ME534 Project: Modeling the Dynamics of a Trebuchet

Mark Colton

Winter 2025

Introduction

Objective

The purpose of this project is to model and simulate the motion of a trebuchet and projectile using Lagrange's method with Lagrange multipliers.

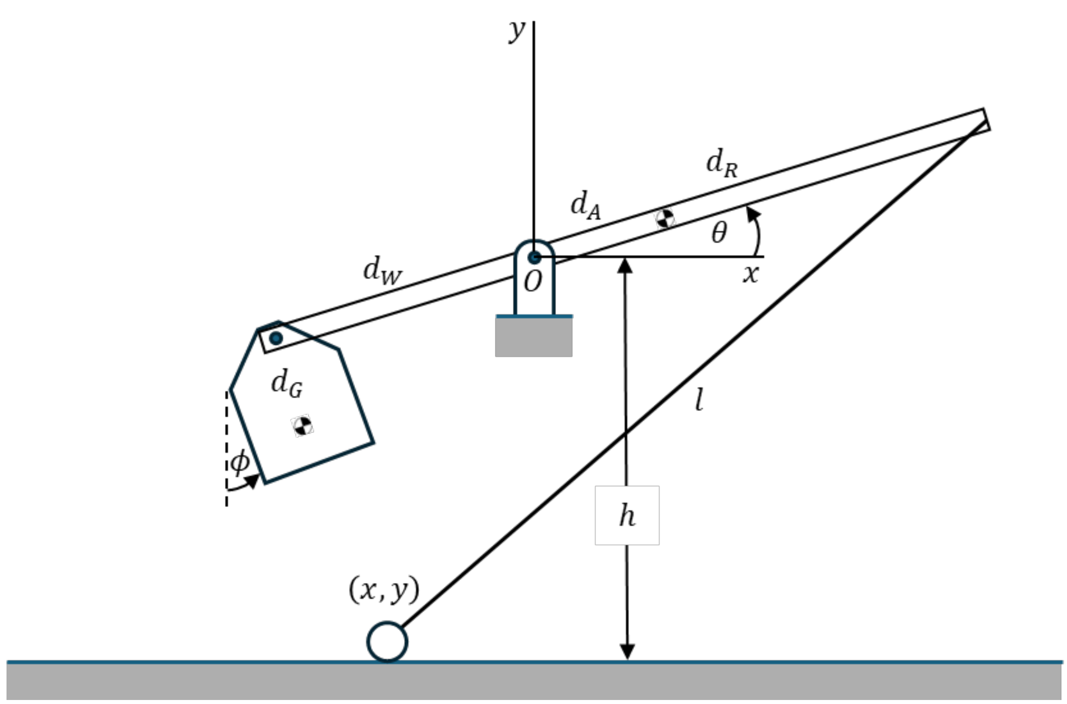
Deliverables

In the end, you will submit the following:

1. A brief memo that includes the equations of motion, constraint equations, plots, and answers to the questions specified in this document.
2. A Matlab or Python file with a function that I can call to test the response of your model to various initial conditions and with various model parameters.

Trebuchet Model

For the purposes of this project, you will use the figure shown below in defining motion variables and model parameters. Note that the trebuchet will launch the projectile in the $-x$ direction.



Model Parameters

The parameters are defined as follows:

m_P = mass of the projectile (kg)

m_A = mass of the arm (kg)

m_W = mass of the swinging counterweight (kg)

I_{AO} = moment of inertia of the arm about the pivot O (kg·m^2)

I_{WG} = moment of inertia of the swinging counterweight about its center of mass (kg·m^2)

l = length of rope (m)

d_R = distance from the pivot O to the connection point of the rope (m)

d_A = distance from the pivot O to the center of mass of the arm (m)

d_W = distance from the pivot O to the pivot of the swinging counterweight (m)

d_G = location of the center of mass of the counterweight relative to its pivot point (kg·m^2)

h = distance of the projectile track below the pivot O (m)

μ = coefficient of friction between the projectile and the track

Motion Variables

The motion variables of interest (and, therefore, the generalized coordinates) in this project are

$q_1 = x$ = horizontal location of the projectile measured from the pivot O (m)

$q_2 = y$ = vertical location of the projectile measured from the pivot O (m)

$q_3 = \theta$ = angle of the trebuchet arm, measured from the horizontal (°)

$q_4 = \phi$ = angle of the swinging counterweight, measured from the vertical (°)

Motion Phases

Definitions

We will define the following three phases of motion of the trebuchet and projectile:

1. Sliding: In this phase, the projectile is sliding along the track. It has not yet left the track.
2. Swinging: In this phase, the projectile has left the track. It is still in the sling at the end of the rope and is swinging relative to the trebuchet arm.
3. Flying: In this phase, the projectile has left the sling and is undergoing unconstrained projectile motion.

Transitions

The transitions between phases occur with the following conditions:

1. Sliding to Swinging: The Sliding Phase ends when the normal force N between the projectile and the track is zero.

2. Swinging to Flying: Trebuchet engineers could adjust the angle at which the projectile was released by changing the angle of a metal pin at the end of the arm. We will assume the pin has been adjusted such that the rope is released when its slope k_{rope} is related to the slope of the arm k_{arm} by this relationship:

$$\frac{|k_{arm} - k_{rope}|}{|k_{arm}|} < 0.7$$

When that condition is met, the projectile is released from the sling and is free to undergo unconstrained projectile motion in the Flying Phase.

Task #1 – Equations of Motion (EOMs)

Develop symbolic equations of motion, not including the generalized constraint forces, for the four generalized coordinates described above, written in matrix form ($[M]$ and $\{F\}$). You are to use Lagrange's method to obtain the EOMs. *There must be a single set of EOMs (a single $[M]$ and a single $\{F\}$) that is valid for the entire motion of the arm, counterweight, and projectile during all three phases of motion.*

Task #2 – Kinematic Constraints and Constraint Forces

Use the method of Lagrange multipliers to account for kinematic constraints and to calculate the normal force N and the tension T in the rope.

Kinematic Constraints

You will express kinematic constraints in matrix form ($[a]$, $\left[\frac{da}{dt}\right]$, and $\{\lambda\}$). Although the EOMs will be unchanged during the three phases of motion, the constraints will change. This is how a single set of EOMs can work for all three phases. For each phase, generate a table that shows the following information, in symbolic form (no numbers plugged in):

Phase:	
n:	
m:	
p:	
Kinematic Constraints (configuration form):	
Kinematic Constraints (velocity form):	
Constraint Coefficients $[a]$:	
Constraint Coefficients $[da/dt]$:	

Constraint Forces

Determine the relationship between the constraint forces (N and T) and the kinematic constraints and Lagrange multipliers ($[a]$ and $\{\lambda\}$). Provide two equations that relate N and T to the Lagrange multipliers and constraint coefficients. For example, $N = 2y\lambda_1 - 4x\lambda_2$ or $N = 2q_2\lambda_1 - 4q_1\lambda_2$, are fictitious examples of the form of the equation that you would use. The equations include the constraint force, the Lagrange multipliers, and the constraint coefficients (not just written as a_{jk}).

Task #3 – Simulation

You are to develop a simulation in Matlab or Python that solves the equations of motion and constraint equations numerically using the *augmented method* described in class and in section 8.2 of Ginsberg. The solution must use the same EOMs (the same $[M]$ and $\{F\}$) for all three phases, but with changing constraints. In the simulation, assume the system always starts from rest, i.e., the initial generalized velocities ($\dot{x}(0)$, $\dot{y}(0)$, $\dot{\theta}(0)$, $\dot{\phi}(0)$) are zero.

Function

You will create and submit a *single* Matlab or Python file that contains a function that I can call to test your results. Your file should be named *your_last_name.m* or *your_last_name.py*. The function within your file should be named "your_last_name" (all lower case). The parameters passed to your function must be the initial conditions of all four generalized coordinates ($\theta(0), \phi(0), x(0), y(0)$), the time vector t, the mass of the projectile m_P, the mass of the counterweight m_W, and the location of the counterweight d_W. Other model parameters must be hardcoded in your function. Your function must return the four generalized coordinates ($\theta(t), \phi(t), x(t), y(t)$), the normal force $N(t)$, and the tension $T(t)$ as functions of the provided time vector t. Everything should be contained in a single m-file or Python file so that I can call your function and compare your simulated results to my simulated results. The specific form of your function should look like this (for Matlab), with my last name replaced by yours:

```
function [theta,phi,x,y,N,T] = colton(theta0,phi0,x0,y0,t,mp,mw,dw)
```

I'll provide more details on what a Python function would need to look like soon.

I will also provide a test driver function, in both Matlab and Python, that can be used to call your function. For your function to work, it must work perfectly with my test driver function, with the units specified below, without any modifications or additional files.

Model Parameters

In your simulations, use the following model parameters:

$m_P = 30$ kg

$m_A = 150$ kg

$m_W = 3000$ kg

$I_{AO} = 1466$ kg·m^2

$I_{WG} = 6750$ kg·m^2

$l = 6$ m

$d_R = 8$ m

$d_A = 1.5$ m

$d_W = 1.5$ m

$d_G = 1.5$ m

$h = 6$ m

$\mu = 2$

Remember that you must be able to change m_P, m_W, and d_W by changing the parameters sent to your function. The other parameters should be hardcoded into your function.

Units

For units, time will be in seconds (s), masses will be passed in units of kilograms (kg), forces will be returned in units of newtons (N), positions/lengths will be passed back and forth in units of meters (m), and angles will be passed back and forth in units of degrees (°). Within your function, it is likely that you will need to do calculations using rad and rad/s.

Test Cases

I will provide some combinations of model parameters and initial conditions for you to try. You will be asked to provide certain plots and/or numerical results for those cases. More on this later.

Memo

You will submit a brief memo that contains the following:

1. The set of EOMs, without constraints, that describe the motion of your system. This must be a single set that works for all three motion phases. Submit your results as the matrices $[M]$ and $\{F\}$, without numerical values plugged in, as described in Task #1 above.
2. The constraints for all three phases of motion. Submit the tables described in Task #2 above, without numerical values plugged in.
3. The symbolic equations for the constraint forces N and T as functions of the coefficients and Lagrange multipliers, as described in Task #2 above.
4. Plots and/or numerical answers for the test cases that I will provide later, as described in Task #3 above.

Milestones

To encourage good progress on the project, I will stop answering questions about certain parts after certain dates. The milestones are posted in the Schedule in Learning Suite.