

```

import math
import numpy as np
import magpylib as mp
from magpylib_force import getFT
import matplotlib.pyplot as plt
import sys

mags = []
m_per_in = 0.0254

# btw, since coils lie in the xy plane by default
# I just rotate the rail to align with the coils of the carriage
dims = (0.768*m_per_in, 0.185*m_per_in, 0.382*m_per_in) # the numbers are
(m, m, m)
pairs = 32
for x in np.linspace(0, (pairs-1)*0.847*m_per_in, pairs):
    mag1 = mp.magnet.Cuboid(
        # TODO fix the approximating the width of the rail
        position = (x, 0, -0.413*m_per_in),
        dimension = dims
    )
    mag2 = mp.magnet.Cuboid(
        position = (x, 0, 0.413*m_per_in),
        dimension = dims
    )

    mags.append(mag1)
    mags.append(mag2)

turns = int(sys.argv[1])
print(f"Using {turns} turns")

coil = mp.Collection()
for i in np.linspace(-0.002, 0.002, turns):
    winding = mp.current.Circle(
        current = 90, # (A) TODO
        diameter = 1*m_per_in,
        position = (0, 0, i),
    )
    winding.meshing = 20
    coil.add(winding)

system = mp.Collection()

```

```

system.add(coil)
system.add(*mags)

system.show()

vel = np.array([0, 0, 0])
mass = 0.15 # (kg)
dt = 1e-5 # (s)
time = 0.50 # (s)

V = [0]

for i in range(round(time / dt)):
    x_pos = coil.children[0].position[0]
    start_ix = 2*math.floor(0.5 + (x_pos / (0.847*m_per_in)))
    connected = mags[start_ix : start_ix+4]
    if len(connected) == 0:
        print("early break")
        time = (i-1)*dt
        break

    # technically, it's the direction of the current in the coil that is
    # changing,
    # but I couldn't figure out how to make that change, so instead
    # I'm changing the polarization (and therefore magnetization) of the
    # magnets
    connected[0].polarization = (1, 0, 0)
    connected[1].polarization = (-1, 0, 0)
    if len(connected) > 2:
        connected[2].polarization = (-1, 0, 0)
        connected[3].polarization = (1, 0, 0)
    else:
        # you are connected to the last contact and nearly launched
        print("almost there")
    # for m in connected:
    #     print(m.position)
    FT = getFT(connected, coil.children)
    # FT is 3 dimensional, they are:
    # the sources of magnetic field,
    # Force and Torque,
    # targets being acted on
    #
    # and each combination is calculated
    net_F = np.sum(FT[:, 0, :], axis=0)
    print(f"t = {round(i*dt, 6)}, F = {net_F[0]}, V = {vel[0]}, S =
{coil.children[0].position[0]}")

```

```
if net_F[0] < 0:
    print("?", net_F)
    break

vel = vel + dt/mass * net_F
coil.move(vel * dt)
V.append(vel[0])

fig, ax = plt.subplots()
X = np.linspace(0, time, len(V))
# X = np.arange(0, time, dt)
print(np.shape(X), np.shape(V))
# ax.set_aspect(1)
ax.plot(X, V)

ax.set(xlabel='time (s)', ylabel='vel (m/s)')
plt.show()
```