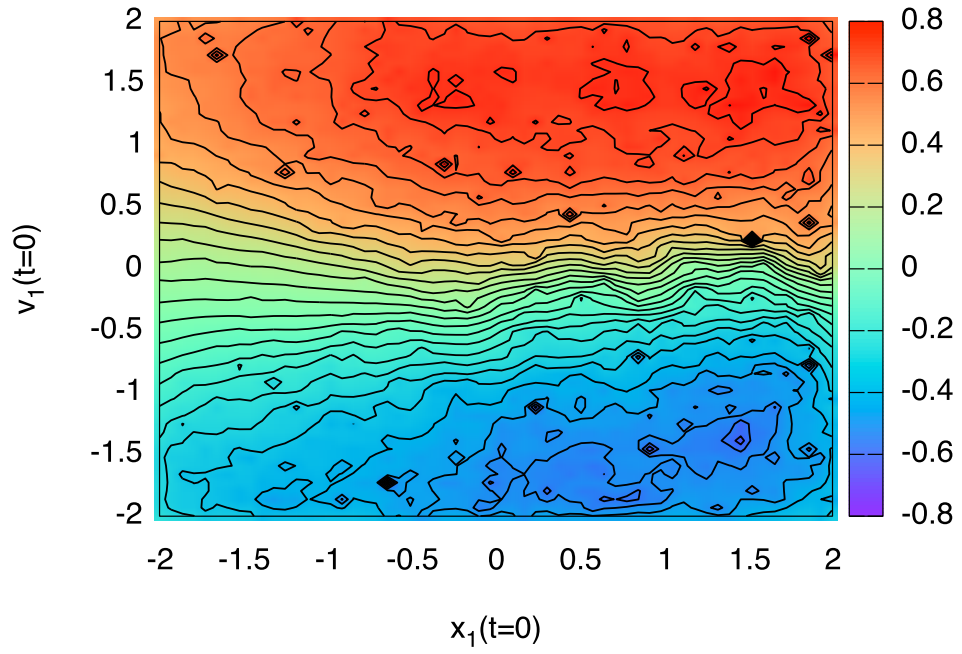


Figure 2.7: Time-averaged velocity $\langle v_0 \rangle$ ***versus*** variation of x_1 and v_1. Result of the simulations for the parameter set: $s_0 = 8$, $a = 0.5$, $b = 1$, $e = 0.1$, $h = 0.1$, $\gamma_0 = 0.02$, $q = 1.0$, $c = 0.1$,$d = 1$,$\Gamma = 0.5$, $D = 0.25$. The time step of simulations is $\Delta t = 10^{-2}$.

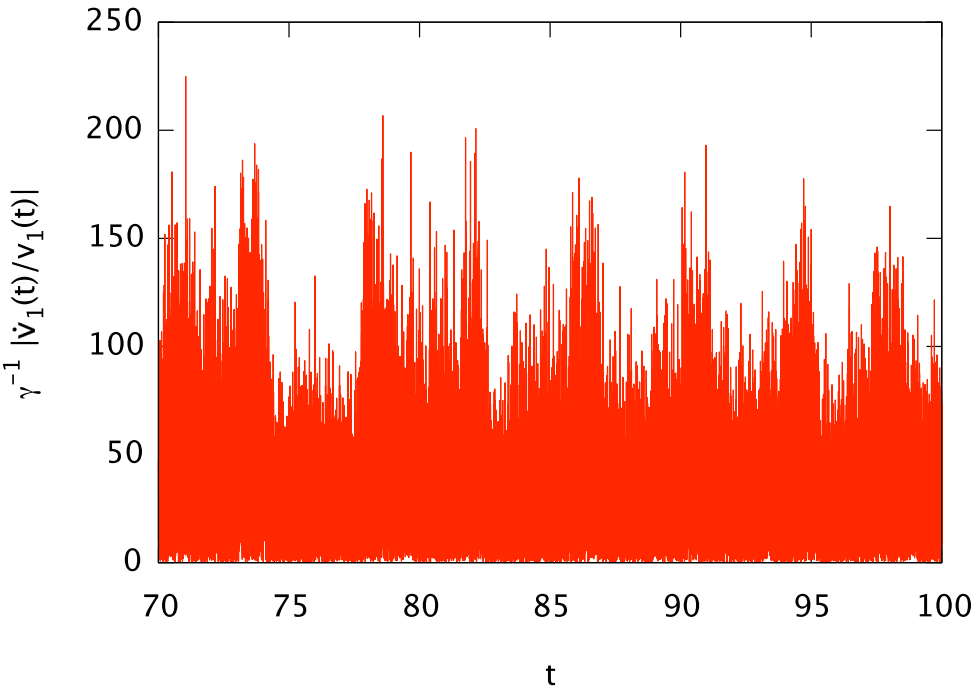


Figure 2.8: The effect of inertia in the model. The inertia term $\dot{v}(t)$ is much larger than the term $\gamma v(t)$ for almost all times.

a whole the molecular motor operates in the crowded environment and should be extensively damped. To analyze this constraint we have made a simulation in which ratio of inertial term dv/dt to the friction term γv has been checked against the time. What we have found out and what is illustrated in Fig. 2.8, is that this ratio is very much above the level that could be called "small". It means that in this context, the model - while possessing most of the properties needed to study molecular motor motility - has a serious drawback. Namely, because of the expected overdamped behavior, low Reynolds number typical for motion in the cell's environment[35], this kind of movement would be unlikely to be realized.

2.3 Separation of mass model — Model 2

2.3.1 From the first to the second model

Drawbacks unveiled in the last part of previous Section have prompted us to introduce some modifications of the presented model. Altogether those changes re-

sulted in a new effective model of molecular motor (**Model 2**).

The founding assumptions remain the same, so are the first equations (2.4 - 2.6). The trick introduced here separates relative kinesin heads movement from the general motion of the molecular motor. Along other things, it gives possibility to use different constants to distinguish two environments.

To get a relative motion formula, one subtracts equations (2.4, 2.5):

$$\frac{mdv(t)}{dt} + m\gamma_c v(t) + (U_1'(x_c + x(t)/2) - U_1'(x_c - x(t)/2) \\ -2TS'(x(t)) = mde(t)v(t) + m\sqrt{2D_v}\xi(t), \tag{2.14}$$

where $x = x_1 - x_2$ is a distance between two heads at the given time and $v = v_1 - v_2$ is a relative velocity. The energy equation after the change of variables gets the form:

$$\frac{de(t)}{dt} = q - ce(t) - mdv^2 e(t). \tag{2.15}$$

As mentioned in previous sections, the author's intention, was to keep inertial terms for the internal motion of the motor (that is for the relative motion of two motor heads) and this has been achieved in equation (2.14). On the other hand, to meet the requirement of the overall motor motion being overdamped in viscous environment, we have analyzed the center of mass motion. Here, by adding equations (2.4, 2.5), we get a formula for CM motion:

$$(\mathbf{2m})\frac{\mathbf{dv_c(t)}}{\mathbf{dt}} + (\mathbf{2m})\gamma_\mathbf{c}\mathbf{v_c(t)} + 2M\Gamma_c\frac{dx_c(t)}{dt} \\ +(U'(x_c + x(t)/2) + U'(x_c - x(t)/2)) \\ = (\mathbf{2m})\mathbf{de(t)v_c(t)} + (\mathbf{2m})\sqrt{\mathbf{2D_v}}(\xi_\mathbf{1}(\mathbf{t}) + \xi_\mathbf{2}(\mathbf{t}))/\mathbf{2}, \tag{2.16}$$

where $x_c = (x_1 + x_2)/2$ stands for the center of mass position and $v_c = (v_1 + v_2)/2$ represents center of mass velocity. Index c in this model stands for the center of mass variables, which is different from index 0 in the previous model, representing point behind moving two-head motor, as in eq. (2.9). Note that in this case, by means of Eq. (2.9), $v_c = v_0$.

With both equations in hand, there are some simplifications that can be introduced just into one of the formulas. The center of mass equation can be compared with models, which do not take into account separate heads motility. It is now possible to neglect inertia term just in the center of mass equation, effectively getting overdamped formula while still retaining inertia term in the relative head motion.

At this point we assume that leading contribution to this motion is coming from the terms that have not been **bolded** in eq. (2.16).

In order to retain the Langevin form, this striped-down equation, needs a replacement for the lost noise term. Therefore, we introduce here a center of mass friction $\Gamma = 2M\Gamma_c$ and corresponding noise intensity D_{x_c}. With approximating the noise level, it must be noted that addition of two separate noise function is not a sum of them. In case of the Gaussian distribution, as discussed here, the square of its variance is the sum of the squares of variances ($\sigma_{1+2}^2 = \sigma_1^2 + \sigma_2^2$). Further, we will write ratchet potential as in eq. (2.7) in a form in which bias force F_c is explicitly present in the center of mass equation:

$$\frac{dx_c(t)}{dt} = \frac{F_c}{\Gamma} - \frac{1}{\Gamma}[(U_1'(x_c + x(t)/2) + U_1'(x_c - x(t)/2)] + \sqrt{2D_{x_c}}\xi_0. \tag{2.17}$$

This model still manages to operate and can perform work against the external force. Figure 2.9 illustrates the process for different external forces. It can be observed that for forces not exceeding absolute value $F_0 = 0.2$, the motor operates in its working regime.

Without going into the details (Chapter 3 is devoted to the broad aspect of efficiency), it should be stated that in terms of the parameter relating the work performed by the motor against the bias force with the power supply — the **thermodynamical efficiency** $\eta = \frac{|F_c v_c|}{q}$ — the second model is less effective than the first one.

2.3.2 Initial parameters test for the model

Like in the case of previous model (see Sec. 2.2.3), space of parameters for the second model is also abounding.

As the first step, we have chosen initial conditions ($x_0(0)$, $v_0(0)$, $e(0)$ and parameters (d, Γ, F_0) to test their possible impact on motor functioning. Secondly, we have defined which parameters to keep fixed. Relative motion friction frequency $\gamma_0 = 0.02$, noise level for the center of mass equation D_{x_0} is 1.0 while for the relative motion D_v is 0.1. In general, kinesin is a subject to the fluctuation that come from the crowded surroundings. On the other hand, "shielded" relative motion of kinesin heads is less exposed to those fluctuations. The other model parameters values come from the findings for the first model: $q = 1$ and $c = 0.1$ while we have also kept elastomer parameters $a = 0.5$ and $b = 1.0$. Simulation were carried out with head masses $m = 1$, time step $dt = 10^{-3}$. The total final time of simulations has been set to $t_{fin} = 50.0$.

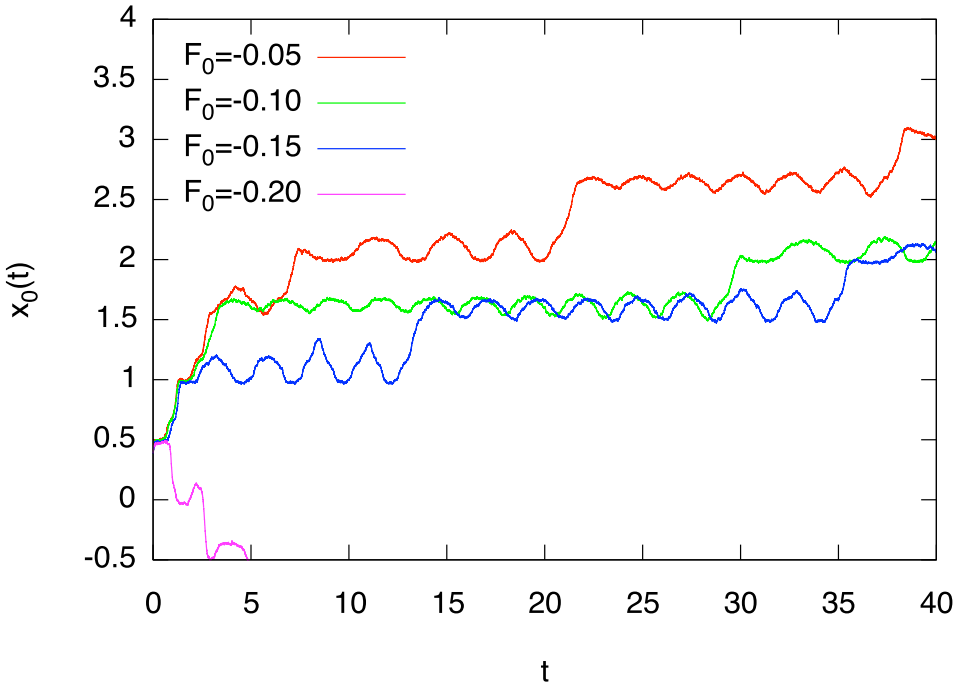


Figure 2.9: Trajectories of the center of mass (above) for various values of the load force: $F_c = -0.05, -0.10, -0.15, -0.20$. At $F_c = -0.20$ the motor is already overloaded and ceases to operate at higher loads. Parameters for the simulation are $a = 0.5$, $b = 1$, $\gamma = 0.02$, $\Gamma = 0.2$, $q = 1.0$, $h = 0.1$, $c = 0.1$, $d = 0.1$.

Here, we propose an alternative method to explore the working regime. The approach is as follows:

1. draw values of 6 parameters in following ranges: $x_0(0) = (0, 6)$, $v_0(0) = (0, 6)$, $e(0) = (0, 1)$ and $d = (0, 1)$, $\Gamma = (0.1, 1)$, $F_0 = (-0.3, 0)$
2. make simulation for $t_{fin} = 50.0$, calculate mean velocity
3. repeat 10^5 times.

The reason that we have chosen this approach originates from the idea of finding really the "best" region of initial parameters. The previous approach that we have used in the study of the first model, have not took into account that other parameters can play the role at the same time, e.g there might be certain value of $x_0(0)$ for which $\Gamma = 0.2$, $F_0 = -0.1$ is the most efficient and for different value of $x_0(0)$ optimal value of Γ and F_0 can be different.

There is also another reason, for which we are not using ensemble-averaged velocity. For systems exhibiting multimodal distributions of velocities, ensamble average would hide characteristic features of motion (i.e. switching of the direction of direction of motion from left to right).

Because we draw independently all 6 parameters and then plot it as 3D plots of mean velocity as a function of 3 pairs of 2 parameters, one can see general trends of how single parameter influence on the overall efficiency. For the sake of clarity, only values of positive mean velocity were taken into account. Further in Figs. 2.10, 2.11, 2.12 we only keep points where the efficiency $\eta_C = \frac{|F_0\langle v_0\rangle|}{q}$ is higher than 1%[2].

2.3.3 Mean velocity distribution as the function of Γ.

Contrary to the statements about calculation ensemble-averaged velocity drawbacks, here we have calculated mean velocity distribution as the function of Γ parameter. The reason for that, is that we wanted to get clear view of the Γ value dependency, even at the cost of potential loss of subtle phenomena occurring in the model. Results are presented in Fig. 2.13. As we have chosen safe value of force $F_0 = -0.1$, for all interesting cases ($\Gamma >> \gamma_0$, that is for $\Gamma > 0.1$), mean velocity is positive and decreases slowly with the increase of Γ.

2.3.4 Mean velocity distribution as a function of the ratchet amplitude h

The last thing that has been checked for the second model is that we have changed the core ratchet parameter h — potential height. Using the same approach as in

[2] "thermodynamical efficiency", detailed discussion about estimating molecular motor performance can be found in the following Chapter

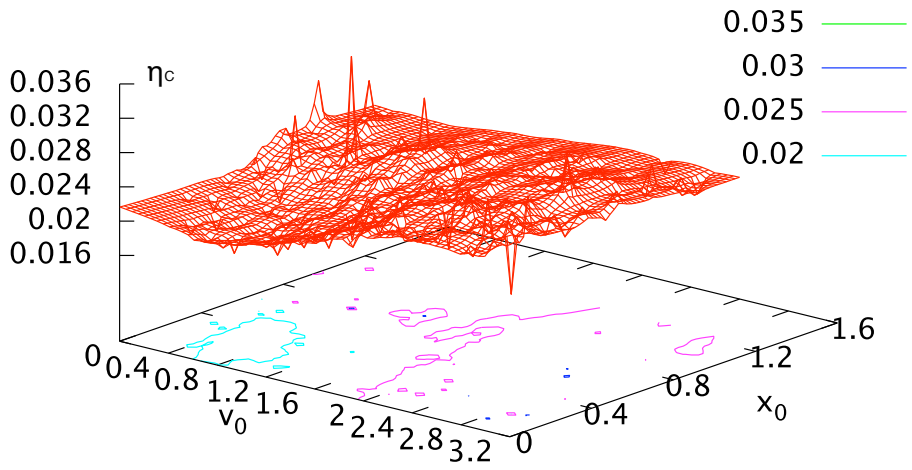


Figure 2.10: Efficiency $\eta_C = \frac{|F_0\langle v_0\rangle|}{q}$ as a function of initial velocity and initial relative position of heads. Note that the plot exhibits flat structure after reaching certain values of $v_0(0) > 2$ and $x_0(0) > 0.8$.

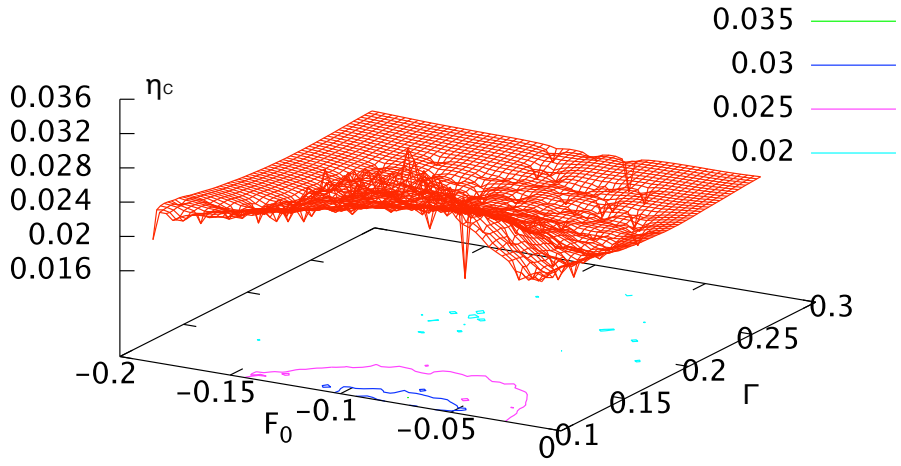


Figure 2.11: Efficiency $\eta_C = \frac{|F_0\langle v_0\rangle|}{q}$ as a function of force F_0 and Γ. There exists a range of F_0 which "provides" high efficiency. The lower Γ, the higher efficiency.

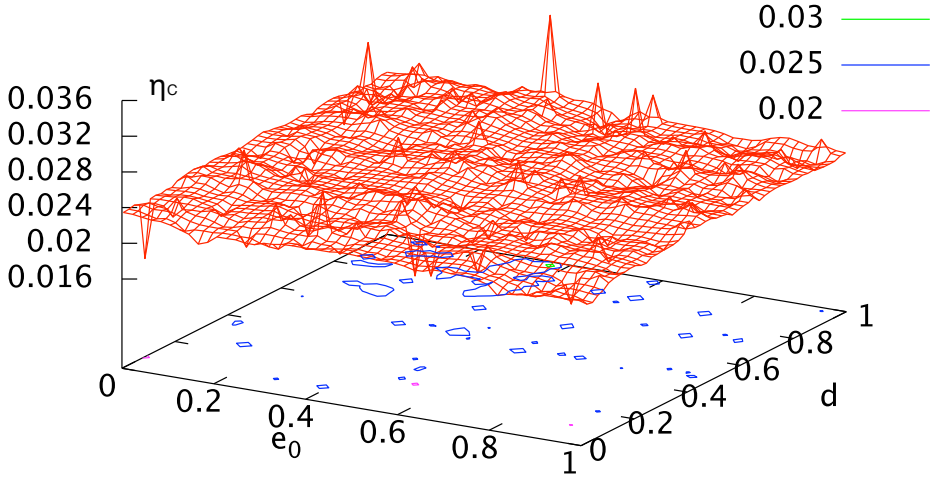


Figure 2.12: Efficiency $\eta_C = \frac{|F_0\langle v_0\rangle|}{q}$ as a function of parameter d and initial value of $e(0)$. Initial value of $e(0)$ seems not to be important, while one can see that for $d > 0.4$ there is much more often occurrence of efficiency over the cut-off point.

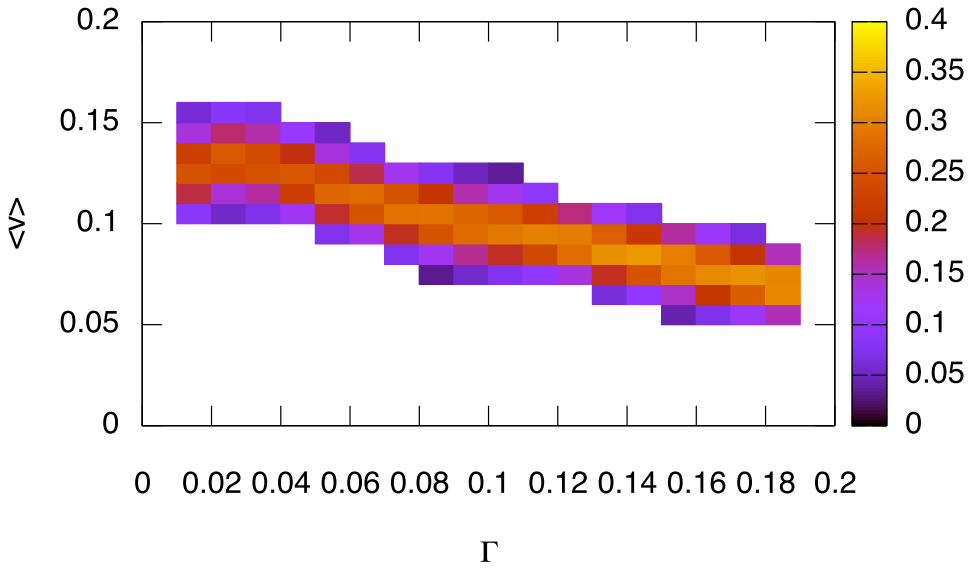


Figure 2.13: Time- and ensemble-averaged velocity distribution after $t = 50$ as the function of Γ. Results obtained for $F_0 = -0.10$ and 100 simulation steps for each point.

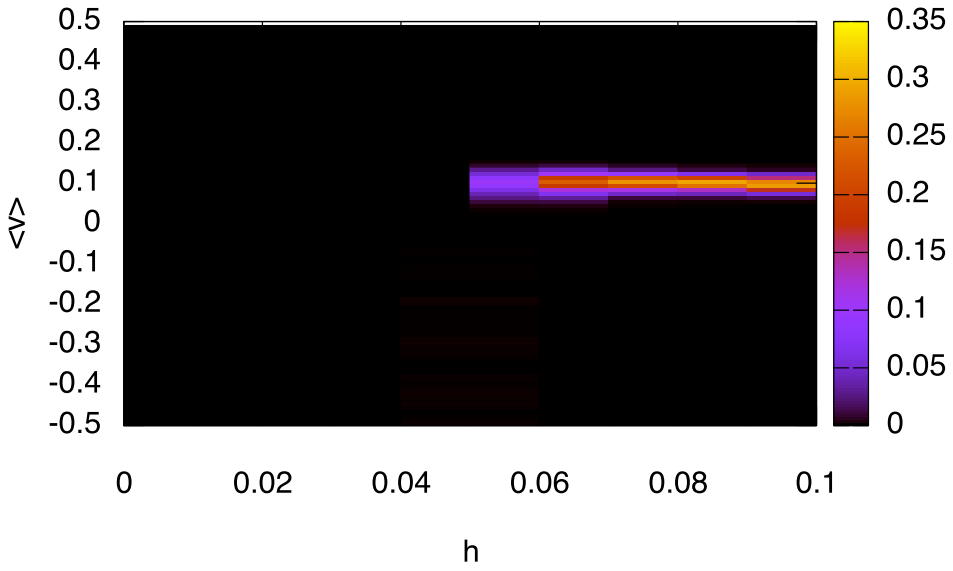


Figure 2.14: Time- and ensamble-averaged velocity distribution after $t = 50$ as the function of h - ratchet amplitude. Results obtained for $\Gamma = 0.101790$, $F_0 = -0.10$, number of simulation steps for each point is 100.

previous section, plot that can be found in Fig. 2.14 has been made.

Nevertheless, what can be observed, is that a change of h has no impact on the mean velocity of the motor. The presence of the ratchet effect itself is limited for values grater than $h > 0.05$. Only in this regime the positive, upward movement appears.

For values below critical value one can find fast negative velocity, far from absolute values characteristic for the model ($|\langle v \rangle| < 0.5$).

Chapter 3

Performance of models

3.1 The concept of the efficiency: Addressing performance of the motor models

In the formal terms, thermodynamic energy conversion efficiency is the *"rate of free energy output divided by the rate of chemical free-energy consumption by the motor"* [36]. In less formal language, efficiency gives the information about how much work done one can get for a given amount of resources. An information about an average MPG (miles per gallon) is used by consumers for some kind of estimation about the efficiency of the car. Of course, to get a proper efficiency, chemical energy released in combustion of fuel should by calculated - as well as a mechanical work performed by the engine. Especially the latter is not an easy task, due to environmental conditions (terrain, weather, driver skills etc). At the end of the day, a driver asks a question - how much fuel I have used to carry my family for a distance of those 1000 miles. He or she does not think about all other factors that a physicist should take into account for calculating a proper, thermodynamical efficiency.

The previous paragraph in some way explained the need for understanding that "efficiency" might be a complex issue to study. In a way, it is a question about how far the term "efficiency" stands from the term "usefulness". Even with a strict physical definition, in the back of the head one thinks about the benefit behind the whole thing.

This Chapter is dedicated to exploration of the efficiency (in a multiple meaning) of both molecular motor models presented in the previous Chapter. The opening sections explain different ways of defining efficiency for molecular motors and the closing ones present those efficiency values calculated using computer simulations.

3.1.1 Efficiency in classical thermodynamics

In this simple approach work done by a motor, working against force F_0 is divided by the chemical energy provided from the ATP hydrolysis:

$$\eta_C \equiv \frac{f\langle v\rangle}{A\langle r\rangle}, \tag{3.1}$$

where f is an external force, A is the chemical free energy consumed by the time of one motor cycle, $\langle r\rangle$ stands for rate of the chemical reaction cycle and $\langle v\rangle$ is an average motor velocity.

According to the definition for both presented models (Model 1 & 2) this can be rewritten as:

$$\eta_C \equiv \frac{|F_0\langle v_0\rangle|}{q}, \tag{3.2}$$

where $q = A(r)$

The problem is, that in an absence of external force, the efficiency is by definition zero. This classical equation evaluates more what is rather considered as a motor usefulness in an imaginative tug-of-war with some other motor. In the case of dense environment in which molecular motors operate, it does not mean taking into account a friction force coming from thermal fluctuations [37].

3.1.2 Generalized efficiency

In the paper of Derényi *et al.* [37], authors introduce the concept of *generalized efficiency*. It is defined as a ratio of minimal energy needed for the task to be accomplished (E_{min}) and the actual energy used to accomplish certain task (E_{in}):

$$\eta_G \equiv \frac{E_{min}}{E_{in}}. \tag{3.3}$$

For molecular motors the minimal energy is used when molecular motor is moving uniformly with an average velocity $\langle v\rangle$ along the track (i.e. microtubule) and in this case:

$$P_{min} = \frac{dE_{min}}{dt} = F_0\langle v\rangle + \gamma\langle v\rangle^2. \tag{3.4}$$

The problem of a generalized efficiency in the context of presented model is that, it is hard to evaluate minimal energy used in a hand-over-hand mechanism.

Intuitively, the minimum is achieved when motor heads are not moving when observed in the center of mass point of view. In consequence, generalized efficiency for both presented models can be defined as:

$$\eta_G \equiv \frac{|F_0\langle v_0\rangle| + \Gamma M\langle v_0\rangle^2}{q}. \tag{3.5}$$

3.1.3 Stokes efficiency

In the paper of Wang *et al.* [36] Stokes efficiency is defined as:

$$\eta_S \equiv \frac{\zeta\langle v\rangle^2}{A\langle r\rangle + f\langle v\rangle}, \tag{3.6}$$

where f is an external force, ζ is a drag coefficient, A is the chemical free energy consumed by the time of one motor cycle, $\langle r\rangle$ stands for rate of the chemical reaction cycle and $\langle v\rangle$ is an average motor velocity. As the Authors of the concept summarize [36]:

> *"The Stokes efficiency can be viewed as a measure of how efficiently the motor can utilize the free energy to drive a load through a viscous medium."*

In terms of models presented in the previous Chapter, Stokes efficiency can be defined as:

$$\eta_{S,1} \equiv \frac{m\gamma_0(\langle v_1\rangle^2 + \langle v_2\rangle^2) + M\Gamma\langle v_0\rangle^2}{q + F_0\langle v_0\rangle} \tag{3.7}$$

for the first model and:

$$\eta_{S,2} \equiv \frac{\Gamma\langle v_c\rangle^2}{q + F_c\langle v_c\rangle} \tag{3.8}$$

for the second model.

3.2 Comparison of efficiency

3.2.1 Velocity distribution

Looking at all the definitions of the efficiency presented in the previous Section it is possible to separate important variables playing role in the efficiency calculations. In fact, there is only one variable — mean velocity $\langle v_c\rangle$, the other parameters in the

context of studied models are the set simulation parameters - as energy input rate q, friction Γ or γ and, finally, opposing force F_0.

Because of randomness of the noise part, every simulation of motor trajectory is different. For certain critical values of parameters it may produce false impression of wether one is upward or downward regime. To study this further, we have made series of simulations and their outcome - mean motor velocity distributions for different external forces is presented in Fig. 3.1 and 3.2 for the first and second model, respectively.

For the first model for all the presented opposing forces, distribution has asymmetric and bimodal character. In all the cases there is one dominant value of the mean velocity, the most probable one. Especially for the force value $F_0 = -0.04$ bimodality has crucial implications. For the same set of parameters, depending on realization (e.g. noise) motor can operate in the upward motor regime or can not overcome the force and is dragged by the force.

Because of that bimodal character, it is hard to precisely define stall force. If one define it as the maximum value of the force for which the most probable mean velocity is positive, then it can be assigned to a value $F_0 = 0.045$.

In the case of the second model for the force F_0 value of -0.2 one can speak about downward regime - in most cases motor will be dragged by the force, effectively moving in opposite direction that one would call the right one.

The middle plots, for the medium force of $F_0 = -0.1$, represent upward regime, where motor goes in almost 100% cases in the right direction.

In the bottom plots average velocity is even higher than 0.2, but one has to remember that for no-force environment this movement is far from one that we call effective.[1] On the right, for all the distributions one can find example trajectories.

Comparing both models in the terms of mean velocity against the opposing force (and taking into the account that initial and running parameters are not exactly the same) key difference can be formulated as follows:

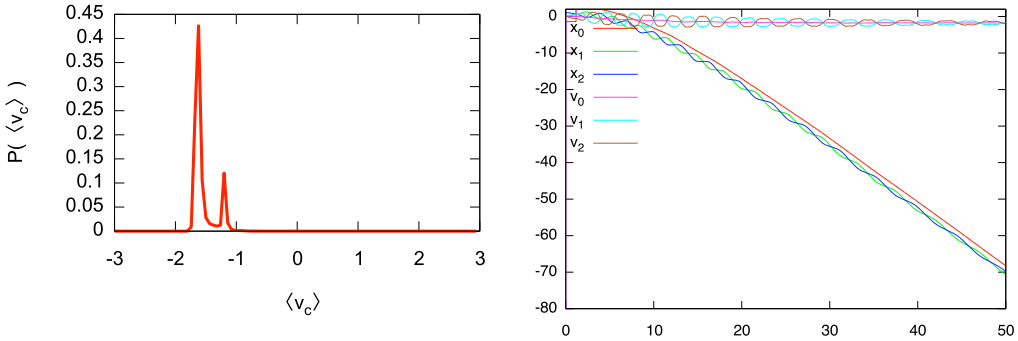
1. Model 1 can operate much faster (about one order of magnitude) than Model 2;

2. Model 2 still operates for the forces over four times higher comparing to the first model;

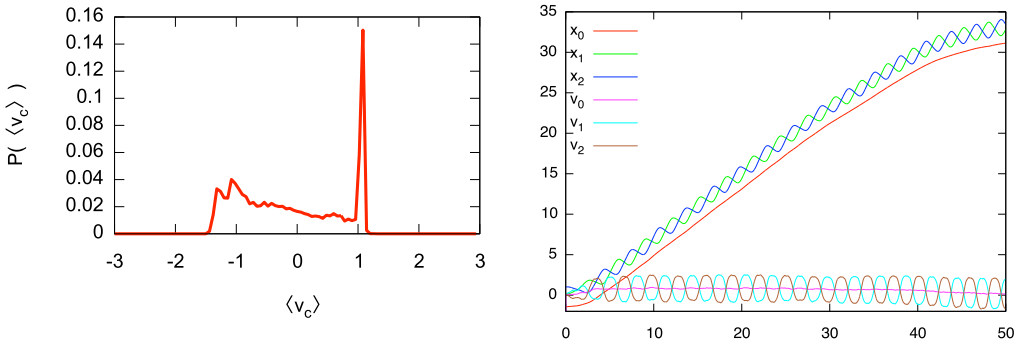
3. Model 1 produces bimodality in the mean velocity distribution — this is not the case for the second model.

Summarizing those findings, both models can be compared to car and the tractor. The first can be driven very fast, but more than bunch of fellow passengers

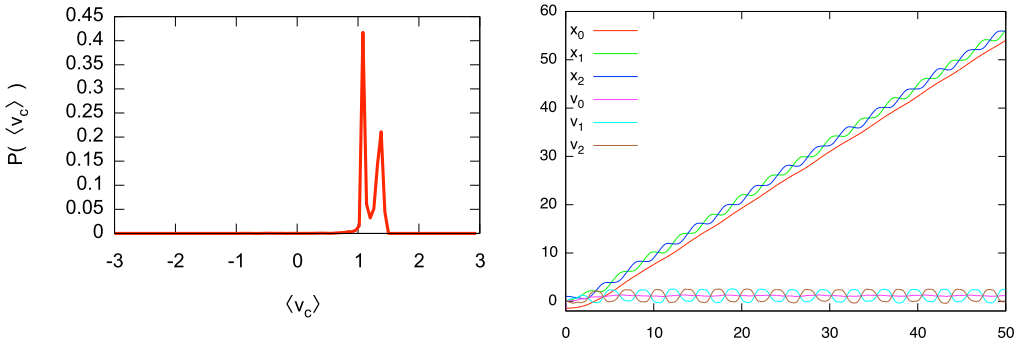
[1] With the respect to efficiency "kinds" presented before, for the classical formula for efficiency it is by definition zero. That is contrary to the other formulations of efficiency.



(a) $F_0 = -0.1$

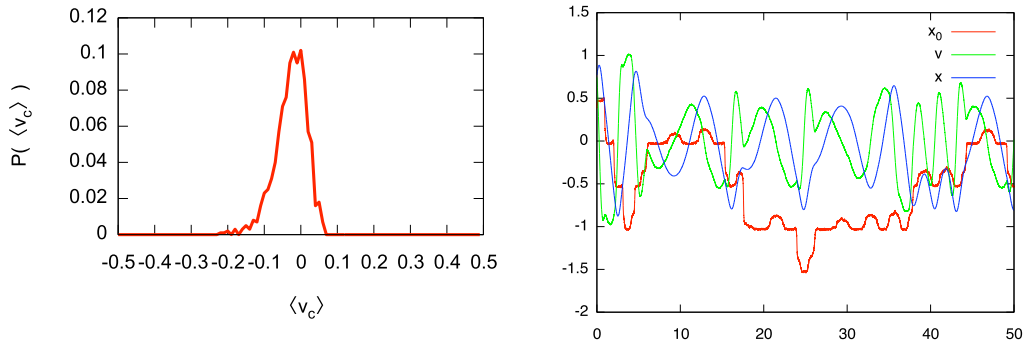


(b) $F_0 = -0.04$

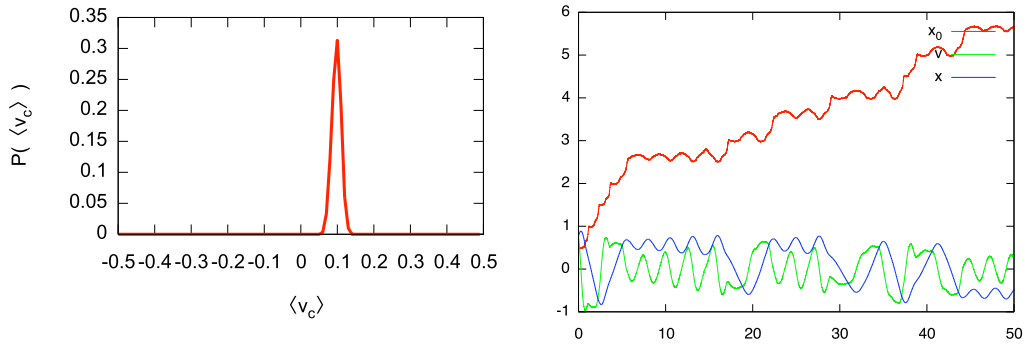


(c) $F_0 = 0$

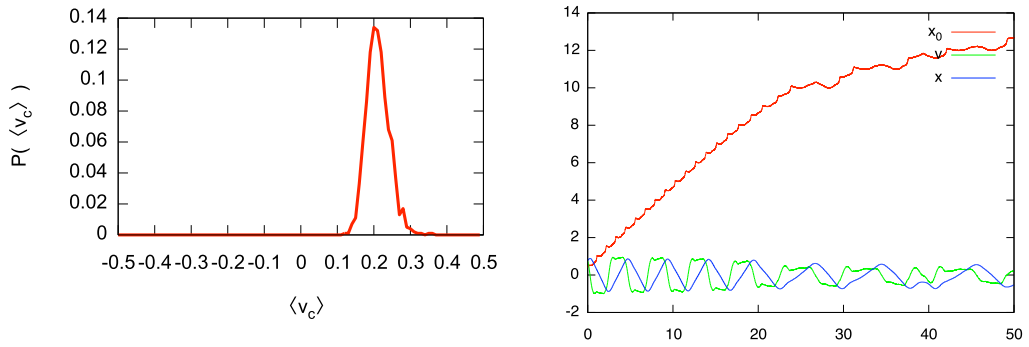
Figure 3.1: **The original Model 1:** Time- and ensemble-averaged mean velocity distribution as a function of force F_0 with example trajectory for $x_1 = 0.1$, $x_2 = 1.0$, $v_2 = 0.2$, $v_1 = 0.1$, $e = 0.1$. Every point has been calculated 100 times. The details of the original model can be found in Chapter 2.



(a) $F_0 = -0.2$



(b) $F_0 = -0.1$



(c) $F_0 = 0$

Figure 3.2: **The separation-of-mass Model 2:** Ensemble average of the time-averaged velocity distribution as a function of force F_0 with example trajectories. For the calculations for the Model 2, we have used fixed parameters as follows: $\gamma_0 = 0.02$, $\Gamma = 0.101790$, $m = 1$, $D_{x_0} = 1.0$, $D_v = 0.1$, time step $dt = 10^{-3}$, final time $t_{fin} = 50.0$, $q = 1$, $c = 0.1$. We also have kept $a = 0.5$, $b = 1.0$, $v = 0.745513$, $x = 0.771748$, $e = 0.023095$ and $d = 0.960061$. Every point has been calculated 100 times. The details of the separation-off-mass model can be found in Chapter 2

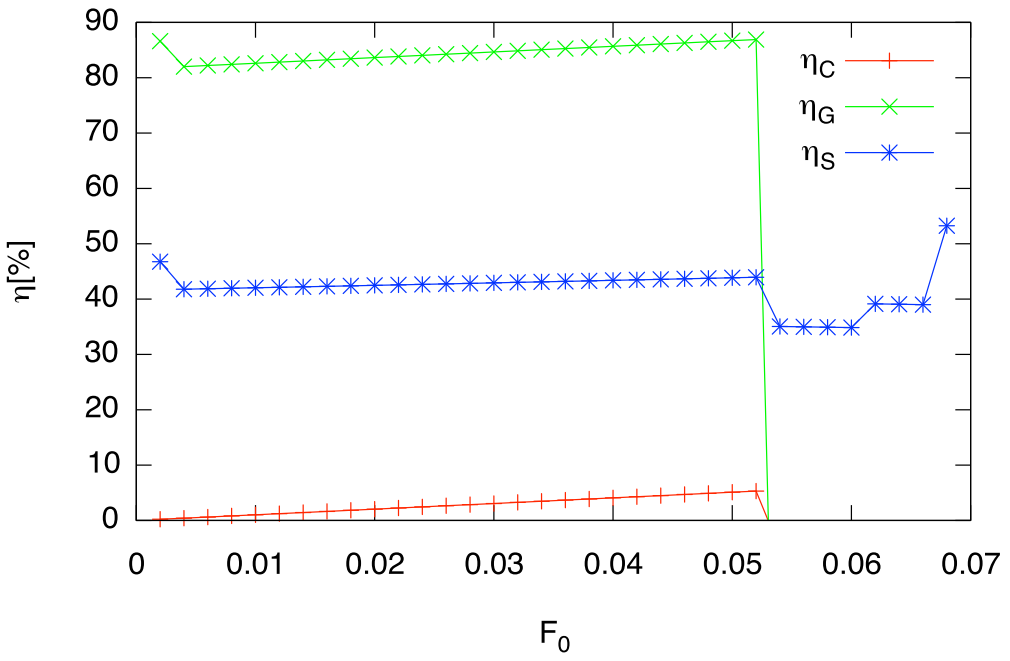


Figure 3.3: **Model 1** efficiency according to different definitions presented in Section 3.1.1–3.1.3 as a function of force F_0. Each refers to the ensemble-avaraged (100 independent trajectories). Friction parameter has been set to $\Gamma = 0.101790$

would impact on its performance. On the other hand, while capable of caring heavy trailer, it is not wise (and against the law) to drive a tractor on an expressway.

3.2.2 Efficiency definitions in use

Efficiency plots according to presented definitions for the first and second model can be found in Fig. 3.3 and Fig. 3.4, respectively.

For the first model in all the calculated definitions efficiency is almost constant (very subtle increase) until reaching stall point. In the case of general and classical themodynamical definitions that also manifest in the drop of the efficiency under the physical value of zero. On the other hand, for the Stokes efficiency because nominator is always positive and denominator changes just a little even with the change of mean velocity sign, the resultant efficiency is almost the same for all the values of opposing force.

In the second model, as it can be thought reading previous section, with the decrease of the opposing force, mean velocity increases. There should be a certain

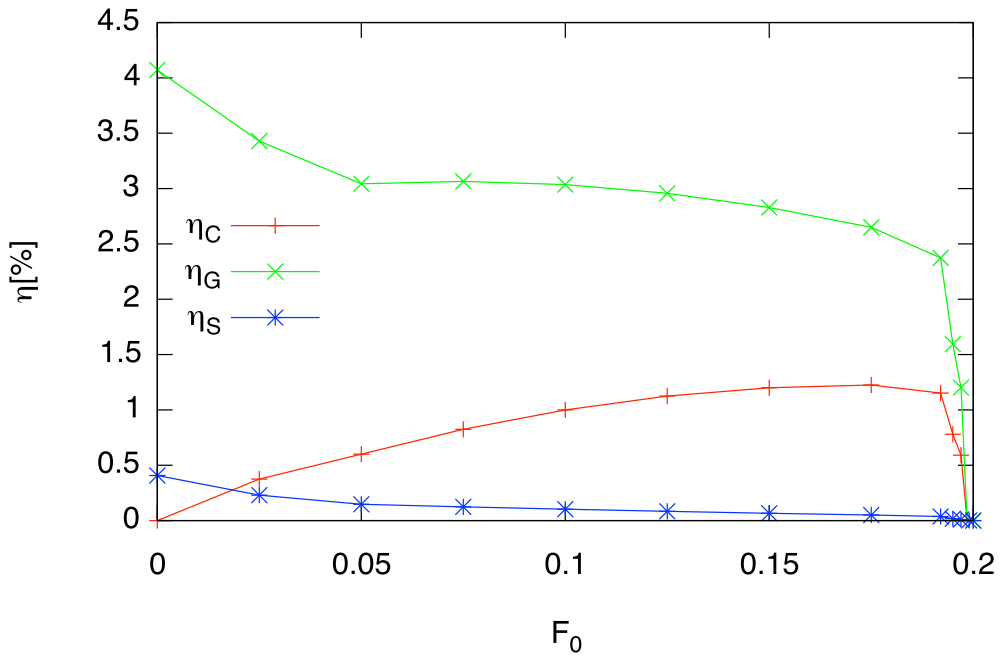


Figure 3.4: **Model 2** efficiency according to different definitions presented in Section 3.1.1–3.1.3as a function of force F_0. Every point is ensemble-average calculated 100 times, crucial friction parameter in this case is $\Gamma = 0.101790$.

force for which velocity is still high, while relatively strong opposing force still exists. In the case of presented simulations for classical definition of efficiency, this point is for $F_0 \approx -0.18$. For higher force values efficiency rapidly drops to zero (just before reaching force of $F_0 = -0.2$) and for lower values decreases gently until reaching zero when opposing force equals that.

In the experimental work of Nishiyama *et al.* [38] very similar behavior of the efficiency as a function of load force has been observed. Critical force value has been measured as 8 pN, which is in agreement with other works.

Situation is slightly different for the remaining two efficiency definitions. As generalized efficiency is the sum of thermodynamical efficiency and the against-friction term it has similar shape for higher force values, while for low force, because of the relatively high velocity it reaches the maximum value. Stokes efficiency for most of the checked force values remain on the low level - that is also the result of the form of definition with the given value of absorbed chemical energy.

3.3 Remarks on Model 2 efficiency

In this short section, we present two results obtained for Model 2 as a result of an inspiration coming from the article of Bormuth *et al.* [39].

3.3.1 Generalized efficiency without external force as the function of friction

As demonstrated in the previous sections, efficiency can be considered as a different thing, depending on the situation in which one asks about it. Here, we present generalized efficiency for the second model without the opposing force. As a result, efficiency now reads $\frac{\Gamma\langle v_0\rangle^2}{q}$. Like before, because of noise randomness, instead of just one simulation for every point, we have made a lot of program runs for different (random) Γ within certain range $0 < \Gamma < 1$. What one can see, is a clear "optimal" value, much easier to accurately "measure". Fig. 3.5 shows the plot for that, as in caption it's for absent force F_0.

3.3.2 Friction force as a function of mean velocity

Mean velocity has been calculated as a simple division of distance by time ($t = 50$). What we have observed (and what could be found in Fig. 3.6) is a kind of branching - some of the points follow linear $F = \Gamma \times v$ formula, while the others follow the as presented in the aforementioned paper [39].

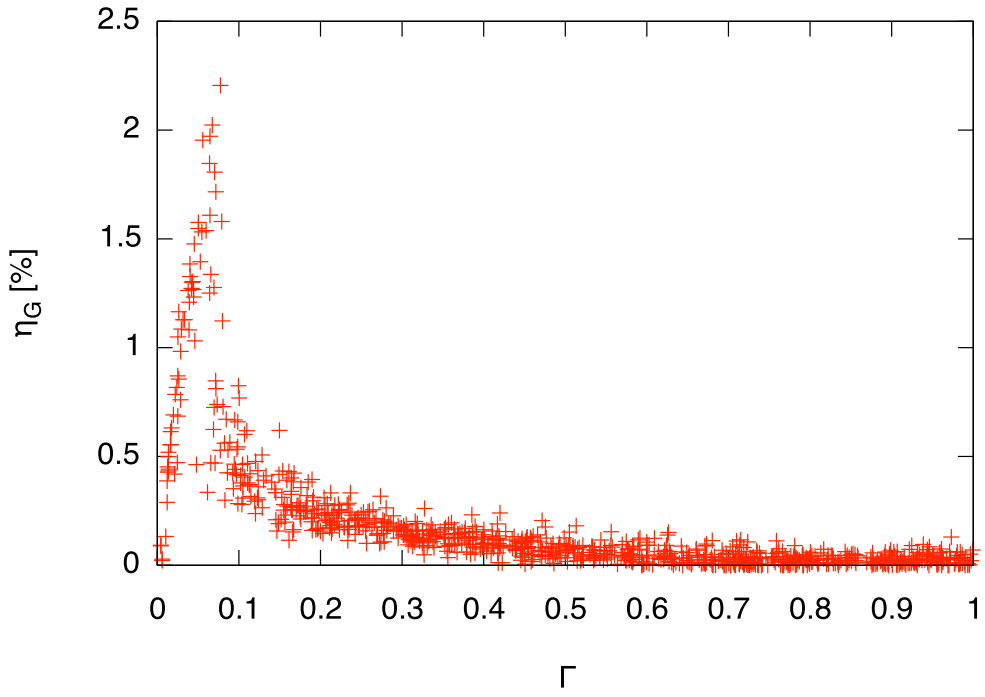


Figure 3.5: Generalized efficiency $\eta_G = \frac{\Gamma \langle v_0 \rangle^2}{q}$ as a function of Γ,$F_0 = 0$. The parameters of the Model 2 are $\gamma 0 = 0.02$, $m = 1$, $D_{x_0} = 1.0$, $D_v = 0.1$, time step $dt = 10^{-3}$, final time $t_{fin} = 50.0$, $q = 1$, $c = 0.1$. We also have kept $a = 0.5$, $b = 1.0$, $v = 0.745513$, $x = 0.771748$, $e = 0.023095$ and $d = 0.960061$.

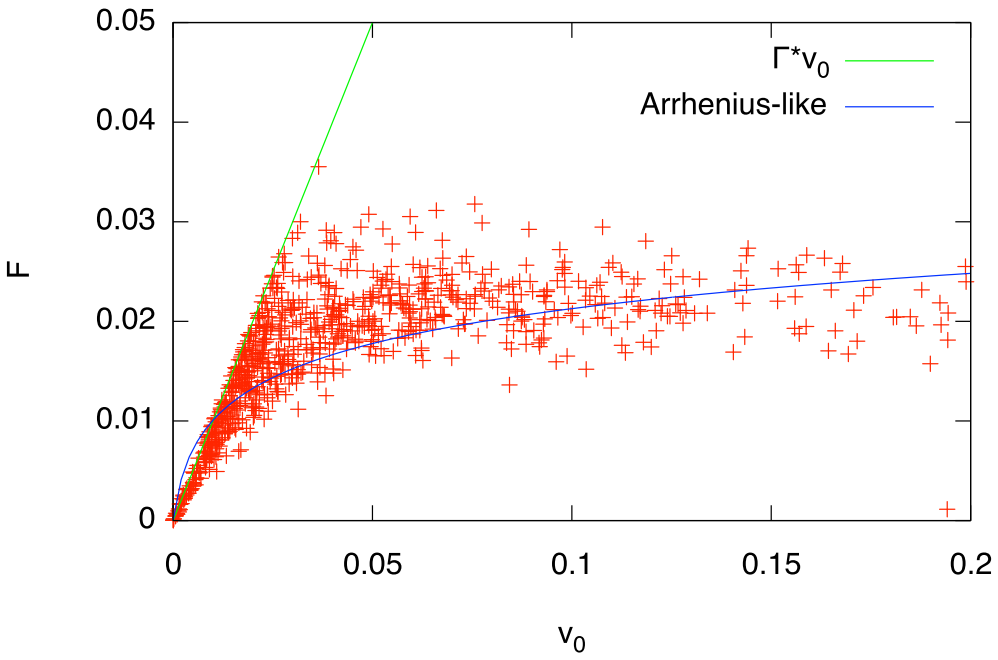


Figure 3.6: Friction force as a function of $\langle v \rangle$ for absent external force ($F_0 = 0$). This figure corresponds to Fig. 2 in [39]. Ratchet nature of the model mimics energy barrier needed to overcome to break molecular bonds.

While some portion of points follow the classical, linear dependence, for the higher velocities, dependency becomes nonlinear. In the article of Bormuth it is proposed that asymmetric energy barrier exists, that needs to be overcame to break the molecular bonds. As a result, using Arrhenius theory exponential dependency has been derived. The striking similarity of the second model with those experimental results is becoming understandable if one recalls ratchet nature of model presented here, being the theoretical equivalent of the energy barrier of protein's molecule.

Chapter 4

Summary and Conclusions

In this thesis I have presented a two-version model of an entropic forces driven molecular motor. In both cases, two motor heads are connected by a contracting and expanding elastomer. On the other hand, track of molecular motor is modeled by ratchet asymmetric potential. Both models are tuned in a way that they perform hand-over-hand motion, in that way they support hand-over-hand hypothesis of kinesin molecular motor movement.

First model, presented along broad discussion of working parameters, consists of inertial motion equations for both heads and the energy depot derivative formula. The measured thermodynamical efficiency, caching up to 30%, has been one of the key model characteristics. Despite that, deeper analyzes revealed that ratio of inertial and friction terms is definitely to large to exist in overcrowded cell environment.

The second model separates relative motion of heads from the center of mass motion. Therefore, while still keeping acceleration terms in the relative motion equations, center of mass motion expression has been formulated in the language of the overdamped Langevin equation. This reasoning allowed the previously proposed two-head mechanism, while keeping the generally accepted assumption of the dense environment. The second model is suffering thermodynamical efficiency loss, comparing to the preceding idea.

Performance issues has been discussed in the context of different efficiency formulations. The most broadly accepted thermodynamical efficiency of the second model is in agreement with the experimental works. As for the other formulations it was not possible to compare them directly with experimental works, general discussion has been enclosed.

Models similar to presented here, containing nonlinear velocity dependent dissipation terms have also been broadly used to describe active motion of cells [40], collective swarm dynamics [20, 41] and coherent changes in direction of moving groups of animals [42].

More recently, the interest in using the active Brownian particle models (ABP) has been boosted [43] by noticing apparent connection between ABP models and dynamics of coupled molecular motors (CMM). This opens a new field of important applications of the ABP methodology in investigations of e.g. bidirectionality of motion and velocity reversals evidenced in experimental motility essays on groups of biological molecular motors [44].

In my opinion, the future of molecular motor modeling lies in the interdisciplinary approach, where presented here Brownian dynamics meets molecular dynamics modeling and chemomechanical technique. Even more insight to the properties of the system could be found by careful studying constantly appearing new experimental works. More than a few groups nowadays are concentrating their efforts on studying the collective behavior of molecular motors. On the other hand, very basic and general terms as friction, while under constant investigation [39], still lack deeper understanding of their molecular origin.

Appendix A

Comparison with experimental data

In this Appendix I present comparison of my model with experimental data enclosed in article of Bormuth *et al.* [39]. As the units used in that work are different, in the beginning I will show conversion of the diffusion equations used there to the ones in this work. Here the equations with dimension (2.4)-(2.6) diffusion coefficient is defined as follows:

$$D_v = \frac{\gamma}{m} k_B T. \tag{A.1}$$

On the other hand, in aforementioned work it is defined as:

$$D_{TIRF} = \frac{k_B T}{\gamma_{TIRF}}, \tag{A.2}$$

where TIRF stands for "total internal reflection fluorescence" experimental method used by the Authors.

Those diffusion constants can be related as:

$$D_{TIRF} \times \gamma_{TIRF} = \frac{D_v m}{\gamma} \tag{A.3}$$

$$D_v = \frac{D_{TIRF} \times \gamma \times \gamma_{TIRF}}{m}. \tag{A.4}$$

As γ_{TIRF} has a dimension of force, contrary to γ (used in this work, which has a dimension of frequency), both friction coefficients can be related by:

$$\gamma = \frac{\gamma_{TIRF}}{m}. \tag{A.5}$$

Applying this to (A.4), one obtains a final relation:

$$D_v = D_{TIRF} \frac{\gamma_{TIRF}^2}{m^2}. \tag{A.6}$$

From Bormuth *et al.* paper [39] I have used data for type kip3p kineisn:

- $D_{TIRF} = 0.0043\mu m^2/s = 0.0043 \times 10^{-12} m^2/s = 4.3 \times 10^{-15} m^2/s$
- $\gamma_{TIRF} = 0.95\mu Ns/m = 0.95 \times 10^{-6} Ns/m$
- whole kip3p weight is $91kDa = 1.51 \times 10^{-22} kg$

The remaining missing value I have obtained from Peters *et al.* paper [45]:

- kinesin motor domain (head) weight is aprox $m = 39kDa = 0.65 \times 10^{-22} kg$

Applying above values to (A.6) one gets:

$$D_v = \frac{4.3 \times 10^{-15} m^2/s \times (0.95 \times 10^{-6} Ns/m)^2}{(0.65 \times 10^{-22} kg)^2} \tag{A.7}$$

It is more convenient for the model to use following formula:

$$\begin{aligned} m\sqrt{2D_v} = \sqrt{2D_{TIRF}\gamma_{TIRF}^2} &= \sqrt{2 \times 4.3 \times 10^{-15} m^2/s \times (0.95 \times 10^{-6} Ns/m)^2} \\ &= \sqrt{7.76 \times 10^{-27}} = 8.8 \times 10^{-14} N\sqrt{s} \\ &= 0.088 pN\sqrt{s} \end{aligned} \tag{A.8}$$

Now lets evaluate the entropic force associated with linker stretching. Data read from Greater *et al.* article [31], indicate that following nonlinear force-length dependency:

$$\begin{aligned} 2TS'(\Delta x = 5.5nm) &= 50pN \\ 2TS'(\Delta x = 6nm) &= 100pN, \end{aligned} \tag{A.9}$$

where $\Delta x = |x_1 - x_2|$ is the relative distance between the motor heads[1].

To keep those values consistent with the proposed elastomer function (2.8) following parameters for my model should be set as follows:

$$\begin{aligned} a &= 7.69 \times 10^{-3} pN/nm \\ b &= 1.647 \times 10^{14} pN/nm^3 \end{aligned} \tag{A.10}$$

[1] It is crucial to note that from some distance of kinesin heads, force of elastomer rises very high. It is similar situation with our model, where stochastic force part dominates over all the other ones (see Tab. A.1).

With those assumptions the ratio of the entropic force and stochastic force can be approximated as follows from the aforementioned works:

$$\frac{2TS'(\Delta x = 6nm)}{m\sqrt{2D_v}} \approx 10^3 \tag{A.11}$$

As one can notice, comparing this ratio with the one from the presented Model 1 (see Tab. A.1), the experimental data suggest even stronger influence of entropic forces in the Models' Langevin equations. Yet, the ratio is similar in the orders of magnitude.

Table A.1: Comparison of numerical values of the forces present in Langevin equations for the Model 1, for the $F = -0.2$

$U'(x)$	ratchet potential part with the load force	|0.6|
$F_S(x_1 - x_2)$	entropic force of an elastomer	|35|
$m\sqrt{2D_{v_i}}\xi_i(t)$	Stochastic force	|0.15|
$mde(t)v_i$	Mechanical energy from the depot	|1|
$m\gamma_0 v_1(t)$	Dissipative forces	|0.1|

On the other hand, in my simulations stall force was about $F = -0.2$ and that value should be compared with about $F_{stall} = 8$ pN read from the experiments [46]. Force value is incorporated into the $U'(x)$ term in model equation so it is possible to directly compare e.g. force to entropic force ratio in the model with similar ratio obtained from the experiment:

$$\frac{F_{stall}}{2TS'(\Delta x = 6nm)} \approx 12.5, \tag{A.12}$$

while for the model this value is approximately 130. Difference here appears to be huge, although the choice of comparison elements is arbitrary and entropic force component can vary significantly even with small change of motor head distance (cf. twofold change of it when distance has been altered by only 0.5 nm).

Summarizing this appendix, while the actual values used in the model simulations cannot be compared 1:1, the general trends and domination of some terms remain similar to the ones observed in the experiment. It might be possible to associate most of the model coefficients with certain laboratory measurements, the problem is however that not all of needed values has been obtained by one group and experiment. Even in this short and simple comparison, the data about elastomer stretching behavior had been gained from alternative source.

Appendix B

Backsteps problem

The molecular motor kinesin has a directed motion along the microtubule. It means that in general steps are taken only in one direction. Occasional steps to the other, "negative" direction are called backstepping. In the recent work of Bier and Cao ??, it has been proposed that occasional backstepping is actually not a disadvantage, but rather a mechanism that by increasing the entropy of the moleculer motor system, increases the free entropy-associated energy and as a consequence speeds up the motor.

Here, in this short appendix I would like to explore some details of my models in the context of aforementioned work. Backstepping of course can be accounted in the easiest way by simple summing all the negative iterations of the simulations. In that way it is possible to distinguish between same average velocity values with a different trajectory "history".

The major disadvantage of this approach I find in counting all the small "fluctuation backsteps" that constantly appear in all crowded, Brownian environments. To address this problem, I have created a simple algorithm that finds and counts only the "real" steps, that means the ones exceeding some given value.

For the backstopping accounting I have taken twenty as much time as in other calculations ($t = 1000$) and on purpose I have chosen critical area of parameters (for friction $\Gamma = 0.1$ and external force $F = -0.18$)).

The result of the first simulation can be found in Fig. B.1. It has been made by 100 simulations for each value of force in the range $-0.17 < F_0 < -0.21$, with the increments of $\Delta F_0 = 0.001$. For initial values of F_0 (still in forward regime) there are no backsteps, at certain point from time to time there are some backsteps. One can see that until mean velocity is positive, number and percent of backsteps are a linear function of mean velocity. Situation changes with negative mean velocities — distribution of points is much broader and it doesn't look linear anymore.

On the other hand, in the second Fig. B.2 external force is kept fixed at $F_0 = -0.19$, while now friction Γ is changing from 0 to 0.2 with $\Delta\Gamma = 0.01$ increments

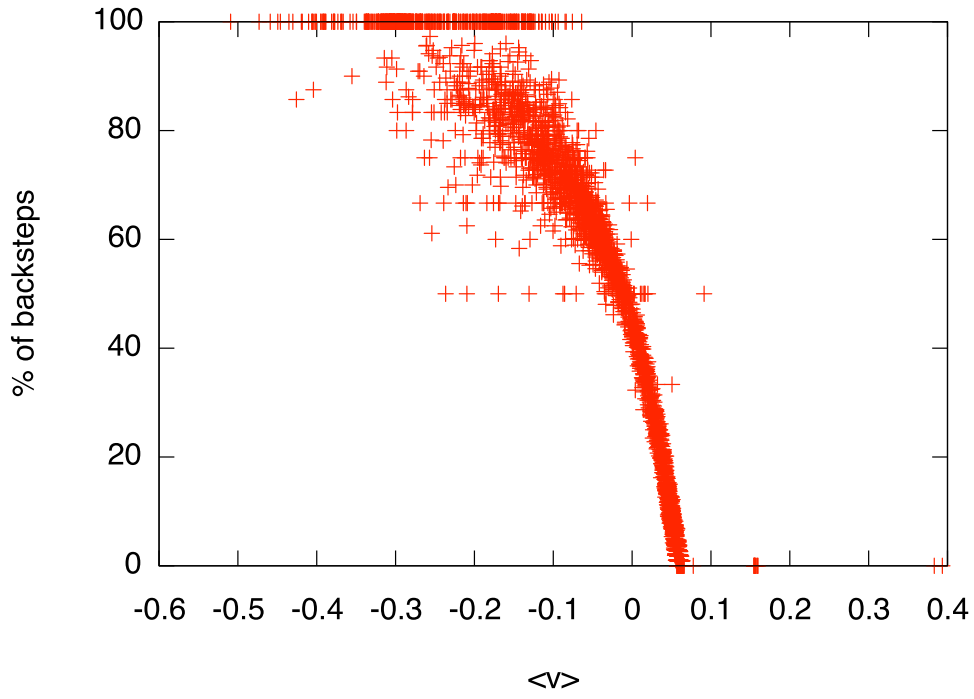


Figure B.1: % of backsteps as a function of $< v >$ for fixed Γ and variable $-0.17 < F_0 < -0.21$.

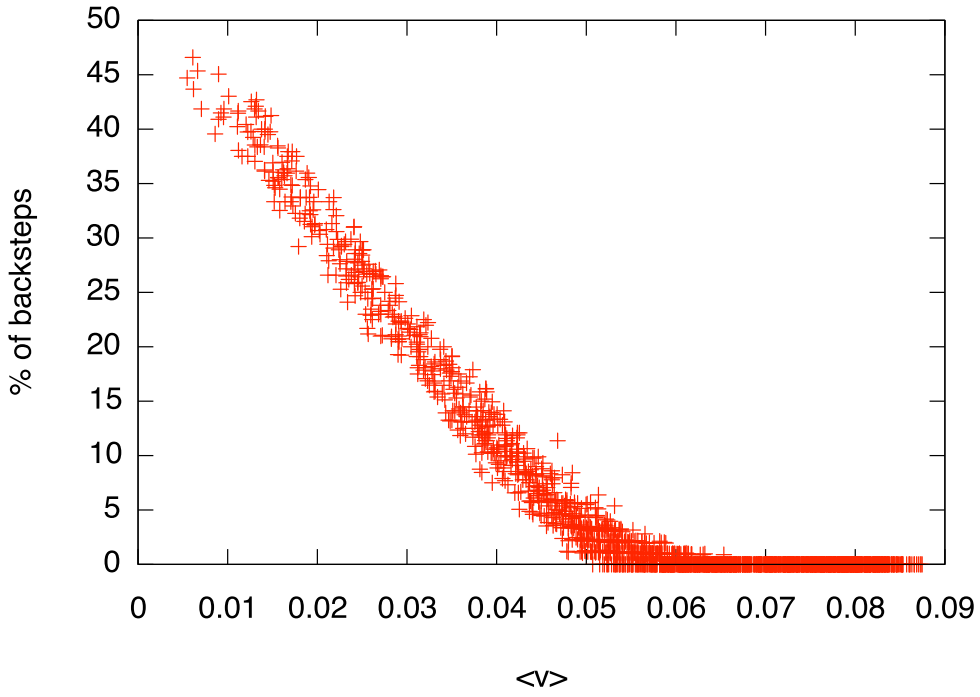


Figure B.2: % of backsteps as a function of $< v >$ for fixed $F_0 = -0.19\,\Gamma$ and variable $0 < \Gamma < 0.2$.

and the number of simulations is still 100. Here it can be observed a linear behavior up to critical value of $\Gamma = 0.05$, and there are no backsteps for higher Γ.

Appendix C

Active Brownian Motion Simulation on Graphic Cards

This appendix is a result of my work on exploring possibilities of a faster method to study molecular motors Brownian dynamics. It has not been published yet, although a modified version of this text can be found on ArXiv repository [47].

C.1 Introduction

Over the last few years General Programming on Graphical Processing Units (GPGPU) has started to spread in all the areas where time consumption and performance of calculations is crucial. Scientific simulation is a perfect example [48–50].

Motion of molecular motors has been simulated using different approaches and models [22, 51–54]. Inherent in the modeling of stochastic processes is to model the noise i.e. non-systematic, fluctuating increments to the process, under the study. To obtain the most probable behavior of the system, it is necessary to repeatedly perform calculations and then calculate their mean values based on stochastic properties of the ensamble. On the other hand, chaotic behavior of models with respect to initial parameters requires simulation for different sets of both starting and constant conditions. All of those tasks can be easily divided into separate threads — problem is highly parallel.

Most of high performance computing (HPC) in science is performed on clusters — sets of single or multiple-core processing units connected by network. Workflow consists of preparation of source code, its transmission over the network using cluster controller, putting *job* to *queue*, some waiting time, execution of the parallel program and finally acquisition of results over the network.

While fast, that approach has several drawbacks. First of all, for many simulations the benefit of faster calculations, if one includes time for "sending", "waiting"

and "acquisition", might be rather insignificant. Secondly, the cost of every computer forming a cluster is much higher than the cost of even high-end graphical processing unit (GPU). Every GPU itself is a multiprocessing unit - even the low-end Nvidia GeForce 320M GPU found on 11" Apple MacBook Air laptop has 48 cores, while SU9400 Intel processor (CPU) found on the same device offers only 2 cores [55].

The most popular technique for performing GPGPU nowadays is NVidia's proprietary technology CUDA. While very efficient, it's vendor agnostic - it runs only on certain Nvidia graphical cards. In 2008 Khronos Group consortium created first specification of OpenCL which has been declared as an "open standard for programming heterogeneous data and task parallel computing across GPUs and CPUs" [56].

Heterogeneous approach allows researchers to write programs managed by *hosts* that can be run on different OpenCL *devices*, even the one where GPUs cannot be used for OpenCL calculations. On that occasion, simulations are performed on all of the CPU cores. Performance is worse, yet calculation can be done without a single change of source code. OpenCL parallel nature also allows using all the CPU processing power without any special multithreading programming — task are dispatched by OpenCL into different threads to all available processor cores.

One of downsides of GPGPU is that both writing data to GPU and reading data back to CPU is carried over a relatively slow system bus. It takes the same time to transfer data from CPU to GPU and to carry a more than a few operations on the GPU on the same set of data. This will be evaluated later in the text. That said, crucial for good overall simulations performance is to carry GPU - CPU data transfer only when it is really necessary.

At this point there is OpenGL, a technology developed over 20 years ago by Silicon Graphics and now maintained by the Khronos Group, which as already mentioned, is responsible for developing OpenCL. OpenGL is a standard specification for writing programs that produces computer graphics. Creating an image on a monitor consists of setting environment, sending instructions to GPU to execute them and finally show the result on the screen. For example, rotation of an object is performed by sending a simple command to rotate by some angle and over some axis, while the calculations of all coordinates are made by GPU, without CPU burden.

OpenCL even in its initial specification mentions the possibility of integration with OpenGL. Sending the results, which are already in the GPU memory, to the CPU and re-sending them back to the GPU to visualize them, seems as an obvious waste of both time and device processing resources.

In this appendix, I present an approach which integrates simulation and presentation of the results on the same processing unit, GPU. If for some reason OpenCL cannot be run on GPU, it still can work on CPU — much more efficiently than by

simple execution of single thread simulation. Because of the little performance impact, results can be presented after trajectories for a given sets of parameters are calculated. Over time, results are averaged and a researcher can terminate computation when he finds results sufficient to prove or invalidate his thesis.

Specific problems of active Brownian motion (ABM) simulations using OpenCL are explained in a subsequent section. Comparison of performance on different hardware and software systems is also provided.

In this thesis I have extensively used plots of average velocity as a function of force F_0. In this appendix this relation is used to demonstrate GPGPU approach and to compare the performance for various software and hardware setups.

Previous approaches used to check similar relations were based on serial multiple program executions, with parameters governed by Perl script. In the OpenCL method, several instances of the same program, different only by force parameter F_0 are run simultaneously. Every instance of that program is called a *work item* in OpenCL. Over the time results for given parameters are averaged to maintain reliable result, independent of a given set of random numbers.

For every set of parameters OpenCL *kernel* as in Algorithm (1) is executed.

Algorithm 1 Pseudo kernel

```
read input parameters from global memory
fill local memory with parameters
for i = 0 to T do
    do Marsaglia xorshift
    do Box-Muller transformation
    calculate new velocity and energy (as in eqs. 2.11,2.12,2.13)
end for
assign local parameters to global memory
```

After kernel finishes calculations, different techniques are used to visualize the results. Their differences and performance are discussed in Sec. C.2.2.

Solving stochastic differential equations requires generating noise (in eq. $\xi(t)$ is understood as a source of a Gaussian white noise), which computer equivalent is a set of pseudo-random numbers. The programmer can either fill OpenCL buffers with random numbers provided by host RNG or write OpenCL implementation of existent RNG algorithms. Downside of a first approach is the time it takes to pass numbers to GPU — in case of OpenCL/OpenGL it is almost always more efficient to send initial number and set of routines which would be executed on the GPU. In this work, I have chosen the second way of generating random numbers, as in [57]. Marsaglia xor-shift algorithm [58] has been used and because white Gaussian noise needs normal number distribution instead of unitary, Box-Muller transformation has been also applied.

The other problem that arises, when one writes a program running on the GPU, is the precision issue. Using double precision (DP) variable types (like *double*) is a standard for most scientific calculations. While the newest graphic cards available on the market allow DP usage, for the sake of compatibility with the ones that does not have this possibility, kernels should use single precision (SP) variable types like *float*. SP programs can be as precise as the DP ones, as long as the technology drawbacks are overcome in a correct manner [59]. This includes avoiding adding very small numbers to very big ones.

While offering less flexibility in setting high precision data types, OpenCL specification encourages programmers to use vector data types. On certain graphic cards, *float4* (consisting of four float numbers) is the "natural" data type and keeping the same structure of data may increase performance. It should be also noted, that if executed on modern CPUs, using vector data types can be also beneficial in shortening calculation time [60].

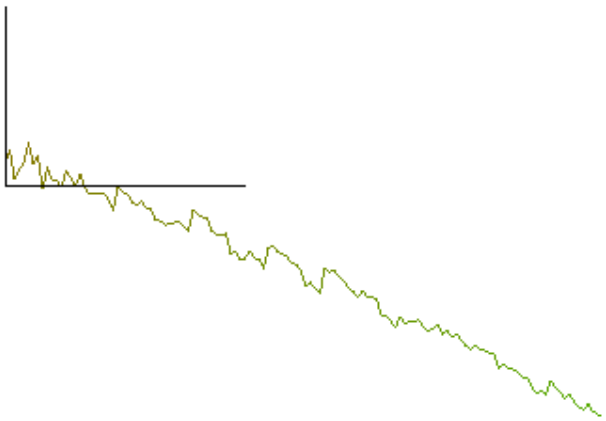


Figure C.1: Example plot of tested program. Mean velocity $< v >$ is plotted against opposing force F_0 (see details in the text).

C.2 Performance issues

A test program has been compiled and run on various Mac OS X 10.6 capable computers. Through the test run all the other user visible applications have been shut down. Every test sequence consisted of running program for 20 steps for every power of 2 from 2^0 to 2^{16} starting parameters, i.e. work items. Then, the last fifteen steps have been taken into account and the average has been plotted (Fig. C.2).

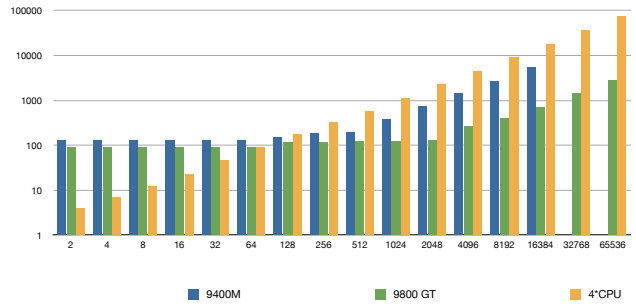


Figure C.2: Comparison of calculation time, depending on used hardware and number of simulation points. The lower, the better performance is.

Depending on a chosen hardware, time used for calculations varies substantially.

While having multiple cores (from few up to thousands), graphic cards suffer delays in every situation which requires transferring data to and from it to the CPU. GPUs usually work on lower frequency clocks. On the other hand, easy memory access and high frequency clocks would not overcome the main CPU drawback - low number of processing cores (from one to six in most cases).

C.2.1 Performance on various simulation setups

Preliminary tests of my program show that, depending on the number of parallel tasks to compute, GPU can be slower or faster, comparing to CPU. In the case of traditional central processor calculation, on 4-core Q6600 it always takes twice the time to compute twice larger set of simulation points.

On the other hand, it does not take significantly longer to compute more simulation points until certain threshold number is reached. The latter is a generic characteristic of a given graphic card. Most obvious threshold should be GPU's core count. As the test results show, it is not always true.

Nvidia GeForce 9400M is a popular graphic card used in laptops until last year. It has 16 processing cores and uses up to 512 MB of system memory (in case of this test - DDR3). While one can expect no difference in time of calculations until number of simulation point reaches number of cores and increase from thereon,

actually this threshold occurs at number of 128. That is 8 times more than core count.

Introduced in 2008, Nvidia GeForce 9800 GT is a PCI-Express standalone card. In this test, version with 512 MB GDDR3 internal memory has been used. It has 112 cores working at 1500 MHz processor clock. Here, there is clear threshold corresponding with the number of cores, when the time of single simulation starts to increase. However, unless the number of simulation points exceeds 2048 every doubling of required simulation points (say, an increase from 2^8 to 2^9) does not result in doubling of the run time.

In other words, it means that there is no difference in simulation time whether or not researcher calculates result for one or thousand parameters — as long as the upper limit is under the threshold, which is dependent on a used graphic card.

To further study OpenCL GPU performance, I have carried out more detailed calculations. Between 50 and 3500 work items, with the increment of 50, I have measured the time of one calculation step. Because for every work item number calculations have been performed at least 20 times, the average with standard deviation has been carried out. The procedure has been done for 10^4 and 10^3 iterations steps and for zero steps. The latter helped to measure offset time for initial variables transfer from global to local memory as well as final transfers from local to global memory and visualization (which performance is discussed in the second part of the article). Subtracting offset time from "regular" calculations time provided a more detailed view of the matter — result can be found in Fig. C.3.

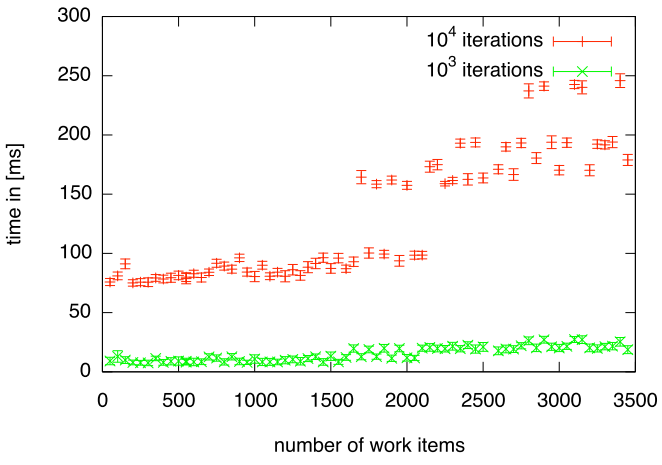


Figure C.3: Time of OpenCL calculations carried on GeForce 9800 GT GPU against number of work items. Band-like structure could be notice instead of linear rise of calculation time.

Most prominent, when one observes plot in Fig. C.3, is the band structure of performance. There is almost no rise in time of calculation until number of work items slightly passes 2000. This is similar observation like in the previous plot. What distinguishes it, is the situation repeats for what is happening over 2000 work items. It can be said that, GPU operates in some certain regimes of performance, and crossing the thresholds results in non-linear rise of calculation times, i.e. while on CPU time of calculation is a linear function of work items (work to be done), on the GPU time of calculation rather can be explained as floor- or ceiling-like functions.

Closer look at the plot in Fig. C.3 can reveal that regardless of number of iterations, bands occur in similar places — for same number of work items. In the authors opinion, for a given graphic card there exists a maximum number of work items that can be done at the same time without any significant performance impact. 9800 GT GPU consists of 14 cores for each 8 streaming processors are provides. That makes 112 processors to operate at the same time. Every core operates in 32 *warps* that help hide latencies of the memory. It appears that this about-2000 is the threshold after reaching it GPU has to employ extra cycle to utilize all the work items. That situation seams to reappear for the aforementioned threshold multiples.

For some near-the-threshold regions one can see that time of calculation is sometimes on the longer time band. One of the explanations could be that GPU has been used at the moment of simulation for some other, most probably, system task.

C.2.2 OpenCL/OpenGL interoperation performance

In this part, I will analize the impact on a performance, when intermediate simulation steps are shown to the user. In all tests the time for one cycle (gathering initial parameters from memory, actual calculations and on-screen presentation of results) has been counted in microseconds.

Three different approaches will be presented and compared.

Presenting results with external software

In the traditional approach calculations are carried out on a fast device (CPU, GPU), final results are saved into files and finally data are plotted with external software (e.g. gnuplot). This procedure may be adequate if one carries long calculations with a single final plot. Impact of memory transfer issues, time of external plotting software to initialize, read data from a file and plot is irrelevant comparing to the time of calculations.

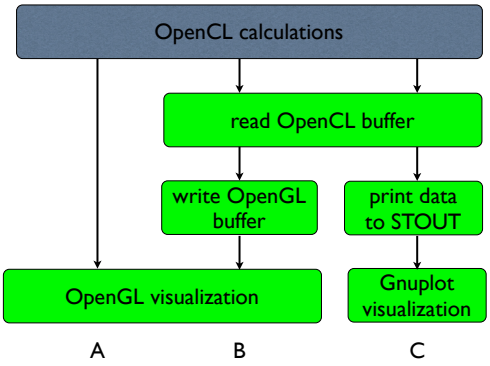


Figure C.4: Different approaches to OpenCL calculation visualization. From left: OpenCL/OpenGL interoperation with shared buffers (a), OpenGL is used to visualize results, but buffers are not shared with OpenCL (b), almost traditional approach where results are send to standard output, captured by gnuplot and visualize there (c).

Intermediate steps with OpenGL

In this strategy, both calculations and visualization of the results are done by the same program. OpenCL device runs the kernel, in which calculations are done. Results are saved into the host memory and then they are plotted onto the screen, using OpenGL. OpenGL is initialized only once at the beginning of a program run. After receiving new data the screen is only updated.

Intermediate steps with OpenGL/OpenCL shared buffers

In the proposed method, there is no transfer of calculation results from OpenCL device to host memory. Both computation and visualization operate on the same buffers.

OpenCL and OpenGL specification requires that if one wants to use shared memory buffer in-between those two frameworks, first OpenGL buffer should be created. Secondly, instead of creating plain OpenCL buffer (*clCreateBuffer*), it has to be created from the OpenGL one (*clCreateFromGLBuffer*). In OpenGL world, buffers hold mostly either information about position or color. For example, color buffer can hold a chain of float numbers representing colors in RGBA scheme. The four numbers ($a_1 ... a_4$) stand for three color intensities (red, green, and blue with $a_i \in [0, 1]$) and a_4 stands for the alpha opacity controller. Similarly, vertex buffer holds a chain of float numbers that every four represent position in homogeneous

coordinates (x,y,z,w).

Comparison of performance

The time of one step has been calculated for every method for three different numbers of simulation iterations and for three different numbers of work itmes (starting parameter of the force F_0). Resultant time has been averaged over 15 consecutive steps and plotted in Fig. C.5.

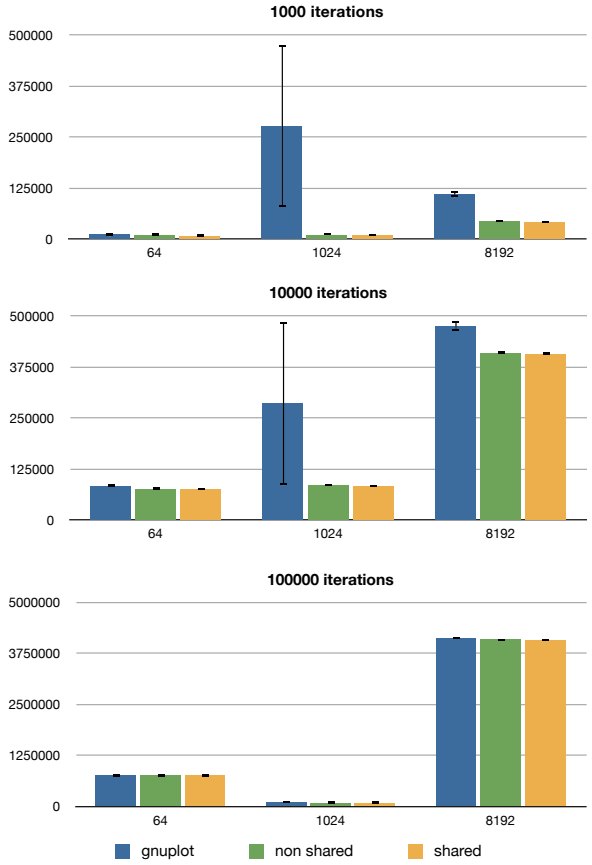


Figure C.5: Time in μs of one calculation cycle for shared buffers method, non-shared buffers method and a reference gnuplot technique

For longer runs (with high number of iterations) the performance of all three methods seems to be comparable. On the other hand, the lower number of itera-

tions, the more striking is the difference between different approaches. It can be found that for given numbers of calculated parameters the time difference between methods is more or less constant.

Using linear regression one can estimate the average time for one iteration for different methods and number of parameters with an offset being time of data preparation, result acquisition and visualization. Results of the latter operation can be found in Fig. C.6.

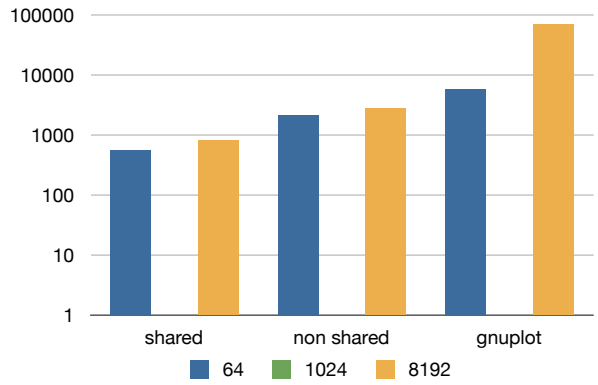


Figure C.6: Time in μs consumed by every calculation step for data preparation, result acquisition and visualization. Notice log scale for time.

It appears that for gnuplot reference technique, time dedicated to visualization can be longer by even two orders of magnitude comparing to the fastest shared buffers method. On the other hand, non shared buffers method can be a few times longer than shared buffers approach. For all tested methods of transferring data to the screen it has always taken more time to show plots of larger data. However, that impact was much more evident for gnuplot approach than for other, OpenGL techniques.

One of the ways to present comparison between different visualization methods is to show the number of possible calculation steps that could have been done in time spent on a visualization step. Data visible in Fig. C.6 has been divided by time of one iteration (that has been calculated from the same linear regression as mentioned before). Results can be found in the Tab. C.1.

For a longer calculation the number of possible iterations that could be taken in the time of visualization even for the slowest gnuplot reference method is neglectable comparing to number of iterations done. On the other hand, for relatively short runs (e.g. screening of data every $1000 - 2000$ steps) for larger sets of

Table C.1: Number of possible iterations that could be taken in time lost on visualization

method	2^6 parameters	2^{13} parameters
share buffers	80 steps	20 steps
non-share buffers	283 steps	69 steps
gnuplot	768 steps	1734 steps

parameters visualization can take more time than calculation itself.

C.3 Conclusions

In this appendix I have shown the approach to visualize OpenCL calculation of stochastic differential equations using co-existing OpenGL framework. That technique simplifies the workflow of SDE calculations, without loosing performance boost from graphic card use. Presented OpenGL approach, especially with shared buffers, can help to gain better insight to calculation in real time.

Vendor agnostic OpenCL can be run on different devices, even rewriting naive C code and running on same CPU can give outstanding improvement of a calculation time.

Bibliography

1. Perlson, E., Maday, S., Fu, M.-m., Moughamian, A. J. & Holzbaur, E. L. Retrograde axonal transport: pathways to cell death? *Trends Neurosci.* **33,** 335–344 (2010).
2. Saha, A. R. *et al.* Parkinson's disease alpha-synuclein mutations exhibit defective axonal transport in cultured neurons. *J. Cell. Sci.* **117,** 1017–1024 (2004).
3. Stokin, G. B. *et al.* Axonopathy and transport deficits early in the pathogenesis of Alzheimer's disease. *Science* **307,** 1282–1288 (2005).
4. Her, L.-S. & Goldstein, L. S. B. Enhanced Sensitivity of Striatal Neurons to Axonal Transport Defects Induced by Mutant Huntingtin. *J. Neurosci.* **28,** 13662–13672 (2008).
5. Hirokawa, N. & Takemura, R. Biochemical and molecular characterization of diseases linked to motor proteins. *Trends Biochem. Sci.* **28,** 558–565 (2003).
6. Seidman, C. Genetic causes of inherited cardiac hypertrophy: Robert L. Frye lecture. *Mayo Clin. Proc.* **77,** 1315–1319 (2002).
7. Petit, C. Usher syndrome: From genetics to pathogenesis. *Annu. Rev. Genomics Hum. Genet.* **2,** 271–297 (2001).
8. Seabra, M. C., Mules, E. H. & Hume, A. N. Rab GTPases, intracellular traffic and disease. *Trends Mol. Med.* **8,** 23–30 (2002).
9. Block, S. M. Kinesin: what gives? *Cell* **93,** 5–8 (1998).
10. Cai, D., Verhey, K. J. & Meyhofer, E. Tracking single kinesin molecules in the cytoplasm of mammalian cells. *Biophys. J.* **92,** 4137–4144 (2007).
11. Purcell, E. M. Life at Low Reynolds Number. *Am. J. Phys.* **45,** 3–11 (1977).
12. Langevin, P. Sur la théorie du mouvement brownien. *C. R. Math. Acad. Sci. Paris* **146,** 530–533 (1908).
13. Smoluchowski, M. Experimentell nachweisbare, der üblichen Thermodynamik widersprechende Molekularphänomene. *Physik. Z.* **XIII,** 1069–1080 (1912).

14. Feynman, R. P., Leighton, R. B. & Sands, M. L. *The Feynman lectures on physics* (Addison-Wesley, 1963).

15. Astumian, R. D. Thermodynamics and Kinetics of a Brownian Motor. *Science* **276,** 917–922 (1997).

16. Magnasco, M. O. Forced Thermal Ratchets. *Phys. Rev. Lett.* **71,** 1477–1481 (1993).

17. Eshuis, P., Van Der Weele, K., Lohse, D. & Van Der Meer, D. Experimental Realization of a Rotational Ratchet in a Granular Gas. *Phys. Rev. Lett.* **104,** 248001 (2010).

18. Reimann, P. Brownian motors: noisy transport far from equilibrium. *Phys. Rep.* **361,** 57–265 (2002).

19. Lindner, B. & Nicola, E. M. Diffusion in different models of active Brownian motion. *Eur. Phys. J. Special Topics* **157,** 43–52 (2008).

20. Schweitzer, F., Ebeling, W. & Tilch, B. Complex motion of Brownian particles with energy depots. *Phys. Rev. Lett.* **80,** 5044–5047 (1998).

21. Klimontovich, Y. L. *Statistical theory of open systems* (Springer, Nov. 1994).

22. Derényi, I. & Vicsek, T. The kinesin walk: A dynamic model with elastically coupled heads. *Proc. Natl. Acad. Sci. USA* **93,** 6775–6779 (1996).

23. Mateos, J. L. Chaotic transport and current reversal in deterministic ratchets. *Phys. Rev. Lett.* **84,** 258–261 (2000).

24. Linke, H., Downton, M. T. & Zuckermann, M. J. Performance characteristics of Brownian motors. *Chaos* **15,** 026111 (2005).

25. Casdagli, M. Chaos and Deterministic versus Stochastic Non-Linear Modelling. *J. R. Statist. Soc. B* **54,** 303–328 (1992).

26. Chang, C.-H. Ratchet models using driving forces generated by deterministic chaotic maps. *Phys. Rev. E* **66,** 015203 (2002).

27. Strogatz, S. H. *Nonlinear dynamics and Chaos: with applications to physics, biology, chemistry, and engineering* (Westview Press, Jan. 1994).

28. Astumian, R. D. & Bier, M. Fluctuation driven ratchets: Molecular motors. *Phys. Rev. Lett.* **72,** 1766–1769 (1994).

29. Svoboda, K., Schmidt, C. F., Schnapp, B. J. & Block, S. M. Direct observation of kinesin stepping by optical trapping. *Nature* **365,** 721–727 (1993).

30. Żabicki, M., Ebeling, W. & Gudowska-Nowak, E. The thermodynamic cycle of an entropy-driven stepper motor walking hand-over-hand. *Chemical Physics* **375,** 472–478 (2010).

31. Graeter, F., Heider, P., Zangi, R. & Berne, B. J. Dissecting entropic coiling and poor solvent effects in protein collapse. *J. Am. Chem. Soc.* **130,** 11578–11579 (2008).

32. Mateos, J. L. Current reversals in chaotic ratchets. *Acta Phys. Pol. B* **32,** 307–320 (2001).

33. Machura, L., Kostur, M. & Luczka, J. Transport characteristics of molecular motors. *Biosystems* **94,** 253–257 (2008).

34. Yildiz, A., Tomishige, M., Vale, R. D. & Selvin, P. R. Kinesin walks hand-over-hand. *Science* **303,** 676–678 (2004).

35. Lauga, E. & Powers, T. R. The hydrodynamics of swimming microorganisms. *Reports on Progress in Physics* **72,** 096601 (2009).

36. Wang, H. & Oster, G. The Stokes efficiency for molecular motors and its applications. *Europhys. Lett.* **57,** 134–140 (2002).

37. Derényi, I., Bier, M. & Astumian, R. D. Generalized efficiency and its application to microscopic engines. *Phys. Rev. Lett.* **83,** 903 (1999).

38. Nishiyama, M., Higuchi, H. & Yanagida, T. Chemomechanical coupling of the forward and backward steps of single kinesin molecules. *Nat. Cell Biol.* **4,** 790–797 (2002).

39. Bormuth, V., Varga, V., Howard, J. & Schaeffer, E. Protein Friction Limits Diffusive and Directed Movements of Kinesin Motors on Microtubules. *Science* **325,** 870–873 (2009).

40. Selmeczi, D., Mosler, S., Hagedorn, P. H., Larsen, N. B. & Flyvbjerg, H. Cell Motility as Persistent Random Motion: Theories from Experiments. *Biophys. J.* **89,** 912–931 (2008).

41. Strefler, J., Ebeling, W., Gudowska-Nowak, E. & Schimansky-Geier, L. Dynamics of individuals and swarms with shot noise induced by stochastic food supply. *Eur. Phys. J. B* **72,** 597–606 (2009).

42. Yates, C. A. *et al.* Inherent noise can facilitate coherence in collective swarm motion. *Proc. Natl. Acad. Sci. USA* **106,** 5464–5469 (2009).

43. Touya, C., Schwalger, T. & Lindner, B. Relation between cooperative molecular motors and active Brownian particles. *Phys. Rev. E* **83,** 051913 (2011).

44. Endow, S. A. & Higuchi, H. A mutant of the motor protein kinesin that moves in both directions on microtubules. *Nature* **406,** 913–916 (2000).

45. Peters, C. *et al.* Insight into the molecular mechanism of the multitasking kinesin-8 motor. *EMBO J.* **29,** 3437–3447 (2010).

46. Svoboda, K. & Block, S. M. Force and Velocity Measured for Single Kinesin Molecules. *Cell* **77,** 773–784 (1994).

47. Żabicki, M. OpenCL/OpenGL aproach for studying active Brownian motion. *arXiv* **physics.comp-ph** (2011).

48. Rossinelli, D, Bergdorf, M, Cottet, G. & Koumoutsakos, P. GPU accelerated simulations of bluff body flows using vortex particle methods. *J. Comput. Phys.* **229,** 3316–3333 (2010).

49. Stone, J. E., Gohara, D. & Shi, G. OpenCL: A parallel programming standard for heterogeneous computing systems. *Comput. Sci. Eng.* **12,** 66 (2010).

50. Sundholm, E. *Distance Fields Accelerated with OpenCL* PhD thesis (Umeå University, 2010).

51. Zhang, Y. Three phase model of the processive motor protein kinesin. *Biophys. Chem.* **136,** 19–22 (2009).

52. Parker, D., Bryant, Z. & Delp, S. L. Coarse-Grained Structural Modeling of Molecular Motors Using Multibody Dynamics. *Cel. Mol. Bioeng.* **2,** 366–374 (2009).

53. Bowling, A. P., Palmer, A. F. & Wilhelm, L. Contact and Impact in the Multibody Dynamics of Motor Protein Locomotion. *Langmuir* **25,** 12974–12981 (2009).

54. Bier, M. & Cao, F. J. How occasional backstepping can speed up a processive motor protein. *Biosystems* **103,** 355–359 (2011).

55. Apple. *MacBook Air performance information* Apple. 2010.

56. Munshi, A. *The OpenCL Specification* Khronos OpenCL Working Group. 2011.

57. Januszewski, M. & Kostur, M. Accelerating numerical solution of stochastic differential equations with CUDA. *Comput. Phys. Commun.* **181,** 183–188 (2010).

58. Marsaglia, G. Xorshift RNGs. *J. Stat. Soft.* **8,** 1–6 (2003).

59. Langou, J. *et al.* in *SC 2006 Conference* (2007).

60. *Optimizing OpenCL on CPUs* Intel Corporation (2010).