Will Legg

04-05-2024

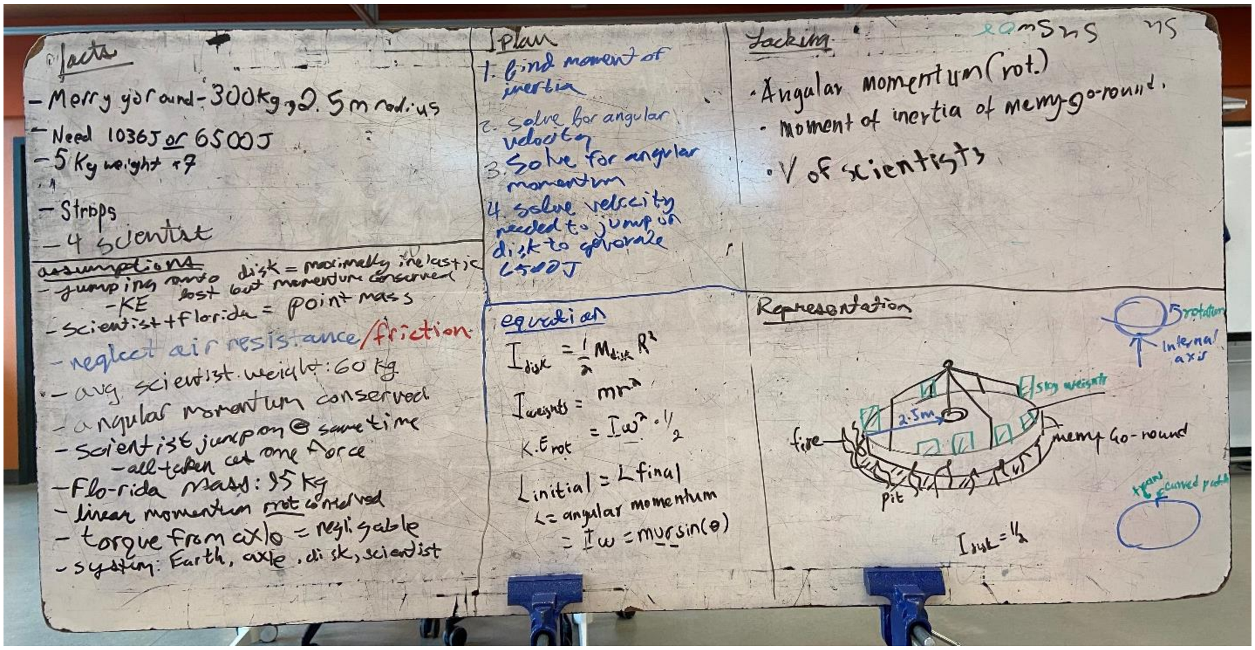
REFLECTION PROMPT:

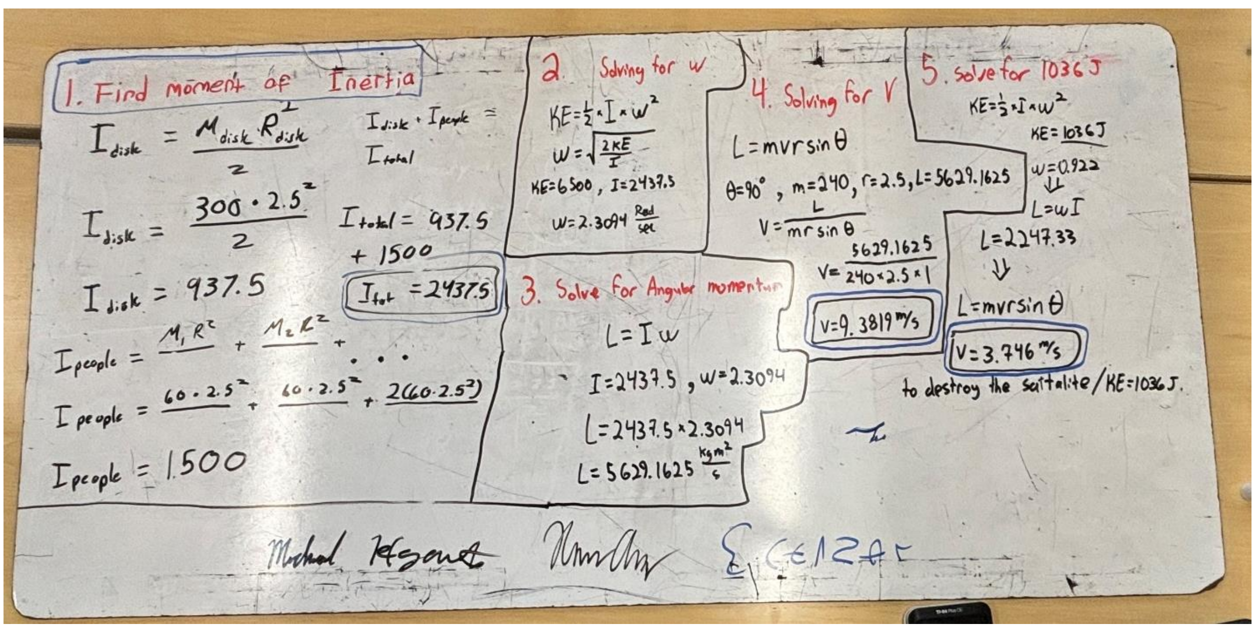
Reflecting on this week in class, were you able to improve on what you said last week? Why or why not? As a group this week we made sure to pay special attention to our assumptions and our organization of our quadrants and solving process. I tried my best to contribute to all parts of our solving process from identifying what we were solving for up to the solving process. I felt I was able to stay involved and engaged as we worked through our problems and tried to ask questions of my groupmates to make sure we were all on the same page and to help everyone try to conceptualize what we needed to do.

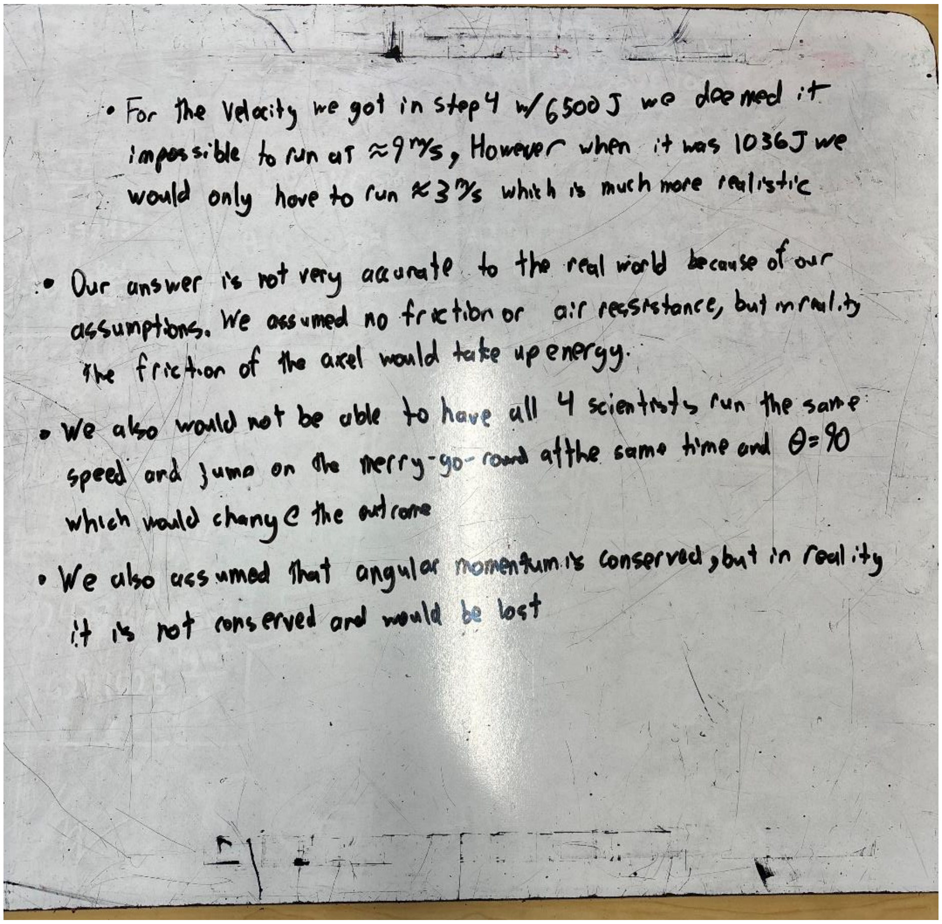
What went well this week? This week I was able to lead our groups zoom call to build our program to represent our model of the merry-go-round and this really helped me gain an even better understanding of the concepts we covered in class on Tuesday by translating them to python. Making sure we built on our assumptions this week also helped me understand some of the short-comings of our model and how it could be improved and made to be more accurate as we gain more physics knowledge.

What area(s) could you improve on for next week related to in class and how might you work to improve those next week? Be specific and include supporting examples from this week's class. What strategies might you try to improve next week? Next week in class I will look to stay involved in all aspects of our groups process. As Richard recommended in class, I'll also be looking to treat our lab as if it were our group project that's due the following week. This would entail an evaluation of our model and assumptions in addition to our usual four quadrants and solving. Also making sure to include all relevant representations such as energy bar graphs & charts as well as force diagrams, if applicable.

PASTE IMAGES OF YOUR TUESDAY WHITEBOARDS HERE:

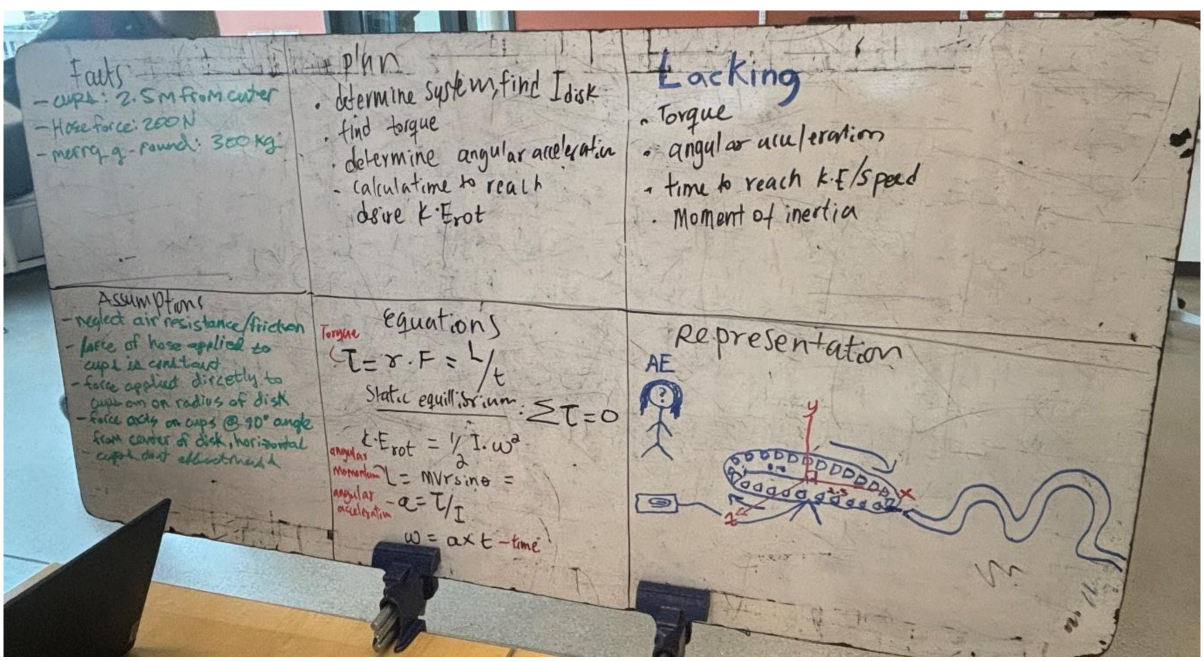






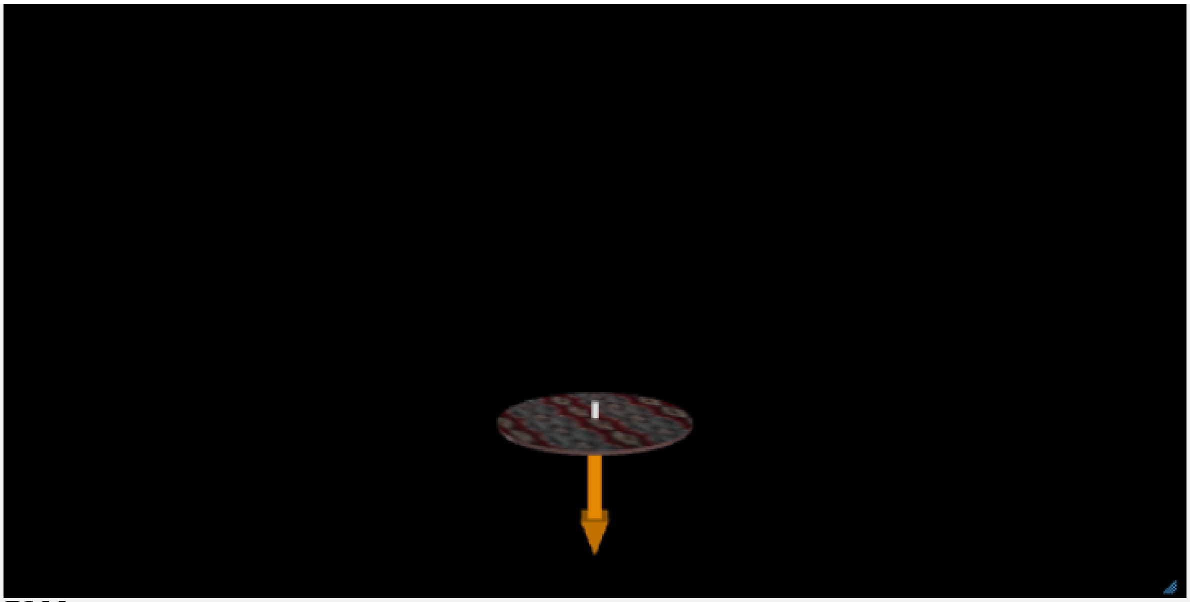
• For the Velocity we got in step 4 w/ 6500 J we deemed it impossible to run at ≈9 m/s, However when it was 1036 J we would only have to run ≈3 m/s which is much more realistic
• Our answer is not very accurate to the real world because of our assumptions. We assumed no friction or air resistance, but in reality the friction of the axel would take up energy.
• We also would not be able to have all 4 scientists run the same speed and jump on the merry-go-round at the same time and θ=90 which would change the outcome
• We also assumed that angular momentum is conserved, but in reality it is not conserved and would be lost

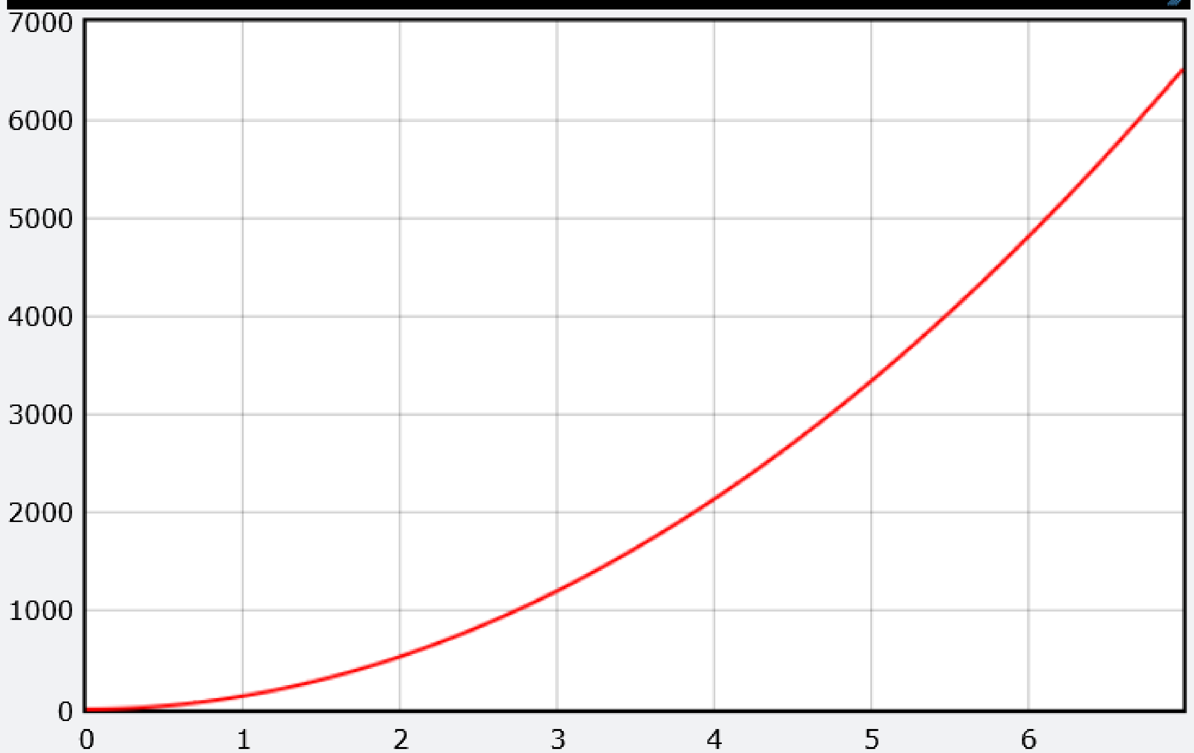
THURSDAY WB:



Facts
- cups: 2.5 m from center
- Hose force: 200 N
- merry-g-round: 300 kg
plan
. determine system, find I disk
. find torque
. determine angular acceleration
- calculate time to reach desire K·E rot
Lacking
. Torque
. angular acceleration
. time to reach K·E/speed
. Moment of inertia
Assumptions
- neglect air resistance/friction
- force of hose applied to cups is constant
- force applied directly to cups on radius of disk
- force acts on cups @ 90° angle from center of disk, horizontal
Equations
Torque τ = r·F = L/t
Static equillibrium: Στ=0
K·E rot = 1/2 I·ω²
angular momentum L = mvrsinθ =
angular acceleration α = τ/I
ω = α x t - time
Representation
AE
y
x
z
2.5

```
Web VPython 3.2
# GlowScript 2.9 VPython

get_library('https://cdn.rawgit.com/PERLMSU/physutil/master/js/physutil.js')

#Objects
axle = cylinder(pos=vector(0,-5,0), axis=vector(0,0.5,0), radius=0.1, color=color.white)
platform = cylinder(pos=vector(0,-5,0), axis=vector(0,0.1,0), radius=2.5, texture=textures.rug)

#Parameters and Initial Conditions
disk_m = 300
I = (disk_m * platform.radius**2) / 2
L = 0
tau = 500
KE = 0


#Time and time step
t=0
tf=10
dt=0.01

#MotionMap/Graph
Larrow = arrow(color=color.orange)

EnergyGraph = PhysGraph(numPlots=1)
#Calculation Loop
while KE <= 6500:
    rate(100)

    taunet = tau
    L += taunet *dt

    platform.rotate(angle=(L/I)*dt, axis=vector(0,1,0), origin=axle.pos)  #DO NOT MODIFY THIS LINE

    omega = L/I
    rotational_energy = 0.5*I*omega**2
    EnergyGraph.plot(t,rotational_energy)
    t = t + dt

    Larrow.pos = axle.pos

    # L scaled down by 1000
    Larrow.axis = vector(0,-L / 1000,0)

    KE = rotational_energy

print(KE + " " + t)
```





KE = 6514.68
time = 6.989999999999895