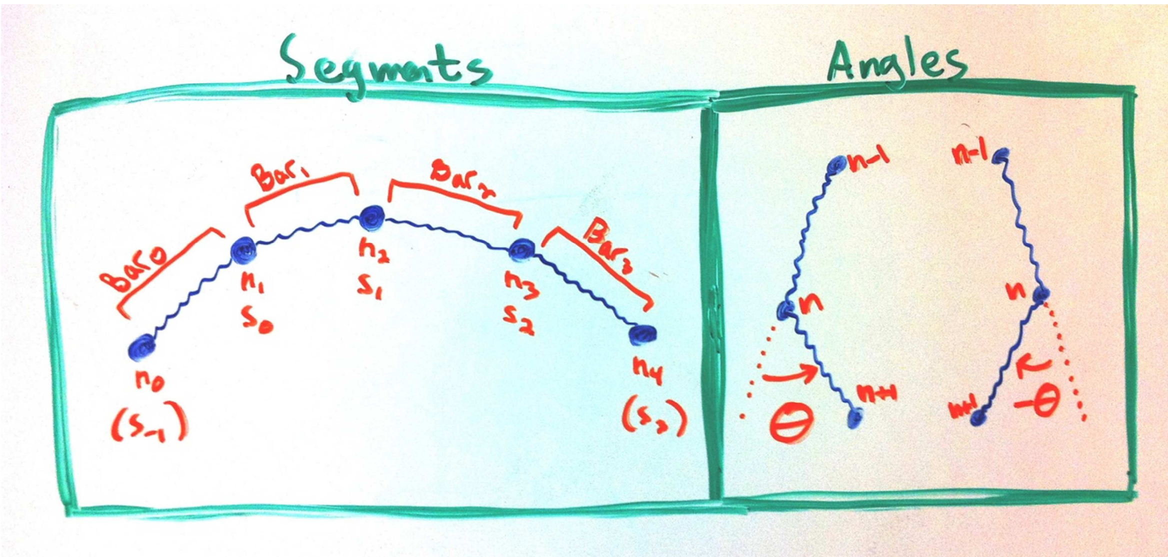
LinearBody Model

Overview

The LinearBody model uses classical mechanics to simulate sinusoidal locomotion observed in nematode worms in Newtonian fluids of high viscosity. The goal of the model is to provide a reasonably accurate measure of thrust, which may be used as a fitness function in evolutionary algorithms. Due to the focused nature of the use case, the model was designed to be as simple as possible, often times at the sacrifice of physical reality. The model consists of a collection of s+2 point masses (called "nodes"), where s is the number of segments. A segment consists of three adjacent nodes, with the center nodes' index relating to the index of the segment. The line between point masses will henceforth be called a "bar." Counter intuitively, there are s+1 bars in a model with s segments. The angle at a given segment is defined as the deviation of bar n+1 from the direction of bar n-1. Below is a diagram to illustrate this:



Forces

The model takes into account four forces: Stoke's drag, kinetic friction, linear springs, and torsional forces (torsional spring and muscle). I explain each force in detail below:

Perpendicular Stoke's Drag:

Stoke's drag is appropriate for objects moving through relatively viscous fluids, at low speeds, with no turbulence. The drag equation reminds one of kinetic friction (also used in this model).

$$\boldsymbol{F} = -b\boldsymbol{v}$$

Where b is a physical constant related to the shape of the object and physical properties of the fluid. Further, I have limited this drag to only act in on direction: Perpendicular to the bar. The drag experienced by any node is therefore the sum of half of the perpendicular stokes drag of adjacent bars.

$$\boldsymbol{F}_n = -\frac{b}{2}\left[dist(n, n-1)\boldsymbol{Norm_{n-1}} \cdot \left(\frac{\boldsymbol{v}_n + \boldsymbol{v}_{n-1}}{2}\right) + dist(n, n+1)\boldsymbol{Norm_{n+1}} \cdot \left(\frac{\boldsymbol{v}_n + \boldsymbol{v}_{n+1}}{2}\right)\right]$$

Where $\boldsymbol{Norm}_n$ denotes the unit normal to the bar between n and the subscripted node, facing towards the left if n-1 is above n, and $dist(n_1, n_2)$ is the distance between nodes $n_1$and n_2. It may be difficult to see how this equation arises from the definition of Stoke's drag, but if you multiply $\frac{b}{2}$ into the equation and recognize that the velocity of a bar is the average of the velocities of its endpoints, it may begin to make sense. There is one non-obvious assumption in this approach:

- No torque is applied to the bar by drag. As the length of a bar gets smaller, this omission becomes less important until, in the limit, it doesn't matter at all. Adding torque due to drag would serve to destabilize the larger body in motion, possibly complicating the evolutionary task at hand. This was seen as a reasonable sacrifice in the pursuit of simplicity.

Kinetic Friction:

Friction was not originally intended to be included in this model, and is not necessary for proper simulation, but proves useful in overcoming certain cases in which the body does not behave as intended. Without kinetic friction, for example, the body can "glide" forever once it has straightened out. I also observed a case where the body simply curled up into a circle and rotated forever. These cases are not dangerous, but can lead to unrealistic solutions with good fitness scores. Friction helps avoid such things, requiring forward thrust to be sustained for long distances to be covered.

$$\boldsymbol{F}_n = -k\boldsymbol{v}_n$$

Where k is the coefficient of friction for each node.

Linear Springs:

All springs in this model are dampened, linear, Hooke's law springs. Linear springs connecting nodes tend to require heavy dampening to avoid chaotic oscillation and cases where adjacent nodes are forced over one another. The force on node n due to a dampened adjacent spring to n-1 is:

$$F_{n,n-1} = K(L - x) - D\frac{dx}{dt}$$

Where *K* is the linear spring constant, *L* is the equilibrium length, and *D* is the dampening coefficient.

Torques:

Torques are calculated only about segment nodes (surrounded by two adjacent nodes). End nodes do experience forces due to torque, but this is discussed later. The torque produced by a torsional spring at segment s is:

$$\tau_{ss} = K(\pi - \theta_s) - D\frac{d\theta_s}{dt}$$

Where θ is the angle created by drawing an arc counterclockwise from the line segment between nodes n -- n-1, and the line segment n -- n+1. The motivation behind the torsional spring was to keep the body straightened when no other forces are acting upon it. The springs also serve as a counterbalance to the muscle force so that muscles encounter a resistance as they bend a segment. The torque of muscle is simply:

$$\tau_{sm} = mi_s$$

Where *m* is a constant that roughly represents the "strength" of the muscle, and *i* is an externally produced input. Do note: The model does not make any assumptions or impose any bounds on this input. Neither does my implementation. There is no functional difference between setting a muscle of strength 2 to input 0.5, and setting a muscle of strength 1 to input 1. The constant multiplier is only there for convenience. Total torque produced at segment s is therefore:

$$\tau_s = \tau_{ss} + \tau_{sm}$$

The torque at a given segment node s is interpreted as a force on each adjacent node s-1 and s+1 normal to the bar between them. This, in turn, produces two reactive forces on s itself. The force on adjacent nodes due to torque at node s is:

$$\boldsymbol{F}_{s-1} = \boldsymbol{Norm}_{\boldsymbol{s-1}} \frac{\tau_s}{dist(s, s-1)}$$

$$\boldsymbol{F}_{s+1} = \boldsymbol{Norm}_{\boldsymbol{s+1}} \frac{\tau_s}{dist(s, s+1)}$$

Where $dist(s_1, s_2)$ and $\boldsymbol{Norm}_{\boldsymbol{s}}$ are defined as in the Stoke's drag force. This equation arises naturally from the definition of torque: $\tau = ||\boldsymbol{F}||d$, where d is the length of the lever arm. The reactive force on $\boldsymbol{F}_s$ is then, trivially, the negative sum of the above forces:

$$\boldsymbol{F}_s = -(\boldsymbol{F}_{s-1} + \boldsymbol{F}_{s+1})$$