

# Efficient sliding locomotion of a two-link scallop

Connor Puritz

University of Michigan – Ann Arbor

July 22, 2019

*Abstract.*— We analyze the locomotion of a two-link scallop-like body and optimize the efficiency of it’s motion. The body consists of two equal-length segments whose opening angle varies periodically with time; the angle function is prescribed by a finite dimensional parameter space. Sliding locomotion is examined in several frictional regimes, and a genetic algorithm is used to find optimal angle functions which give rise to the most efficient movements.

## 1 Introduction

Simulating snake locomotion has been a topic of interest in recent years from both biological and engineering contexts (Silas 2019). Studies have taken several approaches to modeling snake bodies. Hu and Shelley (2012) modeled a snake as a smooth, inextensible curve in the plane, whereas Silas (2019) modeled a snake as a three-link body with changes in two angles defining the motion of the body.

We will examine a similar but simpler multi-link body, which we call a scallop. A scallop is composed of two equal length segments which are symmetric about the  $x$ -axis. The half-angle between the segments is given by the time-varying function  $\theta(t)$  with period  $\tau$ . A scallop with total length  $L$  can be parametrized as

$$\begin{aligned} x(s, t) &= x_{\text{CM}}(t) + (|s| - 1/4) \cos(\theta(t)), \\ y(s, t) &= s \sin(\theta(t)), \quad -L/2 \leq s \leq L/2, \end{aligned} \tag{1}$$

where  $x_{\text{CM}}$  denotes the  $x$ -coordinate of the center of mass of the scallop.

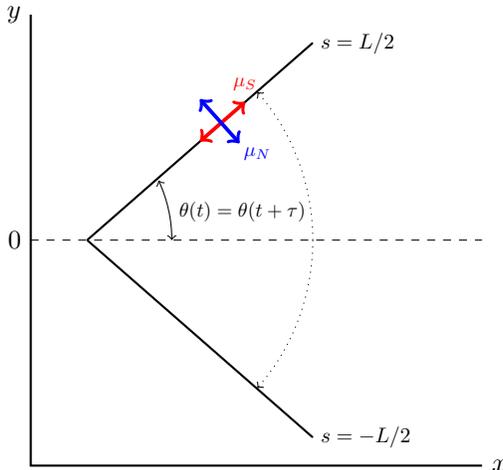


Figure 1: Schematic diagram of a scallop. Normal and tangential forces are shown in blue and red, respectively.

We assume that normal and tangential frictional forces are acting on the scallop. The frictional forces are modeled as spatially distributed Coulomb kinetic frictional forces, with the force per unit length given by

$$\mathbf{f}_\delta(s, t) = -\rho g \mu_n \left( \widehat{\partial_t \mathbf{X}} \cdot \hat{\mathbf{n}} \right) \hat{\mathbf{n}} - \rho g \mu_s \left( \widehat{\partial_t \mathbf{X}} \cdot \hat{\mathbf{s}} \right) \hat{\mathbf{s}}.$$

Here,  $\rho$  is the body mass per unit length (assumed to be uniform) and  $g$  is the gravitational constant. The unit normal and unit tangent vectors to the scallop are

$$\begin{aligned} \hat{\mathbf{n}}(s, t) &= (-\sin \theta(t), \operatorname{sgn}(s) \cos \theta(t)), \\ \hat{\mathbf{s}}(s, t) &= (\operatorname{sgn}(s) \cos \theta(t), \sin \theta(t)). \end{aligned}$$

$\widehat{\partial_t \mathbf{X}}$  is the normalized velocity vector, given by

$$\widehat{\partial_t \mathbf{X}}(s, t) = \frac{(\partial_t x, \partial_t y)}{\sqrt{\partial_t x^2 + \partial_t y^2 + \delta^2}}.$$

The constant  $\delta \ll 1$  is a small regularization parameter used to smooth out the discontinuity at zero velocity. We use a value of  $\delta = 10^{-8}$ .

Since the scallop is bilaterally symmetric about the  $x$ -axis, there is zero net vertical force and zero net torque due to friction. Thus the motion of the scallop is entirely described by the equation of motion for its  $x$ -coordinate, with  $f_{\delta x}$  being the  $x$ -component of  $\mathbf{f}_\delta$ :

$$\int_{-L/2}^{L/2} \rho \partial_{tt} x(s, t) ds = \int_{-L/2}^{L/2} f_{\delta x}(s, t) ds. \quad (2)$$

Due to bilateral symmetry, we can reduce each integral in (2) to twice the integral from 0 to  $L/2$ . Furthermore, since the term  $(|s| - 1/4)$  in  $x(s, t)$  is symmetric about  $1/4$ , that whole term vanishes from the lefthand side upon integration, and thus the lefthand side reduces to  $\ddot{x}_{\text{CM}}$ . Finally, we nondimensionalize the system by scaling  $s$  by  $L$  and  $t$  by  $\tau$ , the period of  $\theta$ . The final nondimensionalized equation of motion is

$$\begin{aligned} \text{Fr} \ddot{x}_{\text{CM}}(t) &= 2 \int_0^{1/2} -\mu_N \left( \widehat{\partial_t \mathbf{X}} \cdot \hat{\mathbf{n}} \right) \hat{\mathbf{n}}_x \\ &\quad - \mu_S \left( \widehat{\partial_t \mathbf{X}} \cdot \hat{\mathbf{s}} \right) \hat{\mathbf{s}}_x ds. \end{aligned} \quad (3)$$

Fr, defined as

$$\text{Fr} = \frac{L}{g\tau^2},$$

is called the Froude number, and is a ratio of body acceleration to gravitational acceleration. For simplicity, we assume the scallop starts at rest and at the origin, so  $x_{\text{CM}}(0) = \dot{x}_{\text{CM}}(0) = 0$ . We will start by taking  $\theta$  to be of the form

$$\theta(t) = A_0 + A_1 \cos(2\pi t). \quad (4)$$

$A_0$ , the half-angle bias, can be restricted to

$$0 \leq A_0 \leq \pi/2. \quad (5)$$

since a scallop with  $A_0 = \pi/2 + \alpha$  has the same motion as a scallop with  $A_0 = \pi/2 - \alpha$ , just reflected about the  $y$ -axis. We restrict  $A_1$  to

$$0 \leq A_1 \leq A_0, \quad (6)$$

since a scallop with  $A_1 > A_0$  has its two links cross over each every period which is physically unrealistic.

## 2 Numerical methods

The equation of motion (3) was solved using a second order midpoint Runge-Kutta method, with a time step of  $\Delta t = 10^{-3}$ . All integrals were evaluated using a second-order trapezoidal rule with 100 partitions. The time step and number of partitions were chosen as a balance between computational efficiency and accuracy.

Due to the constant period of  $\theta$  (nondimensionalized to 1), a scallop converges to a consistent periodic behavior rather quickly. To quantify this steady state, we say that a scallop has reached a steady state once the time averaged speed change by less than 1% between periods:

$$\left| \frac{\int_{t+2}^{t+1} \dot{x}_{\text{CM}} dt' - \int_t^{t+1} \dot{x}_{\text{CM}} dt'}{\int_t^{t+1} \dot{x}_{\text{CM}} dt'} \right| < 0.01.$$

A similar condition is used in Hu and Shelly (2012). In simulations that were run in several frictional regimes for different values of  $A_0$  and  $A_1$ , scallops took no more than 8 periods to reach a steady state. To be safe though, we will always calculate the motion of a scallop through 10 periods.

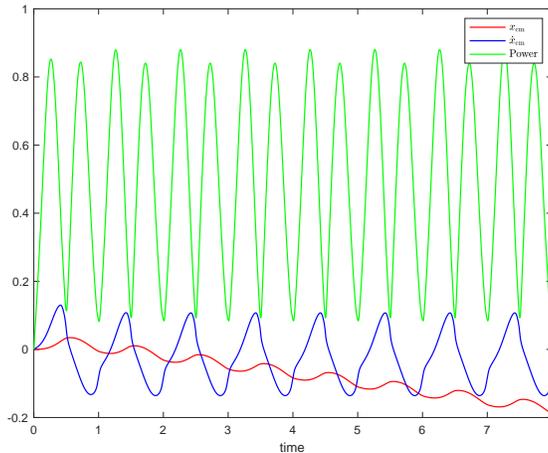


Figure 2: Example motion of a scallop with  $A_0 = \pi/2 - 0.7$  and  $A_1 = 0.5$  in the frictional regime  $\mu_N/\text{Fr} = 1$  and  $\mu_S/\text{Fr} = 2$ . The position of the center of mass is in red, the velocity of the center of mass in blue, and the power exerted by the scallop in green.

### 3 Optimizing locomotion

A common way to define the mechanical efficiency of locomotion is

$$\lambda = \frac{\langle U \rangle}{\langle P \rangle}, \quad (7)$$

where  $\langle U \rangle$  is the time-averaged speed over a single period and  $\langle P \rangle$  is the time averaged power over a single period (Silas 2019). The idea behind this definition is that a particular motion is considered efficient if it moves the body most (large  $\langle U \rangle$ ), while having requiring a small cost to do so (low  $\langle P \rangle$ ). However, a single coefficient is not necessarily the best way to examine efficient classes of locomotions. We may want to make some kind of tradeoff between  $\langle U \rangle$  and  $\langle P \rangle$ , and if the tradeoff does not increase  $\lambda$ , then looking at  $\lambda$  will not tell us the best way to go about making the tradeoff.

We will instead be using a different concept of efficiency. Let  $A \subset \mathbb{R}^n$  be the space of parameters used to define  $\theta$ , and call  $\langle U \rangle, \langle P \rangle^{-1} : A \rightarrow \mathbb{R}$  our objective functions, both of which we want to maximize. A solution  $X \in A$  is said to dominate a solution  $Y \in A - \{X\}$  if  $\langle U \rangle(X) \geq \langle U \rangle(Y)$  and  $\langle P \rangle^{-1}(X) \geq \langle P \rangle^{-1}(Y)$ , with at least one of the inequalities strict. The Pareto frontier is defined as the set of nondominated solutions, or solutions  $X \in A$  which are not dominated by any other solution in  $A$ . By looking at the Pareto frontier of this problem, we can examine all possible tradeoffs we can make with regards to  $\langle U \rangle$  and  $\langle P \rangle$ , and know the best ways to go about making these tradeoffs.

In the simple case of  $\dim(A) = 2$ , we can easily find the Pareto frontier using a brute force approach. However, this process is time consuming and certainly not a viable approach more any more complex systems.

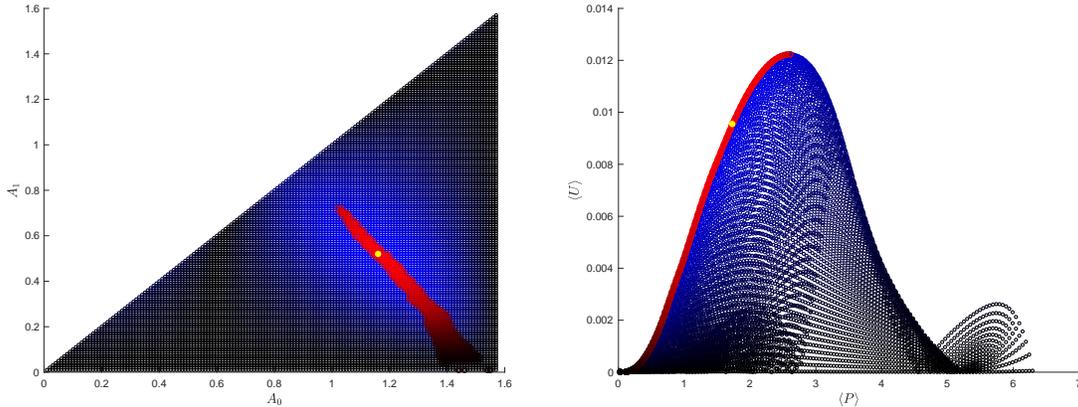


Figure 3: Pareto frontiers in objective and parameter spaces with  $\mu_N/\text{Fr} = \mu_S/\text{Fr} = 5$ . Blue solutions are all solutions calculated using brute force, red solutions are the dominant solutions. Lighter colors indicate higher efficiency, darker colors indicate lower efficiency. The yellow solution has the highest efficiency.

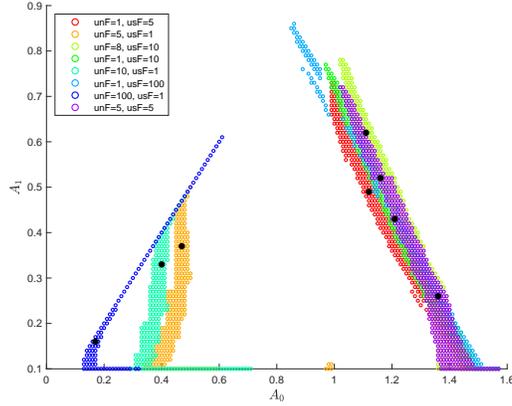


Figure 4: Pareto frontiers in parameter space for various combinations of  $\mu_N/\text{Fr}$  and  $\mu_S/\text{Fr}$ .

## 4 Extending the model

A natural extension to this model is to consider a more complex angle function, such as

$$\theta(t) = A_0 + \sum_{n=1}^M B_n \cos(2\pi nt) + C_n \sin(2\pi nt). \quad (8)$$

With even a small  $M$ , the Pareto frontier for such a problem would be too expensive to compute using a brute force approach. Therefore, we will instead use a genetic algorithm. Genetic algorithms are a metaheuristic which attempt to find optima of a problem using a procedure inspired by the biological process of natural selection. They are popular for finding Pareto frontiers as they can be used to generate sets of optimal solutions rather than converging to a single maximally optimal solution. In particular, we use the popular NSGA-II algorithm (Deb et al. 2002). We give a brief outline of the algorithm below.

### 4.1 Initialization

To setup the algorithm, we first must choose a population size  $N$ . Each individual  $K_i$ ,  $i \leq N$  in the population must be assigned a genetic representation, or chromosome, which encodes its position in the parameter space of the optimization problem. In our case, each gene is a parameter value, and the chromosome is represented as  $H = (A_0, B_1, \dots, B_M, C_1, \dots, C_M)$ . Next, a fitness function must be constructed, which corresponds to the objective functions of the optimization problem. So our fitness function is

$$f(K_i) = (\langle U \rangle|_{H_i}, \langle P \rangle|_{H_i}). \quad (9)$$

The population is then sorted into disjoint subsets (nondomination fronts) based on the following procedure, called nondominated sorting:

1. Initialize a rank counter equal to 1.
2. Select all individuals in the population which are not dominated by any other individual in the population. Set the rank of these individuals as the value of the rank counter.
3. Remove all individuals of the current rank from the population, and increment the rank counter.
4. Repeat steps 2-3 until all individuals have been assigned a rank.

The idea behind this sorting is to assign all individuals to disjoint nondomination fronts. As the algorithm proceeds, more individuals should join the front with rank 1, and this front should approach the true Pareto frontier.

## 4.2 Selection

Once the initial population has been sorted, we select  $N/2$  pairs of individuals, with replacement, to serve as the parents for the next generation. Pairs of parents are chosen using a procedure known as binary tournament selection:

1. Choose two distinct individuals randomly from the population.
2. Select the individual with highest rank to be a parent. If the two chosen individuals have the same rank, choose the individual with the highest crowding distance.
3. Repeat steps 2-3 to choose a second parent.

Crowding distance is a measure of how densely packed individuals are in the objective space. It can be calculated as

$$C(K_i) = \sum_{j=1}^N \|f(K_i) - f(K_j)\|.$$

To avoid premature convergence of the algorithm, we want to maintain some level of diversity in our population. Individuals with larger crowding distances are located in regions of the objective space that are less densely packed, and therefore increase the diversity of the population.

## 4.3 Genetic operators

Once the  $N/2$  pairs of parents have been selected, they are used to generate  $N$  children, two per pair of parents. The parental chromosomes are first copied, and the children are generated from these using two operators: mutation and crossover. There are many possible implementations of these operators; the ones we are currently using are point-wise mutation and crossover operators, although they will likely be changed as we decide what works best for this problem.

For mutation, a low probability of mutation  $p_M$  and a maximum mutation value  $\Delta F$  must be set. Currently, we are using  $p_M = 0.2$  and  $\Delta F = 0.1$ . Each gene  $g$  in each copy of the parental chromosomes is then mutated as  $g = g + BU$ , where  $B \sim \text{Bernoulli}(p_M)$  and  $U \sim \text{Unif}(-\Delta F, \Delta F)$ . For crossover, a high crossover probability  $p_C$  must be set. Currently, we are using  $p_C = 0.6$ . Then for each gene  $g$ , we swap  $g$  between the copies of the parental chromosomes with probability  $p_C$ . If  $g$  goes outside the bounds (equations (5), (6)) for whichever parameter it represents, then we set  $g$  equal to that bound. Once mutation and crossover

have been performed, the altered copies of the parental chromosomes are now the two children of that pair of parents.

#### 4.4 Survivor selection

After  $N$  children have been generated, they are recombined with the parents into a superpopulation of size  $2N$ . We perform a nondominated sort on the superpopulation, and select individuals of smaller ranks until  $N$  have been selected. If a non domination front has more individuals than are left to be selected, than individuals are selected based on maximal crowding distance. These selected  $N$  individuals now make up the next generation, and steps 2-4 are repeated.

#### 4.5 Termination

A genetic algorithm can never guarantee the optimality of its results, so there is not a natural stopping point. Instead, we decide to halt the algorithm in a two stage process. First, a certain number of generations are guaranteed to run, somewhere between 100 and 1000. Afterwards, the algorithm keeps running until the relative change in cumulative fitness between generations falls below 1%:

$$\left| \frac{\text{Gen}_{n+1} (\sum \|f(K_i)\|) - \text{Gen}_n (\sum \|f(K_i)\|)}{\text{Gen}_n (\sum \|f(K_i)\|)} \right| < 0.01.$$

Running a large number of generations first should bring the population close to the Pareto frontier, and the second condition ensures that the population is not changing greatly anymore, which is expected once the population converges to the frontier.

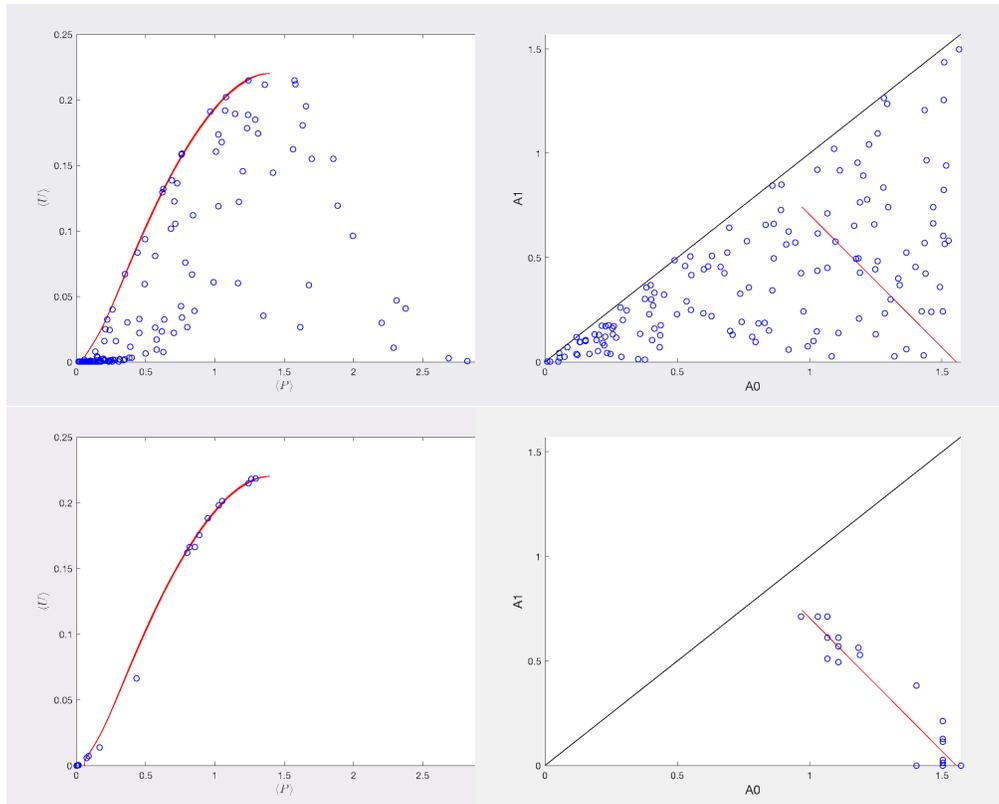


Figure 5: Example plots of NSGA-II running with  $\mu_N/\text{Fr} = 1$ ,  $\mu_S/\text{Fr} = 10$ . Top left) Initial population in objective space; Top right) Initial population in parameter space; Bottom left) Population in objective space after 6 generations; Bottom right) Population in parameter space after 6 generations. The red curve in objective space is the best-fit curve to the Pareto frontier computed using brute force. The red line in parameter space is the best fit line to the Pareto frontier.

## 5 Conclusion

We have several goals for the continuation of this project. First, we aim to improve our implementation of NSGA-II through testing on both two- and three-dimensional parameter spaces in which the Pareto frontiers are easily visualized. Once that is done, we will port our code to the Flux HPC Cluster at The University of Michigan in order to compute Pareto frontiers for much larger parameter spaces than are possible with our current computing resources. In studying larger parameter spaces, we hope to better understand how more complex angle functions affect the efficiency of a scallop's locomotion, and possibly draw more general conclusions about the nature of this system.

## References

- Hu, D. L. and Shelley, M. (2012) D L Hu and M Shelley. Slithering Locomotion. In *Natural Locomotion in Fluids and on Surfaces*, pages 117–135. Springer.
- Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation*. 6 (2): 182.
- Silas Alben. Efficient sliding locomotion with isotropic friction. arXiv preprint arXiv:1902.03163, 2019.