We first set up the same set of basic assumptions and variables.

```
GRAV <- 9.8 # gravity (m/s^2)
MASS <- 1 # mass (kg)
I_CM <- 1/12 # roational inertia (kg m^2)
PHI <- 0.1 # angle of Ft relative to floor (parallel) (rad)
L <- 0.5 # distance from the center of mass (of rotation point) to tenson (m)
FT <- 11 # tension force (N)
OMEGA <- 0.1 # angle of line orthogonal to floor relative to gravity (rad) (because shifted axis)
```

Additionally, we set the time interval and seed values for all values that's tallied:

```
dt <- 0.0001
t_max <- 5

vx <- 0
vy <- 0

time <- 0
theta <- 0
thetadot <- 0
```

Great. Let's start writing the loop now by setting up a bunch of arrays and writing their values in.

```
cTime = NULL
cTorqueNet = NULL
cDDTheta = NULL
cDTheta = NULL
cTheta = NULL

cAccelX = NULL
cAccelY = NULL
cVelX = NULL
cVelY = NULL

cFNetX = NULL
cFNetY = NULL

cKERot = NULL
cKETrans = NULL
```

Awesome, we will start tallying, then!

```
for (i in 0:(t_max/dt)) {
    # write down standard values
    cTime[i] = time
    cTheta[i] = theta

    # torque is calculated via the dot product between the vector of the radius projected out
    # and also the angle at which the thing is at (so like theta + phi)
    #
    # note that, unlike the tabled version, L here represets distance from CoM to tension
    # application
```

```
    torque <- FT*L*cos(theta+PHI)
    cTorqueNet[i] = torque

    # from knowing the torque, we could divide out the rotational inertia to figure the
    # acceleration of rotation
    thetadotdot <- torque/I_CM
    cDDTheta[i] <- thetadotdot

    # from this, we could of course tally for the velocity of theta as well
    thetadot <- dt*thetadotdot + thetadot
    cDTheta[i] <- thetadot

    # After knowing the value for theta, we could use it to calculate the net forces in
    # both components.
    # we define up as +, down as -, right as +, left as -
    fnet_x <- FT*cos(PHI) + MASS*GRAV*sin(OMEGA)
    fnet_y <- FT*sin(PHI) - MASS*GRAV*cos(OMEGA)
    # "I think ax and ay will be constant with time" --- Mark

    cFNetX[i] = fnet_x
    cFNetY[i] = fnet_y

    # Dividing the mass out, we could get accelerations
    ax <- fnet_x/MASS
    ay <- fnet_y/MASS

    # We also tally the components seperately for velocity
    vx <- ax*dt + vx
    vy <- ay*dt + vy

    # And we add them together to tally
    cAccelX[i] = ax
    cAccelY[i] = ay

    cVelX[i] = ax
    cVelY[i] = ay

    cKERot[i] = 0.5 * I_CM * thetadot^2
    cKETrans[i] = 0.5 * MASS * (vx^2+vy^2)

    # We increment the time and theta based on the tallying variable
    time <- dt + time
    theta <- dt*thetadot + theta
}

rotating_link <- data.frame(cTime,
    cTheta,
    cDTheta,
    cDDTheta,
    cTorqueNet,
    cAccelX,
    cAccelY,
    cKERot,
    cKETrans)
```

```
names(rotating_link) <- c("time",
  "theta",
  "d.theta",
  "dd.theta",
  "net.torque",
  "accel.x",
  "accel.y",
  "ke.rot",
  "ke.trans")
```

Let's import some visualization tools, etc.

```
library(tidyverse)
```

Let's first see the head of this table:

```
head(rotating_link)
```

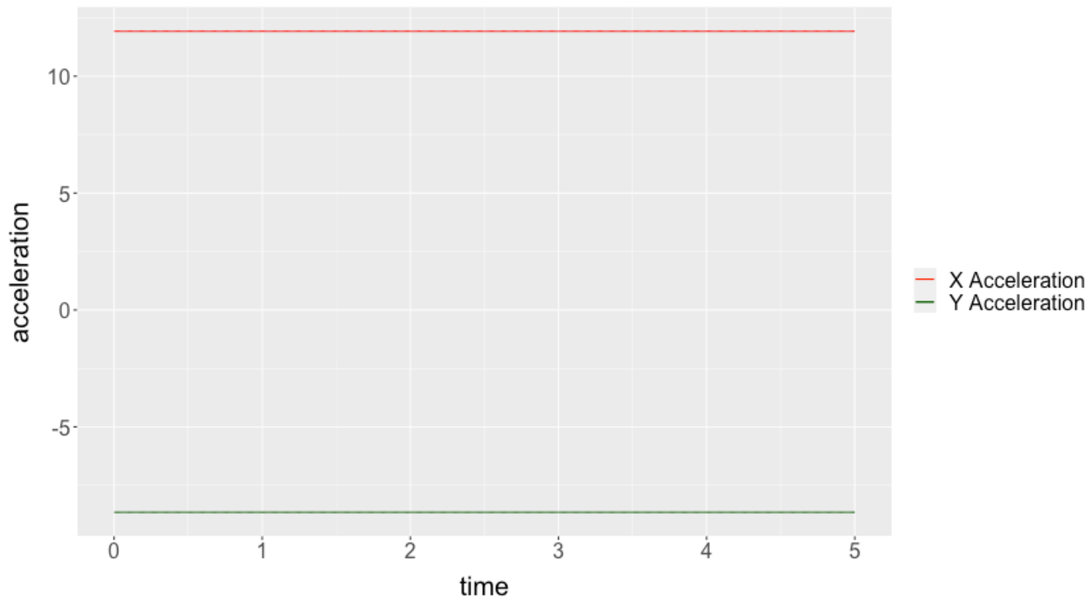
```
1e-04 6.56702749083497e-07 0.0131340545489667 65.6702705813175 5.47252254844313 11.9234133011972
-8.65287323660954 7.18764120396806e-06 4.3408e-06
2e-04 1.97010820398017e-06 0.0197010807416836 65.6702619271685 5.47252182726404 11.9234133011972
-8.65287323660954 1.61721909329306e-05 9.7668e-06
3e-04 3.94021627814853e-06 0.0262681056362569 65.6702489457332 5.47252074547777 11.9234133011972
-8.65287323660954 2.87505572382313e-05 1.73632e-05
4e-04 6.56702684177422e-06 0.0328351287999327 65.6702316367577 5.47251930306314 11.9234133011972
-8.65287323660954 4.4922736804507e-05 2.713e-05
5e-04 9.85053972176748e-06 0.039402149799923 65.6702099999032 5.47251749999193 11.9234133011972
-8.65287323660954 6.46887253689821e-05 3.90672e-05
6e-04 1.37907547017598e-05 0.0459691682033976 65.6701840347464 5.47251533622887 11.9234133011972
-8.65287323660954 8.80485177213443e-05 5.31748e-05
```

Before we start graphing, let's set a common graph theme.

```
default.theme <- theme(text = element_text(size=20), axis.title.y = element_text(margin = margin(t = 0,
```

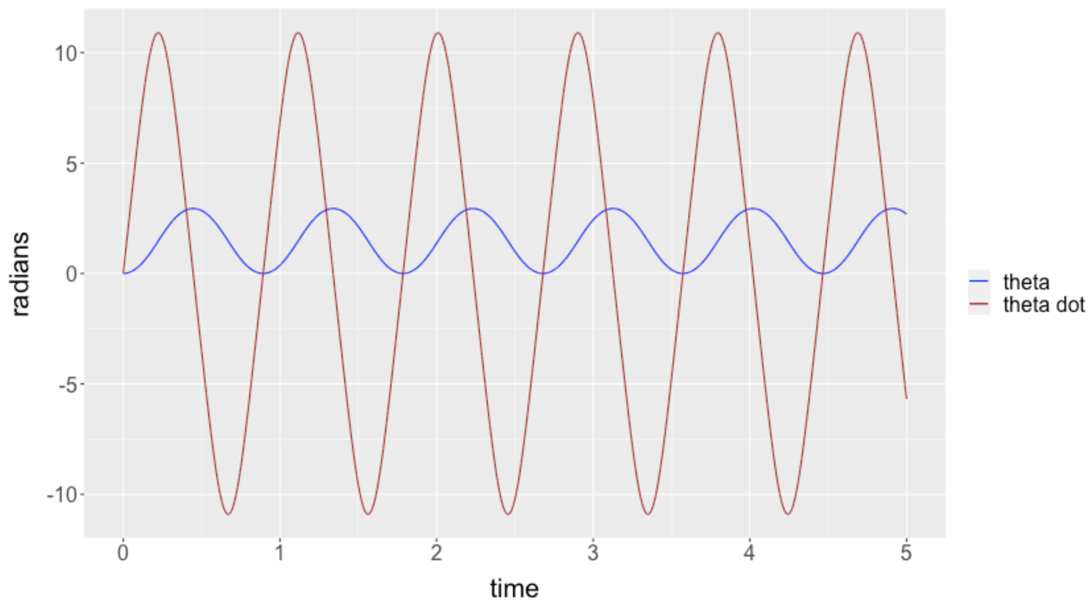
We will graph `ax` and `ay` on top of each other:

```
rotating_link %>% ggplot() + geom_line(aes(x=time, y=accel.x, colour="X Acceleration")) + geom_line(aes
```



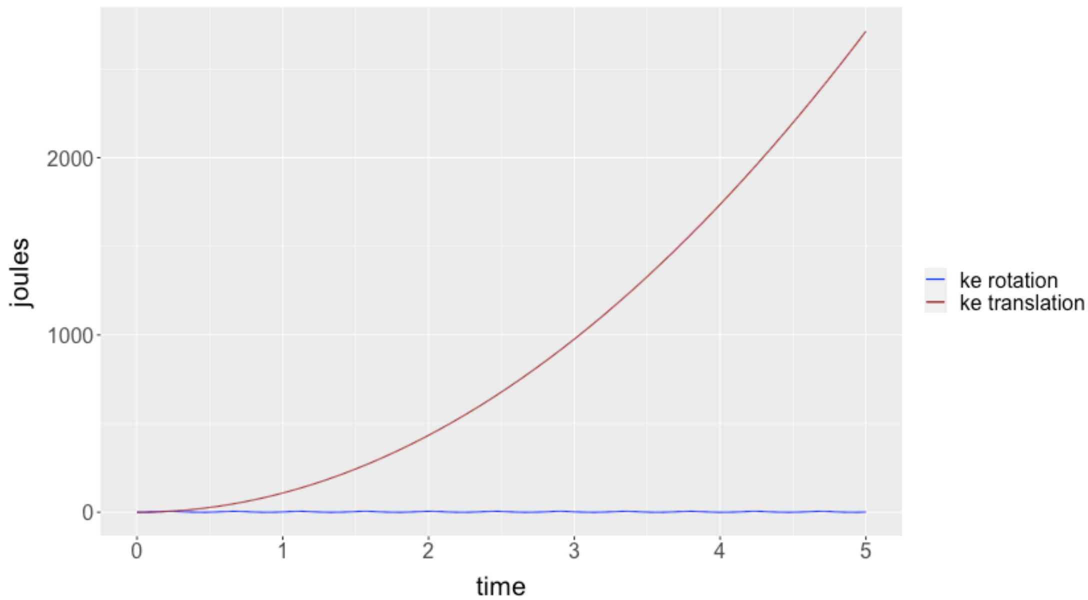
Theta dot atop theta:

```
rotating_link %>% ggplot() + geom_line(aes(x=time, y=theta, colour="theta")) + geom_line(aes(x=time, y=
```



We finally, plot KE rotation and translation

```
rotating_link %>% ggplot() + geom_line(aes(x=time, y=ke.rot, colour="ke rotation")) + geom_line(aes(x=t
```



Let's also plot torque as well.

```
rotating_link %>% ggplot() + geom_line(aes(x=time, y=net.torque)) + default.theme
```