

Simulating dynein's powerstroke using Brownian dynamics

Elliott Capek

May 3, 2017

CONTENTS

1 Chapter: Introduction	4
1.1 Motivation	4
1.2 What is dynein?	5
1.2.1 Basics	5
1.2.2 Why do cells need motor transport?	6
1.2.3 What role does dynein play in the cell?	7
1.2.4 Dynein structure	7
1.2.5 How does it behave?	7
1.2.6 Why is dynein interesting?	7
1.3 Dynein Stepping Background	8
1.3.1 ATP cycle influences conformational state	9
1.3.2 Microtubule and nucleotide states are coupled	9
1.3.3 Stepping pattern	10
1.4 Dynein models	11
1.4.1 Cianfrocco Model	11
1.5 This work	12
1.5.1 Necessary and sufficient criteria for motion	12
1.5.2 Goals	13
2 Chapter: Theory	13
2.1 Model features	13
2.1.1 Spherical domains connected by massless rods	13
2.1.2 Rigid rods	13
2.1.3 Angular springs	14
2.1.4 Two-state model	14
2.1.5 Transitioning between states	14
2.1.6 Brownian dynamics	15
2.1.7 Binding to microtubule	15
2.1.8 2D model	15
2.2 Mathematical models	15
2.2.1 Prestroke onebound model	16
2.2.2 Poststroke bothbound model	18
2.3 Data interpretation	20
2.3.1 Converting experimental rates to preexponential factors A_b and A_{ub}	20
2.3.2 Literature values for experimental binding rates	21
2.3.3 Calculating one/bothbound times of experimental dynein	21

3 Chapter: Methods	22
3.1 Simulation	22
3.1.1 Time evolution	23
3.1.2 Transitioning between states	23
3.1.3 Corrections	24
3.1.4 Forces	25
3.2 Verifying the model	25
3.2.1 Conformational tests	25
3.2.2 Energy conservation	26
3.2.3 Equipartition theorem	27
3.3 Parameter fitting	27
3.3.1 Fitting rate constants	28
3.3.2 Fitting spring constants	28
4 Chapter: Results & Discussion	29
4.1 Validating model	29
4.2 Model achieves processivity	29
4.3 Model step size is highly variant	31
5 Conclusion	32
5.1 Findings	32
5.2 Further work	33
6 Appendix	35
6.0.1 Glossary	35
6.1 Motion equations	35
6.1.1 Onebound equations	35
6.1.2 Bothbound equations	35
7 Questions for Tate/Nicole	36
8 Questions for David	36
9 Todo for project	36

LIST OF FIGURES

1 Artist's rendition of dynein bound to a microtubule.	5
2 Passive diffusion vs motor transport rate	6
3 Dynein localization to axon growth cones	8
4 Artist's rendition of motor-MT complex	9
5 Cryo-EM microscopy of dynein conformational states	10
6 Experimental dynein step size histogram	11
7 Dynein experimental off-axis stepping size	11
8 Cianfrocco mechanochemical cycle	12
9 Dynein model schematics	14
10 Dynein experimental off-axis stepping histogram	16
11 Model bothbound equilibrium angle defintion	18
12 Pseudocode of the bothbound Dynein model	18
13 Sarlah kinetic rates	21
14 Experimental values for fitting	22
15 Code snippet	23
16 Code snip	23
17 Snip	25

18	Literature values for experimental dynein stepping behavior.	27
19	Literature values for dynein model parameters.	28
20	Physical constants of simulation.	28
21	Onebound average spring energy.	29
22	Bothbound average spring energy.	29
23	Parameters used in processive simulations in Figures 24 and 25.	29
24	Near and far binding domains over the course of several microseconds of simulation. Simulation run at standard conditions with $k_b = 123s^{-1}$, $k_{ub} = 456s^{-1}$, $c_b = c_m = c_t = 789\Delta G_{ATP}$	30
25	Model vs experimental step sizes. a.) Step sizes model took at parameters in 23. b.) Experimental stepping behavior of dynein [10].	31

Abstract

Dynein is a motor protein which transports cargo along tracks inside the cell. Like related motor proteins kinesin and myosin, dynein uses cellular energy to take steps with its two foot domains. Unlike the more regularly-stepping kinesin or myosin, dynein's stepping pattern is very random: it can take steps between zero and 60nm in both the forwards and backwards directions. It is believed that dynein takes such broad, stochastic steps because of its large size and several elastic regions, which make it more influenced by Brownian motion. To test this, we model the motor as a 2D system of springy hinges, then simulate this model using Brownian dynamics. Preliminary results indicate such a model is capable of taking steps between zero and 25nm. These results give hope that, with further tweaking, the model may be able to generate both larger and backwards steps.

Keywords: Dynein, powerstroke, Brownian dynamics, coarse-grained, dynamics simulation

CHAPTER: INTRODUCTION

An embryo in its spherical blastula stage has spherical symmetry [...]. But a system which has spherical symmetry, and whose state is changing because of chemical reactions and diffusion, will remain spherically symmetrical for ever [...]. It certainly cannot result in an organism such as a horse, which is not spherically symmetrical. **Alan Turing, The Chemical Basis of Morphogenesis, 1952**

Cells need to create asymmetric, heterogeneous structures inside themselves in order to function. However, as per the above quote, diffusive processes cannot create this asymmetry. How then do cells create heterogeneous structures they need in order to exist? Motor proteins are part of the answer. Motor proteins are structures which use energy to create directed motion in cells. In broad terms, they are mechanical creators of asymmetry. This thesis will explore a hypothesis on how one such motor protein, Dynein, generates directed motion.

1.1 Motivation

Motor proteins are cellular structures which use chemical energy to do work and generate motion. Kinesin, myosin and dynein are the three main motor proteins in cells. Though these proteins have significantly different ancestors ??, they all evolved into a single similar structure. In biology when a structure is evolved three separate times, it is a sign that nature has found an efficient way for solving a problem. How, then, do motor proteins work?

The basic form is simple. Kinesin, myosin and dynein each use two binding domains to attach to a surface. These binding domains are connected via a linker domain. Each motor protein is an engine which undergoes cycles, where each cycle moves the motor across a surface. In a single cycle a binding domain will unbind from a surface, tense its foot to kick forward, rebind, then untense its foot to pull the whole motor forward. This pattern is repeated by both feet to generate motion [1] [17].

Although the three motor protein families operate using the same stepping cycle, their behavior differs significantly. Kinesin and myosin are predictable motors which step in the forwards direction with a regular step size [get kinesin, myosin stepping papers]. In contrast, dynein's stepping pattern varies wildly in both size and direction. The motor steps forward on average, but takes steps backwards and to the side quite regularly [10]. How can motors which implement the same release, lever, bind, lurch stepping mechanism behave so differently?

One possibility is that dynein's size and organization are responsible for the motor's odd stepping pattern. Dynein is much larger than kinesin or myosin, meaning its feet may need to move significantly further over each cycle. Due to the random nature of Brownian motion from water particle collisions, there may be significantly more variance in the path taken by dynein's feet compared to other motor feet. This could explain the large

variance in step size. Another contributing factor could be that dynein is so large that it can twist and contort during its step, allowing varying and even backwards steps, which a small motor would be too tense to achieve.

This thesis describes an experiment which aims to test whether Brownian motion and flexibility are what give rise to dynein's unique stepping pattern. To do this, a mathematical model of a dynein-like motor is constructed which exhibits flexibility and feels Brownian forces. This model is then simulated, and its behavior is compared to the behavior of real dynein.

1.2 What is dynein?

1.2.1 Basics

Dynein is a cellular structure which converts chemical energy to mechanical energy. It does so by reacting with adenosine triphosphate (ATP) to take steps along long cellular tracks known as microtubules (MTs). Dynein's mechanical energy is used for various cellular functions, including to transport molecules around the cell, pull chromosomes during division, and move cellular propellers known as cilia [4].

Dynein is a protein with many smaller domains. A depiction of the protein and its labeled subdomains is shown in Figure (1). Each motor is a homodimer made up of two identical monomer proteins. Each monomer has a binding domain which binds to the microtubule, a motor domain which binds ATP, a tail domain which connects to the other monomer, a stalk which connects the binding and motor domains, and a linker which connects the motor and tail domains. The tail domain is also responsible for binding to cargo.

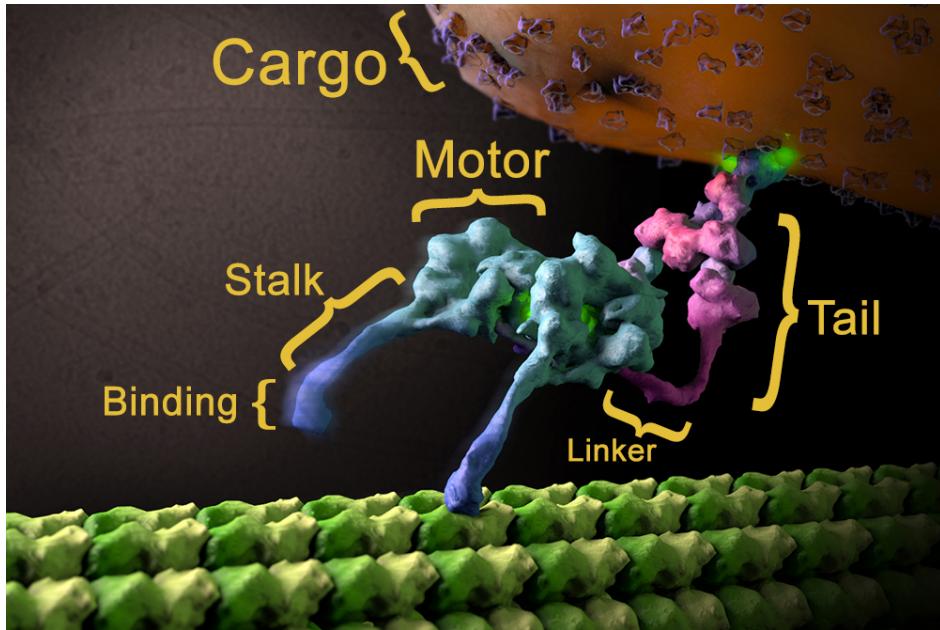


Figure 1: Artist's rendition of dynein bound to a microtubule.

One monomer is bound to the MT, and the other is raised up. Modified. Source: Lander Lab, The Scripps Research Institute.

Dynein is believed to move by using the energy of ATP to cycle through a ring of states, each with different spatial relations to the microtubule. By cycling forward through this ring, the molecule will end up in the same state it began, but moved forward a small amount. This ring of states is known as the mechanochemical cycle [4]. The goal of this project is to create a mathematical model of the mechanochemical cycle and verify that it can reproduce experimental dynein stepping data. There are several types

of dynein which all behave differently. This project will focus on cytoplasmic dynein-1, hereby referred to as dynein.

1.2.2 Why do cells need motor transport?

Cells are organized, heretogenous structures which respond quickly to their environments. This means that cells require a mechanism for rapidly moving components to precise locations within the cell. This can be a challenge, since cells are fairly large compared to the proteins which compose them. A human fibroblast cell has a volume of roughly $2000 \mu\text{m}^3$ ^[18], corresponding to roughly $8\mu\text{m}$ in diameter. In comparison, human hemoglobin has a diameter of roughly 5nm (PDB id 5ME2), a 10^3 factor difference in size.

Diffusion, or random motion of molecules due to collisions with solvent (i.e. Brownian motion), is one possible process cells could use to transport biomolecules. Diffusion has two problems: it is slow and nondirected. The expected distance covered for one-dimensional Brownian motion after time t was found by Einstein to be [5]:

$$\langle x \rangle = \sqrt{2Dt} \quad (1)$$

where D is the diffusion constant describing the diffusability of the biomolecule. For a medium-sized (140kD, 1D = weight of H atom) protein with a diffusion constant $D = .2\mu\text{m}^2/\text{s}$ [?], it would take about a month to travel across a millimeter-sized cell. In contrast, it would take dynein, which travels at roughly 100nm/s [10], a much more reasonable three hours to do the same. Figure (2) demonstrates how motor transport quickly outstrips diffusion in a cell.

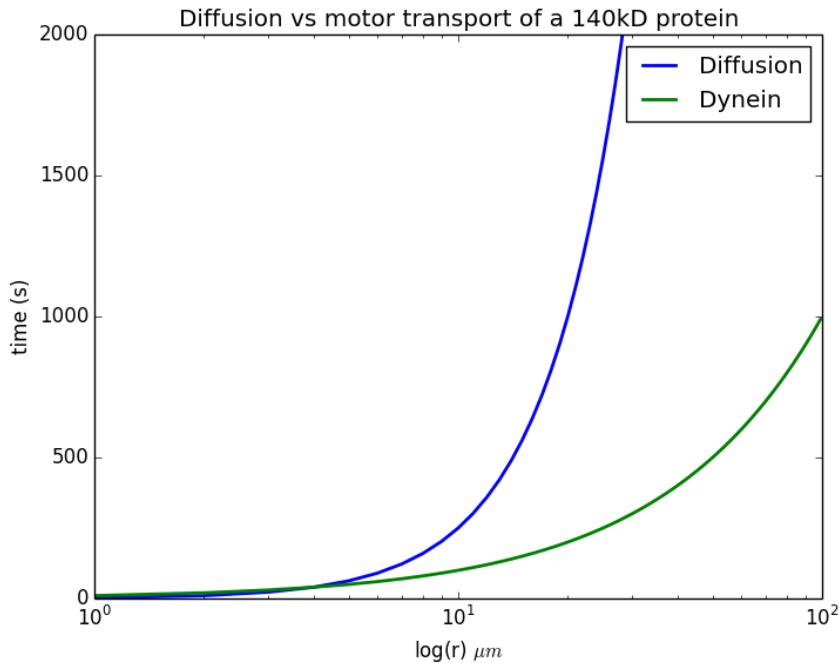


Figure 2: Passive diffusion vs motor transport rate

Plot comparing the diffusion rate of a 140kD protein with a diffusion constant $D = .2\mu\text{m}^2/\text{s}$ with a dynein motor travelling at 100nm/s .

Maybe use the more reasonable $2\mu\text{m/s}$ number for dynein velocity, in-vitro speed may not be applicable since [ATP]-limited

Another advantage of motor transport is that microtubules can be highly specific in their location in the cell. During cell division chromosomes line up at the center of each cell in a plane. Such a geometrically precise shape

requires some sort of directed motion which a random process could never provide.

maybe decorate the above plot with pictures of cells of each size, or how long certain cellular events take? That could be cool and make it actually an interesting plot this argument would be much better if you could find examples of things which need to be done quickly in the cell, like responding to foreign particles or signalling molecules or cell division or neural events

1.2.3 What role does dynein play in the cell?

Some of dynein's primary roles in the cell are to transport cellular cargos such as organelles, vesicles, mRNA, cytoskeletal filaments and certain proteins. Dynein also plays a role in positioning and breaking down the nucleus, cell death, spindle formation and the placement of other important cellular structures [16].

Microtubules are long polymers of alternating α - and β - tubulin subunits. MTs are polar, meaning they have a distinct directionality based on the orientation of the tubulin proteins which comprise them. One pole, the minus end, is where they initiate formation, typically at a MTOC (microtubule organizing center) around the center of the cell. The other pole, the plus end, is where they grow and shrink from. Kinesin and myosin, two other families of motor protein, typically walk towards the plus end of the MT. Dynein is unique in that it walks towards the minus end, making it a minus end-directed motor. This lends it to a particularly important function inside neurons known as retrograde transport.

Neurons have cell bodies at their centers and axons which grow outwards. Axons are long, narrow structures extending up to a meter in length in humans. Cell bodies contain nuclei and important organelles for synthesizing proteins, but growth of axon tips is vital for neuron function. This means bidirectional motor transport between axon tips and cell bodies is very important in neurons, since diffusion would not be quick enough. Retrograde transport is transport from the axon tip to the cell body. Cargo includes vesicles full of proteins ready to be broken down in the cell body and microtubule fragments to be returned from the axon tip [9]. Because microtubule minus ends are found in the cell body, dynein's minus end-directed nature makes it the only protein capable of retrograde transport. An interesting question is, if proteins like dynein are synthesized in the cell body but needed at the axon tip to perform retrograde transport, how do dynein get to the tip? The answer is shown in 3: kinesin, a plus end-directed motor, takes dynein to the tip from the cell body.

1.2.4 Dynein structure

eventually find a different figure

As shown in the crystal structure in Figure XXX, a dynein motor is a fusion of two identical subunits, or monomers, each with binding, stalk, motor and tail subdomains. Each of these subunits has a unique purpose for the motor.

Binding domain Microtubule binding domains (MTBDs) allow dynein to bind to the microtubule.

This could be fixed up...probably remove multiple of these sections

1.2.5 How does it behave?

Dynein's stepping pattern is distinct from its partner, kinesin. Kinesin takes very precise 8-nm steps [3]. In contrast, dynein takes a large range of steps, as shown in Figure

1.2.6 Why is dynein interesting?

Studying how nature creates directed motion is interesting in itself. Another interesting question is how, given the large number of similarities between motors kinesin and dynein, does dynein achieve such a radically different stepping pattern than kinesin? Elucidating how the structural differences between the motors give rise to the

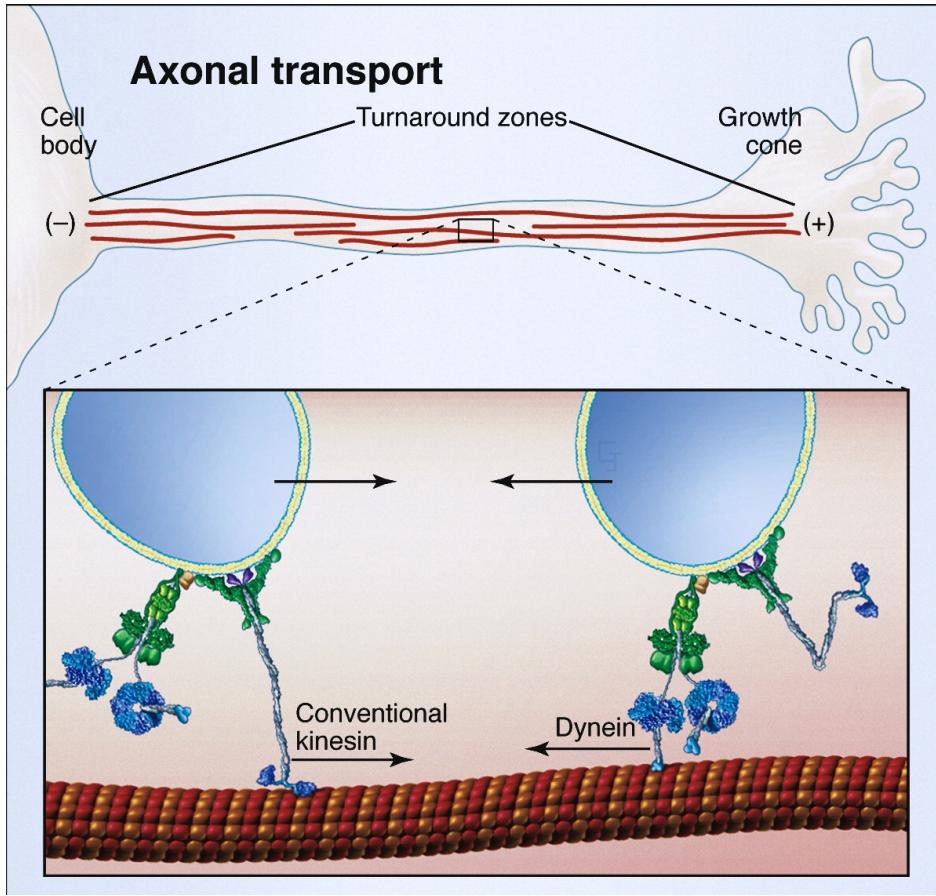


Figure 3: Dynein localization to axon growth cones.

Mechanism of dynein localization to axon tips, or growth cones, in retrograde transport mediated by kinesin anterograde transport. Modified from [16].

differences in dynamics would provide lots of important information on how the machines fine-tune their stepping.

The major difference between kinesin and dynein is their size. Kinesin proteins are generally 100-200 kD in mass, whereas dynein is roughly 1mD. As shown in Figure ??, the size and volume differences between proteins is very apparent. Dynein is about 20nm from MTBD to tail, whereas kinesin is —. Another key difference is the separation between active sites on either protein. Kinesin binds to the MT with a foot domain, colored blue in Figure ???. It hydrolyzes ATP at the residues colored pink. The separation between these domains is roughly —nm. In contrast, Dynein binds to the MT with its MTBD, colored blue, and hydrolyses ATP in its motor domain, residues colored pink. This distance is close to 20nm in length - much longer than kinesin. Another difference is the way the motors interact with the microtubule. Both motors bind to the MT using domains. However, for dynein the motor-MTBD-MT angle is —, whereas for kinesin, it is —.

Despite these differences, the two motors implement very similar chemical cycles to achieve motion. Both feet begin bound to the microtubule, then one unbinds. This foot diffuses forward, then rebinds to the microtubule

1.3 Dynein Stepping Background

Biophysicists and molecular biologists have conducted experiments over the past twenty years which shed light on how the dynein motor might walk. The full mechanism is not known, but results of such experiments suggest a certain vague picture of how dynein generates motion.

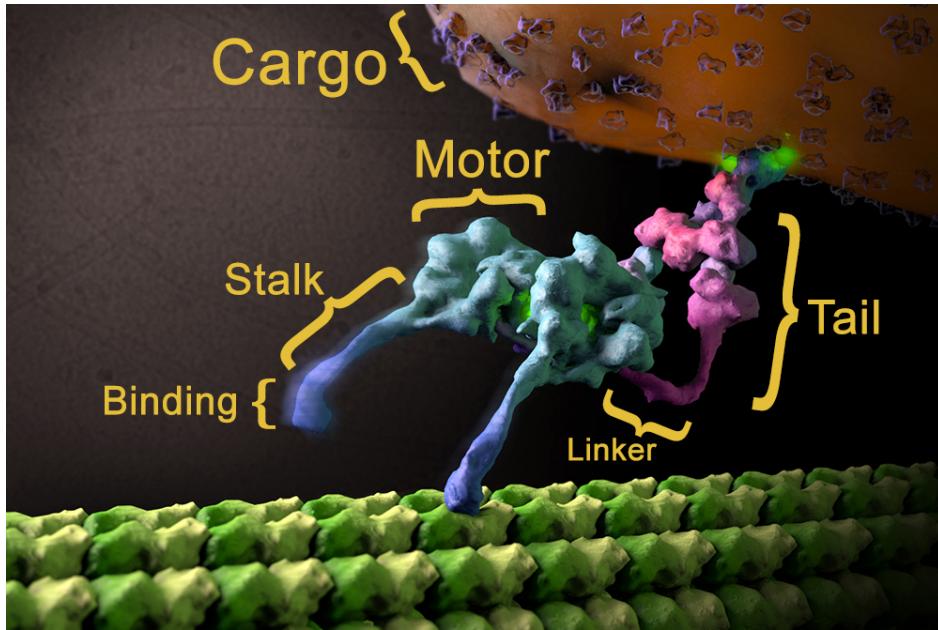


Figure 4: Artist's rendition of motor-MT complex.

Artist rendition of the dynein motor bound to a microtubule. One monomer is bound to the MT, and one is raised up. Modified. Source: Lander Lab, The Scripps Research Institute.

1.3.1 ATP cycle influences conformational state

Dynein is known to occupy multiple discrete conformational states depending on its environment. In particular, the concentration of ATP and microtubules is known to alter the shape dynein takes on. A FRET (fluorescence resonance energy transfer) study attached fluorophores to dynein's motor and tail and measured the average distance between the two [7]. It was found that average distance increased or decreased depending on the nucleotide bound to dynein, indicating the motor takes on different conformations for different bound nucleotides. Dynein unbound to nucleotide, or frozen in ATP or ADP-bound states, had a similar tail-motor distance. However, when the motor was stuck in an ADP-Pi state, with ATP hydrolyzed but phosphate not yet released, the tail-motor distance was increased. This suggests that dynein starts its cycle with no nucleotide, binds ATP, hydrolyzes ATP, enters the ADP-Pi state and changes conformation to a far tail-motor distance, then releases phosphate, moves back to its original close tail-motor conformation, then ejects ADP and starts over.

Studies suggest that near/far tail-motor conformation changes are not due to lengthening of the stem linking the two domains, but rather a change in position of the head relative to the trail [7] [2]. Cryo-electron microscopy images of the dynein motor shown in Figure (5) demonstrate the differences in conformation in the nucleotide-free and ADP-Pi state.

Crystal structures of cytoplasmic dynein-2, a relative of dynein, indicate that ATP-Pi binding to the motor domain induces a conformation change in the motor domain, which alters the angle at which the motor binds to the tail [15] (make a Chimaera figure where the pre/poststroke motors are docked showing this). Taken together, these results indicate that as the motor binds and hydrolyzes ATP, it undergoes a conformation change which physically moves the motor relative to the tail.

TODO: look at [8] and [14] to see if they can add to this discussion

1.3.2 Microtubule and nucleotide states are coupled

Dynein is known to bind to the microtubule at a location known as the α - β polymerization site [12]. This site occurs roughly every 8nm, indicating dynein is restricted to step sizes which are multiples of 8nm.

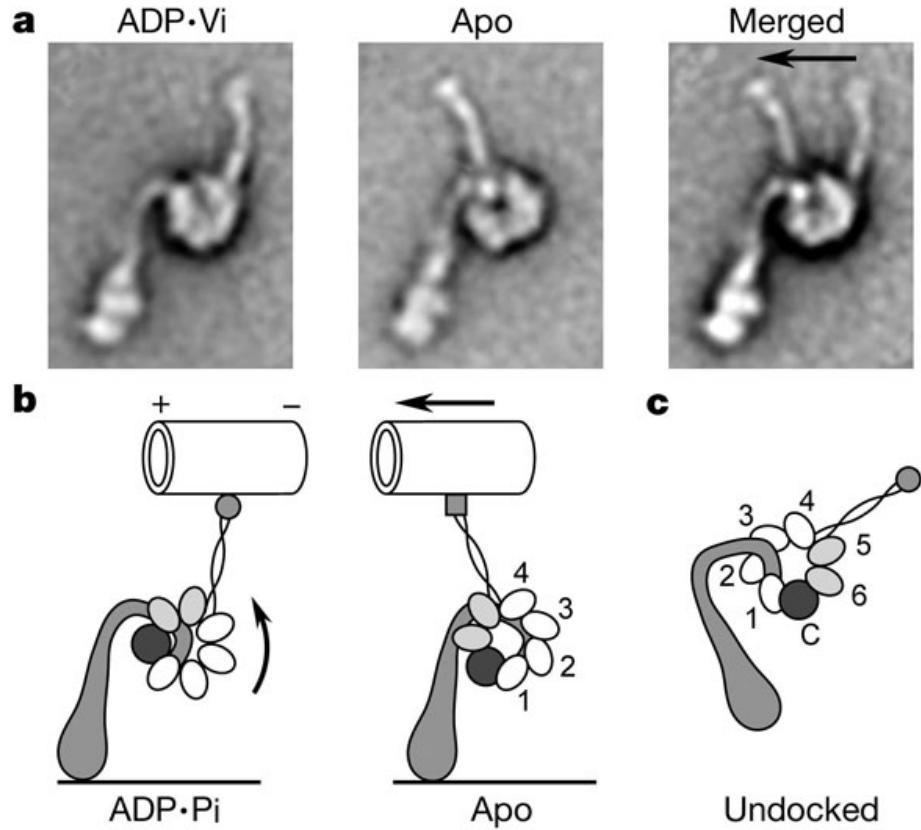


Figure 5: Cryo-EM microscopy of dynein conformational states.

Cryo-electron microscopy images of dynein in two different nucleotide states. Left: dynein frozen in ADP-Pi state. Middle: dynein without nucleotide. Right: two states superimposed on one another. Bottom: cartoons representing how dynein may bind to the MT in either state. From [2].

The nucleotide state of the motor is not the only determiner of dynein conformation; microtubule binding state has also been shown to influence conformation. A study on the MTBD-MT complex indicates the stalk connecting the motor and MTBD can take on two major conformations, depending on whether the MTBD is bound to the MT or not [6]. The conformation of the stalk has been shown to communicate information between the motor and MTBD. In the presence of microtubules, the ATPase rate of the motor is increased 25-fold. This indicates the stalk communicates MT-binding state of the MTBD to the motor, increasing its ATP hydrolysis rate. The communication happens the other way, as well. Treating dynein with ATP causes MT-binding rate to drop 15-fold, indicating ATP-binding at the motor decreases MT-binding rate at the MTBD.

In addition to a bidirectional linking between ATPase rate and MT-binding, stalk state has been shown to directly influence motor conformation [6]. When the stalk is allowed to freely transition between bound and unbound states, the motor can also transition between near and far tail/motor states. However, when the stalk is frozen in a particular conformation, the motor loses its transitioning ability. This suggests that the conformations of the two systems are tightly linked.

1.3.3 Stepping pattern

Several studies have been done on the dynein motor's stepping pattern using selectively-labeled motor domains [11] [10]. These studies show dynein's stepping pattern to be stochastic and weakly forward-directed. As shown in Figure (1.3.3), dynein takes many backwards steps, and also many steps with components perpendicular to the microtubule. This indicates the process dynein uses to generate motion has a stochastic component to it. It

is likely that the physical states of subcomponents in the motor, and/or the effects of Brownian motion due to water particle collision, play a role in dynein's stepping mechanism.

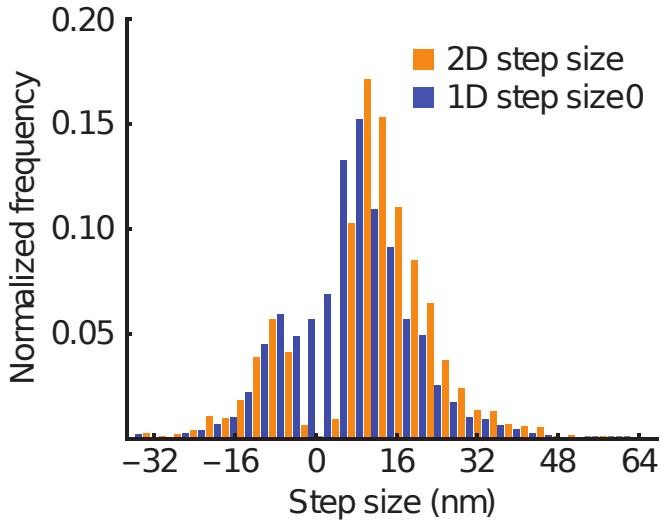


Figure 6: Experimental dynein step size histogram. Size of steps taken by dynein motor parallel to microtubule. Positive axis corresponds to steps towards the microtubule minus-end. Figure modified from [10].

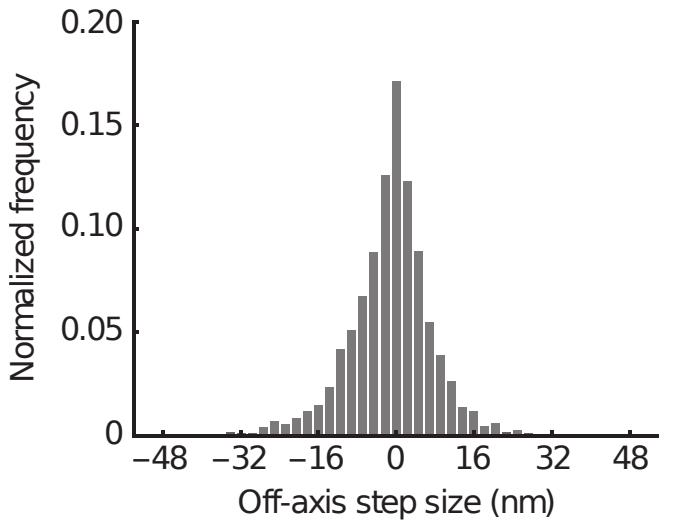


Figure 7: Dynein experimental off-axis stepping size. Size of “off-axis” steps taken by the dynein motor perpendicular to the microtubule. Figure modified from [10].

1.4 Dynein models

Data presented in the previous sections paints a picture of dynein as a collection of interconnected parts, each cycling through its own collection of states and influencing other parts. Somehow the protein interacts with nucleotides and microtubules in a way that allows it to reliably move through the cell. Models presented below take these facts and string them together into cohesive cycles which might explain how dynein walks. It is important to note that these models are hypotheses.

1.4.1 Cianfrocco Model

Cianfrocco *et. al* [4] propose dynein achieves motion via transitioning through the cycle of states shown in Fig. 8. This model will hereon be referred to as the Cianfrocco model. The model claims the primary events in the dynein step are as follows. First, starting with both binding domains attached to the MT, ATP binds to the motor domain, which causes it to change conformation. ATP hydrolysis to ADP-Pi causes further conformation changes. These changes cause the binding domain to release the microtubule, meaning only one binding domain is MT-bound. The free binding domain then undergoes another conformation change known as the “priming stroke,” which changes the orientation of the binding domain with respect to the MT. Finally the binding domain rebinds to the MT, causing a final conformation change in the motor which returns it to the beginning of the cycle. The exact order of these events is not precisely known.

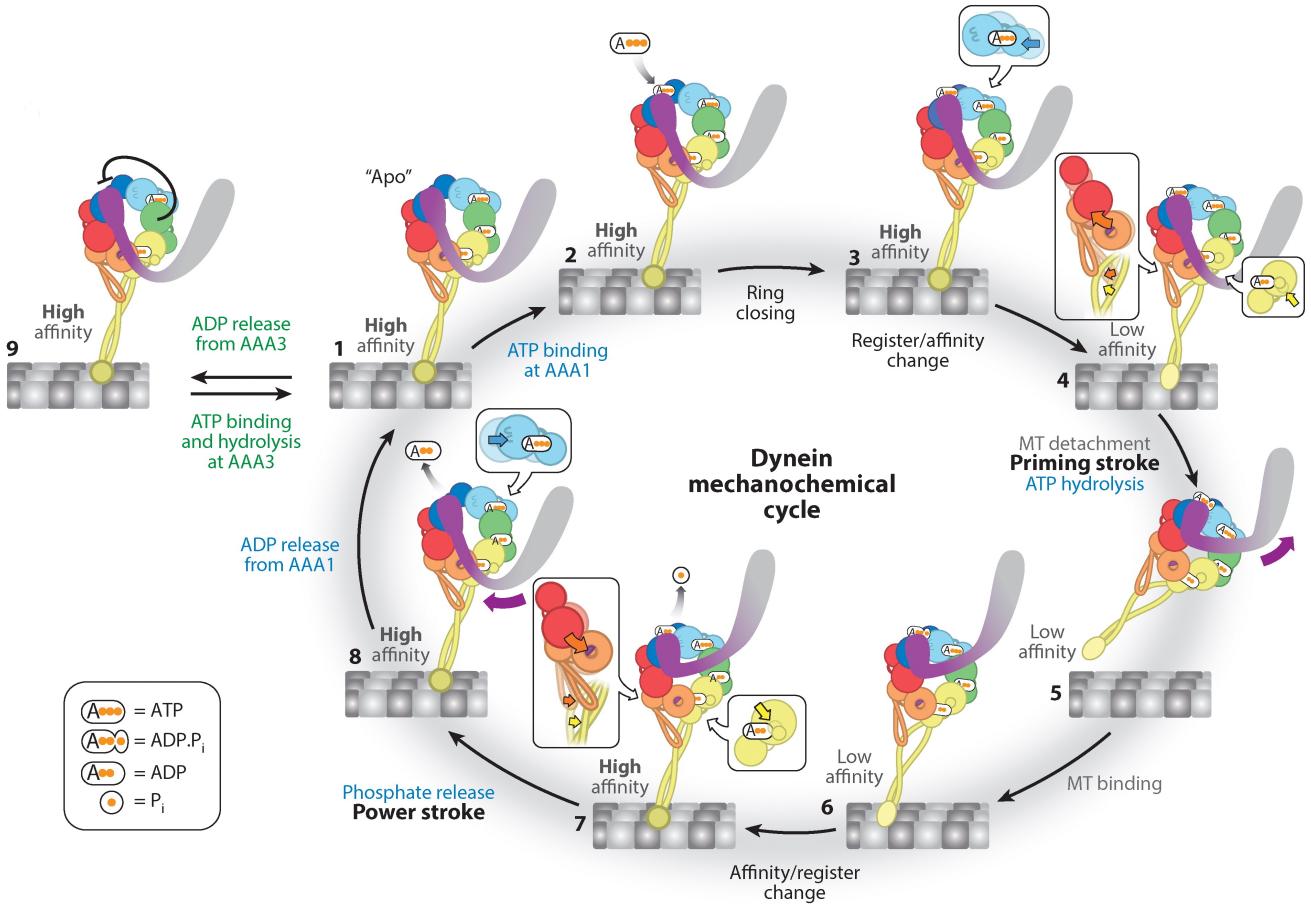


Figure 8: Cianfrocco mechanochemical cycle.

Cycle of states the dynein motor goes through during motion in the Cianfrocco model [4].

1.5 This work

1.5.1 Necessary and sufficient criteria for motion

One of the principle goals of researchers studying the dynein motor is establishing how it walks. This can be a difficult task, since direct observation of the motor walking is not currently experimentally possible. The next best thing, then, is building a model which both only does all the things dynein does, and generates motion. Finding such a system would be very useful for forming hypotheses on how dynein walks. Another requirement of the system would be that it be as simple as possible while still explaining all the data. By Occam's Razor, such systems are the most likely to represent real dynein. Thus, finding the necessary and sufficient criteria for dynein's motion is a big goal in the field.

A good starting place for finding the minimum set of motile criteria would be preexisting models, like the Cianfrocco model. This model describes a set of states which logically lead to each other and are supported by experiment. It seems intuitive that a motor which obeyed the Cianfrocco cycle would walk. There are multiple ways to verify this. A completely synthetic biological motor could be designed which obeys the cycle, and its motile properties could be studied. This seems incredibly difficult. Much easier would be to create a physical model of such a cycle, simulate it, and see if it obeys similarly to experimentally observed dynein. If the physical model both behaves according to the laws of physics and captures experimental dynein behavior well, this would be good evidence that such a model describes how dynein actually walks.

1.5.2 Goals

The goal of this work is to construct a minimum physical model of dynein protein based on the theoretical Cianfrocco model. The model should behave as physics predicts a protein to behave; namely, it should obey Brownian dynamics and conservation of energy. It should also roughly capture the state transitions and conformation changes of the Cianfrocco model. Also the model should be as simple as possible, to assure that each part of the model is necessary. If too many extra features are added, the model will not be minimal and not achieve its purpose.

This model will then be converted into a computer program, tested for compliance with the laws of physics, and simulated. Information on the long-term behavior of such a simulation will then be used to evaluate how well the model describes experimental dynein's behavior.

CHAPTER: THEORY

2.1 Model features

The purpose of this project is to find if a minimal set of features is capable of reproducing dynein's stepping behavior. To test this hypothesis, these features are put into a model and simulated. A schematic of the model is pictured in Figure ?? and described below.

Add a figure (maybe here?) of all the events which are known to happen in dynein, along with our mechanochemical cycle. It would basically show all events occurring at two times: unbinding or rebinding.

2.1.1 Spherical domains connected by massless rods

A key feature of the model is that each domain of the protein (two feet, two motors and a tail) is treated as a sphere. Each sphere is connected to one or two other spheres, as shown in Figure (??). This simplification is made to make the math easier and allow a simple definition of dynein's conformation.

One quickly notices by glancing at Figure (??), a visualization of actual dynein, that modelling the tail and feet domains as spheres is a stretch. The main point of the simplification is not to capture the shape of dynein, but the conformation. It is undefined to discuss the distance between the two motor domains in real dynein, because the motors are made up of hundreds of amino acids, and the average inter-domain distance between amino acids is comparable to the intra-domain distance. However it is very easy to describe the difference between two spheres.

Each sphere has an associated drag factor γ , which describes the resistance it feels to motion. This drag factor is calculated based on the radius R of the spheres, and is given by Stoke's Law:

$$\gamma = 6\pi\nu R \quad (2)$$

Thus R becomes a parameter input to the model.

2.1.2 Rigid rods

The model assumes connected domains are rigidly attached to one another by rods. This assumption makes the math easier, and is not necessarily based on biology. One study found that the standard deviation of stalk and stem lengths in *in vitro* dynein was roughly 1nm ?? . However a contrasting piece of evidence showed the coiled coil, a common protein structure which links the MTBD and motor domains in dynein, can stretch in myosin, another motor protein, by 2.5x ?? .

In reality, all molecules have a degree of elasticity. A dynein model by Sarlah *et. al.* models the dynein stalk as an elastic rod [13]. However, the timescale of harmonic stalk oscillations is likely much quicker than that of the mechanochemical cycle, so it is reasonable to model the stalk as a rigid rod at the stalk's average extension

length.

2.1.3 Angular springs

Each domain has a spring constant, c_b, c_m or c_t for binding, motor or tail, which decides the energy associated with that domain angle. For example, the tail domain energy is given by

$$\Delta G_t = \frac{1}{2}c_t (\theta_t - \theta_t^{eq})^2 \quad (3)$$

2.1.4 Two-state model

The model assumes MT-bound dynein can assume one of two states: one where both MTBDs are bound, and one where only one MTBD is bound to the MT. These states are also differentiated by different motor equilibrium angles, Θ_m^{bb} and Θ_m^{ob} , and different transition rates. Biological dynein likely passes through a series of permutations of nucleotide, MT affinity and conformational states as it travels through its mechanochemical cycle. This model makes the assumption that the mechanochemical cycle can be broken down into a cycle of prestroking to move the unbound foot forward, brownian walking to place the foot on the microtubule, then powerstroking to move the other domains forward.

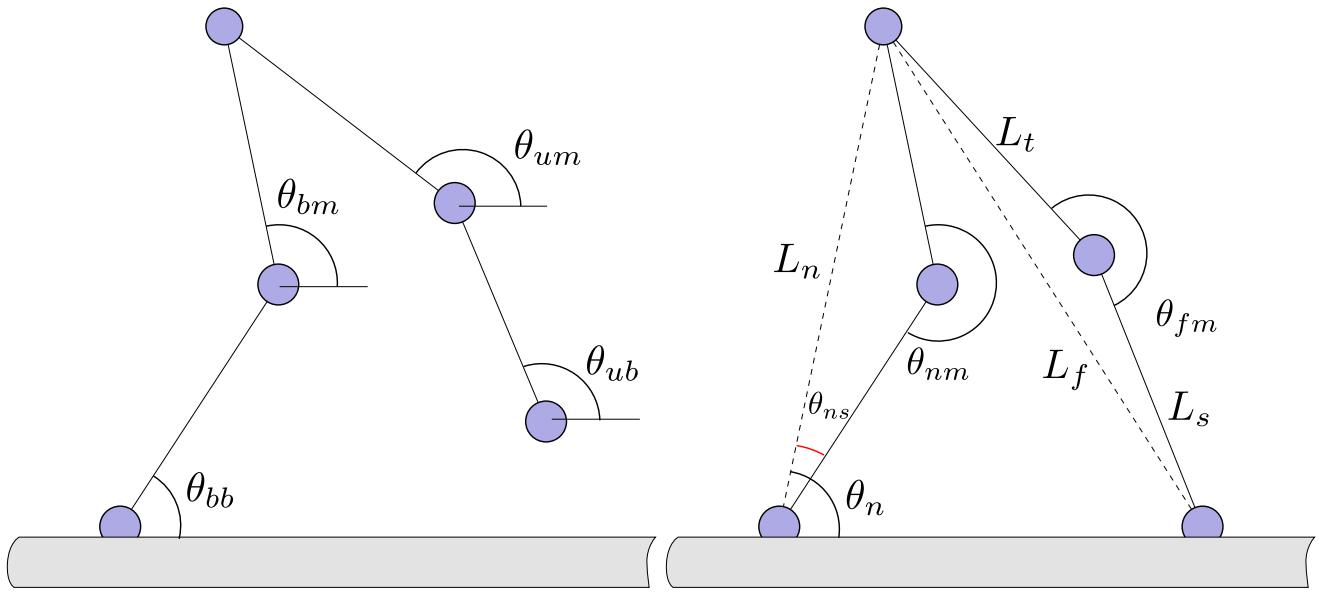


Figure 9: Dynein model schematics.
Representations of onebound (left) and bothbound (right) dynein models.

todo: Add springs to joints!

2.1.5 Transitioning between states

The model transitions between states with a probability dependent on both the energy difference between the two states and the physical separation between the MT and binding MTBD. The energy difference between states A and B is given by:

$$\Delta G_{A \rightarrow B} = \sum_{i=0 \dots N_B}^B \frac{1}{2} c_i (\theta_i - \Theta_i^B)^2 - \sum_{j=0 \dots N_A}^A \frac{1}{2} c_j (\theta_j - \Theta_j^A)^2 \quad (4)$$

Transition energies are used to calculate transition rates, as given by the Arhennius equation $k = Ae^{-\beta\Delta G}$, where $\beta = (k_B T)^{-1}$. In this equation ΔG is the total energy of transition between states and A is a constant known as

the preexponential factor with units of s^{-1} .

$$k_b = Ae^{\frac{-\Delta G}{k_B T}} = Ae^{\frac{-\Delta G_{conf}}{k_B T}} e^{\frac{-\Delta G_{bind}}{k_B T}} \quad (5)$$

where ΔG_{conf} is the change in spring energy in going from bothbound to onebound and ΔG_{bind} is the energy gained from binding to the microtubule. Because binding energy for the MT-MTBD complex is hard to find, the following substitution is made:

$$k_b = Ae^{\frac{-\Delta G}{k_B T}} = Be^{\frac{-\Delta G_{conf}}{k_B T}} \quad (6)$$

Thus B becomes a parameter of the model.

2.1.6 Brownian dynamics

The Brownian dynamics equation describes the motion of high-drag objects which spend most of their time at terminal velocity, and is given by:

$$\dot{x} = \frac{1}{\gamma} (F_{net} + R) \quad (7)$$

where γ is the drag constant of the object and R is a random Brownian force which describes the effects of water particle collision. R is a Gaussian-sampled zero-centered force with variance given by:

$$R_{\sigma^2} = \sqrt{\frac{2 * kb * T * \gamma}{dt}} \quad (8)$$

Dynein is in the range of objects which BD describes well. (citation needed)

2.1.7 Binding to microtubule

The model “steps” whenever an MTBD diffuses near the microtubule. *In vivo*, dynein binds to a site on the $\alpha - \beta$ tubulin polymer microtubule which repeats roughly every 8nm. Thus, all dynein steps should be multiples of 8nm. This model ignores this requirement and allows stepping to occur at any length. The assumption is that *in vivo* dynein takes 8nm steps because of its size and stepping angle. Thus the 8nm step length should be reproduced by the model without explicit coding.

2.1.8 2D model

The coordinate system for this model is two-dimensional. Thus interactions between the two motors and MTBDs are ignored. This assumption is made to simplify the model. Other dynein models, like the winch model ??, add in attractive and repulsive forces between the two motors which act in three dimensions. It is not clear why such forces are necessary for dynein motion, and so they are omitted.

Side stepping of dynein is ignored in this model. Stepping perpendicular to the MT occurs in dynein [10], as shown in Figure 10. This model assumes it is a symptom, and not an influencer of, of dynein’s forward-backward stepping mechanism.

2.2 Mathematical models

To simulate a protein’s passage through the above mechanochemical cycle, two mathematical models are needed: one for the prestroke, MTBD-detached state, and one for the poststroke, MTBD-bound state. The purpose of each model is to calculate the position and velocity of each domain at an arbitrary time.

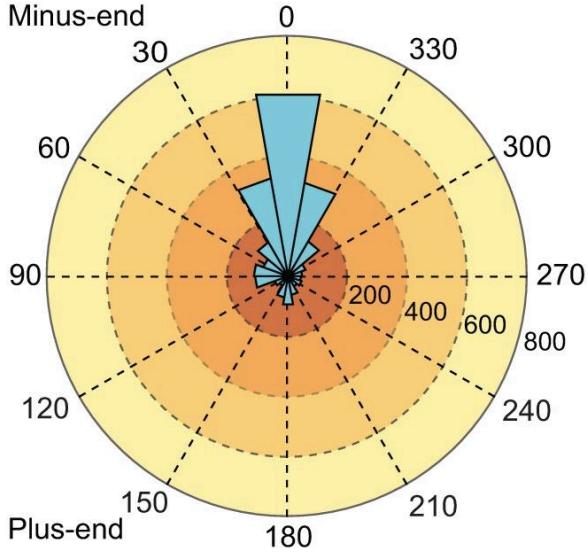


Figure 10: Dynein experimental off-axis stepping histogram.

Angular histogram of off-axis stepping of dynein, as found by [10]. 0 degrees corresponds to a step parallel to the MT, and 270 degrees a step purely to the right. Modified.

2.2.1 Prestroke onebound model

Prestroke dynein is referred to as onebound, since one motor domain is bound to the MT in this state, and the other unbound. A spatial representation of onebound dynein is shown in Figure 9.

TODO: make onebound model figure, label domain names (MTBD motor tail), Ls, Lt, microtubule

Each domain is represented by a point, and is connected to other domains via fixed-length rods. Angles represent the degrees of freedom of the model. Since the protein is a dimer, there are two copies of the MTB, motor and tail domains. The tails are fused together and so represented as a single point. The motor and MTB domains are given a “bound” and “unbound” designation to differentiate them. The “bound” domains are the MTBD attached to the MT and its corresponding motor domain. The unbound domains are the unbound MTBD and its corresponding motor domain.

2.2.1.1 Angular and Cartesian coordinates The four angles $\Theta_{bb}, \Theta_{bm}, \Theta_{um}$ and Θ_{ub} , corresponding to bound binding, bound motor, unbound motor and unbound binding angle, together describe any possible conformation the system can take on. The domain of Θ_{bb} has a restricted domain of $[0, \pi]$ to prevent below-MT conformations, but the other angles have domains of $[0, 2\pi]$. Each angle is defined relative to the horizontal axis. The position of the bound binding domain X_{bb} and the four domain angles are all that is needed to determine the position of the protein at any time.

2.2.1.2 $\dot{\Theta}_{ub}, \dot{\Theta}_{um}, \dot{\Theta}_{bm}, \dot{\Theta}_{bb}$ calculation The following equations allow recursive calculation of the positions of each domain in cartesian coordinates:

$$X_{bm} = X_{bb} + L_s \cos(\Theta_{bb}) \quad (9) \qquad Y_{bb} = 0 \quad (13)$$

$$X_t = X_{bm} + L_t \cos(\Theta_{bm}) \quad (10) \qquad Y_{bm} = Y_{bb} + L_s \sin(\Theta_{bb}) \quad (14)$$

$$X_{fm} = X_t - L_t \cos(\Theta_{fm}) \quad (11) \qquad Y_t = Y_{bm} + L_t \sin(\Theta_{bm}) \quad (15)$$

$$X_{fb} = X_{fm} - L_s \cos(\Theta_{fb}) \quad (12) \qquad Y_{fm} = Y_t - L_t \sin(\Theta_{fm}) \quad (16)$$

$$Y_{fb} = Y_{fm} - L_s \sin(\Theta_{fb}) \quad (17)$$

L_s and L_t correspond to lengths of each interdomain rod, and the t subscripts refer to tail domain coordinates.

2.2.1.3 Angular and Cartesian velocities The goal is to express angular velocities $\dot{\Theta}_{bb}$, $\dot{\Theta}_{bb}$, $\dot{\Theta}_{bb}$ and $\dot{\Theta}_{bb}$ in terms of known quantities. To begin, the cartesian velocities of each domain can be calculated from the above position equations in a similar recursive manner. A subset of the equations are shown here:

$$\dot{X}_{bb} = 0 \quad (18)$$

$$\dot{X}_{bm} = \dot{X}_{bb} - L_s \sin(\Theta_{bb})\dot{\Theta}_{bb} \quad (19)$$

$$\dot{X}_t = \dot{X}_{bm} - L_t \sin(\Theta_{bm})\dot{\Theta}_{bm} \quad (20)$$

$$\dot{Y}_{bb} = 0 \quad (21)$$

$$\dot{Y}_{bm} = \dot{Y}_{bb} + L_s \cos(\Theta_{bb})\dot{\Theta}_{bb} \quad (22)$$

$$\dot{Y}_t = \dot{Y}_{bm} + L_t \cos(\Theta_{bm})\dot{\Theta}_{bm} \quad (23)$$

Another way to express these cartesian velocities is using the Brownian dynamics equation $\dot{X} = \frac{1}{\gamma}F_{net} + R$:

$$\dot{X}_{bm} = \frac{1}{\gamma_m} \left(F_{xml} + -\lambda_{bs}(X_{bm} - X_{bb}) + \lambda_{bt}(X_t - X_{bm}) \right) + R_{xml} \quad (24)$$

$$\dot{X}_t = \frac{1}{\gamma_t} \left(F_{xt} + -\lambda_{bt}(X_t - X_{bm}) + \lambda_{ft}(X_{fm} - X_t) \right) + R_{xt} \quad (25)$$

where γ_n is a drag coefficient for the binding, motor or tail domain with units of mass per second. F_{xn} is the x component of external force on each domain n due to various factors. $\lambda_{12}(X_1 - X_2)$ is the x component of internal force on domain 2 due to domain 1, where λ is a tension coefficient with units of mass per second per second. R_{xn} is the Brownian coefficient representing motion due to solvent collision, with units of velocity.

These two sets of equations, cartesian and Brownian, can be equated to get more interesting equations. For example, Eq (19) and Eq (24) can be equated. This equating, combined with expanding the recursive velocity definitions leads to new equations, some of which are shown here:

$$-L_s \sin(\theta_{bb})\dot{\theta}_{bb} = \frac{1}{\gamma_m}F_{xml} + -\frac{1}{\gamma_m}\lambda_{bs}(X_{bm} - X_{bb}) + \frac{1}{\gamma_m}\lambda_{bt}(X_t - X_{bm}) + R_{bmx} \quad (26)$$

$$-L_s \sin(\theta_{bb})\dot{\theta}_{bb} - L_t \sin(\theta_{bm})\dot{\theta}_{bm} = \frac{1}{\gamma_t}F_{xt} + -\frac{1}{\gamma_t}\lambda_{bt}(X_t - X_{bm}) + \frac{1}{\gamma_t}\lambda_{ft}(X_{fm} - X_t) + R_{tx} \quad (27)$$

A total of eight coupled differential equations are formed from this procedure. These eight equations form a system of equations with eight unknowns: $\dot{\Theta}_{bb}$, $\dot{\Theta}_{bb}$, $\dot{\Theta}_{bb}$ and $\dot{\Theta}_{bb}$, and the four tension coefficients λ_{bs} , λ_{bt} , λ_{um} and λ_{ub} . This system is more compactly represented as:

$$\begin{pmatrix} L_s \sin(\theta_{bb}) & 0 & 0 & 0 & -\gamma_m(X_{bm} - X_{bb}) & \gamma_m(X_t - X_{bm}) & 0 & 0 \\ L_s \sin(\theta_{bb}) & L_t \sin(\theta_{bm}) & 0 & 0 & 0 & -\gamma_t(X_t - X_{bm}) & \gamma_t(X_{um} - X_t) & 0 \\ L_s \sin(\theta_{bb}) & L_t \sin(\theta_{bm}) & -L_t \sin(\theta_{um}) & 0 & 0 & 0 & -\gamma_m(X_{um} - X_t) & \gamma_m(X_{ub} - X_{um}) \\ L_s \sin(\theta_{bb}) & L_t \sin(\theta_{bm}) & -L_t \sin(\theta_{um}) & -L_s \sin(\theta_{ub}) & 0 & 0 & 0 & -\gamma_b(X_{ub} - X_{um}) \\ -L_s \cos(\theta_{bb}) & 0 & 0 & 0 & -\gamma_m(Y_{bm} - Y_{bb}) & \gamma_m(Y_t - Y_{bm}) & 0 & 0 \\ -L_s \cos(\theta_{bb}) & -L_t \cos(\theta_{bm}) & 0 & 0 & 0 & -\gamma_t(Y_t - Y_{bm}) & \gamma_t(Y_{um} - Y_t) & 0 \\ -L_s \cos(\theta_{bb}) & -L_t \cos(\theta_{bm}) & L_t \cos(\theta_{um}) & 0 & 0 & 0 & -\gamma_m(Y_{um} - Y_t) & \gamma_m(Y_{ub} - Y_{um}) \\ -L_s \cos(\theta_{bb}) & -L_t \cos(\theta_{bm}) & L_t \cos(\theta_{um}) & L_s \cos(\theta_{ub}) & 0 & 0 & 0 & -\gamma_b(Y_{ub} - Y_{um}) \end{pmatrix} \begin{pmatrix} \dot{\theta}_{bb} \\ \dot{\theta}_{bm} \\ \dot{\theta}_{um} \\ \dot{\theta}_{ub} \\ \lambda_{bs} \\ \lambda_{bt} \\ \lambda_{ut} \\ \lambda_{us} \end{pmatrix} = \begin{pmatrix} -F_{bmx} + \gamma_m R_{bmx} \\ -F_{tx} + \gamma_t R_{tx} \\ -F_{umx} + \gamma_m R_{umx} \\ -F_{ubx} + \gamma_b R_{ubx} \\ -F_{bmy} + \gamma_m R_{bmy} \\ -F_{ty} + \gamma_t R_{ty} \\ -F_{umy} + \gamma_m R_{umy} \\ -F_{uby} + \gamma_b R_{uby} \end{pmatrix}$$

This matrix is then solved using the Mathematica computer algebra system, resulting in a set of motion equations which describe the model's trajectory over time, eg $\dot{\Theta}_{bm}(\Theta_{bb}, \Theta_{bm}, \Theta_{um}, \Theta_{ub})$. These motion equations are further described in (6.1.1).

2.2.2 Poststroke bothbound model

The main difference between the prestroke and poststroke mathematical models is that the latter has a further constraint: both its MTB domains are attached to the MT. This has the effect of diminishing the degrees of freedom of the system from four to two, assuming the motor remains above the MT. A spatial representation of the bothbound model is shown in Figure (11).

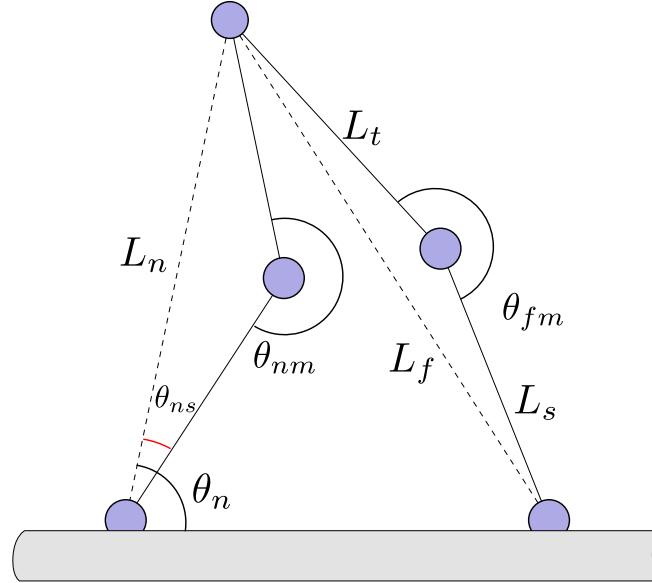


Figure 11: Model bothbound equilibrium angle definition.

The bothbound model is very similar in assumptions to the onebound: domains are represented as points connected by rigid rods, and the tails are still fused into a single point. The main differences are the naming scheme and spatial constraints. Dimers can no longer be differentiated by being un/bound, so are now differentiated by their z-position: being near or far from the “camera.” This is similar to labeling them as the right or left dimer. Thus there are the near binding (nb), near motor (nm), tail (t), far motor (fm) and far binding (fb) domains. Both the near and far binding domains have their x-coordinate fixed at zero. The system is described completely by two angles: Θ_{nm} and Θ_{fm} . Angles Θ_t , Θ_{nb} and Θ_{fb} are dependent on these two motor angles.

The second MTBD constraint adds another required piece of information the model must track: the x position of the second MTBD. This position is described by L , the distance between the near and far binding domains. The pseudocode used to describe the bothbound model are shown in Figure (??).

```

class Dynein_bothbound{
    double Θnm, Θfm
    double ḡnm, ḡfm
    double Xnb,L
}

```

Figure 12: Pseudocode of the bothbound Dynein model.

2.2.2.1 $\dot{\Theta}_{nm}, \dot{\Theta}_{fm}$ calculation The first step in calculating angular velocities is to define new variables L_n and L_f using the Law of Cosines:

$$L_n = \sqrt{L_s^2 + L_t^2 - 2L_s L_t \cos \theta_{nm}} \quad (28)$$

$$L_f = \sqrt{L_s^2 + L_t^2 - 2L_s L_t \cos \theta_{fm}} \quad (29)$$

These new artificial stalks are not physically relevant, but allow the definition of new angles $\Theta_n, \Theta_{ns}, \Theta_f$ and Θ_{fs} , which are useful for calculating the position of the motor domain. The treatment of the near and far versions of these angles is very similar, so only the near angles will be dealt with here. The angles themselves are never dealt with, only their sines and cosines:

$$\cos \theta_n = \frac{L^2 + L_n^2 - L_f^2}{2LL_n} \quad (30)$$

$$\sin \theta_n = \sqrt{1 - \cos^2 \theta_n} \quad (31)$$

$$\cos \theta_{ns} = \frac{L_s^2 + L_n^2 - L_t^2}{2L_s L_n} \quad (32)$$

$$\sin \theta_{ns} = \begin{cases} +\sqrt{1 - \cos^2 \theta_{ns}} & \theta_{nm} < \pi \\ -\sqrt{1 - \cos^2 \theta_{ns}} & \theta_{nm} > \pi \end{cases} \quad (33)$$

These values are arrived at through the Law of Cosines and a trigonometric identity. The sign of $\sin(\Theta_n)$ is restricted to positive values due to the angle's $[0, \pi]$ domain. Θ_{ns} 's domain is $[-\pi, \pi]$, meaning its sign is not restricted. The partial derivatives of these values are used in future calculations, and some are shown here:

$$\frac{\partial \cos \theta_n}{\partial L_n} = \frac{1}{L} - \frac{L^2 + L_n^2 - L_f^2}{2LL_n^2} \quad (34)$$

$$\frac{\partial \cos \theta_n}{\partial L_f} = -\frac{L_f}{LL_n} \quad (35)$$

$$\frac{\partial \sin \theta_n}{\partial L_n} = \frac{-\cos \theta_n}{\sqrt{1 - \cos^2 \theta_n}} \left(\frac{1}{L} - \frac{L^2 + L_n^2 - L_f^2}{2LL_n^2} \right) \quad (36)$$

The position of the motor and tail domains is calculated as follows, using the angle addition trigonometric identity:

$$X_{nm} = L_s (\cos \theta_n \cos \theta_{ns} - \sin \theta_n \sin \theta_{ns}) \quad (37)$$

$$Y_{nm} = L_s (\cos \theta_n \sin \theta_{ns} + \sin \theta_n \cos \theta_{ns}) \quad (38)$$

$$X_t = L_n \cos \theta_n \quad (39)$$

$$Y_t = L_n \sin \theta_n \quad (40)$$

The partial derivatives of these values are used later. A selection are shown here:

$$\frac{dX_{nm}}{dL_n} = L_s \left(\cos \theta_n \frac{d \cos \theta_{ns}}{dL_n} + \cos \theta_{ns} \frac{d \cos \theta_n}{dL_n} - \sin \theta_n \frac{d \sin \theta_{ns}}{dL_n} - \sin \theta_{ns} \frac{d \sin \theta_n}{dL_n} \right) \quad (41)$$

$$\frac{dY_{nm}}{dL_n} = L_s \left(\cos \theta_n \frac{d \sin \theta_{ns}}{dL_n} + \sin \theta_{ns} \frac{d \cos \theta_n}{dL_n} + \sin \theta_n \frac{d \cos \theta_{ns}}{dL_n} + \cos \theta_{ns} \frac{d \sin \theta_n}{dL_n} \right) \quad (42)$$

$$\frac{dX_{nm}}{dL_f} = L_s \left(\cos \theta_n \frac{d \cos \theta_{ns}}{dL_f} + \cos \theta_{ns} \frac{d \cos \theta_n}{dL_f} - \sin \theta_n \frac{d \sin \theta_{ns}}{dL_f} - \sin \theta_{ns} \frac{d \sin \theta_n}{dL_f} \right) \quad (43)$$

These partial derivatives are then used in a system of equations very similar to that in Equations (26-??), combining Brownian and coordinate system definitions of domain velocities:

$$\dot{X}_{nm} = \frac{1}{\gamma} \left(F_{xml} - \lambda_{ns}(X_{bm} - X_{bb}) + \lambda_{nt}(X_t - X_{bm}) \right) + R_{xml} = \frac{\partial X_{nm}}{\partial L_n} \dot{L}_n + \frac{\partial X_{nm}}{\partial L_f} \dot{L}_f \quad (44)$$

$$\dot{X}_t = \frac{1}{\gamma} \left(F_{xt} - \lambda_{nt}(X_t - X_{bm}) + \lambda_{ft}(X_{fm} - X_t) \right) + R_{xt} = \frac{\partial X_t}{\partial L_n} \dot{L}_n + \frac{\partial X_t}{\partial L_f} \dot{L}_f \quad (45)$$

$$\dot{X}_{fm} = \frac{1}{\gamma} \left(F_{xmr} - \lambda_{ft}(X_{fm} - X_t) + \lambda_{fs}(X_{fb} - X_{fm}) \right) + R_{xmr} = \frac{\partial X_{fm}}{\partial L_n} \dot{L}_n + \frac{\partial X_{fm}}{\partial L_f} \dot{L}_f \quad (46)$$

$$\dot{Y}_{nm} = \frac{1}{\gamma} \left(F_{yml} - \lambda_{ns}(Y_{bm} - Y_{bb}) + \lambda_{nt}(Y_t - Y_{bm}) \right) + R_{yml} = \frac{\partial Y_{nm}}{\partial L_n} \dot{L}_n + \frac{\partial Y_{nm}}{\partial L_f} \dot{L}_f \quad (47)$$

$$\dot{Y}_t = \frac{1}{\gamma} \left(F_{yt} - \lambda_{nt}(Y_t - Y_{bm}) + \lambda_{ft}(Y_{fm} - Y_t) \right) + R_{yt} = \frac{\partial Y_t}{\partial L_n} \dot{L}_n + \frac{\partial Y_t}{\partial L_f} \dot{L}_f \quad (48)$$

$$\dot{Y}_{fm} = \frac{1}{\gamma} \left(F_{ymr} - \lambda_{ft}(Y_{fm} - Y_t) + \lambda_{fs}(Y_{fb} - Y_{fm}) \right) + R_{ymr} = \frac{\partial Y_{fm}}{\partial L_n} \dot{L}_n + \frac{\partial Y_{fm}}{\partial L_f} \dot{L}_f \quad (49)$$

The matrix version of this system looks like:

$$\begin{pmatrix} -\frac{X_{nm}}{L_n} & -\frac{X_{nm}}{L_f} & -\frac{X_{nm}-X_{nb}}{\gamma} & \frac{X_t-X_{nm}}{\gamma} & 0 & 0 \\ -\frac{X_t}{L_n} & -\frac{X_t}{L_f} & 0 & -\frac{X_t-X_{nm}}{\gamma} & \frac{X_{fm}-X_t}{\gamma} & 0 \\ -\frac{X_{fm}}{L_n} & -\frac{X_{fm}}{L_f} & 0 & 0 & -\frac{X_{fm}-X_t}{\gamma} & \frac{X_{fb}-X_{fm}}{\gamma} \\ -\frac{Y_{nm}}{L_n} & -\frac{Y_{nm}}{L_f} & -\frac{Y_{nm}-Y_{nb}}{\gamma} & \frac{Y_t-Y_{nm}}{\gamma} & 0 & 0 \\ -\frac{Y_t}{L_n} & -\frac{Y_t}{L_f} & 0 & -\frac{Y_t-Y_{nm}}{\gamma} & \frac{Y_{fm}-Y_t}{\gamma} & 0 \\ -\frac{Y_{fm}}{L_n} & -\frac{Y_{fm}}{L_f} & 0 & 0 & -\frac{Y_{fm}-Y_t}{\gamma} & \frac{Y_{fb}-Y_{fm}}{\gamma} \end{pmatrix} \begin{pmatrix} \dot{L}_n \\ \dot{L}_f \\ \lambda_{ns} \\ \lambda_{nt} \\ \lambda_{ft} \\ \lambda_{fs} \end{pmatrix} = \begin{pmatrix} -\frac{1}{\gamma} F_{nmx} - R_{nmx} \\ -\frac{1}{\gamma} F_{tx} - R_{tx} \\ -\frac{1}{\gamma} F_{fmx} - R_{fmx} \\ -\frac{1}{\gamma} F_{nmy} - R_{nmy} \\ -\frac{1}{\gamma} F_{ty} - R_{ty} \\ -\frac{1}{\gamma} F_{fmy} - R_{fmy} \end{pmatrix}$$

This system of equations is then solved and used to calculate \dot{L}_n and \dot{L}_f , as shown in Appendix (6.1.2).

2.3 Data interpretation

Metrics for quality are required to assess how well the model fits reality. These metrics include time spent in one/bothbound states and average step size. These metrics are developed here.

2.3.1 Converting experimental rates to preexponential factors A_b and A_{ub}

Rate constants k_b and k_{ub} govern the probability of transitioning from onebound to bothbound and vice-versa, respectively. One issue is that when these parameters are measured experimentally, the values represent the average rate for an ensemble of dynein in many different conformations. In contrast, the rates the model uses represent the probability of unbinding of a specific conformation. Thus, it is necessary to find a way to convert between experimental rates k_b^{exp} and k_{ub}^{exp} and conformation-specific rates k_b and k_{ub} . As shown in Section 3.1.2.1, rates can be expressed using the Arrhenius equation:

$$k_b = A'_b e^{\frac{-\Delta G_{bind}}{k_B T}} = A'_b e^{\frac{-\Delta G_{conf}^{ob \rightarrow bb}}{k_B T}} e^{\frac{-\Delta G_{bind}}{k_B T}} = A_b e^{\frac{-\Delta G_{conf}^{ob \rightarrow bb}}{k_B T}}$$

$$k_{ub} = A'_{ub} e^{\frac{-\Delta G_{unbind}}{k_B T}} = A'_{ub} e^{\frac{-\Delta G_{conf}^{bb \rightarrow ob}}{k_B T}} e^{\frac{\Delta G_{bind}}{k_B T}} = A_{ub} e^{\frac{-\Delta G_{conf}^{bb \rightarrow ob}}{k_B T}}$$

where the constant Boltzmann factor for binding energy are hidden inside the unprimed preexponential factors. The experimental values are calculated the same way, with the free energy changes replaced with expectation values. Since by the equipartition theorem $\langle G_{conf}^{ob} \rangle = 2k_B T$ and $\langle G_{conf}^{bb} \rangle = k_B T$, the equations can be simplified:

$$k_b^{exp} = A_b e^{\frac{-(2k_B T - k_B T)}{k_B T}} = \frac{A_b}{e}$$

$$k_{ub}^{exp} = A_{ub} e^{\frac{-(k_B T - 2k_B T)}{k_B T}} = A_{ub} e$$

Thus experimental rates can be converted easily into the model parameters A_b and A_{ub} .

2.3.2 Literature values for experimental binding rates

Sarlah *et. al.* [13] perform a similar simulation of dynein and provide the reaction rates shown in Figure 13.

Parameter	Description	Parameter value	Experimental value	Source
k^0_{+ATP}	ATP binding	$2 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$	$2.2 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$	(58)
k^0_{-ATP}	ATP release	50 s^{-1}	175 s^{-1}	(58)
k^0_{-MT}	MT release, D	500 s^{-1}	460 s^{-1}	(26)
k^0_{+MT}	MT binding, D	100 s^{-1}		
k^0_{+RS}	ATP hydrolysis, linker swing to prestroke	1000 s^{-1}	180 s^{-1}	(58)
k^0_{-RS}	ATP synthesis, linker swing to poststroke	100 s^{-1}	$4 \text{ s}^{-1}; 30 \text{ s}^{-1}$	(47,58)
k^0_{+MT}	MT binding, D^*	$10,000 \text{ s}^{-1}$		
k^0_{-MT}	MT release, D^*	500 s^{-1}		
k^0_{-Pi}	Pi release	5000 s^{-1}		
k^0_{+Pi}	Pi binding	$10^4 \text{ M}^{-1} \text{ s}^{-1}$	$8000 \text{ M}^{-1} \text{ s}^{-1}$	(47)
k^0_{+PS}	Power stroke	5000 s^{-1}		
k^0_{-PS}	Reverse stroke	10 s^{-1}		
k^0_{-ADP}	ADP release	160 s^{-1}	$\sim 1000 \text{ s}^{-1}$	(27)
k^0_{+ADP}	ADP binding	$2.7 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$	$2.7 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$	(27)

Figure 13: Sarlah kinetic rates.
Kinetic rates used in Sarlah *et. al.*'s model [13].

The assumption of this project is that many of the states in this table can be combined into a single state without loss of predictive power of dynein behavior. Specifically, the assumption is that ejecting phosphate, conducting the powerstroke, releasing ADP, binding ATP and dissociating from the microtubule and all occur during the bothbound state, and hydrolyzing ATP, conducting the prestroke and rebinding the MT occur during onebound. These states are roughly grouped as MT-bound and MT-unbound states in the Cianfrocco mechanochemical cycle [4].

The rate-limiting transitions in the Cianfrocco model are used as the guess k_b and k_{ub} values. That means that, since ATP hydrolysis has the slowest rate constant in onebound and MT release has the slowest rate in bothbound, $k_b^{exp} = 180 \text{ s}^{-1}$ and $k_{ub} = 460 \text{ s}^{-1}$ serve as good initial guesses.

2.3.3 Calculating one/bothbound times of experimental dynein

Rate constants can be fit to experimentally measurable values, such as velocity and the time dynein is bound to the microtubule (processivity time). We define the following values:

- $\langle t_{bb} \rangle$: Time in bothbound per step
- $\langle t_{ob} \rangle$: Time in onebound per step
- $\langle t_{step} \rangle$: Time of a single step. $\langle t_{step} \rangle = \langle t_{bb} \rangle + \langle t_{ob} \rangle$.
- P_{bb} : Probability of being bothbound per unit time. $P_{bb} = \frac{\langle t_{bb} \rangle}{\langle t_{step} \rangle}$
- P_{ob} : Probability of being onebound per unit time. $P_{ob} = \frac{\langle t_{ob} \rangle}{\langle t_{step} \rangle}$

- k_{ub} : Rate of unbinding per unit time while in bothbound. $k_{ub} = \langle t_{bb} \rangle^{-1}$
- k_b : Rate of binding per unit time while in onebound. $k_b = \langle t_{ob} \rangle^{-1}$
- k_{dis} : Rate of unbinding per unit time while in onebound. $k_{dis} = P_{ob}k_{ub}$
- $\langle t_{processivity} \rangle$: Time bound to microtubule before dissociation. step length / velocity = k_{dis}^{-1}

$\langle t_{bb} \rangle$ and $\langle t_{ob} \rangle$ can thus be defined in terms of observables $\langle t_{step} \rangle$ and $\langle t_{processivity} \rangle$:

$$\begin{aligned}
k_{dis} &= P_{ob}k_{ub} = \frac{\langle t_{ob} \rangle}{\langle t_{step} \rangle \langle t_{bb} \rangle} \\
&= \frac{\langle t_{step} \rangle - \langle t_{bb} \rangle}{\langle t_{step} \rangle \langle t_{bb} \rangle} \\
&= \frac{\langle t_{step} \rangle - \langle t_{bb} \rangle}{\langle t_{step} \rangle \langle t_{bb} \rangle} \\
\langle t_{processivity} \rangle &= \frac{\langle t_{step} \rangle \langle t_{bb} \rangle}{\langle t_{step} \rangle - \langle t_{bb} \rangle} \\
\langle t_{bb} \rangle &= \frac{\langle t_{processivity} \rangle \langle t_{step} \rangle}{\langle t_{processivity} \rangle + \langle t_{step} \rangle} \\
\langle t_{ob} \rangle &= \langle t_{step} \rangle - \langle t_{bb} \rangle \\
&= \langle t_{step} \rangle - \frac{\langle t_{step} \rangle \langle t_{processivity} \rangle}{\langle t_{step} \rangle + \langle t_{processivity} \rangle} \\
&= \frac{\langle t_{step} \rangle^2}{\langle t_{step} \rangle + \langle t_{processivity} \rangle}
\end{aligned}$$

Thus, by the above equations and using data from [10], values for $\langle t_{bb} \rangle$ and $\langle t_{ob} \rangle$ can be found:

Velocity	$\langle L_{step} \rangle$	$\langle t_{step} \rangle$	$\langle t_{processivity} \rangle$	$\langle t_{bb} \rangle$	$\langle t_{ob} \rangle$
134nm/s	8nm	0.06s	7.9s	0.0595s	4.52e-4

Figure 14: Experimental values for fitting.
Table calculating estimates for $\langle t_{bb} \rangle$ and $\langle t_{ob} \rangle$.

CHAPTER: METHODS

This section will go into detail on how the model was implemented in code, verified for accuracy, and how its parameters were fit to experiment.

3.1 Simulation

To generate stepping data, the models were coded in C++ and time-evolved using Euler's method to generate reasonable dynamical behavior. When in a position deemed worthy of a state transition a conversion, the model was converted to the respective other model, representing part of a "step" through the mechanochemical cycle. Some unphysical but minor corrections at points in the simulation were needed to rescue the model from entering incalculable positions.

Code is available at https://github.com/elliottc12/dynein_walk.git.

todo: come up with a better word than "model" to describe the state of the system at any time

3.1.1 Time evolution

Euler's method for solving differential equations was used to calculate domain positions through time. Because each model is a set of multiple coupled differential equations, care was taken to update each domain position at the same time, and not in a cascade. A code snippet is shown in Figure (??) for the onebound updating scheme.

```
def simulate(dyn_ob):
    t = 0
    while (t < runtime):
        double temp_bba = dyn_ob->get_bba() + dyn_ob->get_d_bba() * dt
        double temp_bma = dyn_ob->get_bma() + dyn_ob->get_d_bma() * dt
        double temp_uma = dyn_ob->get_uma() + dyn_ob->get_d_uma() * dt
        double temp_uba = dyn_ob->get_uba() + dyn_ob->get_d_uba() * dt

        dyn_ob->set_bba(temp_bba)
        dyn_ob->set_bma(temp_bma)
        dyn_ob->set_uma(temp_uma)
        dyn_ob->set_uba(temp_uba)

    t += dt
```

Figure 15: Code snippet

Code snippet of Euler's method to update the positions of the onebound model simultaneously.

The `dt` used to update was 10^{-11} seconds.

3.1.2 Transitioning between states

3.1.2.1 Transition rate Transitions from one state to another are accomplished by calculating the probability of a transition at a given timestep, given by $P_{ob \rightarrow bb} = k_b dt$. Each timestep a value in $[0, 1]$ is sampled from a random number generator and, if lower than P , the respective transition is made. A code snippet illustrating how unbinding occurs is shown in Figure (17).

```
def get_near_unbinding_rate(dyn_bb) {
    dyn_ob_test = make_onebound_from_bothbound(dyn_bb) // do a fake unbind to find
    dG_spring = dyn_ob_test.get_PE() - dyn_bb.get_PE()
    low_affinity_unbinding_preexponential_factor = low_affinity_unbinding_rate / exp
    return low_affinity_unbinding_preexponential_factor*exp(-dG_spring/kb/T)
}
def transition_bothbound_to_onebound(dyn_bb) {
    random_number = rand(0,1)
    if (random_number < get_unbinding_rate(dyn_bb)) {
        return make_onebound_from_bothbound(dyn_bb)
    }
    else return 0
}
```

Figure 16: Code snip

Code snippet of transition rate calculation using the Arrhenius equation.

3.1.2.2 OB \rightarrow BB Transition calculation When binding occurs, conversion between OB and BB models is done by creating a bothbound dynein which is as similar in terms of domain position as possible to the onebound

dynein.

Due to the nature of the simulation, binding attempts must occur above, but very close to, the microtubule. This is because *in vivo*, electrostatic interactions between MTBD and MT bring the two together when they are near. These electrostatic interactions are not present in the model, and so are simulated by allowing binding attempts to occur when $Y_{ub} \leq 0.2\text{nm}$ above the MT. When a binding attempt is successful, the unbound binding domain of the OB dynein is “teleported” such that $Y_{ub} = 0.0$. This is to enforce the constraint that both binding domains of the BB model are directly on the MT.

To create a BB model from OB, the following intermediate variables are first computed:

$$L_n^2 = (X_t - X_{nb})^2 + Y_t^2 \quad (50)$$

$$L_f^2 = (X_t - X_{fb})^2 + Y_t^2 \quad (51)$$

$$\cos(\Theta_{nm}) = \frac{L_s^2 + L_t^2 - L_n^2}{2L_s L_t} \quad (52)$$

$$\cos(\Theta_{fm}) = \frac{L_s^2 + L_t^2 - L_f^2}{2L_s L_t} \quad (53)$$

todo: figure out why this new method is used instead of just computing the bad nma angles

3.1.2.3 BB → OB Transition calculation Similar to the binding transition, unbinding involves creating a new onebound dynein from the old bothbound. To accomplish a transition to the farbound state, the following equations are used:

$$\Theta_{new,bba} = \Theta_{old,nba} \quad (54)$$

$$\Theta_{new,bma} = \Theta_{old,nma} + \Theta_{old,nba} - \pi \quad (55)$$

$$\Theta_{new,uma} = \Theta_{old,fma} + \Theta_{old,fba} - \pi \quad (56)$$

$$\Theta_{new,uba} = \Theta_{old,fba} \quad (57)$$

To do a nearbound transition, the “f” and “n” subscripts in the above equations are swapped.

As shown in Section ??, the MT binding rate is much quicker than the unbinding rate. This means that, when just unbound, the rate of rebinding is very high. To prevent immediate rebinding, the model is given a “rebinding immunity” of a single timestep. During this immunity period, rebinding is not allowed.

3.1.3 Corrections

Certain configurations of the model, such as in the bothbound model when $\Theta_{nm} = \pi$ or $\Theta_{fm} = \pi$, it becomes impossible to calculate intermediate variables due to values going to infinity (name for this??). For example when $\Theta_{nm} = \pi$, $L_n = L_s + L_t$ and so:

$$\cos \theta_{ns} = 1 \quad (58)$$

$$\sin \theta_{ns} = 0 \quad (59)$$

$$\text{some val?} \rightarrow \infty \quad (60)$$

Alternative problem states are when $L = 0$ and ???. These states are corrected similarly.

3.1.4 Forces

Spring and Brownian forces are used to properly move the model through time. Appendices (??-6.1.2) show angular velocities are functions of the x- and y-components of both spring forces F and Brownian forces γR .

3.1.4.1 Brownian forces Each Brownian force $R_{n,m}$ on domain n in direction m is sampled from a Gaussian with standard deviation $\sqrt{\frac{2k_B T \gamma_n}{dt}}$.

3.1.4.2 Spring forces Spring forces are calculated by first finding the torque τ on each angle, then converting this to a force. For example, the onebound near motor angle feels a force $\tau_{nm} = c_m(\Theta_{nm} - \Theta_{nm,eq})$, where c_n is the spring constant of the motor domain and $\Theta_{nm,eq}$ the equilibrium angle. The magnitude of force on the adjacent binding and tail domains is thus τ/L_s and τ/L_t , pointing in the direction adjacent the Ls and Lt rods, respectively. The force on the motor domain is found as the equal-and-opposite force to these two imposed forces. In code:

```

T = cm*(nma - nma_eq);
f1 = T/Ls;
f2 = T/Lt;
f1x = f1 * sin(bba);
f1y = f1 * -cos(bba);
f2x = f2 * sin(bma);
f2y = f2 * -cos(bma);
f.bbx += f1x;
f.bby += f1y;
f.tx += f2x;
f.ty += f2y;
f.bmx += -(f1x + f2x); // equal and opposite forces
f.bmy += -(f1y + f2y);

```

Figure 17: Snip
Code snippet of internal force calculation.

3.2 Verifying the model

The models were verified to follow physical laws by checking their behavior in various tests: general conformational tests, energy conservation and obeying the equipartition theorem. The purpose of these tests was to verify that equations had been entered properly (primarily the conformational and energy conservation tests), and that the model obeyed the laws of physics (all tests).

3.2.1 Conformational tests

The models were tested in the following different conformations to verify they behaved as expected:

Onebound tests

- **Onebound: Upwards line conformation with no forces.** Check that all domain x-values are zero* and all y-values are the proper addition of L_s and L_t lengths. Also checks that all angular velocities are zero. This test verifies that the angle scheme is defined properly, that the model can handle straight lines without generating NaNs and that there are no velocities (angular or cartesian) without forces.
- **Onebound: Horizontal line conformation with no forces.** Check that all domain y-values are zero and all x-values are the proper addition of L_s and L_t lengths. Also check that all velocities are zero. This test

double-verifies that the coordinate system behaves as expected and that velocities are zero in the absence of forces.

- **Onebound:** *Upwards line conformation with +x forces* Check that cartesian x-velocities are positive and y-velocities are zero. Verifies that the cartesian velocities have the expected sign.
- **Onebound:** *Upwards line conformation with +y forces* Check that all cartesian velocities are zero. Verifies that upwards forces in an already-vertical stalk have no effect on velocity.
- **Onebound:** *Prepowerstroke opposing internal/brownian forces* Check that, for two onebound models, one with +x spring forces and no Brownian forces and the other with -x Brownian forces and no spring forces, that the velocities of each unbound domain are equal and opposite. Verifies that equal Brownian and internal forces have the same magnitude and direction of effect on velocities.

*: checks for equality are within a margin of error of $\epsilon = 1e^{-5}$.

These tests together verify that the onebound model's internal coordinate system's directions match the XY coordinate system directions, that the angular directions in the model properly align with the $\pm x$ and $\pm y$ directions of the XY coordinate system, that velocities have the proper magnitude at zero force and proper sign at nonzero force, and that Brownian and internal forces are in the same unit system. There are still errors which could exist in the model pertaining to magnitude of velocity, which will be tested for in the energy conservation section.

Bothbound tests

- **Bothbound:** *Upwards line conformation with no Brownian forces.* Both MTB domains are very close together ($L = 10^(-25)$) and motor domains nearly vertical ($\Theta_{n/f} = \pi \pm 10^(-15)$). Equilibrium angles set to have a vertical line as the lowest energy point. Model is not exactly vertical for NaN concerns explained in section 3.1.3. Checks that domain x-values are zero*, domain y-values are the correct combination of L_s and L_t lengths, all angles are at equilibrium and all forces are zero. Verifies that the domains are properly oriented by the angles, the angles are defined properly and forces are calculated from angles properly.
- **Bothbound:** *Equilateral house conformation with outward x-forces.* The model forms the shape of a square with an equilateral triangle on top, with the tail domain as the triangle. Checks if all domains are at the desired locations and the derivatives are the proper sign and are equal and opposite to their counterparts on the opposing side of the house. This test verifies that the angular system behaves as expected and forces are symmetric about the two dimers.
- **Bothbound:** *Two table conformations with near/far domains flipped.* Two models are put in the same table-like conformation, but the near/far domains are flipped. Checks if opposite domains have the same positions and forces. Re-verifies that the model is symmetric in position and force between the two dimers.

These tests verify the bothbound model's angle system behaves as expected, forces are zero when expected and have the proper signs, and that the whole system is symmetric with either dimer.

3.2.2 Energy conservation

To verify that the magnitudes of forces are calculated properly, a conservation of energy test was performed. A very slight angular nudge was applied to either model. This nudge caused a small change in energy. This change in energy was compared with the work the forces on each domain caused over the nudge. For energy to be conserved, these values were required to be very similar. Mathematically the test required the following:

$$\sum_n \frac{1}{2} (\Theta_{nudge} - \Theta_n - \Theta_{n,eq}) = \sum_n \vec{F}_n \cdot \vec{r}_{n,nudge} \quad (61)$$

where Θ_{nudge} was the angular nudge applied, F_n the net force on domain n and $\vec{r}_{n,nudge}$ the cartesian displacement of each domain n due to the nudge. This test is powerful since it links the force-calculation system of

either model to the energy calculation system. Energy is calculated by looking at displacements of angles from equilibrium, whereas forces are calculated using displacement of angles from equilibrium, but also orientation and position of each domain. Having the force system and energy system calculate the same results indicates that they are both likely correct.. **make this more precise.**

is there a test to be run which could verify the velocity calculation is correct?

3.2.3 Equipartition theorem

The equipartition theorem (ET) is a thermodynamic statement which says that all systems with solely quadratic energy dependence on N degrees of freedom will have $\langle E \rangle = \frac{N}{2}k_bT$. For the dynein models to display ET behavior, they would have to both properly calculate forces and properly calculate velocities from forces. Brownian forces are also required to properly explore the different states of the system, but they need not be the correct magnitude for ET agreement.

To test if the models obey the ET, the total potential energy of each model was logged over **ten million** (???) timesteps of simulation. The results are reported in Figure ??, where the PE scale is in units of $\frac{1}{2}k_bT$. Energies are reported as a running average over the simulation from 0 to time t . As shown, both models have their average potential energy converge to the expected amount, where the onebound model has four degrees of freedom and the bothbound two.

This test verifies both force and velocity are calculated properly for either model. If either force or velocity was wrong, then the model would not obey the Boltzmann hypothesis $P(s) = e^{-\Delta G_s/K_B T}$, where s is a state of the system. The transition rate from s_1 to s_2 is $P(s_1 \rightarrow s_2) = e^{-\Delta G_2 - \Delta G_1 / K_B T}$. Wrong velocities are those that do not transition the system from s_1 to s_2 at this rate.

would it be a stronger test to also calculate the standard deviation of $\langle PE \rangle$? It seems like $\langle PE \rangle$ would still equal $N/2K_B T$ as long as the forces/velocities were symmetrically wrong. Calculating the stdev would require the state occupation to actually the ET. Or, maybe just plot the angle occupation and show that it is what it should be?

Possibly do mutations on motion equations to show that tests are no longer behaved

Author	Conditions	Velocity	Run length
Qiu [10]	1mM [ATP]	134 ± 60.4 nm	$1.06 \pm 0.044 \mu m$
Reck-Peterson [11]	1mM [ATP]	85 ± 30 nm/s	$1.9 \pm 0.2 \mu m$

Figure 18: Literature values for experimental dynein stepping behavior.
Table of literature values for dynein behavior

3.3 Parameter fitting

There are 14 free variables in the dynein model: rigid rod lengths L_s and L_t , domain radii R_b , R_m and R_t , spring constants c_b , c_m and c_t , equilibrium angles $\theta_{b,eq}$, $\theta_{t,eq}$, $\Theta_{m,eq}^{Pre}$, $\Theta_{m,eq}^{Post}$, onebound binding rate k_b , and bothbound unbinding rate k_{ub} . Some of these values are found easily from literature or by looking at crystal structures of the motor protein, as shown in Table ??.

Others are more difficult to find. These values, including k_b , k_{ub} and the three spring constants, are fit by an optimization process. First un/binding rates are fit to one/bothbound times, then spring constants are fit to average step length. This process is chronicled below.

In addition, the physical conditions the simulations were run at are shown in Table ???. These conditions are meant to reflect the conditions found inside a typical cell. **of what species?**

	Winch (Sarlah)	Schmidt	Lin	PyMol 3VKH	PyMol 4RH7	Redwine	Kon	Burgess
L_s	12nm			21.0nm	22.1nm			
L_t	7nm				11.15nm			
R_b				1.57nm	1.45nm			
R_m	7nm			7.36nm	6.3nm			
R_t					2.16nm			
θ_m^{Pre}	250°		171°					160
θ_m^{Post}	330°		137.5°					136
θ_b	56°		63.5°					
θ_t	0°							
k_{ub}	$180s^{-1,a}$						90.2 ± 4.5	
k_b	$460s^{-1,b}$							
c_t								
c_m								
c_b						140 kJ/mol		

Figure 19: Literature values for dynein model parameters.
Table of literature values for various free parameters in dynein model.

T	[ATP]
310.15 K	1mM

Figure 20: Physical constants of simulation.
Table of physical constants used in simulation.

3.3.1 Fitting rate constants

To find the ideal binding and unbinding rates of our model, we ran simulations at various k_b and k_{ub} conditions. The time each simulation spent in each state, t_{bb} and t_{ob} , was compared with the experimental data shown in Table 14. Simulation behavior for a wide range of k_b and k_{ub} values is shown as a heat map in Figure ??.

Way to represent heatmap: show a blowup of the most promising section of the first heatmap, then a blowup of the second, etc until the optimum is found

3.3.2 Fitting spring constants

This section will be written once it is conducted. This is contingent on getting rate fitting done, and preferably high-dt simulations stable so an optimization process can be run.

CHAPTER: RESULTS & DISCUSSION

4.1 Validating model

The simulation was first validated for obeying the laws of physics, and in particular, the equipartition theorem. Figure 4.1 shows how both models have their angular potential energy converge to the expected $\frac{n}{2}k_B T$ energy over a long time. This is evidence that both the forces and velocities of the model are properly calculated. Grossly incorrect velocity or force calculations would lead to an incorrect occupation distribution of angles, and hence an incorrect average energy. The model's force calculations are further validated in Section 3.2.1, and energy conservation is validated in Section 3.2.2.

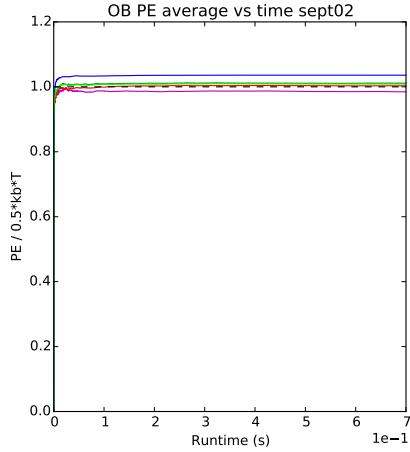


Figure 21: Onebound average spring energy.
Average domain PE of onebound model converging to $\frac{1}{2}k_B T$ energy over time.

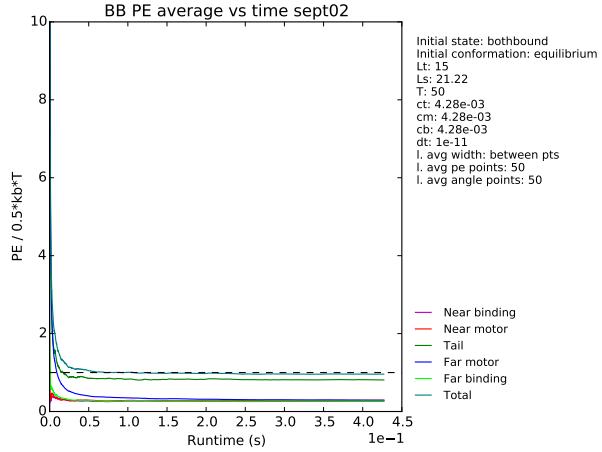


Figure 22: Bothbound average spring energy.
Total domain PE of bothbound model converging to $\frac{1}{2}k_B T$ energy over time.

4.2 Model achieves processivity

When run at the proper parameters, the dynein model can step processively. Figure 24 shows the dynamics of a motor when run at the parameters shown in Table 23.

k_b	k_{ub}	c_b	c_m	c_t	dt
$10^{18}s^{-1}$	$10^{20}s^{-1}$	$2.0\Delta G_{ATP}$	$2.0\Delta G_{ATP}$	$1.0\Delta G_{ATP}$	$10^{-10}s$

Figure 23: Parameters used in processive simulations in Figures 24 and 25.

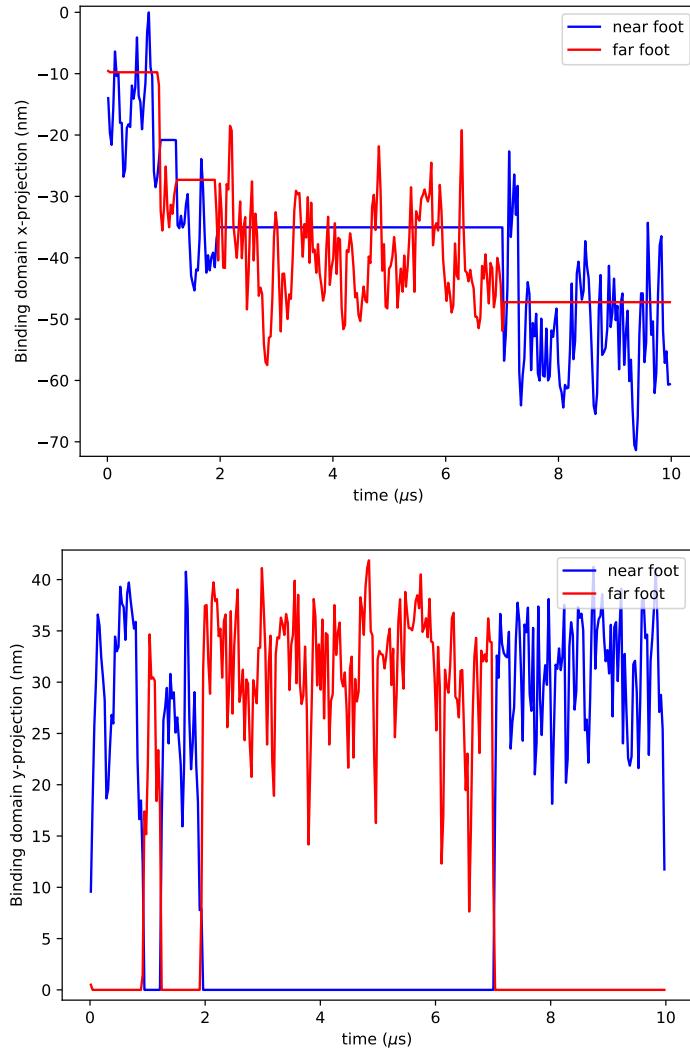


Figure 24: Near and far binding domains over the course of several microseconds of simulation. Simulation run at standard conditions with $k_b = 123\text{s}^{-1}$, $k_{ub} = 456\text{s}^{-1}$, $c_b = c_m = c_t = 789\Delta G_{ATP}$.

Figures 24.b and 24.c demonstrate the model stepping five times, where a step constitutes lifting a binding domain off the microtubule for any amount of time. This behavior indicates that a springy hinge model with shifting equilibrium angles is capable of generating processive stepping behavior. Now that the model has demonstrated stepping behavior, it must be seen if it can generate steps at a size- and time-scale similar to that which real dynein takes in experiment.

Another important determination from Figure 24.c is the average position of the unbound foot is about 30nm above the microtubule. This is a preliminary finding which may not represent how the model will behave once fit to stepping behavior, so drawing conclusive conclusions from it is impossible. However, it does initially suggest how dynein searches for the microtubule. Instead of hovering near the microtubule looking for a binding site, the motor pulls its foot away from the microtubule, diffusing until pushed downward far enough to bind. This behavior could explain why dynein's stepping pattern is so large: diffusing downward from 30nm would allow significant x-diffusion along the microtubule. This could be contrasted with kinesin, which is smaller and could not move its foot so high off the microtubule.

Importantly, the physical parameters causing processive motion are reasonable physically. The spring constants are roughly the same size as the energy of hydrolysis of ATP, which is reasonable. By a very rough argument, the dynein must move its tail angle by roughly one radian to diffuse enough for a step. This would

correspond to a $\Delta G_{spring} = \frac{1}{2}c_t\Delta\theta_t \approx \Delta G_{ATP}$, which is very reasonable. In addition, the tail spring constant is less than the motor and binding spring constants, which is in line with the notion that the tail is much more elastic than the motor or binding domains. CITE

4.3 Model step size is highly variant

Figure 25.a shows the stepping pattern of the model at the parameters shown in Table 23. This simulation was run over a much longer time period, and hence shows much more significant information on the model's stepping behavior.

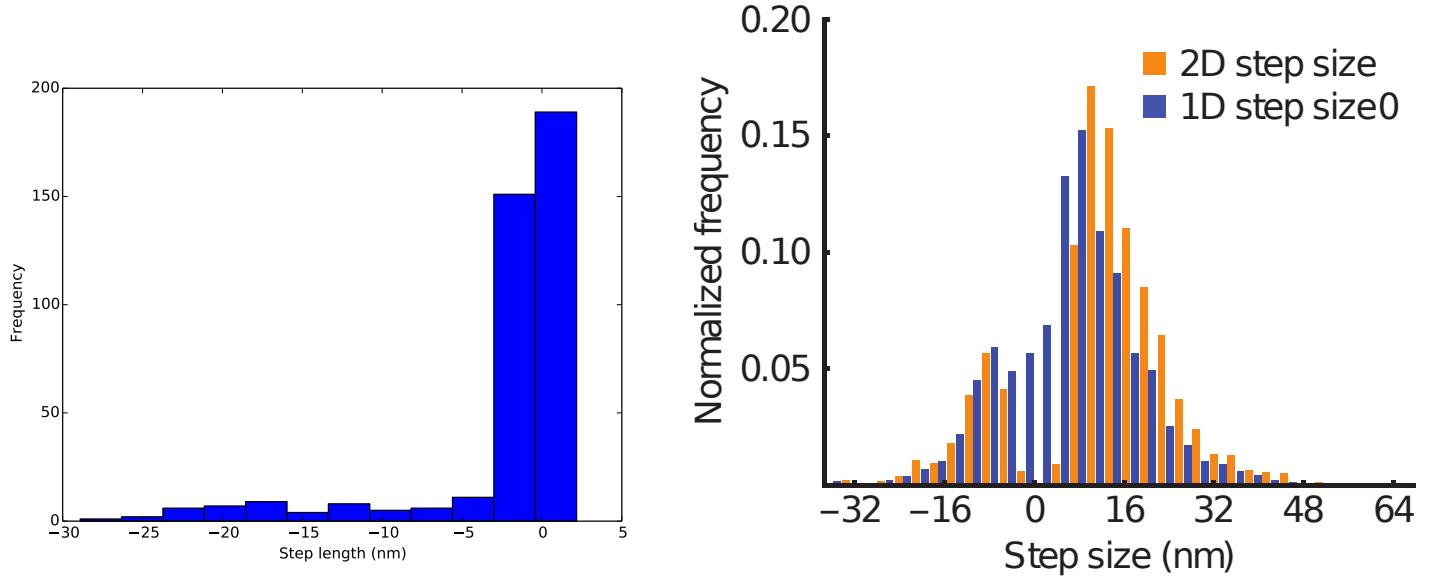


Figure 25: Model vs experimental step sizes. **a.)** Step sizes model took at parameters in 23. **b.)** Experimental stepping behavior of dynein [10].

Figure 25.a shows the model has a strong preference for $< 5\text{nm}$ steps, but is capable of generating steps up to 25nm in length, with a fairly even distribution of step frequency between 5 and 25nm . This is a very important result, since it indicates that the model is capable of taking a varied step size. This behavior is characteristic of dynein, and hence very important to reproduce.

The preference for $< 5\text{nm}$ steps is very un-dynein-like. One explanation is that steps may be defined differently between experimental and modeled dynein. In the simulation, every state transition where the motor unbinds from the microtubule is considered a step and plotted on the stepping histogram. This may not be how the experiment in 25.b records steps. Another possibility is that the binding rate used for these simulations is far too high, and thus, immediate rebinding is very likely. A solution would thus be to lower the microtubule binding rate and see if the stepping histogram would level out. This may help, but may not, since lowering the binding rate would affect all binding probabilities, not just immediate rebindings.

The ability of the model to generate near-experiment stepping behavior is imperative if the model is to be considered a proper explanation of dynein's stepping. In the case that changing the binding rate does not remove the small-step preference of the model, it may be necessary to change how the model works. For example, using the total spring-energy-of-binding change to calculate the binding probability may be invalid, or perhaps the entire two-state model will need to be thrown out. The inability of either of these assumptions to predict dynein's behavior would themselves be important findings.

Another issue with the simulated histogram is that it shows no steps over 30nm. This is a problem, since taking $40 + nm$ steps is characteristic of dynein. Ideally, tuning the parameters of dynein would increase the step size. However, in the case that it can't, this might be an important result. The current model assumes connections between domains are rigid. This may not be a valid assumption, since the coiled coil region connecting the binding and motor domains has some elasticity to it CITE. Thus it may be necessary to add in elasticity to the model to reproduce long steps.

Comparing the simulation results in 25.a with the experimental results in 25.b, it is clear that the model is still far from experimental dynein. However, this is not indicative of a failure of the model, since the simulation parameters have not yet been tuned to dynein. It is very possible that changing the binding rate, unbinding rate and spring constants will change the stepping behavior of the model dramatically. Lowering the binding rate may even out the histogram and prevent instant rebinding. It may also allow the model to diffuse along the microtubule longer, generating longer steps than 25nm. Another issue is the lack of backwards steps, which may also be solved by altering the spring constants.

CONCLUSION

5.1 Findings

The most important findings of this work are that a springy hinge model of dynein can take steps of a similar size to real dynein. What's more, it is likely further tweaking of the model will improve the model's behavior to the point of generating backwards and large steps. This would be a very important result which would strongly support the linker-swing/diffusive-step model of dynein stepping and guide further research.

Many studies have been done on dynein, resulting in a large body of literature about the motor's behavior and features. Sifting through this information to find what is important for the motor's step is a difficult task. Mutational studies can remove traits of the motor, deducing what traits are necessary for stepping. However, this process is slow, and it can be difficult to remove large traits of the motor while keeping the engineered dynein representative of the natural motor. This is where simulations come in. Minimal features can be put into a physical model, which can then be tested for behavior similar to real dynein. In this study, we created a mathematical model which simplifies the motor to a rigid hinge structure with harmonic spring energies between the hinges. The equilibrium angles change dynamically to reflect structural changes during dynein's mechanochemical cycle. The model was then subjected to Brownian forces and evolved. If the model generates a stepping pattern similar to real dynein, this would demonstrate that dynein can be thought of as a rigid hinged structure. This would mean that many other traits of dynein may not be vital for stepping, such as stalk elasticity [13], motor-motor interaction [13], nucleotide state [4], nucleotide state of non-AAA1 nuclease sites [4], and changing stiffness of the foot domain [2]. Thus future research may want to focus more on linker swing and binding interactions, and less on these other traits.

Other findings cast doubt on some aspects of the model. The vast preference of the model for $< 5nm$ steps suggests that the model may handle binding events unlike real dynein. When binding, the total spring energy of the transition from a onebound to a bothbound state is calculated, then used to calculate the likelihood of transition via a Boltzmann factor. This approach may be overly simple, or take into account the wrong information. It may be that it is necessary to expand the two-state model into a three-state model with a low-microtubule-affinity onebound state immediately after unbinding, to prevent instant rebinding. This state would eventually transition to the high-microtubule-affinity state after some time. If this proves to be necessary to decrease small steps, this would be an interesting result showing that the motor might rely on its low-microtubule-affinity ADP-Pi state to prevent early binding.

5.2 Further work

The goal of studying dynein's mechanochemical cycle is to find an explanation for how it works. That is, to find the necessary and sufficient properties of dynein that cause the cycle to happen. Simulations are only part of the recipe for creating a scientific theory. Simulations are easy to implement and can study phenomena which are difficult to design experiments for. However, when experimental data is available, it is generally more reliable than a simulation of the same system. This is because a simulation is only as good as the strength of correspondence between the mathematical model and the system it represents. With dynein, this correspondence can be hard to achieve because the protein is complicated. The model predicts, according to the laws of physics for small objects, how a small machine composed of springy hinges would behave in an aqueous environment. This machine can be made the same size and general shape as dynein, and its conformational angles can be made the same as dynein's, and its state transitions can happen in the same way dynein's do. However, it is never possible to say that this machine "is" dynein or represents how dynein actually behaves. Thus, it is only possible to say that the simulation provides sufficient criteria for dynein stepping, not necessary criteria. The simulation shows that the linker-swing/diffusive-step model is sufficient for motion, but it cannot show that this is needed for dynein's walk.

To more strongly support the linker-swing/diffusive step model of dynein motion, more experiments should be done. Demonstrating that dynein requires flexible domain connections in order to take such broad steps would be one powerful experiment. Mutating the foot and tail domains to be more rigid, then examining stepping behavior would be one way to see if rigidity is necessary for random steps. Another experiment could use FRET (Forster Resonance Energy Transfer) and fluorophores on the microtubule and foot domain to track how the distance between the two regions evolves with time. This could provide information about the average distance between the MTBD and MT, which would validate this work's finding that the MTBD mostly hovers. More importantly, FRET data could be fit to models to predict the amount of diffusion the foot domain undergoes over the course of a step. Significant diffusion would indicate that diffusion may be very important for the step. Experiments with both feet labeled could get similarly good information.

Talk to Dr. Hsu about this sort of thing...what experiments could be used to validate the model?

If the model were validated by FRET and rigidity mutation studies, then there are many other interesting things which could be found from it. First, the amount of work done by the motor could be found by inducing an artificial drag force pulling the protein's motor domains against its direction of travel. This force could be integrated over dynein's trajectory to find the work the model exerted on its artificial cargo. This could be used to estimate the efficiency of the motor. Another interesting study would be to add binding sites to the model, only allowing it to bind at $\alpha - \beta$ tubulin dimerization sites, like real dynein. This would mandate re-fitting the binding constants to the new environment. It would be interesting to know if this model behaves more, or less dynein-like than the original. This would help answer the question of whether dynein searches for a binding site on the microtubule, or if it is naturally tuned to bind at the right site.

Another very interesting simulation which could be run could help answer the open question of why kinesin and dynein, though using similar mechanisms, produce such different steps. Kinesin, unlike dynein, conducts all of its chemistry directly on its foot. This translates to effectively having two fewer joints than dynein. Kinesin's stroke is believed to involve a similar foot unbinding, kicking forward, rebinding, then lurching the motor forward mechanism. The dynein model could be used to construct a very rough model of kinesin by shrinking the tail linkers and increasing the motor domain spring constant c_m high enough to simulate a rigid hinge. This model could then be reduced to the size of kinesin, given the proper equilibrium angles, and simulated with Brownian dynamics. If this new model even roughly manages to mimic experimental kinesin stepping, that would be fascinating. This result would show that the differences between kinesin and dynein can be explained exclusively through the values of spring constants c_t , c_m , c_b , lengths L_s and L_t , equilibrium angles and un/binding constants. This would mean the differences between the two motors are primarily due to the conformational angles the motors take with the microtubule and between their own domains, not because of differences in their mechanism.

Next: add Big Questions to introduction, then answer them here.

REFERENCES

- [1] J. Andreasson, S. Shastry, W. Hancock, and S. Block. The mechanochemical cycle of mammalian kinesin-2 kif3a/b under load. *Curr Biol.*, 9(25):1166–1175, 2015.
- [2] S. Burgess, M. Walker, Sakakibara, P. H., Knight, and K. Oiwa. Dynein structure and power stroke. *Nature*, 421:715–718, 2003.
- [3] N. J Carter and R. A. Cross. Mechanics of the kinesin step. *Nature*, (435):308–312, 2005.
- [4] Michael A. Cianfrocco, Morgan E. DeSantis, Andres E. Leschziner, and Samara L. Reck-Peterson. Mechanism and regulation of cytoplasmic dynein. *Annual Reviews of Cell and Developmental Biology*, 31(Nov):83–108, 2015.
- [5] A. Einstein. On the movement of small particles suspended in a stationary liquid demanded by the molecular-kinetic theory of heat. 1989.
- [6] T. Kon, K. Imamura, A. Roberts, R. Ohkura, P. Knight, IR. Gibbons, S. Burgess, and K. Sutoh. Helix sliding in the stalk coiled coil of dynein couples atpase and microtubule binding. *Nat Struct Mol Biol.*, 16:325–333, 2009.
- [7] Takahide Kon, Toshifumi Mogami, Reiko Ohkura, Masaya Nishiura, and Kazuo Sutoh. Atp hydrolysis cycle-dependent tail motions in cytoplasmic dynein. *Nature Structure and Molecular Biology*, 12:513–519, 2005.
- [8] J. Lin, K. Okada, M. Raytchev, M. Smith, and D. Nicastro. Structural mechanism of the dynein power stroke. *Nature Cell Biology*, 16:479–485, 2014.
- [9] Emin Oztas. Neuronal tracing. *Neuroanatomy*, 2(1):2–5, 2003.
- [10] Weihong Qiu, Nathan D. Derr, Brian S. Goodman, Elizabeth Villa, David Wu, William Shih, and Samara L. Reck-Peterson. Dynein achieves processive motion using both stochastic and coordinated stepping. *Nat Struct Mol Biol.*, 19(2):193–200, 2012.
- [11] Samara L Reck-Peterson, Ahmet Yildiz, Andrew P Carter, Arne Gennerich, Nan Zhang, and Ronald D Vale. Single-molecule analysis of dynein processivity and stepping behavior. *Cell*, 126(2):335–348, 2006.
- [12] W. B Redwine, R. Hernandez-Lopez, S Zou, J. Huang, S. L Reck-Peterson, and A. E. Leschziner. Structural basis for microtubule binding and release by dynein. *Science*, 337(6101):1532–1536, 2012.
- [13] Andreja Sarlah and Andrej Vilfan. The winch model can explain both coordinated and uncoordinated stepping of cytoplasmic dynein. *Biophys J*, 3(107):662–671, 2014.
- [14] H. Schmidt, ES. Gleave, and AP. Carter. Insights into dynein motor domain function from a 3.3a crystal structure. *Nat Struct Mol Biol.*, 19:492–497, 2012.
- [15] H*. Schmidt, R*. Zalyte, L. Urnavicius, and AP. Carter. Structure of human cytoplasmic dynein-2 primed for its power stroke. *Nature*, 518:435–438, 2014.
- [16] Ronald D. Vale. The molecular motor toolbox for intracellular transport. *Cell*, 112(4):467–480, 2003.
- [17] S. Wulfa, V. Ropars, S. Fujita-Becker, M. Oster, G. Hofhaus, L. Trabuco, O. Pylypenko, L Sweeney, A. Houdesse, and R. Schröder. Force-producing adp state of myosin bound to actin. *Proceedings of the National Academy of Sciences*, 113(3):E1844–E1852, 2016.
- [18] Mitsui Y and Schneider EL. Relationship between cell replication and volume in senescent human diploid fibroblasts. *Mech Ageing Dev.*, 5(1):45–56, 1976.

APPENDIX

6.0.1 Glossary

ATP Adenosine triphosphate is a small organic molecule which very spontaneously breaks apart into ADP and a phosphate. This reaction is used to power many energy-requiring mechanisms within the cell. This reaction has a $\Delta G^\circ \approx -36\text{kJ/mol}$ at physiological conditions ??.

Protein Dynein is a protein, or long chain of amino acids linked together and folded into a particular shape.

6.1 Motion equations

6.1.1 Onebound equations

The following is a continuation of the derivation for the full motion equations used to time evolve the onebound model:

$$AA = L_s \sin \theta_{bb}$$

$$BB = L_t \sin(\theta_{bm})$$

$$CC = -L_t \sin(\theta_{um})$$

$$DD = -L_s \sin(\theta_{ub})$$

$$LL = -L_s \cos(\theta_{bb})$$

$$MM = -L_t \cos(\theta_{bm})$$

$$NN = L_t \cos(\theta_{um})$$

$$OO = L_s \cos(\theta_{ub})$$

$$EE = -\gamma_m(X_{bm} - X_{bb})$$

$$FF = \gamma_m(X_t - X_{bm})$$

$$GG = -\gamma_t(X_t - X_{bm})$$

$$HH = \gamma_t(X_{um} - X_t)$$

$$II = -\gamma_m(X_{um} - X_t)$$

$$JJ = \gamma_m(X_{ub} - X_{um})$$

$$KK = -\gamma_b(X_{ub} - X_{um})$$

$$PP = -\gamma_m(Y_{bm} - Y_{bb})$$

$$QQ = \gamma_m(Y_t - Y_{bm})$$

$$RR = -\gamma_t(Y_t - Y_{bm})$$

$$SS = \gamma_t(Y_{um} - Y_t)$$

$$TT = -\gamma_m(Y_{um} - Y_t)$$

$$UU = \gamma_m(Y_{ub} - Y_{um})$$

$$VV = -\gamma_b(Y_{ub} - Y_{um})$$

$$\begin{pmatrix} AA & 0 & 0 & 0 & EE & FF & 0 & 0 \\ AA & BB & 0 & 0 & 0 & GG & HH & 0 \\ AA & BB & CC & 0 & 0 & 0 & II & JJ \\ AA & BB & CC & DD & 0 & 0 & 0 & KK \\ LL & 0 & 0 & 0 & PP & QQ & 0 & 0 \\ LL & MM & 0 & 0 & 0 & RR & SS & 0 \\ LL & MM & NN & 0 & 0 & 0 & TT & UU \\ LL & MM & NN & OO & 0 & 0 & 0 & VV \end{pmatrix} \begin{pmatrix} \dot{\theta}_{bb} \\ \dot{\theta}_{bm} \\ \dot{\theta}_{um} \\ \dot{\theta}_{ub} \\ \lambda_{bs} \\ \lambda_{bt} \\ \lambda_{ut} \\ \lambda_{us} \end{pmatrix} = \begin{pmatrix} X1 \\ X2 \\ X3 \\ X4 \\ X5 \\ X6 \\ X7 \\ X8 \end{pmatrix}$$

Solving the above system for $\dot{\theta}_{bb}, \dot{\theta}_{bm}, \dot{\theta}_{um}$ and $\dot{\theta}_{ub}$ gives the desired velocities. Solution was done via Mathematica.

6.1.2 Bothbound equations

The following are the full motion equations used to time evolve the bothbound model:

$$\begin{aligned}
a &= \frac{-dXnm}{dLn} & m &= \frac{-dYnm}{dLn} \\
b &= \frac{-dXnm}{dLf} & n &= \frac{-dYnm}{dLf} \\
c &= -(x_{nm} - x_{nb})/\gamma_m & p &= -(y_{nm} - y_{nb})/\gamma_m \\
d &= (x_t - x_{nm})/\gamma_m & q &= (y_t - y_{nm})/\gamma_m \\
e &= \frac{-dXt}{dLn} & r &= \frac{-dYt}{dLn} \\
f &= \frac{-dXt}{dLf} & s &= \frac{-dYt}{dLf} \\
g &= -(x_t - x_{nm})/\gamma_t & t &= -(y_t - y_{nm})/\gamma_t \\
h &= (x_{fm} - x_t)/\gamma_t & u &= (y_{fm} - y_t)/\gamma_t \\
i &= \frac{-dXfm}{dLn} & v &= \frac{-dYfm}{dLn} \\
j &= \frac{-dXfm}{dLf} & w &= \frac{-dYfm}{dLf} \\
k &= -(x_{fm} - x_t)/\gamma_m & x &= -(y_{fm} - y_t)/\gamma_m \\
l &= (x_{fb} - x_{fm})/\gamma_m & y &= (y_{fb} - y_{fm})/\gamma_m
\end{aligned}$$

Solving the above system for $\dot{\Theta}_{nm}$ and $\dot{\Theta}_{fm}$ gives the desired velocities. Solution was done via Mathematica.

QUESTIONS FOR TATE/NICOLE

QUESTIONS FOR DAVID

- Would it be possible to, for the ET test, also verify the standard deviation of energy is what it should be? Since just verifying the energy averages to $N/2KbT$ doesn't verify the forces are correct, but the stdev would?
- What is the value of $\langle \Delta G_{conf} \rangle$ in going from bothbound to onebound?

TODO FOR PROJECT

- Do calculations to turn processivity into state time ratios
- Fit the state times to above number by tweaking un/binding rates
- Fit spring constants to step length histograms