Theoretical Mechanics

Assignment 2

Mirna Alnoukari

Task Description

In this assignment, we are asked to conduct an experiment and develop a mathematical model for the yo-yo. In order to prove that our solution is correct, it is necessary to compare the simulation result and experimental results.

Tools

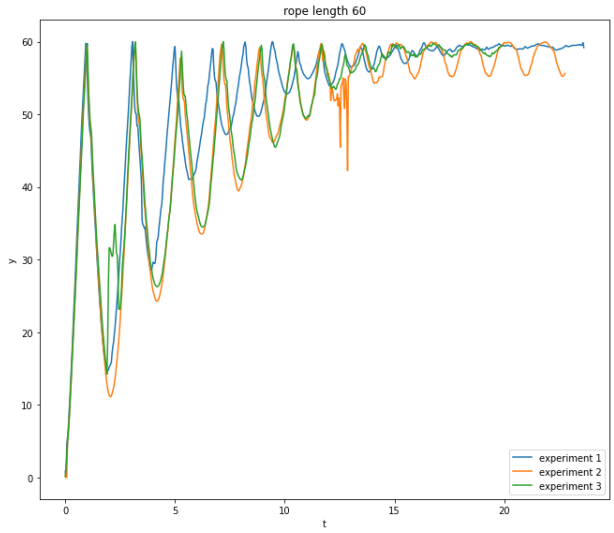
- Python (Colab, matplotlib, OpenCV, numpy)
- Camera
- Markdown editor

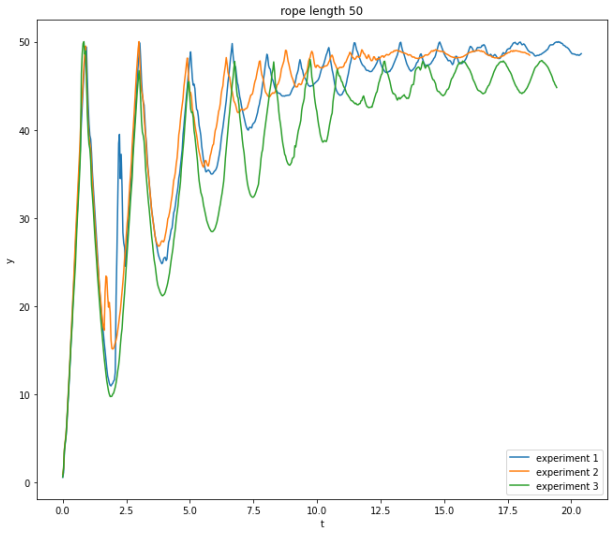
Experiments

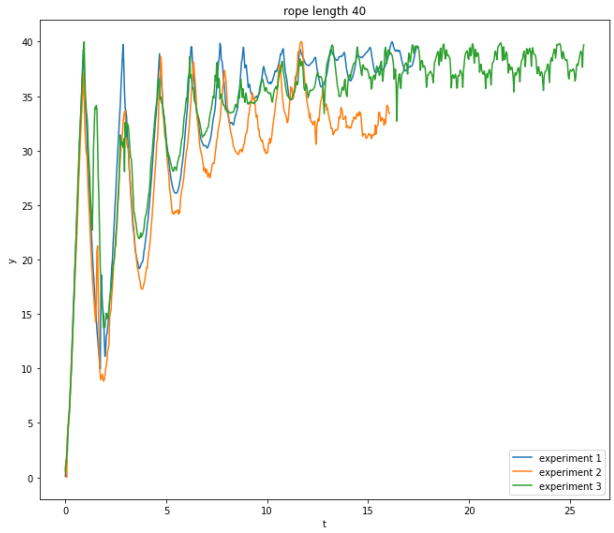
Five experiments has been done and each experiment has been repeated 3 times in
order to remove some non-deterministic errors in the measurements.

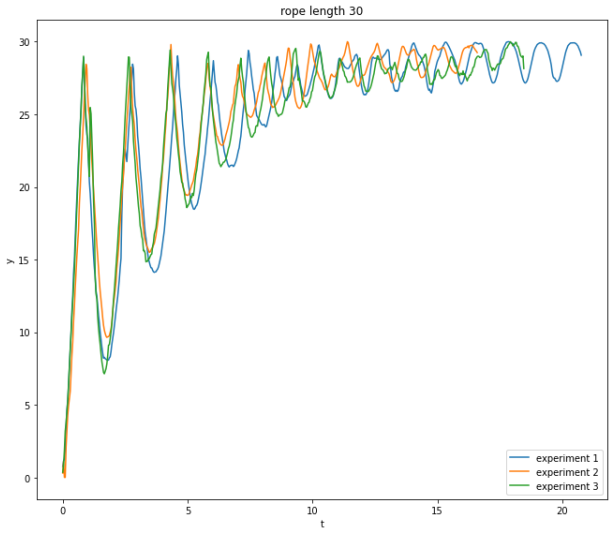
The recording of the experiment was done using phone's camera, using opencv and matplotlib libraries, we created a function to do the following steps for each video experiment: resized the video, converted it to grayscale, and create blur using GaussianBlur to make the yoyo easier to identify from other objects in the video frame. We assumed a threshold in process_motion function, and then found the center of mass in intensity_center function to identify the movement easily. Finally, we could be able to know the motion of a yoyo through yoyo_motion function that takes a video path and rope length as a parameters and call all the previously mentioned functions to find the motion of every video experiment.

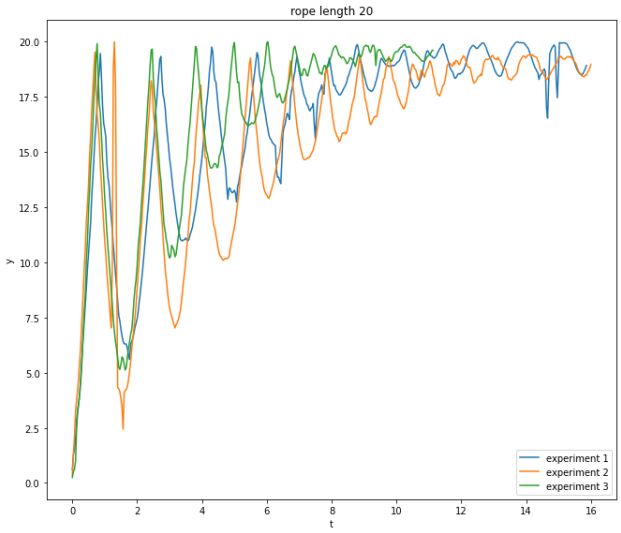
The following are the plots for each experiment with different rope sizes repeated 3 times:











In the following tables are the measurements required for the experiment:

Table 1

Initial string length over the number of oscillations:

# Experiment	String length	Oscillations Mean	Oscillations STD
1	60 cm	14	0.816

# Experiment	String length	Oscillations Mean	Oscillations STD
2	50 cm	12.7	0.471
3	40 cm	12.4	0.816
4	30 cm	13	1.41
5	20 cm	9.7	0.471

Table 2

Maximum distance from bottom position (oscillation amplitude) over oscillation:

# Exp/ #Osc	1	2	3	4
1	(13.3, 2.3)	(26.6, 1.6)	(36.3, 3.3)	(42.3, 4.0)
2	(12.0, 2.1)	(24.0, 2.1)	(33.6, 4.0)	(38.3, 3.8)
3	(11.6, 2.4)	(19.3, 2.6)	(26.3, 1.2)	(29.6, 1.2)
4	(7.6, 1.6)	(14.6, 0.4)	(18.6, 0.4)	(22.0, 0.8)
5	(4.1, 1.1)	(9.5, 1.4)	(12.1, 1.6)	(14.0, 1.2)

Modeling

Research object is the yo-yo, which can be considered as a particle moving in a translation motion and rotational motion, consists of two disks (for approximation) and cylinder connection between the disks

Using the principle of conservation of energy

$$\frac{1}{2}mv^2 + \frac{1}{2}Iw^2 - mgL = 0$$

I is inertia around *x-axis* of rotation

$$v = \dot{L} = wr_0$$

Substituting *v* in the equation we get

$$\dot{L}^2 = \frac{2gL}{1 - \frac{I}{mr_0}}$$

Solving the previous ODE, we obtain the solution of L.

Note: This method will not give us a close expression of our experiment when the yo-yo is at the maximum length and it will jump directly to the other direction.

We have chosen to use numerical solution for its simplicity to derive from our code.
First, let us derive the equations of our motion using Newton's second law:

$$\Sigma f_y = ma_y$$
$$-T + mg - kv = ma_y$$

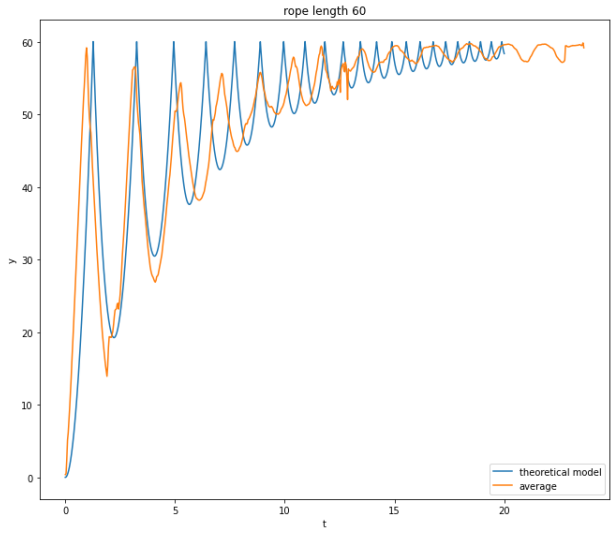
where *k* is the coefficient of friction

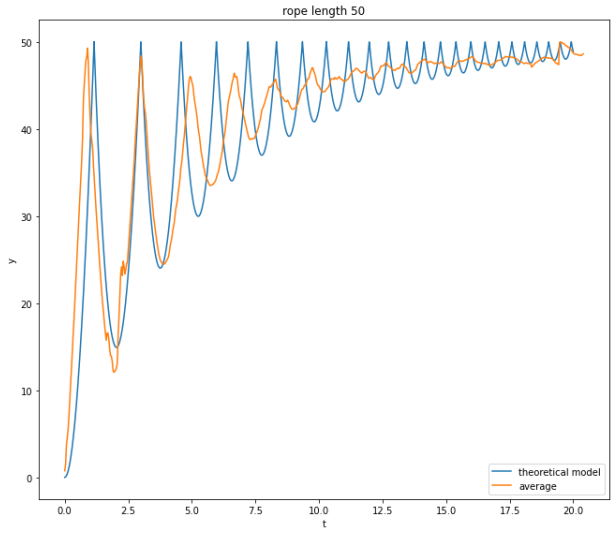
$$\Sigma\Gamma_0 = I.\alpha = I\frac{a_y}{r}$$
$$-T.r = I\frac{a_y}{r}$$
$$T.r^2 = I.a_y$$
$$a_y = \frac{mg - kv}{m + \frac{I}{r^2}}$$

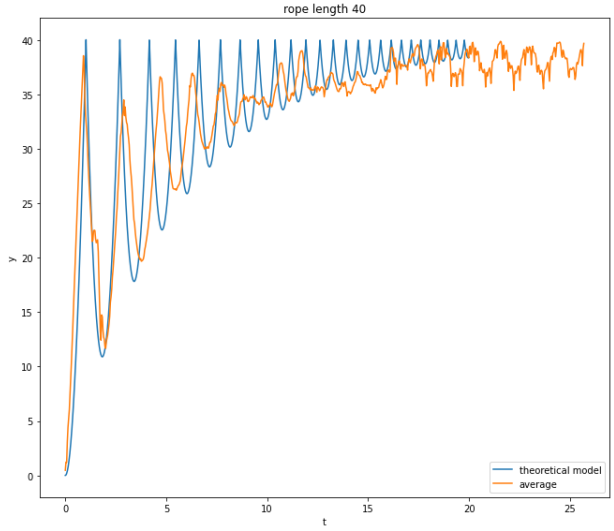
The following is a function that expresses the model of motion for the yo-yo; the function takes time and rope length as its parameters, all the values from yo-yo parameters are taken, *tyv* is a list of tuples of *t, y, v*. In the *for loop*, before we reach the maximum length *L* of the rope, we calculate the new position *y_i* (tyv[i, 1] as represented in code), and the new velocity *v_i* (tyv[i, 2]) based on their old values (old position and velocity).

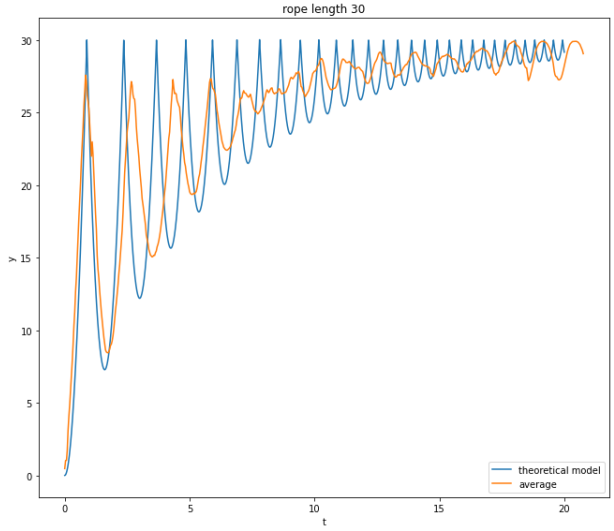
```
def yoyo_motion_theoretical(time, rope_length):
  m = 0.06385
  r = 0.006
  I = 0.0000248
  g = 9.81
  L = rope_length / 100
  k = 0.2 # manually calibrated

  dt = 0.001

  tyv = np.zeros((int(time / dt), 3))

  for i in range(1, tyv.shape[0]):
    tyv[i, 0] = dt * i
    if tyv[i - 1, 1] < L:
      tyv[i, 1] = tyv[i - 1, 1] + tyv[i - 1, 2] * dt
      tyv[i, 2] = tyv[i - 1, 2] + (m * g - k * tyv[i - 1, 2]) / (m + I / r ** 2) * dt
    else:
      tyv[i, 1] = 2 * L - tyv[i - 1, 1]
      tyv[i, 2] = -tyv[i - 1, 2]

  tyv[:, 1] *= 100
  return tyv[:, [0, 1]]
```

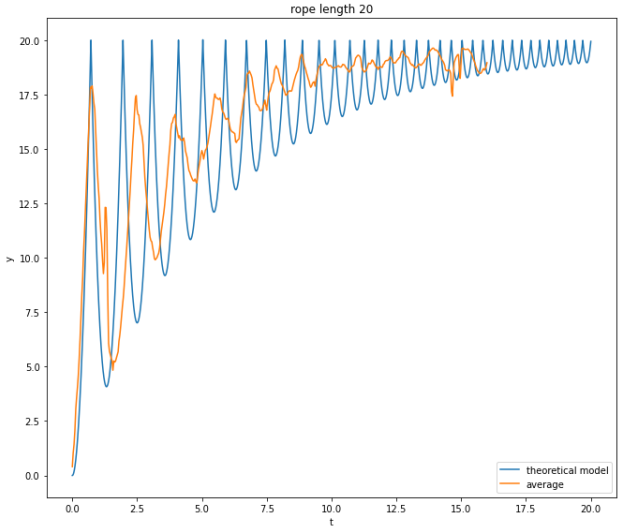
Comparison between the theoretical model and the experiment:


rope length 60
theoretical model
average
y
t


rope length 50
theoretical model
average
y
t


rope length 40
theoretical model
average
y
t


rope length 30
theoretical model
average
y
t


rope length 20
theoretical model
average
y
t