Journal of Computational Science 2 (2011) 324–334

Contents lists available at SciVerse ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Exact stochastic simulations of intra-cellular transport by mechanically coupled molecular motors

Jo A. Helmuth [a,b], Sylvain Reboux [a,b], Ivo F. Sbalzarini [a,b,*]

[a] MOSAIC Group, Institute of Theoretical Computer Science, ETH Zurich, 8092 Zurich, Switzerland
[b] Swiss Institute of Bioinformatics, ETH Zurich, 8092 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 27 April 2011
Received in revised form 25 August 2011
Accepted 31 August 2011
Available online 14 September 2011

Keywords:
Hybrid stochastic–deterministic model
Hybrid continuous–discrete simulation
Stochastic simulation algorithm
Intra-cellular transport
Motor proteins

ABSTRACT

Numerous processes in live cells depend on active, motor-driven transport of cargo and organelles along the filaments of the cytoskeleton. Understanding the resulting dynamics and the underlying biophysical and biochemical processes critically depends on computational models of intra-cellular transport. A number of motor–cargo models have hence been developed to reproduce experimentally observed transport dynamics on various levels of detail. Computer simulations of these models have so far exclusively relied on approximate time-discretization methods. Using a consensus motor–cargo model that unites several existing models from the literature we demonstrate that this simulation approach is not correct. The numerical errors do not vanish even for arbitrarily small time steps, rendering the algorithm inconsistent. We propose a novel exact simulation algorithm for intra-cellular transport models that is also computationally more efficient than the approximate one. Furthermore, we introduce a robust way of analyzing the different time scales in the model dynamics using velocity autocorrelation functions.

1. Introduction

Many intra-cellular cargos such as large molecules, vesicles, organelles, or virus particles are transported along the filaments of the cytoskeleton by motor proteins. Due to their polarization and highly ordered arrangement in the cell, microtubules can support directed and targeted transport in order to, e.g., shuttle cargo between the plasma membrane and the peri-nuclear region. Complex intra-cellular distributions and motion patterns have been observed for various cargos, including virus particles, lipid droplets, vesicles, and small organelles; see for example Refs. [1–4] and references in Ref. [5].

In order to quantify and explain these patterns, computational studies of intra-cellular transport have been invaluable [6]. These simulations require accurate yet computationally efficient models of single motor–cargo complexes. A number of such models have been proposed in which a motor is described by a few discrete chemical states and its discrete location on a microtubule, rather than its full atomic structure. In these models coarse-grained motors are mechanically coupled to a cargo that is typically modeled as a rigid sphere immersed in a highly viscous medium. Elastic motor–cargo links transmit the forces that drive cargo motion. The chemical state-changes of the motors – such as binding to, stepping on, and unbinding from microtubules – are described by Poisson processes with rates that depend on the environment and on the forces exerted on the motor [7–11].

We combine previous models into a "consensus model" that is formally described by sets of coupled stochastic chemical reactions where binding, stepping, and unbinding of different motors are discrete reaction events. Coupling between different motors is provided through the dynamics of the cargo. Between individual reaction events this dynamics is governed by a deterministic law, hence defining a hybrid stochastic–deterministic model. State-of-the-art studies of intracellular transport simulate such models using iterative algorithms with discrete time steps of size Δt [10–13]. Due to this time discretization, the probabilities of the reactions and the position of the cargo exhibit a truncation error that decreases with some power of Δt. This power is called the *convergence order* of the simulation and it depends on the specific time-discretization scheme used. The computational cost of these approximate simulation methods is inversely proportional to the time step Δt. The choice of Δt hence entails a tradeoff between numerical accuracy and computational performance.

Analytical [8,14] and numerical [9–11,13,15,16] studies of transport models have revealed a number of non-trivial effects, such as cascades of correlated unbinding events and fast switching of motion direction, which are closely related to stochastic barrier crossing. In systems exhibiting such behavior, tiny errors in the probabilities of the basic state transitions (i.e., binding, stepping, unbinding) can be amplified to large errors in the crossing rates and time correlations. Due to their non-linear nature, these errors

* Corresponding author at: ETH Zurich, Universitätstr. 6, CH-8092 Zurich, Switzerland.
E-mail address: ivos@ethz.ch (I.F. Sbalzarini).

1877-7503/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.jocs.2011.08.006

may remain significant even for the smallest feasible time step Δt.

The stochastic kinetics of chemical reactions coupled to a deterministic dynamical system can be simulated exactly with a hybrid variant of Gillespie's stochastic simulation algorithm [17–19]. Unlike approximate time-discretization algorithms, this exact method does not involve any discretization error and hence samples cargo trajectories from the correct probability distribution as defined by the transport model.

In this paper we describe how the consensus transport model can be simulated exactly using a novel hybrid continuous–discrete simulation algorithm. The presented exact simulation algorithm does not require choosing a time step. It is parameter free and does, by construction, not entail any tradeoff between efficiency and accuracy. Nevertheless, any quantity computed from Monte Carlo simulations is uncertain due to finite sample size. We compare the computational performance of the presented exact algorithm to that of the approximate one and study the convergence properties of the approximate method. Hereby, the finite-sample Monte Carlo error naturally defines the scale of an acceptable numerical error for the approximate algorithm. We find that some transport properties simulated using the approximate method are systematically wrong, i.e., the error shows no sign of convergence within feasible bounds for the time step. Moreover, the exact algorithm is typically between one and two orders of magnitude faster than the approximate one for biologically relevant motor numbers.

Previous studies have reported high sensitivities of the transport properties to the model parameters within their physiological bounds. Several mobility regimes with qualitatively different behaviors have been identified in the parameter space of intracellular transport models. Refining this picture requires screening the parameter space with high resolution, which critically depends on the availability of computationally efficient and accurate simulation algorithms. The present exact simulation method is computationally efficient and provides full control over the Monte Carlo error.

2. A consensus motor–cargo model

A large number of coarse-grained models of motor–cargo complexes have been proposed in the literature [8–11]. We present here a "consensus model" that unites the concepts, state variables, and the laws for their dynamics that have been used in most related studies.

We consider the situation where a single, rigid cargo is moved by several motors of possibly opposite movement directions. We only resolve the 1D position of the cargo along the microtubule filament; models that resolve the cargo's orientation and its diffusive motion perpendicular to the filament also exist [10]. Motor proteins stochastically step on, bind to, and unbind from microtubules with rates that may depend on the force acting on the motor–filament bond. The physical properties of the motor–cargo links play a key role for the cargo dynamics, as they define the magnitude of forces in the system and their fluctuations in response to stochastic events. Frequently, the motor–cargo links are modeled as linear elastic springs, parameterized by their elasticity κ. In response to the forces transmitted by these springs, the cargo moves in the viscous environment of the cytoplasm. Due to the low Reynolds number, this movement is governed by Stokes' law of drag, parameterized by the drag coefficient γ. Inertial forces and thermal diffusion of the cargo are neglected. The latter is justified since the mechanical energy released by a single motor protein is one order of magnitude larger than the average thermal energy of the cargo [20]. Moreover, thermal fluctuations are already implicitly accounted for in the reaction model where they provide the activation energy for the motor protein reactions, leading to stochastic forces acting on the cargo. Since motors do not directly "see" each other, coupling between different motors is exclusively due to cargo motion. An overview of the model state variables, their relations, and the admissible state transitions is given in Fig. 1.

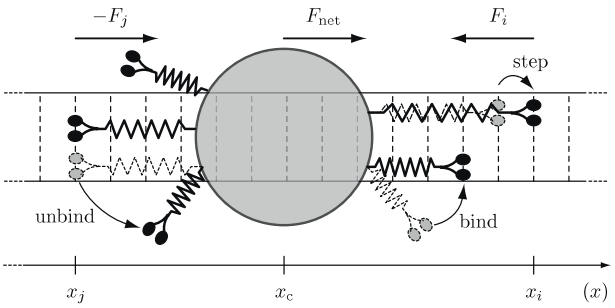

Fig. 1. Illustration of the consensus motor–cargo model. Molecular motors (black) are attached to a rigid cargo (gray disk) through linear elastic springs (zig-zag lines). Motors stochastically bind to, unbind from, and step along discrete binding sites (dashed lines) on a microtubule. The cargo moves in response to the net force F_{net} transmitted by the set of motor–cargo links.

2.1. Time scales

Molecular motors transform chemical energy released by ATP hydrolysis to work in a mechano-chemical cycle [21–23]. In the motor kinesin, this is achieved by a conformational change in the protein structure, pulling the motor forward along the microtubule in discrete steps of 8 nm [7,24]. The time required to complete one such "power stroke" is much shorter than the time between subsequent steps of the same motor and too short for any significant cargo movement to happen meanwhile [7]. Individual steps can therefore be modeled as stochastic events that complete instantaneously at discrete times points. From the perspective of the cargo, these events instantaneously bring the motor–cargo connection to a new state, i.e., the motors jump to the next binding site in their movement direction, elongating the spring. Binding and unbinding of motors are chemical reactions that are also modeled as instantaneous stochastic state changes.

Depending on the properties of the motors and the cargo, the time between individual motor steps and the viscous relaxation time of the cargo position can be of similar order. The cargo dynamics hence has to be explicitly resolved and considered in the stochastic unbinding and stepping processes. Nevertheless, the separation of time scales into instantaneous stochastic events with small rates and slow continuous movement of the cargo allows building a mathematical model from well-known and well-characterized components.

2.2. Mathematical model description

The motion of motors along microtubules is inherently discrete, since the motor–filament interaction responsible for the stable bond is localized to specific parts of the motor and the microtubule. The microtubule consists of α- and β-tubulin dimers, arranged in 13 linear polymer chains called protofilaments. Motors rarely switch between protofilaments. Step lengths of motors are thus integer multiples of the 8 nm distance between two tubulin dimers on the same protofilament. Kinesin almost exclusively makes steps of 8 nm, while cytoplasmic dynein has been observed to also make steps of an integer multiple of that [25,26]. This motivates the use of $L = 8$ nm as the unit of length.

Each motor i is described by the tuple $m_i = (x_i, b_i, d_i)$, where $x_i \in \mathbb{Z}$ is its position, $b_i \in \{\text{bound}, \text{unbound}\}$ its binding state, and the constant $d_i \in \{-1, +1\}$ its movement direction along the filament

(x coordinate). The cargo is described by the position of its center of mass $x_c \in \mathbb{R}$. N is the total number of motors bound to the cargo, which is assumed constant (but not all of them need to be bound to the filament at any given time). This assumption is equivalent to considering binding of motors from an unlimited reservoir of freely diffusing motors to be much faster than unbinding of motors from the cargo [15].

The tension in each motor–cargo link depends on the relative positions of the bound motors with respect to the cargo. According to the linear spring model, a force

$$F_i(t) = \kappa(x_i(t) - x_c(t)) \tag{1}$$

acts on motor i and on the cargo, provided $b_i(t) = \text{bound}$. For unbound motors the force F_i is zero. The dynamic state changes of the motor–cargo complex crucially depend on these forces.

During the time τ after the latest discrete state change (binding, unbinding, or stepping) of any motor, the cargo position evolves according to a deterministic ordinary differential equation:

$$\dot{x}_c(\tau) = \frac{F_{\text{net}}}{\gamma} = \frac{\sum_{i=1}^{N} F_i}{\gamma}, \quad x_c(\tau = 0) = x_c^0. \tag{2}$$

This equation describes the dynamic balance of forces in which the forces exerted on the cargo by the motors are balanced by the viscous drag as parameterized by the drag coefficient γ. While this yields deterministic cargo motion between any two reaction events, the net force F_{net} acting on the cargo changes stochastically at the discrete reaction events of the motor proteins as driven by the thermal fluctuations in the system. The velocity of the cargo responds instantaneously to changes in motor forces since inertia is neglected. The cell cytoplasm is a non-Newtonian fluid in which the drag coefficient γ may depend non-trivially on the radius and the velocity of the cargo. For simplicity, however, it is often assumed that the viscous drag for a given cargo–fluid combination can be parameterized by a single constant γ. Using Eq. (1), the solution of the initial-value problem in Eq. (2) is then given by:

$$x_c(\tau) = (x_c^0 - \langle x \rangle) \exp\left(-\frac{\kappa N_b}{\gamma}\tau\right) + \langle x \rangle, \tag{3}$$

where N_b denotes the number of motors bound to the filament and $\langle x \rangle$ is the bound motors' mean position. The force on a single bound motor i is:

$$F_i(\tau) = \kappa(x_i - \langle x \rangle) - \kappa\left(x_c^0 - \langle x \rangle\right) \exp\left(-\frac{\kappa N_b}{\gamma}\tau\right). \tag{4}$$

The first term on the right-hand side is the force when the cargo is in its equilibrium position. The second term is the transient component of the force, which decays exponentially. Eqs. (3) and (4) are only valid between events of binding, unbinding, or stepping of motors, since these events may change the states of the motors, their average position $\langle x \rangle$, or N_b. In the following, τ is used as the primary time variable, i.e., times are measured relative to the latest event. The real system time t is the cumulative sum of the times between past events.

Binding, unbinding, and stepping of motors are modeled as inhomogeneous Poisson processes. Each event μ is described by a tuple $\mathcal{R}_\mu = (\Psi_\mu, i_\mu)$, where the operator Ψ_μ defines the type of the event. Whenever an event occurs, the state m_{i_μ} of the corresponding motor i_μ is instantaneously changed by applying the operator Ψ_μ. The operators are one of $\{\Psi^{\text{on}}, \Psi^{\text{off}}, \Psi^{\text{st}}\}$:

- For *binding events* (if $\Psi_\mu = \Psi^{\text{on}}$) the associated motor binds to the filament site closest to the current cargo position, hence $\Psi_\mu : b_{i_\mu} \leftarrow \text{bound},\ x_{i_\mu} \leftarrow]x_c[$, where $]\cdot[$ rounds to the nearest integer.
- For *unbinding events* (if $\Psi_\mu = \Psi^{\text{off}}$) the associated motor unbinds from the filament, hence $\Psi_\mu : b_{i_\mu} \leftarrow \text{unbound}$.
- For *stepping events* (if $\Psi_\mu = \Psi^{\text{st}}$) the associated motor moves to the next binding site on the filament in its movement direction, hence $\Psi_\mu : x_{i_\mu} \leftarrow x_{i_\mu} + d_{i_\mu}$.

The rates k_μ at which these events occur generally depend on the forces in the system, which change continuously over time due to cargo motion.

- *Binding events* occur at a constant rate that *does not* depend on the cargo position, the positions of the other motors, or any force in the system [8].

  $$k_\mu = k_{\text{on}}^0. \tag{5}$$

  This reflects the assumption that a motor that is bound to the cargo, but not to the filament, rapidly explores a space on the filament that contains a large number of free binding sites.
- According to Kramer's law of force-assisted barrier crossing [23,27] for chemical reactions, *unbinding events* occur at higher rates when the motor–filament bond is under load, hence:

  $$k_\mu = k_{\text{off}}(F_{i_\mu}(\tau)) = k_{\text{off}}^0 \exp\left(\frac{|F_{i_\mu}(\tau)|}{F_d}\right). \tag{6}$$

  This model is well supported by experimental observations [7]. The unbinding force F_d is a motor-specific parameter that sets the force scale; it does not define a maximum force that bound motors can sustain.
- *Stepping events* occur at rates that reflect the force–velocity relation of a specific motor type. A motor that makes steps of length L at rate k_{st} moves *on average* with a velocity $v = Lk_{\text{st}}$. For kinesin, for example, the velocity (and therefore the rate of stepping) decreases approximately linearly with external load (see Fig. 16.7 in Ref. [21]), hence:

  $$k_\mu = k_{\text{st}}(F_{i_\mu}(\tau)) = \begin{cases} k_{\text{st}}^0 & \text{if} \quad F \le 0 \\ k_{\text{st}}^0(1 - F_{i_\mu}(\tau)/F_s) & \text{if} \quad 0 < F \le F_s \\ 0 & \text{else.} \end{cases} \tag{7}$$

The rates k of impossible events (e.g., binding of an already bound motor) are zero.

3. Exact simulation of the consensus model

We present a hybrid continuous–discrete algorithm to exactly simulate the dynamics of the above motor–cargo model, i.e., the time trajectories $x_c(t)$ and $m_i(t)$ for all motors i. This algorithm is free of discretization errors and does not require choosing a fixed time step.

The position of the cargo at time τ after the latest reaction event is given by Eq. (3). Between reaction events, the rates of the events are not constant since they depend on the dynamic motor forces. These forces, as given by Eq. (4), however, only vary in function of the cargo position. The rates of all stochastic events can therefore be evaluated at any τ.

Formally, the above model of motor–cargo dynamics is a stochastic system of coupled chemical reactions, where binding, stepping, and unbinding events of different motors are different reactions. Simulating the discrete events in the motor–cargo dynamics thus amounts to finding the *next event* and the *next event time*. The exact stochastic simulation algorithm (SSA) is the standard algorithm for simulating such systems [28] as governed by the chemical master equation [29] and driven by the intrinsic thermal fluctuations. In its original formulation, however, SSA requires that rates remain constant between events, whereas in the present system they are continuous functions of time. The extensions to SSA that allow simulating such hybrid systems [19] are given in Section

3.1 below. The complete step-by-step simulation algorithm is then presented in Section 3.2.

3.1. Hybrid stochastic–deterministic cargo dynamics

SSA samples from the joint probability density for the next event μ and the time τ after the last event at which it will happen. This probability density is the exact solution of the chemical master equation [29]. Other than the Monte Carlo sampling error, SSA thus involves no additional discretization errors, which is why the method is termed "exact".

Consider the above model with N^+ plus-end and N^- minus-end directed motors attached to the cargo. In this case, $3N$ different events k_μ, $\mu = 1, \ldots, 3N$, are distinguished: $N^+ + N^- = N$ binding, stepping, and unbinding events. All events except the binding events depend on the corresponding motor forces. The joint probability density for the next event and the time to the next event is a time-dependent exponential distribution [19,30] conditional on the current system state $S = \{m_1, \ldots, m_N, x_c\}$:

$$p(\tau, \mu|S) = k_\mu(S, \tau) \exp\left(-\sum_{\mu=1}^{3N} \int_0^\tau k_\mu(S, \tau')\mathrm{d}\tau' \right) . \tag{8}$$

Without any explicit time dependence of k_μ, finding the time of the next event amounts to sampling from an exponential distribution, while the next event itself can be selected with a probability proportional to its rate. For hybrid system with time-dependent rates, sampling the time increment is less trivial. τ has to be chosen such that

$$\sum_{\mu=1}^{3N} \int_0^\tau k_\mu(S, \tau')\mathrm{d}\tau' = -\log(u_1) , \tag{9}$$

where u_1 is a uniform random number in [0, 1]. This can be done in two ways: The first option is to numerically evaluate the integral from 0 to increasing τ's and to stop integration once the equation is fulfilled. This requires "dense output" integrators with embedded event detection [31,32]. The most commonly used solver for this purpose is "lsodar" as available in the "ODEPACK" software library [33]. The second option is to iteratively solve the equation for τ, for example using Newton's root-finding algorithm. This, however, requires that the integrands and integrals are known in closed form. If they are, the iterative solution of Eq. (9) should be favored as it is computationally more efficient. Once the time τ of the next event has been found, the event itself is selected among all possible events with each event's probability proportional to its rate at time τ. This is done in the standard SSA fashion [28].

For the present model, the linear force–velocity relation and the linear drag force permit finding the integrals of the propensity functions in closed form. For impossible events, the rates (and hence the integrals) are zero. Derivations of the integrals of all others events are given in Appendix B.

3.2. The exact simulation algorithm

The time t, the position x_c of the cargo, and the states m_i of all motors need to be tracked. Starting from an initial system state at time $t = 0$, the simulation algorithm loops through the following steps:

1. Compute the random time τ to the next event. This requires solving Eq. (9) for the upper bound τ of the sum of integrals using Newton's root-finding algorithm. This algorithm solves problems of the type $f(\tau) = 0$. This requires the function value $f(\tau_n)$ and its derivative $f'(\tau_n)$ at a tentative solution τ_n. Here, the function value is
$$f(\tau_n) = \sum_{\mu=1}^{3N} \int_0^{\tau_n} k_\mu(S, \tau')\mathrm{d}\tau' + \log(u_1) . \tag{10}$$
Recall that u_1 is a uniform random number in [0, 1]. The function's derivative is
$$f'(\tau_n) = \sum_{\mu=1}^{3N} k_\mu(S, \tau_n) . \tag{11}$$
Each iteration of Newton's algorithm consists of the following steps:
   (a) For all stepping events, check whether the force acting on the corresponding motor crosses 0 or F_s between 0 and τ_n. If so, determine the crossing times τ_1 and τ_2 using Eqs. (B.8) and (B.9) in Appendix B.
   (b) For each event μ, compute the integrand k_μ and evaluate the integral for the current τ_n. The integral can consist of up to three parts (0 to τ_1, τ_1 to τ_2, and τ_2 to τ_n), depending on whether the force crossed 0, F_s, or both (see Appendix B for details).
   (c) Compute the sums over μ in Eqs. (10) and (11).
   (d) Do a Newton step to determine the new τ_{n+1}:
   $$\tau_{n+1} = \tau_n - \frac{f(\tau_n)}{f'(\tau_n)} . \tag{12}$$
   (e) If τ_{n+1} is not significantly different from τ_n (e.g., $|\tau_{n+1} - \tau_n| < 10^{-9}$ s) stop Newton's algorithm and set $\tau = \tau_{n+1}$; else set $\tau_n = \tau_{n+1}$ and loop back to (a). The algorithm is started with $\tau_n = 0$.
2. Select the next event. This is either a binding, unbinding, or stepping event for one of the motors:
   (a) Compute the rates $k_\mu(S, \tau)$ of all events $\mu = 1, \ldots, 3N$ using $k_\mu = k_{on}^0$ for all binding events, $k_\mu = k_{off}(\tau)$ (Eq. (6)) for all unbinding events, and $k_\mu = k_{st}(\tau)$ (Eq. (7)) for all stepping events.
   (b) Use linear search to select the next event as the smallest μ for which
   $$\sum_{i=1}^{\mu} k_i(S, \tau) \geq u_2 \sum_{i=1}^{3N} k_i(S, \tau) , \tag{13}$$
   where u_2 is a uniform random number in [0, 1].
3. Update time: $t \leftarrow t + \tau$.
4. Update the cargo position: $x_c^0 \leftarrow x_c(t) \leftarrow x_c(\tau)$ where $x_c(\tau)$ is computed from Eq. (3) using the system state prior to the current event.
5. Update the state of motor i_μ corresponding to the event μ by applying the corresponding operator Ψ_μ.
6. If $t \geq t_{max}$ stop; else loop back to Step 1.

Although finding the time to the next event typically requires less than 10 iterations of Newton's algorithm, Step 1 accounts for the vast majority of the computational cost of the present algorithm. A further complication arises from the piece-wise linear force–velocity relation, which requires crossing detection and case distinctions. Relative to Step 1, the computational cost of selecting the next event is negligible. Using more sophisticated event selection schemes is thus not expected to provide any speedup. The overhead introduced by the more complex data structures in these schemes will likely not be amortized for such small N. Moreover, the present time-varying propensity functions prohibit the use of partial-propensity methods [34,35].

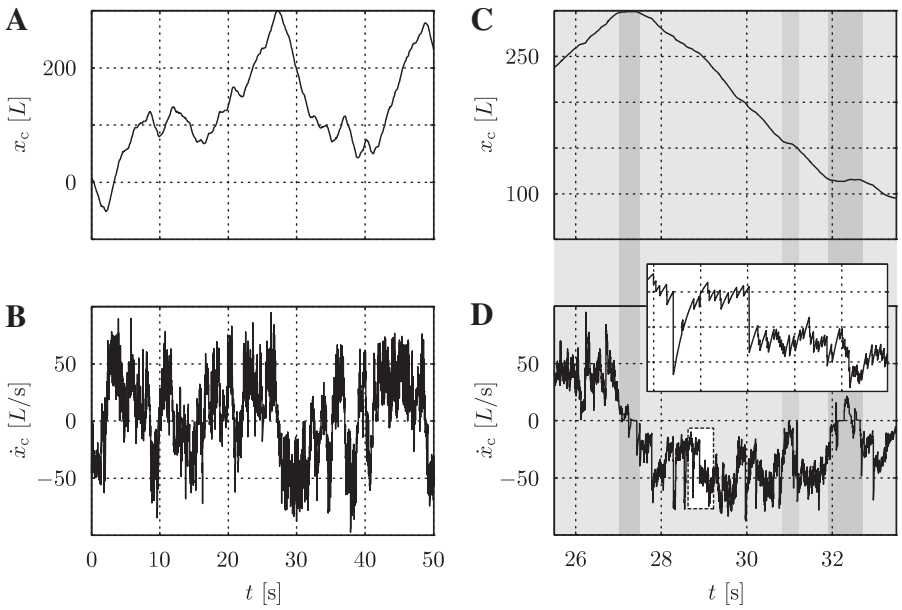

Fig. 2. Multi-scale dynamics of a symmetric tug of war variant of the consensus motor–cargo model. (A and C) Cargo position; (B and D) cargo velocity. (C) and (D) are magnifications of the data shown in (A) and (B). The shading separates phases of uni-directional motion (light gray) and mutual blocking of opposing motors (dark gray). (D) illustrates the different time scales for changes in the velocity. The values of model parameters are given in Tables 1 and 2 ("multi-scale").

4. Benchmarks and results

We characterize the cargo dynamics described by the above consensus model and compare the computational efficiency and accuracy of the exact simulation algorithm with those of the standard approximate algorithm as used in previous studies [10,11]. Details of the standard approximate time-discretization algorithm are described in Appendix A.

4.1. Multi-scale dynamics of cargo motion

Cargo trajectories generated by exact simulations of the present model show complex multi-scale behavior. For suitable parameters, long periods of fast uni-directional motion can develop. Reversal of the motion direction is frequently, but not always, accompanied by a transient deadlock situation in which the motors of opposing movement directions neutralize each other. Fig. 2 shows example time series of the cargo position $x_c(t)$ and its velocity $\dot{x}_c(t)$. On the time scale of tens of seconds, the most significant feature of the trajectories is the emergence of bi-directional saltatory motion (Fig. 2A). Phases during which the cargo moves uni-directionally with an apparently constant velocity of about $50\,\mathrm{L\,s^{-1}}$ typically last 1–5 s (Fig. 2B).

On the time scale of seconds, deadlocks between reversals of the motion direction become apparent (Fig. 2C and D). Phases of directed motion (light gray shading in Fig. 2C and D) end abruptly with sharp drops in the velocity, while fluctuations of the velocity about zero remain during the subsequent deadlocks (dark gray shading). Such deadlocks, however, are not always followed by a reversal of the movement direction, as seen for example at $t = 31$ s and $t = 32$ s. Fig. 2C and D further reveals that the velocity during uninterrupted phases of directed motion is not necessarily constant. Between $t = 27.5$ s and $t = 31$ s the step velocities appear to cluster around $-20\,\mathrm{L\,s^{-1}}$ and $-50\,\mathrm{L\,s^{-1}}$, which is reflected in changes of the slope in Fig. 2C.

The inset in Fig. 2D spans a total of 0.5s and shows the dynamics of the velocity on the time scale of tenths of seconds. Between instantaneous jumps generated by discrete stochastic events, the velocity magnitude decays exponentially. Large jumps originate from unbinding, small jumps from stepping events. The exponential decay is given by the derivative of Eq. (3). Both stepping and unbinding events essentially change $\langle x \rangle$, but the effect of unbinding is typically larger (hence the larger jump). Unbinding events

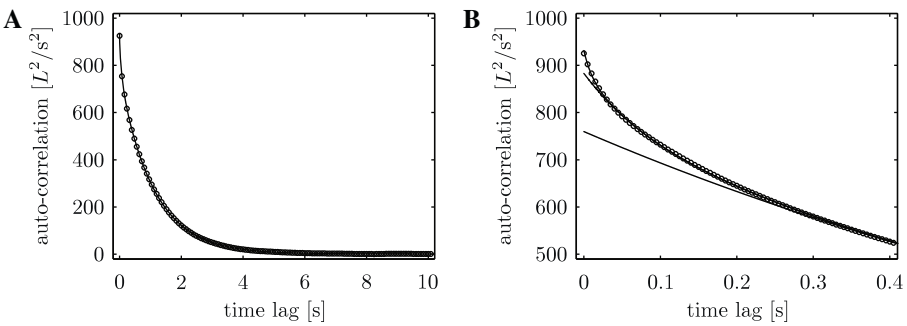

Fig. 3. (A) Empirical cargo velocity autocorrelation function (circles) and its fit (line) with Eq. (15) for the trajectory shown in Fig. 2. The fitted parameters are: $\{a_1, a_2, a_3, a_4\} = \{48.2, 123.3, 758.5, 1.3\}$ and $\{\lambda_1, \lambda_2, \lambda_3\} = \{0.01\,\mathrm{s}, 0.08\,\mathrm{s}, 1.09\,\mathrm{s}\}$. (B) Magnification of the short time-lag region from (A). Upper line: the mixture of all three exponentials is used for fitting; middle line: the fastest time scale is left out; lower line: only the slowest time scale is shown. Note that while some time scales are not displayed, they were still considered during fitting.

additionally change N_b, thus affecting the time constant of the exponential decay between events.

In summary, three time scales can be identified in the cargo trajectories. The slowest time scale is often of highest interest, as it reflects the lengths of phases of directed motion. In order to quantify the persistence of uni-directional motion, we first compute the autocorrelation function of the cargo velocity and then fit it with a mixture of three exponentials and a constant in order to obtain parameters that describe cargo dynamics on the three different time scales (see Section 6.2 and Appendix C for details).

For the parameter settings used in Fig. 2, the empirical autocorrelation (circles in Fig. 3A) decays to practically zero, with a decay time constant on the order of 1 s. This time constant matches well the typical duration of uni-directional motion as shown in Fig. 2A. The slowest time scale is $\lambda_3 = 1.09$ s, which is in good agreement with visual assessment of the sample trajectory. Although the exponential corresponding to the slowest time constant, λ_3, dominates, the other two exponentials and the constant a_4 are still necessary for a robust fit. Fig. 3B shows how successively adding time scales corresponding to faster processes increases the quality of the fit. When all time scales are used, the fit is visually perfect, highlighting the practical value of the proposed autocorrelation analysis. From such a fit, the slowest time scale in variations of the cargo velocity, here represented by λ_3, can be used to robustly quantify correlations of the velocity as caused by persistent uni-directional motion.

4.2. Comparison of the exact and the approximate simulation algorithms

We compare the accuracy and computational cost of the exact simulation algorithm given in Section 3 to those of the standard approximate simulation algorithm as given in Appendix A. The time step Δt is a free parameter of the approximate algorithm, which allows balancing numerical accuracy and computational efficiency. Time-discretization algorithms for deterministic systems converge with some power of Δt, i.e., the numerical error can be made arbitrarily small by reducing Δt. For stochastic systems, however, the numerical error does not vanish for $\Delta t \to 0$ if the magnitude of the intrinsic fluctuations becomes comparable to the mean [36,37]. Moreover, the number of iterations required to simulate a certain amount of real time grows inversely proportional to Δt, which imposes a practical lower bound for Δt and hence for the numerical error. For the present model, Δt is furthermore bounded from above by the condition $\Delta t k_\mu \ll 1$, for all events μ (see Appendix A). For physiologically feasible parameters the largest rates are typically on the order of $100\,\mathrm{s}^{-1}$. Previous studies used $\Delta t = 10^{-5}$ s [10,11], which makes simulations very costly.

A single iteration of the exact algorithm is computationally much more expensive than a single iteration of the approximate algorithm. This is due to the fact that all propensities and their time-integrals need to be computed multiple times until Newton's algorithm converges. In the exact algorithm, however, each iteration yields one event. The approximate algorithm typically requires many iterations per event, as a result of the condition $k_\mu \Delta t \ll 1$. For systems containing few motors, the number of distinct events $3(N^+ + N^-)$ is small and in most iterations no event happens. The exact algorithm therefore outperforms the approximate one by orders of magnitude, as shown in Fig. 4. For biologically relevant motor numbers, the exact algorithm is roughly two orders of magnitude faster than the approximate one with $\Delta t = 10^{-5}$ s. The computational cost of the exact algorithm scales less favorably with the total number of motors in the system ($\approx N^2$ as opposed to $\approx N$; because the total propensity, and hence the number of reactions firing per unit time, is proportional to N and the cost of executing a single reaction is also proportional to N), but the break-even point

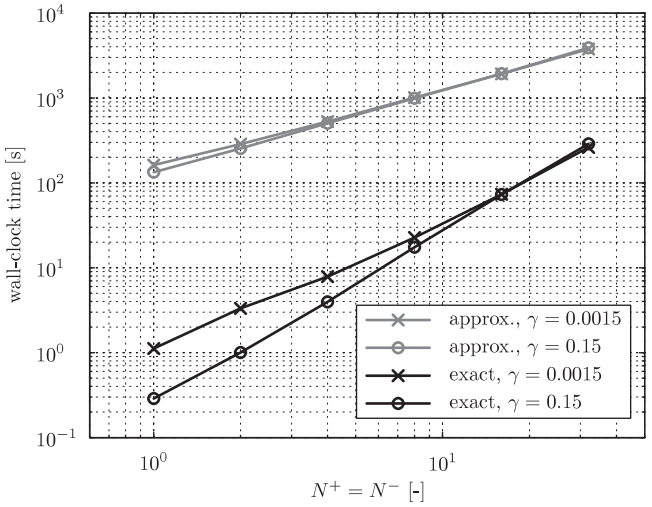

Fig. 4. Computer time needed to simulate $t_{max} = 10^4$ s of the asymmetric tug of war (see Tables 1 and 2) for different motor numbers on a standard laptop with a 2.66 GHz Intel Core 2 Duo processor. Black: exact algorithm; gray: approximate algorithm with $\Delta t = 10^{-5}$ s. Crosses: low drag ($\gamma = 0.0015$ pN s L^{-1}); circles: high drag ($\gamma = 0.15$ pN s L^{-1}).

is at biologically irrelevant motor numbers. This picture is largely unaffected by the details of the system, such as the drag coefficient γ (crosses versus circles in Fig. 4) or the motor types (data not shown).

Increasing Δt improves the performance of the approximate algorithm at the expense of reduced accuracy. We perform a convergence analysis in order to investigate to what extent the approximations of the time-discretization algorithm introduce errors in the simulated cargo dynamics and how they depend on Δt. We use the exact algorithm, which is free of discretization errors, to generate reference results. The significance limit of the errors is given by the Monte Carlo sampling error.

We consider two different systems that have previously been studied [9–11] (see Tables 1 and 2): (i) In the asymmetric tug of war, equal numbers of motors of different type and opposite movement direction compete. Since the opposing motors are of different types, this leads to a net drift of the cargo. (ii) In the symmetric tug of war, both the number and type of motors are identical on both sides, but their movement directions are opposite. On long time scales,

Table 1
Parameters of the model motors.

	k^0_{on} [s^{-1}]	k^0_{off} [s^{-1}]	k^0_{st} [s^{-1}]	F_s [pN]	F_d [pN]
Multi-scale					
Plus motors	1.00	0.20	100	6.00	3.00
Minus motors	1.00	0.20	100	6.00	3.00
Symmetric tug of war [9–11]					
Plus motors	5.00	1.00	125	6.00	3.00
Minus motors	5.00	1.00	125	6.00	3.00
Asymmetric tug of war [9,11]					
Plus motors	1.60	0.26	70	1.10	0.85
Minus motors	1.60	0.78	80	0.45	0.75

Two standard settings (asymmetric/symmetric tug of war) are taken from previous simulation studies [9–11].

Table 2
Model system parameters. Note that $N^{+/-}$ is varied for the timing analysis shown in Fig. 4.

	N^+ [–]	N^- [–]	γ [pN s L^{-1}]	κ [pN L^{-1}]
Multi-scale	5	5	0.24	1.438
Symmetric tug of war	2	2	0.0015 or 0.15	2.4
Asymmetric tug of war	2	2	0.0015 or 0.15	2.4

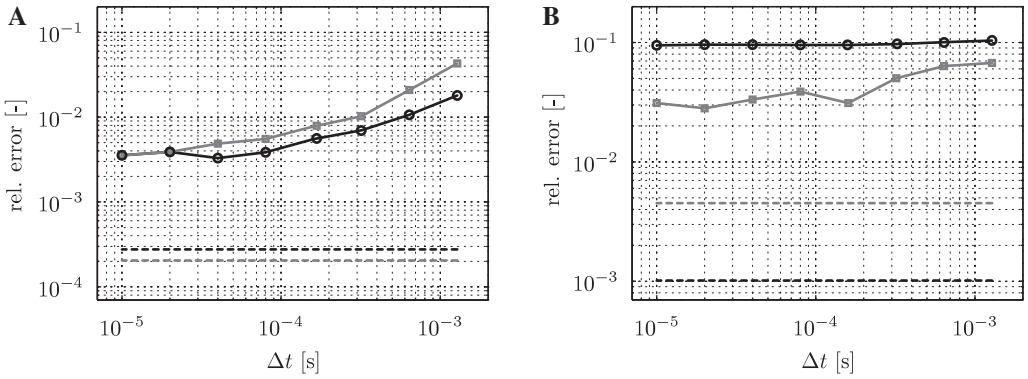

Fig. 5. Relative error versus time-step size of the approximate algorithm for the asymmetric tug of war (see Tables 1 and 2). (A) Relative error of the cargo drift velocity; (B) relative error of the slowest time scale of the velocity autocorrelation. Black lines with circles: $\gamma = 0.15$ pN s L^{-1}; gray lines with squares: $\gamma = 0.0015$ pN s L^{-1}. Dashed lines of the same color indicate the level of the Monte Carlo sampling error for the two cases, defined as the standard deviation of the mean of the $M = 10$ reference simulations.

this causes a random walk of the cargo on the filament, but yields no net drift. We consider two variants of each system: one with high ($\gamma = 0.15$ pN s L^{-1}) and one with low ($\gamma = 0.0015$ pN s L^{-1}) drag. For all systems, we measure the relative errors of the drift velocity, $(x_c(t_{max}) - x_c(0))/t_{max}$, and the slowest time scale of the velocity autocorrelation, $\max_i\{\lambda_i\}$ (see Section 6.2), as a function of Δt. All experiments – both for the approximate and for the exact algorithm – are repeated $M = 10$ times. The relative errors are then defined with respect to the two methods' mean results.

Fig. 5 shows the results for the asymmetric tug of war. The relative error of the drift velocity decreases with Δt (panel A, solid lines). For $\Delta t < 10^{-4}$ s, however, the error remains practically constant at about 4×10^{-3}, which is well above the Monte Carlo error limit (dashed lines). This picture depends only weakly on the drag coefficient. Remarkably, the relative error of the slowest time scale of the velocity autocorrelation shows no sign of convergence (panel B, solid lines). For high drag forces ($\gamma = 0.15$ pN s L^{-1}, black lines with circles) it is larger and constant at about 10 % for all Δt.

Fig. 6 shows the results for the symmetric case. Since in this case the drift velocity should be zero, we report the absolute error. As seen from Fig. 6A, this error is comparable to the Monte Carlo error for all Δt and both γ's and hence insignificant. The slowest time scale of the velocity autocorrelation, however, again significantly deviates from the correct value and the error is largely unaffected by Δt. As in the asymmetric case, higher drag forces worsen the accuracy.

We conclude that the approximate algorithm fails to reproduce the correct cargo dynamics within feasible bounds for the time step Δt. Neither the velocity nor its autocorrelation are free of systematic errors. The magnitude of the errors, however, depends on the specific system studied and the parameters used, such as drag coefficient γ. We furthermore find that the errors are sensitive also to all other model parameters, in particular to the number of motors N^+ and N^- and their binding and unbinding rates k^0_{on} and k^0_{off} (data not shown).

5. Discussion and conclusions

We have introduced an exact hybrid algorithm for simulating the motor-driven motion of intra-cellular cargo. Unlike previously used schemes, the present method does not rely on time-discretization approximations. Instead, it is based on an exact SSA variant for hybrid stochastic–deterministic kinetics that draws samples from the correct probability densities as defined by the transport model.

We have demonstrated the performance of the method on a consensus model of motor–cargo dynamics. This model unites the concepts and features of several state-of-the-art models from the literature in a common description. We have used the time autocorrelation function of the cargo velocity to separate and quantify the three dominant time scales in the model dynamics.

The results have shown that the exact simulation method offers significant advantages over approximate approaches for two reasons: (i) For biologically relevant systems with less than a few hundred motors per cargo and physiological parameter values, the exact simulation method is between one and two orders of magnitude faster than the approximate one. (ii) The numerical approximations entailed in the time-discretization scheme introduce spurious cargo dynamics that remain significant even for the smallest tested time steps. The approximate simulation

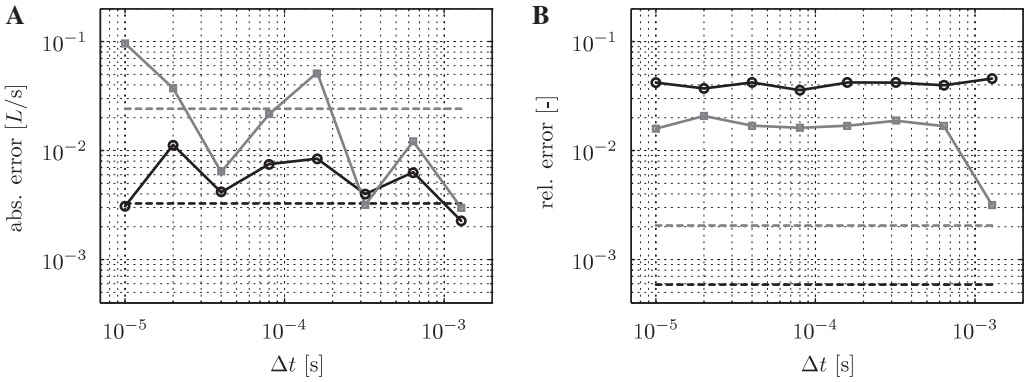

Fig. 6. Error versus time-step size of the approximate algorithm for the symmetric tug of war (see Tables 1 and 2). (A) Absolute error of the drift velocity; (B) Relative error of the slowest time scale of the velocity autocorrelation. Black lines with circles: $\gamma = 0.15$ pN s L^{-1}; gray lines with squares: $\gamma = 0.0015$ pN s L^{-1}. Dashed lines of the same color indicate the level of the Monte Carlo sampling error for the two cases, defined as the standard deviation of the mean of the $M = 10$ reference simulations.

scheme does not converge. Moreover, the residual error non-trivially depends on the model parameters and can hence not be bounded, predicted, or corrected for in practical applications. This indicates that model inference based on fits of experimental observations with approximate simulations may lead to arbitrarily biased parameter estimates. Furthermore, the range of possible model behaviors may not be correctly reproduced, which can lead to spurious rejection of a correct model. For sufficiently well-behaved systems in which the Monte Carlo error is dominating, the approximate time-discretization scheme may outperform the exact algorithm. Without exact reference results, however, it is virtually impossible to predict which systems these will be.

The improved computational performance of the exact algorithm opens new possibilities for detailed parameter screens and compensates for the higher implementation effort. This is also significant for simulations of whole-cell transport patterns. Such simulations often require coarse-graining the dynamics of individual motor–cargo complexes. Calibrating and validating coarse-grained whole-cell models clearly benefits from efficient and exact simulations of single motor–cargo complexes.

Previous studies have reported distributions of run-lengths and velocities of uni-directional motion segments. This requires segmenting the trajectories prior to analysis [4]. Trajectory segmentation, however, needs to be regularized by a prior on segment length or cargo velocities. We therefore proposed a different approach to cargo dynamics analysis that is based on the velocity autocorrelation function. This approach is robust without any prior and yields a few simple statistics, rather than whole empirical distributions, that can easily be interpreted and compared across experiments. In particular, we emphasize the importance of the slowest time scale of the velocity autocorrelation in order to quantify phenomena related to, e.g., reversal of motion direction and cargo unbinding. These processes are of great biological interest since their properties suggest that many non-trivial physical and bio-chemical mechanisms are relevant for cargo transport [3,5,9,16,38–41].

It has previously been argued that the ability of the model to generate unintuitive dynamics is related to the non-trivial coupling between motors. This can, e.g., lead to cascades of unbinding events [9] that often precede reversals or deadlocks. Such sequences of events are "rare" in the sense that they represent excursions far from the system's mean behavior. The frequency of these events thus critically depends on correct reproduction of the system's fluctuations, including higher moments of the intrinsic noise. We believe that the non-converging errors introduced by the approximate time-discretization algorithm are due to its failure to correctly produce higher-order statistics of the model dynamics.

We restricted our analysis to a consensus model of intra-cellular cargo transport. More specialized models have, however, been proposed in the literature. These include models that account for the position of the cargo perpendicular to the filament [10], explicitly include details of the mechano-chemical cycle of the motor proteins [11,13], use more realistic representations of the motor–cargo link, and consider motors that can back-step or have non-linear force–velocity relations [9–11,13]. The exact simulation method presented here can also be used for those extensions of the consensus model. Some of the extensions that are straightforward to implement in the approximate algorithm may, however, require more work in the exact algorithm. Nevertheless, we believe that this effort is justified since the error of the approximate algorithm is not controllable, significant in many cases, and depends on the parameters of the model in an unpredictable way.

We have considered the case where the cargo motion between individual reaction events is governed by a deterministic law. The driving force in this law, however, is stochastic since it is created by motor stepping, binding, and unbinding reactions that are driven by intrinsic thermal fluctuations as governed by the chemical master equation [29]. It is nonetheless conceivable that in some applications also the cargo dynamics *between* reaction events has to be modeled stochastically. The governing equation for the cargo position between any two reaction events (Eq. (2)) then becomes a stochastic differential equation, and the propensity functions k_μ and their integrals can not be written in closed form any more. Using exact SSA to sample the next reaction then requires numerically integrating the stochastic differential equation until the integral value crosses the given threshold for the first time. This is related to determining the first passage time of a random walk, and techniques such as those presented by Buchmann might need to be adopted [42]. This would result in a fully stochastic, yet still hybrid continuous–discrete simulation algorithm.

A C++ implementation of the present algorithm is available free of charge and as open source from the web page of the authors.

6. Materials and methods

6.1. Simulation protocols

Simulations are initialized with $x_c^0 = 0$ and all motors bound at $x = 0$. The system is equilibrated in a burn-in phase of 10^3 s prior to trajectory recording. Simulations run $t_{max} = 10^6$ s and the cargo position is sampled and written to a file every 10^{-3} s, except for timing measurement runs, which do not produce any disk output. Both simulation algorithms are implemented in C++ and use identical code wherever possible. Simulations were run on the "brutus" cluster of ETH Zurich, a 10 000-core AMD Opteron system. Velocity autocorrelations are computed offline using the MATLAB function `xcorr` and fitted with the exponential model using the MATLAB function `fmincon`.

6.2. Model-free analysis of cargo dynamics

We use the *autocorrelation function* of the cargo velocity to describe the stochastic multi-scale dynamics of the motor–cargo complex. Given a finite trajectory $x_c(t)$, sampled at discrete times $t_i = i\Delta t_s$, the autocorrelation function $R_{\dot{x}_c\dot{x}_c}(\tau) = \mathbb{E}\{\dot{x}_c(t)\dot{x}_c(t+\tau)\}$ of the velocity $\dot{x}_c(t)$ is approximated by the time average:

$$\widehat{R_{\dot{x}_c\dot{x}_c}}(j) = (\dot{x}_c \star \dot{x}_c)(j) = \sum_i \dot{x}_c(t_i)\dot{x}_c(t_{i+j}), \tag{14}$$

where the sum is taken over all possible i. $j\Delta t_s$ is the time lag at which the autocorrelation is evaluated. In order to obtain summary statistics for entire cargo trajectories we fit the sample autocorrelation function with a sum of up to three exponential functions plus a constant (see Appendix C for a justification of this model):

$$R_m(j) = \sum_{i=1}^{3} a_i \exp\left(-\frac{j\Delta t_s}{\lambda_i}\right) + a_4 . \tag{15}$$

Each exponential captures the dynamics of a different time scale. The three time scales in the model correspond to the dynamics of motor stepping, motor unbinding, and reversal of the cargo movement direction. The fitting can be stabilized by formulating constraints on the parameters based on prior knowledge.

Acknowledgements

We thank Christoph Burckhardt (Harvard University) and Mattia Gazzola (CSE-Lab, ETH Zurich) for many useful and inspiring discussions. We furthermore thank Janick Cardinale, Christian Müller, Grégory Paul, Rajesh Ramaswamy, and Birte Schrader (all MOSAIC Group, ETH Zurich) for their comments and suggestions. JAH was

financed by the ETH Research Commission under grant TH-1007-1. SR was financed by a grant from the Swiss SystemsX.ch initiative, grant LipidX, evaluated by the Swiss National Science Foundation.

Appendix A. Approximate time-discretization algorithm

The deterministic cargo dynamics (described by Eq. (2)) and the stochastic dynamics of individuals motors (described by Eqs. (5)–(7)) can be approximately simulated using a forward time-integration scheme with fixed time step Δt. Starting from an initial system state $S = S^0$ at $t = 0$ the algorithm cycles through the following steps:

1. For each possible event μ, compute the probability $P(\mu|S, t)$ that it happens between $[t; t+\Delta t)$: With $k_\mu(t+\tau) \approx k_\mu(t)$ for $\tau \in [0; \Delta t)$ and $k_\mu(t)\,\Delta t \ll 1$:

$$P(\mu|S, t) \approx k_\mu(t)\,\Delta t\,, \tag{A.1}$$

   where $k_\mu = k^0_{\text{on}}$ for all binding events, $k_\mu = k_{\text{off}}(\tau_n)$ (Eq. (6)) for all unbinding events, and $k_\mu = k_{\text{st}}(\tau_n)$ (Eq. (7)) for all stepping events. The approximation in Eq. (A.1) follows from a Taylor expansion of the cumulative distribution function of the exponentially distributed waiting times to the next event.
2. Update cargo position: $x_c \leftarrow x_c + \dot{x}_c(t)\,\Delta t$, where $\dot{x}_c(t)$ is computed using Eq. (2).
3. Update time: $t \leftarrow t + \Delta t$.
4. For each possible event μ, draw a uniform random number $u_\mu \in [0; 1]$. If $u_\mu \le P(\mu|S, t)$, update the system state S by applying operator Ψ_μ to motor i_μ.
5. If $t \ge t_{\max}$ stop; else loop back to 1.

Except for model-specific adaptations, this algorithm is identical to the one used in Refs. [10,11]. Due to the lack of time-resolution below Δt, all events are executed quasi-simultaneously at $t + \Delta t$. The algorithm hence does not explicitly resolve potential conflicts between events, e.g., if a motor is found to unbind *and* step during a single iteration. The time step limit $k_\mu(t)\,\Delta t \ll 1$ is imposed in order to render such situations unlikely.

Appendix B. Integrals of the propensity functions

We report below the analytical solutions of the integrals of the propensity functions of possible events, i.e., events with a non-zero propensity. These integrals are required to evaluate the left-hand side of Eq. (9).

- *Binding event*: k_{on} does not depend on the cargo position x_c, hence:

$$\int_0^\tau k_{\text{on}}(S, \tau')\mathrm{d}\tau' = k^0_{\text{on}}\,\tau. \tag{B.1}$$

- *Unbinding event*: The absolute value in Eq. (6) requires special attention. Assume the unbinding event concerns motor *i*. First, one has to check whether the force F_i changes sign between 0 and τ. If so, let τ_1 denote the time of the sign change. If no sign change takes place, $\tau_1 = \tau$. The integral is then split as:

$$\int_0^\tau k_{\text{off}}(S, \tau')\mathrm{d}\tau' = \int_0^{\tau_1} k_{\text{off}}(S, \tau')\mathrm{d}\tau' + \int_{\tau_1}^\tau k_{\text{off}}(S, \tau')\mathrm{d}\tau'\,. \tag{B.2}$$

The variable $s = \text{sign}(F_i(0))$ and the movement direction $d = d_i$ of the concerned motor i are introduced, and Eqs. (6) and (4) are inserted:

$$\int_0^{\tau_1} k_{\text{off}}(S, \tau')\mathrm{d}\tau' = k^0_{\text{off}} \int_0^{\tau_1} \exp\left[\frac{\kappa d s}{F_{\text{d}}}(x_i - \langle x\rangle) - \frac{\kappa d s}{F_{\text{d}}}(x^0_{\text{c}} - \langle x\rangle)\exp\left(-\frac{\kappa N_{\text{b}}}{\gamma}\tau'\right)\right]\mathrm{d}\tau'\,.$$

With $c_1 = k^0_{\text{off}}$, $c_2 = \frac{\kappa d s}{F_{\text{d}}}(x_i - \langle x\rangle)$, $c_3 = \frac{\kappa d s}{F_{\text{d}}}(x^0_c - \langle x\rangle)$, and $c_4 = \kappa N_{\text{b}}/\gamma$, this can be written more compactly as:

$$\int_0^{\tau_1} k_{\text{off}}(S, \tau')\mathrm{d}\tau' = c_1 \int_0^{\tau'_1} \exp[c_2 - c_3\exp(-c_4\tau')]\mathrm{d}\tau'\,. \tag{B.3}$$

If $c_3 \neq 0$ (true whenever the cargo is not in its equilibrium position, i.e., it is still moving), then:

$$\int_0^{\tau_1} k_{\text{off}}(S, \tau')\mathrm{d}\tau' = \frac{c_1\exp(c_2)}{c_4}(-\text{Ei}(-c_3\exp[-c_4\tau_1]) + \text{Ei}(-c_3)), \tag{B.4}$$

where $\text{Ei}(\cdot)$ denotes the exponential integral function. If $c_3 = 0$, the integral reduces to the simpler form:

$$\int_0^{\tau_1} k_{\text{off}}(S, \tau')\mathrm{d}\tau' = c_1 + \exp(c_2)\tau_1\,. \tag{B.5}$$

The second half of the integral (from τ_1 to τ) is solved analogously, but using $s = -\text{sign}(F(0))$ in c_2 and c_3, leading to:

$$\int_{\tau_1}^{\tau} k_{\text{off}}(S, \tau')\mathrm{d}\tau' = \frac{c_1\exp(c_2)}{c_4}(-\text{Ei}(-c_3\exp[-c_4\tau]) + \text{Ei}(-c_3\exp[-c_4\tau_1]))$$

for $c_3 \neq 0$. The simpler result for $c_3 = 0$ is:

$$\int_{\tau_1}^{\tau} k_{\text{off}}(S, \tau')\mathrm{d}\tau' = c_1 + \exp(c_2)(\tau - \tau_1)\,. \tag{B.6}$$

The function $\text{Ei}(\cdot)$ is, e.g., implemented in the GNU Scientific Library (GSL) [43] and can be conveniently evaluated in a computer program. Some care must be taken when $-c_3\exp(-c_4\tau)$ is algebraically close to zero. In these cases, it may evaluate to zero in a computer program, which is not a valid argument for $\text{Ei}(\cdot)$. Furthermore, evaluating $\text{Ei}(\cdot)$ for small arguments is numerically inaccurate. We therefore use a series expansion of $\text{Ei}(z)$, $z = -c_3\exp(-c_4\tau)$:

$$\text{Ei}(z) = \log(|z|) + \gamma_{\text{EM}} + z + \frac{z^2}{4} + \frac{z^3}{18} + \mathcal{O}(z^4) \tag{B.7}$$

$$= \log(|c_3|) - c_4\tau + \gamma_{\text{EM}} + z + \frac{z^2}{4} + \frac{z^3}{18} + \mathcal{O}(z^4)\,, z \ll 1, \tag{B.7}$$

whenever $z < 10^{-10}$. The constant $\gamma_{\text{EM}} = 0.577\ldots$ is the Euler-Mascheroni number.

- *Stepping event*: Due to the piece-wise definition of the force–velocity relation it is necessary to check whether the concerned motor i stalls ($dF_i > F_s$) or is working at full speed ($dF_i < 0$). Let τ_1 denote the time at which dF_i crosses 0 from below and τ_2 the time at which dF_i crosses F_s from below. The cases when dF_i crosses F_s or 0 from above are analogous, but the bounds ofthe

integrals have to be exchanged accordingly. The integral is split into three parts:

$$\int_0^{\tau} k_{\mathrm{st}}(S,\tau')\mathrm{d}\tau' = \int_0^{\tau_1} 0\,\mathrm{d}\tau' + \int_{\tau_1}^{\tau_2} k_{\mathrm{st}}^0\left[1-\frac{d\kappa}{F_{\mathrm{s}}}(x_i-\langle x\rangle)\frac{d\kappa}{F_{\mathrm{s}}}(x_{\mathrm{c}}^0-\langle x\rangle)\exp\left(-\frac{\kappa N_{\mathrm{b}}}{\gamma}\tau'\right)\right]\mathrm{d}\tau' + \int_{\tau_2}^{\tau} k_{\mathrm{st}}^0\mathrm{d}\tau'.$$

Using $c_1 = k_{\mathrm{st}}^0$, $c_2 = 1-(d\kappa/F_{\mathrm{s}})(x_i-\langle x\rangle)$, $c_3 = (d\kappa/F_{\mathrm{s}})(x_{\mathrm{c}}^0-\langle x\rangle)$, and $c_4 = (\kappa N_{\mathrm{b}})/\gamma$ the solution is found as:

$$\int_0^{\tau} k_{\mathrm{st}}(S,\tau')\mathrm{d}\tau' = c_1\left(c_2(\tau_2-\tau_1)-\frac{c_3}{c_4}(\exp(-c_4\tau_2)-\exp(-c_4\tau_1))\right)+c_1(\tau_2-\tau_1).$$

The times τ_1 and τ_2 at which the force crosses 0 and F_{s}, respectively, are found by solving Eq. (4) as:

$$\tau_1 = \log\left(\frac{(x_i-\langle x\rangle)}{(x_{\mathrm{c}}^0-\langle x\rangle)}\right)\frac{\gamma}{\kappa N_{\mathrm{b}}}, \tag{B.8}$$

$$\tau_2 = \log\left(\frac{\kappa(x_i-\langle x\rangle)-F_{\mathrm{s}}}{\kappa(x_{\mathrm{c}}^0-\langle x\rangle)}\right)\frac{\gamma}{\kappa N_{\mathrm{b}}}. \tag{B.9}$$

Note that the times τ_1 and τ_2 only exist if the corresponding crossings indeed take place. Because the forces change monotonically between events, this can easily be tested by inspecting $F(0)$ and $F(\tau)$ from Eq. (4).

The calculations above are only valid for the specific force–velocity relation used here (Eq. (7)). For other force–velocity relations they must be adapted and it may not always be possible to express all integrals in closed form, in which case they have to be computed numerically.

Appendix C. Autocorrelation function of the cargo velocity

Unless the cargo undergoes a random walk, the velocities at neighboring time points are correlated with a correlation time that decays slower the more persistent the motion is. This motivates the use of the autocorrelation function of the cargo velocity to quantify the persistence of uni-directional motion [44]. For the present model, a plot of the autocorrelation versus time lag shows an exponential decay (see Fig. 3). The time constant of the decay corresponds to the time scale over which the time-averaged velocity of the motor–cargo complex significantly changes, e.g., due to reversal of the movement direction. Additional time scales may be present in the autocorrelation function. These correspond to the unbinding and stepping processes, which cause sharp jumps in the velocity followed by an exponential decay. The rate and height of the jumps, and time constant of the subsequent decay, are reflected in the faster modes of the autocorrelation function.

The exponential model used here for the velocity autocorrelation directly derives from the discrete stochastic events in the model. Consider a stationary Poisson process with exponentially distributed waiting times $\tau_i = t_i - t_{i-1}$ that triggers exponentially decaying jumps in the velocity for each event, i.e.:

$$\dot{x}(t) = \sum_i \Delta\dot{x}\,\exp\left(-\frac{t}{\lambda}\right)H(t-t_i), \quad \lambda > 0. \tag{C.1}$$

$H(\cdot)$ denotes the Heaviside step function. It can be shown that the autocorrelation function of $\dot{x}$ is given by:

$$R_{\dot{x}\dot{x}}(\tau) = \lambda^2\nu^2(\Delta\dot{x})^2 + \frac{\nu(\lambda\Delta\dot{x})^2}{2}\exp\left(-\frac{|\tau|}{\lambda}\right), \tag{C.2}$$

with $\nu > 0$ the intensity of the Poisson process. Eq. (C.2) admits two observations: (i) The autocorrelation decays exponentially with a time constant that is identical to that of the decay of the velocity itself; (ii) The autocorrelation does not drop to zero.

The autocorrelation of the sum of two (or more) uncorrelated signals is the sum of the autocorrelations of these signals. This motivates fitting empirical velocity autocorrelation functions with a sum of exponentials as discussed in Section 6. Also the constant term is necessary as can be seen from Eq. (C.2). Even though the assumption that the different types of events (stepping, binding, and unbinding) are uncorrelated and occur at constant rates does not hold for the present model, the velocity autocorrelation is still well described by a sum of exponentials. Interpreting the values of the fitted exponents and pre-factors on the basis of Eq. (C.2), however, may be misleading.

References

[1] M.A. Welte, Bidirectional transport along microtubules, Curr. Biol. 14 (13) (2004) 525–537, doi:10.1016/j.cub.2004.06.045.
[2] R. Mallik, D. Petrov, S.A. Lex, S.J. King, S.P. Gross, Building complexity: an in vitro study of cytoplasmic dynein with in vivo implications, Curr. Biol. 15 (23) (2005) 2075–2085.
[3] V. Levi, A.S. Serpinskaya, E. Gratton, V. Gelfand, Organelle transport along microtubules in xenopus melanophores: evidence for cooperation between multiple motors, Biophys. J. 90 (1) (2006) 318–327, doi:10.1529/biophysj.105.067843.
[4] J.A. Helmuth, C.J. Burckhardt, P. Koumoutsakos, U.F. Greber, I.F. Sbalzarini, A novel supervised trajectory segmentation algorithm identifies distinct types of human adenovirus motion in host cells, J. Struct. Biol. 159 (3) (2007) 347–358.
[5] S.P. Gross, Hither and yon: a review of bi-directional microtubule-based transport, Phys. Biol. 1 (1-2) (2004) R1–R11, doi:10.1088/1478-3967/1/2/R01.
[6] F. Nédélec, Computer simulations reveal motor properties generating stable antiparallel microtubule interactions, J. Cell Biol. 158 (6) (2002) 1005–1015.
[7] N. Carter, R. Cross, Mechanics of the kinesin step, Nature 435 (7040) (2005) 308–312, doi:10.1038/nature03528.
[8] S. Klumpp, R. Lipowsky, Cooperative cargo transport by several molecular motors, Proc. Natl. Acad. Sci. U.S.A. 102 (48) (2005) 17284–17289., doi:10.1073/pnas.0507363102.
[9] M.J.I. Müller, S. Klumpp, R. Lipowsky, Tug-of-war as a cooperative mechanism for bidirectional cargo transport by molecular motors, Proc. Natl. Acad. Sci. U.S.A. 105 (12) (2008) 4609–4614.
[10] C.B. Korn, S. Klumpp, R. Lipowsky, U.S. Schwarz, Stochastic simulations of cargo transport by processive molecular motors, J. Chem. Phys. 131 (2009) 245107, doi:10.1063/1.3279305.
[11] A. Kunwar, A. Mogilner, Robust transport by multiple motors with nonlinear force-velocity relations and stochastic load sharing, Phys. Biol. 7 (1) (2010) 016012, doi:10.1088/1478-3975/7/1/016012.
[12] F. Nedelec, D. Foethke, Collective Langevin dynamics of flexible cytoskeletal fibers, New J. Phys. 9 (2007) 427.
[13] A. Kunwar, M. Vershinin, J. Xu, S.P. Gross, Stepping, strain gating, and an unexpected force–velocity curve for multiple-motor-based transport, Curr. Biol. 18 (16) (2008) 1173–1183, doi:10.1016/j.cub.2008.07.027.
[14] J. Beeg, S. Klumpp, R. Dimova, R.S. Gracia, E. Unger, R. Lipowsky, Transport of beads by several kinesin motors, Biophys. J. 94 (2) (2008) 532–541, doi:10.1529/biophysj.106.097881.
[15] M. Gazzola, C.J. Burckhardt, B. Bayati, M. Engelke, U.F. Greber, P. Koumoutsakos, A stochastic model for microtubule motors describes the in vivo cytoplasmic transport of human adenovirus, PLoS Comp. Biol. 5 (12) (2009) e1000623, doi:10.1371/journal.pcbi.1000623.
[16] V. Soppina, A.K. Rai, A.J. Ramaiya, P. Barak, R. Mallik, Tug-of-war between dissimilar teams of microtubule motors regulates transport and fission of endosomes, Proc. Natl. Acad. Sci. U.S.A. 106 (46) (2009) 19381–19386, doi:10.1073/pnas.0906524106.
[17] A. Alfonsi, E. Cancès, G. Turinici, B. Di Ventura, W. Huisinga, Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems, in: in:E. Cancès, J.-F. Gerbeau (Eds.), ESAIM: Proceedings, vol. 14, 2005, pp. 1–13.
[18] H. Salis, Y. Kaznessis, Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions, J. Chem. Phys. 122 (5) (2005) 054103, doi:10.1063/1.1835951.
[19] D.T. Gillespie, Stochastic simulation of chemical kinetics, Ann. Rev. Phys. Chem. 58 (2007) 35–55.
[20] G. Holzwarth, K. Bonin, D.B. Hill, Forces required of kinesin during processive transport through cytoplasm, Biophys. J. 82 (4) (2002) 1784–1790.

[21] J. Howard, in: Mechanics of Motor Proteins and the Cytoskeleton, Sinauer Associates, Sunderland, 2001.
[22] D. Keller, C. Bustamante, The mechanochemistry of molecular motors, Biophys. J. 78 (2) (2000) 541–556.
[23] C. Bustamante, Y.R. Chemla, N.R. Forde, D. Izhaky, Mechanical processes in biochemistry, Annu. Rev. Biochem. 73 (2004) 705–748, doi:10.1146/annurev.biochem.72.121801.161542.
[24] S. Rice, A.W. Lin, D. Safer, C.L. Hart, N. Naber, B.O. Carragher, S.M. Cain, E. Pechatnikova, E.M. Wilson-Kubalek, M. Whittaker, E. Pate, R. Cooke, E.W. Taylor, R.A. Milligan, R.D. Vale, A structural change in the kinesin motor protein that drives motility, Nature 402 (6763) (1999) 778–784.
[25] R. Mallik, B.C. Carter, S.A. Lex, S.J. King, S.P. Gross, Cytoplasmic dynein functions as a gear in response to load, Nature 427 (6975) (2004) 649–652, doi:10.1038/nature02293.
[26] J.R. Kardon, R.D. Vale, Regulators of the cytoplasmic dynein motor, Nat. Rev. Mol. Cell Bio. 10 (12) (2009) 854–865, doi:10.1038/nrm2804.
[27] H.A. Kramers, Brownian motion in a field of force and the diffusion model of chemical reactions, Physica 7 (1940) 284–304.
[28] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (25) (1977) 2340–2361.
[29] D.T. Gillespie, A rigorous derivation of the chemical master equation, Physica A 188 (1-3) (1992) 404–425.
[30] M.A. Gibson, J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, J. Phys. Chem. A 104 (2000) 1876–1889.
[31] L.F. Shampine, I. Gladwell, R.W. Brankin, Reliable solution of special event location problems for ODEs, ACM Trans. Math. Softw. 17 (1) (1991) 11–25.
[32] G. Grabner, A. Kecskeméthy, An integrated Runge-Kutta root finding method for reliable collision detection in multibody systems, Multibody System Dynamics 14 (2005) 301–316.
[33] A.C. Hindmarsh, L.R. Petzold, Algorithms and software for ordinary differential equations and differential- algebraic equations, part II: higher-order methods and software packages, Comput. Phys. 9 (2) (1995) 148–155.
[34] R. Ramaswamy, N. González-Segredo, I.F. Sbalzarini, A new class of highly efficient exact stochastic simulation algorithms for chemical reaction networks, J. Chem. Phys. 130 (24) (2009) 244104.
[35] R. Ramaswamy, I.F. Sbalzarini, A partial-propensity variant of the composition-rejection stochastic simulation algorithm for chemical reaction networks, J. Chem. Phys. 132 (4) (2010) 044102.
[36] D.T. Gillespie, The multivariate Langevin and Fokker–Planck equations, Am. J. Phys. 64 (10) (1996) 1246–1257.
[37] N.G. van Kampen, in: Stochastic Processes in Physics and Chemistry, 3rd ed., North Holland, 2007.
[38] R. Mallik, S.P. Gross, Molecular motors: strategies to get along, Curr. Biol. 14 (22) (2004) 971–982, doi:10.1016/j.cub.2004.10.046.
[39] R. Mallik, S.P. Gross, Molecular motors as cargo transporters in the cell – the good, the bad and the ugly, Physica A 372 (1) (2006) 65–69, doi:10.1016/j.physa.2006.05.017.
[40] Y. Shtridelman, T. Cahyuti, B. Townsend, D. DeWitt, J.C. Macosko, Force–velocity curves of motor proteins cooperating in vivo, Cell Biochem. Biophys. 52 (1) (2008) 19–29, doi:10.1007/s12013-008-9021-8.
[41] A. Gennerich, R.D. Vale, Walking the walk: how kinesin and dynein coordinate their steps, Curr. Opin. Cell Biol. 21 (1) (2009) 59–67., doi:10.1016/j.ceb.2008.12.002.
[42] F.M. Buchmann, Simulation of stopped diffusions, J.Comput. Phys. 202 (2005) 446–462.
[43] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, GNU Scientific Library Reference Manual, 3rd ed., Network Theory Ltd, Bristol, UK, 2009.
[44] D.B. Hill, M.J. Plaza, K. Bonin, G. Holzwarth, Fast vesicle transport in PC12 neurites: velocities and forces, Eur. Biophys. J. Biophy. 33 (7) (2004) 623–632, doi:10.1007/s00249-004-0403-6.

Jo A. Helmuth is a postdoctoral researcher with the MOSAIC Group at ETH Zurich, Switzerland. He obtained his Ph.D. in Computer Science from ETH Zurich in 2010 and his Diploma in Mechanical Engineering from ETH Zurich in 2005, majoring in nanotechnology and computational science. His research interests are in developing computational methods for image analysis and simulation of intra-cellular transport processes in biology.

Sylvain Reboux is a postdoctoral researcher with the MOSAIC Group at ETH Zurich, Switzerland. He received his Ph.D. from the School of Mathematical Sciences at the University of Nottingham, UK, in 2008. He holds a M.Sc. degree in Fluid Dynamics from the Université Pierre et Marie Curie, Paris, which he completed in 2005. His research interests are in adaptive and hybrid simulation algorithms for complex systems and in parallel particle methods.

Ivo F. Sbalzarini is Assistant Professor of Computational Science at ETH Zurich, Switzerland, and founder and head of the MOSAIC Group. He received his Ph.D. in Computer Science and his Diploma in Mechanical Engineering from ETH Zurich in 2006 and 2002, respectively. His research interests include multi-resolution adaptive particle methods, hybrid and stochastic simulation algorithms, parallel high-performance computing, and image analysis as applied to complex real-world systems, currently mostly from biology.