Rotating-Top Conical Pendulum

Faiyaz Siddiquee Taaroop

June 2023

1 Introduction

In this short note, we will explore the following problem:

As in Figure 1, consider an inextensible string of length l attached to a disk of radius r. At the other end of the string, we hang a mass m. Now, imagine rotating the disk with an angular velocity of ω (See Figure 2). We are required to find the tilt angle of the string, θ.

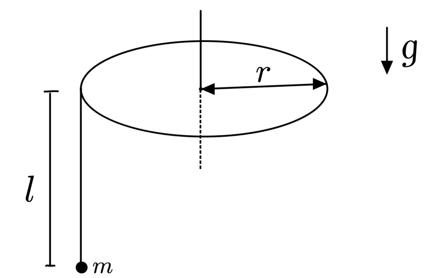

Figure 1: The Problem

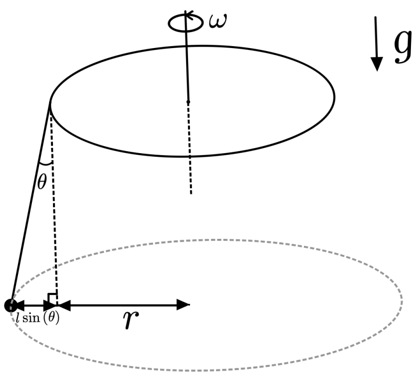

Figure 2: Rotating the Disk

2 Solution

As we can see, the mass m revolves in a circle with radius $R = r + lsin(\theta)$. Moreover, notice that the angular velocity of mass m around the larger circle (radius R) is the same as ω, since the string attaches it to the disk of radius r, which is rotating at ω. So, when the disk completes a full revolution around the circle of radius r, the mass m also completes a full revolution around the circle of radius R.

Now, observe the following free body diagram of mass m:

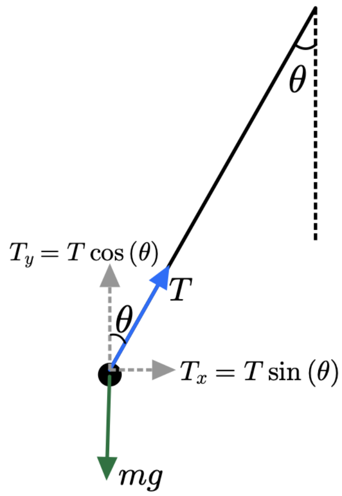

Figure 3: Free Body Diagram

Obviously, the mass cannot have any vertical acceleration. So:

$T_y = Tcos(\theta) = mg$ (i)

And, the horizontal force T_x must provide the centripetal force required by the mass m to rotate in a circle of radius R with angular velocity ω. Hence:

$T_x = Tsin(\theta) = \frac{mv_{tangential}^2}{R} = m\omega^2 R$ (ii)

Dividing (ii) by (i):

$tan(\theta) = \frac{\omega^2 R}{g}$

Substituting $R = r + lsin(\theta)$, we get:

$tan(\theta) = \frac{\omega^2(r+lsin(\theta))}{g}$

This final equation is of the form:

$tan(\theta) = a + bsin(\theta)$

As it turns out, these sorts of equations are quite cumbersome to solve in general. In particular, it involves the use of Euler's Formula, and raising expressions to large powers to get rid of imaginary terms.

For our purposes, however, we will assume $0 \leq \theta \leq \pi/2$ (which makes physical sense), and use the following Taylor series approximations for $sin(\theta)$ and $tan(\theta)$:

$sin(\theta) \approx \theta - \frac{\theta^3}{6} + \frac{\theta^5}{120}$

$\tan(\theta) \approx \theta + \frac{\theta^3}{3} + \frac{2\theta^5}{15}$

Where θ is of course in radians. The approximations hold true mostly in the $0 \leq \theta \leq \pi/2$ region.

Applying these approximations to $tan(\theta) = a + bsin(\theta)$ and rearranging:

$(1 - b)\theta + (\frac{b+2}{6})\theta^3 + (\frac{8-5b}{60})\theta^5 - a = 0$

At this point, we will have to rely on numerical techniques such as the Newton-Raphson method. It might be unsatisfying to find that we have to approximate results rather than finding exact solutions, but remember, dealing with equations involving multiple trigonometric functions is generally quite difficult (especially when they can't be simplified using any neat trigonometric identities). For our case, typing the general form of the equation in Wolfram Alpha results in this monstrosity: https://shorturl.at/jsCY0.

A short recap of the Newton-Raphson Method is given below:

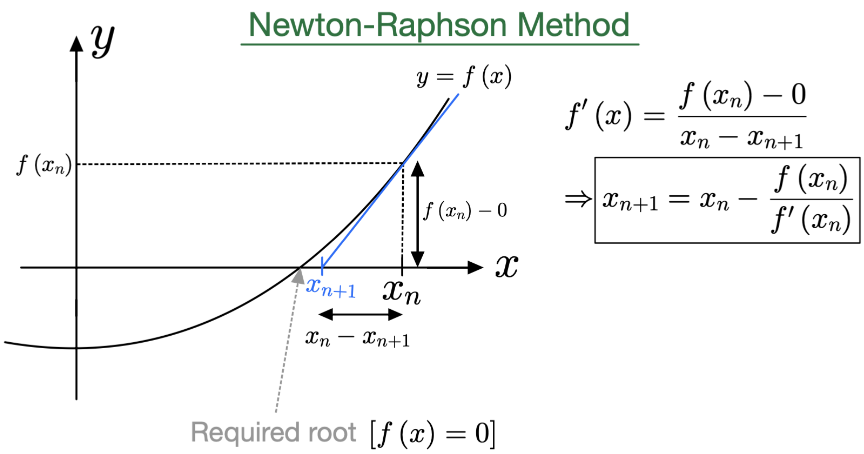

Figure 4: Newton-Raphson Method

In short, we input a seed value (initial guess), x_0. Then, the iteration proceeds as:

$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

It can be shown that the convergence of the Newton-Raphson Method is quadratic (meaning that the difference between our x_n and required root decreases quadratically for each iteration).

For our situation, we can let:

$f(x) = (1-b)x + (\frac{b+2}{6})x^3 + (\frac{8-5b}{60})x^5 - a$

$\implies f'(x) = (1-b) + (\frac{b+2}{2})x^2 + (\frac{8-5b}{12})x^4$

Just to avoid the messy algebra, we can use the following Python code to find numerical answers to the initial problem:

```
# Rotating-Top Conical Pendulum
# Newton Raphson Method

def func(a, b, x):
    result = (1-b)*x + ((b+2) / 6)*(x**3) + ((8 - 5*b)/60)*(x**5) - a
    return result

def func_prime(a, b, x):
    result = (1 - b) + ((b+2) / 2)*(x**2) + ((8-(5*b))/12)*(x**4)
    return result

def solve(omega, r, l, g):
    a = omega**2 * r / g
    b = omega**2 * l / g

    x = 0 # initial guess
    runs = 1000

    for i in range(runs):
        x = x - (func(a, b, x)/func_prime(a, b, x))

    return x

print(solve(0.74, 6, 5.5, 9.8))
```

The last line can be modified according to the numerical values of ω, r, l, and g respectively. The output should be an approximate value for θ in radians (where $0 \le \theta \le \pi/2$). Obviously, the answer can be made more accurate by adding more higher order terms in the Taylor series of $tan(\theta)$ and $\sin(\theta)$.

■