Optimizing a Siege: Mathematically Maximizing Catapult Range with Lagrangian Mechanics

S. Carter and R. Jordan

***Abstract*—Using Lagrangian mechanics to derive the motion of a simple lever-arm catapult, we will calculate the speed of the projectile just before release. This will allow us to determine the optimal ratio of counterweight to thrown mass, and position of the pivot point to maximize the range of the catapult.**

lagrangian, optimization, pendulum, lever, acceleration, siege engine, projectiles

I. Introduction

Catapults are one of the earliest and simplest siege engines. They were used in wars from 400 B.C. until the invention of gunpowder.In modern times, they are used for recreation, education, and reenactment. The modern catapult-builder has an abundance of design decisions to make. We will focus on two: the ratio of the counterweight to the mass being thrown, and the position of the pivot along the beam. Our goal, catering to modern catapult-builders looking for awe-inspiring performance rather than effective siege, will be maximizing the distance the projectile can be thrown.

The simplest catapult design consists of a lever arm with two masses attached to opposite ends, a counterweight and a projectile, as seen in Figure 1. While this is not a design seen historically, it is easy to both build and model, and can throw projectiles impressive distances when properly optimized.

This paper is organized as follows: We begin with an analysis of the natural coordinate system and abstractions we used to model the lever catapult. We then explain the derivation of our equations of motion, implement our model, and examine the results.

II. Modeling the Lever Catapult

A. Abstraction and Coordinate Systems

Designing for modern catapult-builders allows us to make several useful simplifying assumptions. Because most of the mass is concentrated in the counterweight and projectile, we modeled the beam as a massless and rigid constraint with no friction about its fixed pivot point. This is roughly analogous to having very heavy masses for the counterweight and projectile relative to a strong but lightweight rod supporting them. A physical catapult's range will be less than our model predicts, but the optimal mass ratio and pivot point will be nearly the same.

For our model comparison, we used the system described above and depicted in Figure 1, varying the location of the pivot, and the mass ratio between the projectile and the counterweight. It consists of a single beam, anchored at a pivot somewhere in the middle of the beam. The beam itself is of length L (broken into sections L_1 and L_2, one on each side

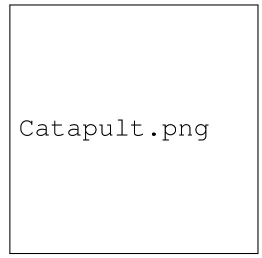

Fig. 1. Illustration of the simplified catapult. The full length of the lever is L1 + L2. The mass of the counterweight attached on the right side is M, while the projectile secured on the left is m. The angle of the beam is measured from the horizontal.

of the pivot, with L_1 to the right and L_2 to the left), the counterweight is a point mass M, while the projectile is point mass m on the opposite side. While the beam is massless, it still serves as a workable abstraction for a lever with a lightweight beam and very heavy masses on each end. It would also serve little purpose to give it mass. While the model would be slightly more accurate, we are asking where to put our pivot, and the locations of the masses on the end of the beam should have the greatest effect on where that location should be.

The coordinate system for the lever-arm has two main unit vectors in its natural coordinate system, $\hat{l}$ and $\hat{\theta}$ as seen in Figure 1. $\hat{l}$ is along the beam towards counterweight M, and $\hat{\theta}$ is in the direction of increasing theta (the angle between the horizontal and the lever), perpendicular to the beam. This is the same as the standard polar coordinate system. This means that in our coordinate system, we can get the two positions of our masses (counterweight M and projectile m), so long as we know L_1 and L_2, their distance from the center of the beam. Their positions are

$$\vec{P_M} = L_1\hat{l} \tag{1}$$

and

$$\vec{P_m} = -L_2\hat{l}. \tag{2}$$

Because we will need to know the velocity terms of both, we should also calculate them here. For a lever arm, the derivatives of equations 1 and 2 simply become

$$\dot{\vec{P_M}} = L_1\theta\hat{\theta} \tag{3}$$

and

$$\dot{\vec{P_m}} = -L_2\theta\hat{\theta}, \tag{4}$$

standard velocity derivatives in a polar coordinate system. Because we are using Lagrangians to do the modeling, we do not need to calculate their accelerations. It is also worth noting that because gravity is involved, we will need to know what $\hat{l}$, and $\hat{\theta}$ are in Cartesian coordinates. The equation is a straightforwards trigonometric calculation from the unit circle, making

$$\hat{l} = \cos\theta\hat{i} + \sin\theta\hat{j}, \tag{5}$$

and

$$\hat{l} = -\sin\theta\hat{i} + \cos\theta\hat{j}, \tag{6}$$

B. Modeling and Lagrangian Equations

The easiest way to analyze a system with constraint forces is Lagrangian mechanics. The lever-arm, while a simple system, can still be complicated by the variable length of the beam, and the Lagrangian equation still makes for less work. That equation, at its simplest, is

$$L = KE - PE. \tag{7}$$

Potential Energy is a function of the height of the object, which means we need to know what the position of each mass is in the $\hat{j}$ (vertical) direction. Since the usual equation is

$$PE = mgh, \tag{8}$$

we can plug in $P_M\hat{j}$ and $P_m\hat{j}$ for h. This leaves us with

$$PE = MgL_1\hat{l} * \hat{j} - mgL_2\hat{l} * \hat{j}, \tag{9}$$

or

$$PE = MgL_1 \sin\theta - mgL_2 \sin\theta. \tag{10}$$

Kinetic Energy can be found the same way, beginning with the equation

$$KE = \frac{1}{2}mv^2. \tag{11}$$

Plugging in both masses, as well as equations 3 and 4, we get

$$KE = \frac{1}{2}ML_1^2\dot{\theta}^2 + \frac{1}{2}mL_2^2\dot{\theta}^2. \tag{12}$$

Finally, we have what we need to compute the Lagrangian, making it

$$L = \frac{1}{2}ML_1^2\dot{\theta}^2 + \frac{1}{2}mL_2^2\dot{\theta}^2 - MgL_1\sin\theta + mgL_2\sin\theta \tag{13}$$

We know that

$$\frac{dL}{d\theta} = \frac{d}{dt}\frac{dL}{\dot{\theta}}. \tag{14}$$

Which gives us the equations

$$\frac{dL}{d\theta} = -MgL_1\cos\theta + mgL_2\cos\theta \tag{15}$$

and

$$\frac{d}{dt}\frac{dL}{\dot{\theta}}. = ML_1^2\ddot{\theta} + mL_2^2\ddot{\theta}. \tag{16}$$

Finally, we can put them together, and get

$$\ddot{\theta} = \frac{-MgL_1\cos\theta - mgL_2\cos\theta}{ML_1^2 + mL_2^2}. \tag{17}$$

With that equation, we have everything that we need to model the system in Matlab, and compute the distance that a projectile can be thrown.

III. Results

IV. Discussion

Sections 3 and 4, if they existed, would go on to present a set of results for the figures of merit, and would interpret those results.

Acknowledgments

The author would like to acknowledge the advice of ninjas, the suggestions of many imaginary friends, the Adobe Illustrator button-finding help of Alyssa Guo, a real friend and graphic design major in the faraway land of California, and the time zones that mean people in California are still awake at 3AM Olin time.

References