linear_sideways

June 16, 2022

[ ]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from datetime import datetime, timedelta
from pd_support import read_df_csv
from scipy.interpolate import interp1d
```

[ ]:
```
cart_df = pd.read_csv("..//straight_line_sideways.csv")
```

[ ]:
```
cart_df
```

[ ]:

	sys_time	e_fr	e_fl	e_rr	e_rl
0	2022-06-15 12:13:22.489984	13	-5	7	3
1	2022-06-15 12:13:37.124103	20	1	14	10
2	2022-06-15 12:13:37.124103	20	1	14	10
3	2022-06-15 12:13:37.124604	20	1	14	10
4	2022-06-15 12:13:37.124604	20	1	14	10
…	…	…	…	…	…
3529	2022-06-15 12:14:09.344631	-7439	8332	8437	-8135
3530	2022-06-15 12:14:09.345196	-7439	8332	8437	-8135
3531	2022-06-15 12:14:09.345196	-7439	8332	8437	-8135
3532	2022-06-15 12:14:09.345196	-7439	8332	8437	-8135
3533	2022-06-15 12:14:09.345671	-7439	8332	8437	-8135

[3534 rows x 5 columns]

[ ]:
```
mc, mc_dt = read_df_csv("..//cart_linear_sideways_mc.csv")
mc_dt
```

[ ]:
```
datetime.datetime(2022, 6, 15, 12, 13, 29, 772000)
```

[ ]:
```
type(mc["center_y"][0])
```

[ ]:
```
numpy.float64
```

[ ]:
```
mc = mc.rename(columns={"time":"seconds"})
```

[ ]:
```
_t = []
for i in list(mc["seconds"]):
    _t.append(mc_dt + timedelta(0,float(i)))

mc["time"] = _t
```

[ ]:
```
mc
```

[ ]:
```
      frame  seconds  center_x  center_y  center_z     org_x     org_y  \
0         0     0.00  0.376711  0.061965  0.048003  0.336798  0.061981
1         1     0.01  0.376718  0.061963  0.048005  0.336790  0.061988
2         2     0.02  0.376716  0.061963  0.048015  0.336800  0.062010
3         3     0.03  0.376717  0.061982  0.048010  0.336802  0.061984
4         4     0.04  0.376725  0.061983  0.048010  0.336806  0.062019
…       …       …         …         …         …         …         …
3771   3771    37.71 -0.259718  0.074456  0.032517 -0.295468  0.075046
3772   3772    37.72 -0.259718  0.074461  0.032516 -0.295468  0.075018
3773   3773    37.73 -0.259725  0.074467  0.032514 -0.295458  0.075047
3774   3774    37.74 -0.259725  0.074449  0.032517 -0.295471  0.075023
3775   3775    37.75 -0.259726  0.074454  0.032511 -0.295475  0.075036

         org_z    xdir_x    xdir_y    xdir_z    zdir_x    zdir_y    zdir_z  \
0    -0.017959  0.416365  0.060203 -0.015434  0.334189  0.063507  0.109515
1    -0.017958  0.416369  0.060193 -0.015421  0.334194  0.063522  0.109507
2    -0.017967  0.416390  0.060189 -0.015428  0.334197  0.063521  0.109513
3    -0.017958  0.416364  0.060215 -0.015432  0.334195  0.063524  0.109503
4    -0.017969  0.416386  0.060197 -0.015427  0.334199  0.063509  0.109510
…          …         …         …         …         …         …         …
3771 -0.035858 -0.216485  0.072623 -0.028880 -0.305695  0.075331  0.091150
3772 -0.035852 -0.216473  0.072621 -0.028879 -0.305703  0.075329  0.091161
3773 -0.035849 -0.216485  0.072634 -0.028877 -0.305710  0.075306  0.091175
3774 -0.035861 -0.216482  0.072616 -0.028889 -0.305722  0.075303  0.091165
3775 -0.035838 -0.216481  0.072625 -0.028884 -0.305703  0.075339  0.091154

                        time
0    2022-06-15 12:13:29.772
1    2022-06-15 12:13:29.782
2    2022-06-15 12:13:29.792
3    2022-06-15 12:13:29.802
4    2022-06-15 12:13:29.812
…                        …
3771 2022-06-15 12:14:07.482
3772 2022-06-15 12:14:07.492
3773 2022-06-15 12:14:07.502
3774 2022-06-15 12:14:07.512
3775 2022-06-15 12:14:07.522
```

```
[3776 rows x 15 columns]
```

[ ]:
```
mc["time"] = pd.to_datetime(mc["time"])
```

[ ]:
```
"""getting initial values of motion capture data"""

marker_cen = np.array(mc[["center_x", "center_y", "center_z"]].iloc[0]).T
marker_xvec = np.array(mc[["xdir_x", "xdir_y", "xdir_z"]].iloc[0]).T
marker_zvec = np.array(mc[["zdir_x", "zdir_y", "zdir_z"]].iloc[0]).T
marker_org = np.array(mc[["org_x", "org_y", "org_z"]].iloc[0]).T
marker_org[0]
```

[ ]:
```
0.336798
```

[ ]:
```
v1 = marker_xvec - marker_org #v1
v2 = marker_zvec - marker_org #v2

v1 = v1.reshape(3,1)
v2 = v2.reshape(3,1)
v3 = marker_org.reshape(3,1)
```

[ ]:
```
def calculate_rotmat(xdir,zdir,org):
    """
    this function calculates rotation matrix
    """
    v1 = xdir - org #v1
    v2 = zdir - org #v2

    vxnorm = v1/np.linalg.norm(v1)

    vzcap = v2 - (vxnorm.T @ v2) * vxnorm
    vznorm = vzcap/ np.linalg.norm(vzcap)

    vynorm = np.cross(vznorm.T[0], vxnorm.T[0]).reshape(3,1)
    rotMat = np.hstack((vxnorm, vynorm, vznorm))
    return rotMat
```

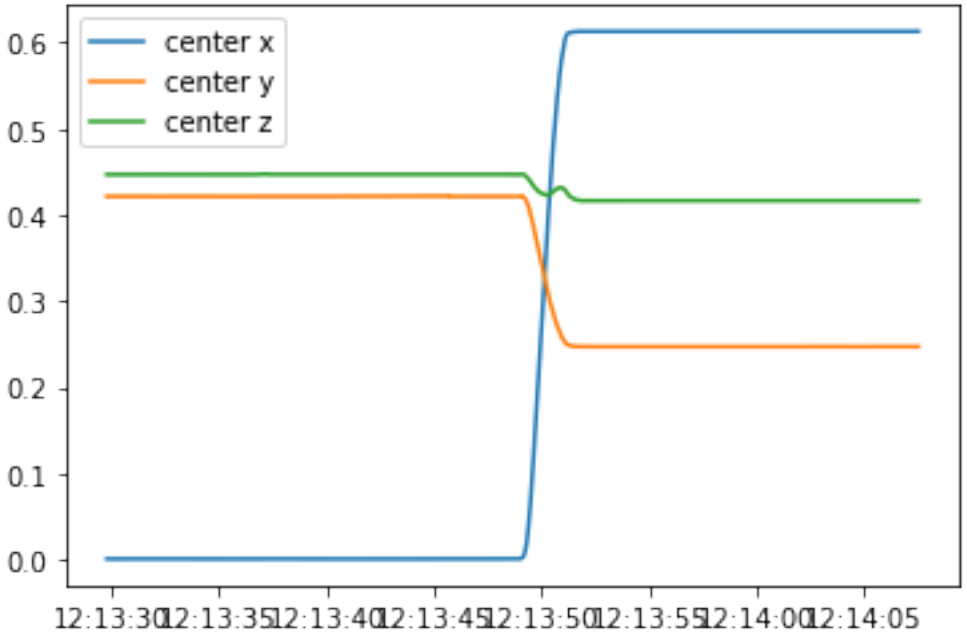
[ ]:
```
"""getting initial rot mat in mc data"""

rot_mat = calculate_rotmat(v1, v2, v3)
rot_mat
```

[ ]:
```
array([[-0.96774126,  0.25042626,  0.02763225],
       [-0.23987084, -0.94934797,  0.20297885],
       [ 0.07706385,  0.18980284,  0.97879316]])
```

[ ]:
```
temp_list = []
for i in range(len(mc["xdir_x"])):
    center_val =  np.array(mc[["center_x", "center_y", "center_z"]].iloc[i])
    center_val = center_val.reshape(3,1)
    transformed_center = rot_mat.T @ center_val + marker_cen
    transformed_center = transformed_center.T[0]
    temp_list.append(transformed_center)
mc[["cen_x", "cen_y", "cen_z"]] = temp_list
```

[ ]:
```
plt.plot(mc["time"], mc["cen_x"], label="center x")
plt.plot(mc["time"], mc["cen_y"], label="center y")
plt.plot(mc["time"], mc["cen_z"], label="center z")
plt.legend()
```

[ ]:
```
<matplotlib.legend.Legend at 0x1e7e49a7688>
```



[ ]:
```
"""resetting cart values to zero"""
cart_df["e_fr"] = cart_df["e_fr"]- cart_df["e_fr"].iloc[0]
cart_df["e_fl"] = cart_df["e_fl"]- cart_df["e_fl"].iloc[0]
cart_df["e_rr"] = cart_df["e_rr"]- cart_df["e_rr"].iloc[0]
cart_df["e_rl"] = cart_df["e_rl"]- cart_df["e_rl"].iloc[0]

cart_df
```

[ ]:

	sys_time	e_fr	e_fl	e_rr	e_rl
0	2022-06-15 12:13:22.489984	0	0	0	0
1	2022-06-15 12:13:37.124103	7	6	7	7
2	2022-06-15 12:13:37.124103	7	6	7	7
3	2022-06-15 12:13:37.124604	7	6	7	7
4	2022-06-15 12:13:37.124604	7	6	7	7
…	…	…	…	…	…
3529	2022-06-15 12:14:09.344631	-7452	8337	8430	-8138
3530	2022-06-15 12:14:09.345196	-7452	8337	8430	-8138
3531	2022-06-15 12:14:09.345196	-7452	8337	8430	-8138
3532	2022-06-15 12:14:09.345196	-7452	8337	8430	-8138
3533	2022-06-15 12:14:09.345671	-7452	8337	8430	-8138

[3534 rows x 5 columns]

[ ]:
```
mils = np.arange(0, len(cart_df["e_fr"])*10, 10)
cart_df["mils"] = mils
cart_df["sys_time"] = pd.to_datetime(cart_df["sys_time"])
```

These are the parameters of the cart

Diameter = 95 mm radius = 47.5 wheel thickness = 45 mm gap between wheel and chassis = 6.5 mm angle between center of chassis and z-dir vector = 51.21 degrees distance between the wheel = 158 mm distance between the wheel and the center of the robot = 101.36 (li)

[ ]:
```
"""circumfrence of the wheel"""

cir_wheel = np.degrees(2*np.pi*(47.5/1000))
cir_wheel
radius = 47.5/1000

lx = 79 #half of the distance between the wheels
ly = 122.5/2

lx = lx/1000
ly = ly/1000
```

The encoder values will give 4000 values per revolution $360/4000 = 0.09$ degrees per encoder rotation value

[ ]:
```
""" angular velocity"""
cart_df["av_fr"] = (cart_df["e_fr"]*0.09).diff()/0.01
cart_df["av_fl"] = (cart_df["e_fl"]*0.09).diff()/0.01
cart_df["av_rr"] = (cart_df["e_rr"]*0.09).diff()/0.01
cart_df["av_rl"] = (cart_df["e_rl"]*0.09).diff()/0.01
cart_df['av_fr'] = cart_df["av_fr"].fillna(0)
cart_df['av_fl'] = cart_df["av_fl"].fillna(0)
cart_df['av_rr'] = cart_df["av_rr"].fillna(0)
```

```
cart_df['av_rl'] = cart_df["av_rl"].fillna(0)

#converting them to radians
cart_df['av_fr'] = cart_df["av_fr"]* np.pi/180
cart_df['av_fl'] = cart_df["av_fl"]* np.pi/180
cart_df['av_rr'] = cart_df["av_rr"]* np.pi/180
cart_df['av_rl'] = cart_df["av_rl"]* np.pi/180

cart_df["av_fl"]
```

[ ]:
```
0       0.000000
1       0.942478
2       0.000000
3       0.000000
4       0.000000
          …
3529    0.000000
3530    0.000000
3531    0.000000
3532    0.000000
3533    0.000000
Name: av_fl, Length: 3534, dtype: float64
```

[ ]:
```
# g1 = np.radians(-51.21)
# g2 = np.radians(51.21)

g1 = np.radians(-np.pi/2)
g2 = np.radians(np.pi/2)

g3 = g2
g4 = g1


b1 = np.pi/4
b2 = -np.pi/4
b3 = b1
b4 = b2


a1 = np.pi/4
a2 = -np.pi/4
a3 = 3*np.pi/4
a4 = -3*np.pi/4


li = 101.36/1000
```

[ ]:
```
t = (-1/radius)*np.array([[np.cos(b1 - g1)/ np.sin(g1), np.sin(b1 - g1)/np.
↪sin(g1), li * np.sin(b1 - g1 - a1)/np.sin(g1)],
                          [np.cos(b2 - g2)/ np.sin(g2), np.sin(b2 - g2)/np.
↪sin(g2), li * np.sin(b2 - g2 - a2)/np.sin(g2)],
                          [np.cos(b3 - g3)/ np.sin(g3), np.sin(b3 - g3)/np.
↪sin(g3), li * np.sin(b3 - g3 - a3)/np.sin(g3)],
                          [np.cos(b4 - g4)/ np.sin(g4), np.sin(b4 - g4)/np.
↪sin(g4), li * np.sin(b4 - g4 - a4)/np.sin(g4)]]
                          )
t
pseudo_t = np.linalg.pinv(t)
pseudo_t
```

[ ]:
```
array([[ 4.47561242e-04, -4.47561242e-04, -4.72799786e-04,
         4.72799786e-04],
       [ 8.73787444e-04,  8.73787444e-04, -2.39613824e-05,
        -2.39613824e-05],
       [ 5.92854944e-03,  5.92854944e-03,  6.26286783e-03,
         6.26286783e-03]])
```

[ ]:
```
val = pseudo_t @ np.array([[cart_df["av_fr"][0]],[cart_df["av_fl"][0]],␣
↪[cart_df["av_rr"][0]], [cart_df["av_rl"][0]]])
val.T
```

[ ]:
```
array([[0., 0., 0.]])
```

[ ]:
```
np.array([[cart_df["av_fr"][0]],[cart_df["av_fl"][0]], [cart_df["av_rr"][0]],␣
↪[cart_df["av_rl"][0]]])
```

[ ]:
```
array([[0.],
       [0.],
       [0.],
       [0.]])
```

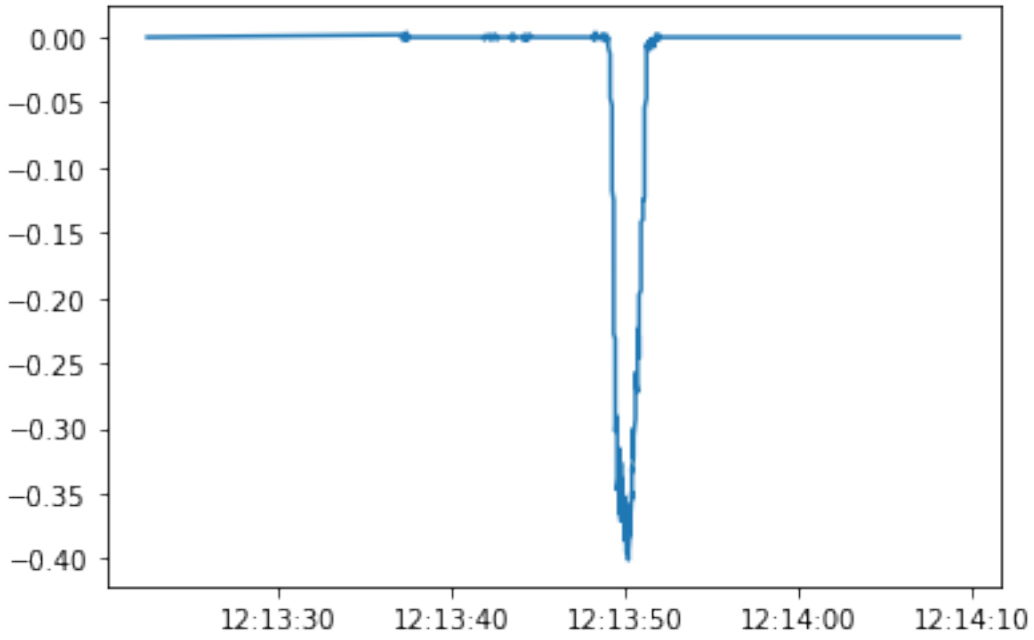
[ ]:
```
_val = []
for i in range(len(cart_df["e_fl"])):
    _v = pseudo_t @ np.array([[cart_df["av_fr"][i]],[cart_df["av_fl"][i]],␣
↪[cart_df["av_rr"][i]], [cart_df["av_rl"][i]]])
    _val.append(_v.T[0])

cart_df[["cal_vx", "cal_vy", "cal_w"]] = _val
# _val
```

[ ]:
```
"""finding vx, vy, w"""

cart_df["vx"] = (cart_df["av_fl"] + cart_df["av_fr"] + cart_df["av_rl"] +␣
↪cart_df["av_rr"])*(radius/4)
```

```
cart_df["vy"] = (- cart_df["av_fl"] + cart_df["av_fr"] + cart_df["av_rl"] -␣
 ↪cart_df["av_rr"])*(radius/4)
cart_df["w"] = (- cart_df["av_fl"] + cart_df["av_fr"] - cart_df["av_rl"] +␣
 ↪cart_df["av_rr"])*(radius/(4*(lx + ly)))
```

[ ]:
```
plt.plot(cart_df["sys_time"], cart_df["vy"])
# plt.plot(cart_df["vx"])
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x1e7e1741b08>]
```



[ ]:
```
"""calculating displacement
s=(1/2)* (v+u)t
v = current velocityn
u = initial velocity
t = time
s = displacement
"""

_xval = []
_yval = []
xf_disp = 0
yf_disp = 0
for i in range(len(cart_df["vx"])):
    if i == 0:
        _xval.append(0)
```

```
        _yval.append(0)
    else:
        x_disp = 0.5*(cart_df["vx"].iloc[i] + cart_df["vx"].iloc[i-1])*0.01
        y_disp = 0.5*(cart_df["vy"].iloc[i] + cart_df["vy"].iloc[i-1])*0.01
        print(y_disp)
        xf_disp = xf_disp+x_disp
        yf_disp = yf_disp+y_disp
        _xval.append(xf_disp)
        _yval.append(yf_disp)

cart_df["x_val"] = _xval
cart_df["y_val"] = _yval
# cart_df["y_val"]
```

```
9.326603190344698e-06
9.326603190344698e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
9.326603190344695e-06
9.326603190344695e-06
-9.326603190344695e-06
-9.326603190344695e-06
0.0
0.0
0.0
9.326603190344695e-06
9.326603190344695e-06
0.0
0.0
0.0
0.0
0.0
0.0
-9.326603190344695e-06
-9.326603190344695e-06
0.0
9.326603190344695e-06
0.0
```

-9.326603190344695e-06
9.326603190344705e-06
9.326603190344705e-06
0.0
0.0
-9.887923813067802e-21
-9.326603190344705e-06
9.326603190344701e-06
1.8653206380689396e-05
-9.326603190344698e-06
5.4210108624275225e-21
9.326603190344701e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
-9.326603190344701e-06
-9.326603190344701e-06
0.0
0.0
0.0
0.0
9.326603190344701e-06
9.326603190344701e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

0.0
0.0
0.0
9.326603190344698e-06
3.252606517456513e-21
-9.326603190344695e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
-9.326603190344698e-06
-9.326603190344698e-06
9.326603190344695e-06
9.326603190344695e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
9.326603190344698e-06
9.326603190344698e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
-9.326603190344695e-06
-9.326603190344695e-06

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
-3.2959746043559335e-21
-3.2959746043559335e-21
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
-9.326603190344701e-06
-9.326603190344701e-06
0.0
0.0
0.0
9.326603190344701e-06
9.326603190344701e-06

```
0.0
0.0
0.0
0.0
9.326603190344698e-06
0.0
-9.326603190344698e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
-9.326603190344701e-06
-9.326603190344701e-06
0.0
0.0
9.326603190344701e-06
9.326603190344701e-06
9.326603190344698e-06
0.0
-9.326603190344698e-06
0.0
0.0
0.0
9.326603190344698e-06
9.326603190344698e-06
0.0
0.0
0.0
0.0
0.0
0.0
-9.326603190344698e-06
-9.326603190344698e-06
0.0
0.0
0.0
0.0
0.0

```
0.0
0.0
0.0
9.326603190344698e-06
3.252606517456513e-21
-9.326603190344695e-06
-9.326603190344698e-06
-9.326603190344698e-06
9.326603190344698e-06
9.326603190344698e-06
-9.326603190344698e-06
0.0
9.326603190344698e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
-9.326603190344698e-06
-3.252606517456513e-21
9.326603190344695e-06
-9.326603190344701e-06
9.326603190344696e-06
9.326603190344698e-06
-9.326603190344698e-06
0.0
0.0
-1.6479873021779667e-21

-9.326603190344698e-06
5.4210108624275225e-21
1.86532063806894e-05
0.0
-9.326603190344698e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
9.326603190344698e-06
0.0
-9.326603190344698e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
9.326603190344698e-06

```
9.326603190344698e-06
0.0
0.0
-9.326603190344698e-06
0.0
0.0
-9.326603190344698e-06
9.326603190344701e-06
3.252606517456513e-21
-9.326603190344698e-06
9.326603190344695e-06
9.326603190344693e-06
-4.943961906533901e-21
0.0
-9.326603190344695e-06
-9.326603190344698e-06
-9.326603190344695e-06
0.0
9.326603190344701e-06
-9.326603190344695e-06
-1.86532063806894e-05
-9.326603190344698e-06
-9.326603190344698e-06
-1.8653206380689396e-05
0.0
-9.326603190344698e-06
-1.8653206380689393e-05
-1.8653206380689393e-05
-2.7979809571034103e-05
-1.86532063806894e-05
-9.326603190344695e-06
-1.865320638068939e-05
-1.865320638068938e-05
-2.79798095710341e-05
-4.6633015951723495e-05
-4.663301595172349e-05
-6.52862223324129e-05
-0.00010259263509379165
-0.00013989904785517048
-0.0001772054606165493
-0.00019585866699723867
-0.00020518527018758333
-0.00022383847656827269
-0.0002611448893296516
-0.00029845130209103 04
-0.0003171045084717197
-0.00037306412761378796
-0.00041969714356551144
```

-0.00044767695313654553
-0.0004943099690882691
-0.0005689227946110265
-0.0006435356201337842
-0.0006994952392758521
-0.0007368016520372311
-0.0007927612711792995
-0.0008673740967020568
-0.0009233337158441253
-0.0009979465413668827
-0.0010725593668896403
-0.0011378455892220533
-0.0012124584147448106
-0.001315051049838602
-0.0014176436849323943
-0.0015109097168358413
-0.0016041757487392878
-0.0016694619710717013
-0.0017534013997848033
-0.0018186876221172154
-0.0019212802572110086
-0.0020518527018758338
-0.0021451187337792805
-0.0022383847656827277
-0.0022943443848247955
-0.0023689572103475537
-0.002462223242251
-0.0025181828613930674
-0.0025648158773447917
-0.0026674085124385833
-0.0027798095710341
-0.002872593782626167
-0.0028632671794358243
-0.0028819203858165133
-0.0029751864177199566
-0.002975186417719958
-0.00291922679857789
-0.0029658598145296134
-0.003012492830481338
-0.003003166227290993
-0.0031710450847171967
-0.003329597338953058
-0.0033762303549047796
-0.0034321899740468466
-0.0034508431804275395
-0.0034508431804275395
-0.0034508431804275365
-0.0034135367676661576

-0.00331094413257237
-0.0032829643230013346
-0.00322704703859264
-0.0032736377198109906
-0.003413536767666162
-0.00354109212330983
-0.00363737524423443
-0.0036000688314730546
-0.0035720890219020163
-0.00346949638680823 3
-0.003413536767666162
-0.0033762303549047736
-0.00339483561285472
-0.0034415165772371976
-0.00350680279956960 4
-0.00360006883147305 1
-0.00364670184742477 7
-0.0036467018474247827
-0.00349747619637926 5
-0.00333892394214339 4
-0.0033575771485240863
-0.0034135367676661654
-0.00346016978361788 75
-0.0035720890219020198
-0.0035627624187116714
-0.00358141562509236 12
-0.0036467018474247736
-0.00360939543466339 4
-0.0035627624187116805
-0.0035627624187116805
-0.00358141562509236 12
-0.00360006883147305 8
-0.0036280486410440837
-0.00367468165699580 3
-0.0036560284506151276
-0.0036840082601861633
-0.0036373752442344347
-0.0035720890219020107
-0.00360006883147304 9
-0.0036840082601861633
-0.00376794768889926
-0.00380525410166064
-0.0038612137208027097
-0.0037586210857089115
-0.00363737524423442 3
-0.0036280486410440807
-0.00373064127613788 5
-0.00380525410166064

-0.0037959274984702856
-0.0038145807048509815
-0.0038891935303737423
-0.003889193530373737
-0.0038518871176123565
-0.003842560514422014
-0.0038798669271833948
-0.0038891935303737423
-0.0037959274984703043
-0.0037492944825185752
-0.0038705403239930403
-0.003935826546325459
-0.0038798669271833995
-0.003870540323993046
-0.003851887117612356
-0.00384256051442202
-0.0037679476888992603
-0.003721314672947538
-0.0037306412761378793
-0.0038332339112316666
-0.0038518871176123626
-0.0037772742920895957
-0.00380525410166064
-0.0037586210857089176
-0.0036933348633764996
-0.0036373752442344287
-0.003628048641044086
-0.0036467018474247766
-0.0036467018474247706
-0.003730641276137885
-0.0037772742920896074
-0.0036280486410440868
-0.0035161294027599523
-0.0034881495931889197
-0.003376230354904779
-0.0033669037517144312
-0.0033762303549047913
-0.003320270735762727
-0.003394883561285452
-0.0034881495931889084
-0.0034601697836178992
-0.0033948835612854695
-0.0033482505453337292
-0.0032736377198109867
-0.00312441206876549
-0.0030497992432427113
-0.003031146036862015
-0.003031146036862027

-0.0030871056560040915
-0.0031710450847172
-0.0031990248942882437
-0.0030871056560040915
-0.00308710565600408
-0.003115085465575136
-0.00299383962410065 8
-0.0028819203858165177
-0.002779327750722714
-0.00272336813158064 9
-0.0027233681315806725
-0.00268606171881925 7
-0.0027047149251999535
-0.00274202133796136 9
-0.00274202133796134 5
-0.0027140415283903 9
-0.00269614488932965202
-0.0026462232422509996
-0.0026415590226299 2714
-0.00239693701991858 7
-0.00238761041672825 1
-0.0023409774007765224
-0.0023036709880151 31
-0.0022850177816344463
-0.0022756911784441 22
-0.0023036709880151 31
-0.0023782838135378913
-0.0023782838135378913
-0.0022756911784440987
-0.0022197315593020 34
-0.0021451187337792 97
-0.0020984857178275 81
-0.0020705059082565015
-0.0020891591146371975
-0.0021078123210179173
-0.0020331994954951565
-0.0019772398763530 57
-0.0019585866699723 84
-0.0019212802572110277
-0.0018466674316882674
-0.0018093610189268753
-0.0017720546061654712
-0.0017254215902137667
-0.0016508087646910183

-0.0015295629232165299
-0.0014736033040744535
-0.0014642767008841173
-0.001445623494503433
-0.0014083170817420529
-0.001371010668980673
-0.0013337042562192808
-0.001315051049838608
-0.001315051049838608
-0.0013337042562192925
-0.0013710106689806723
-0.001380337272171009
-0.0013337042562192925
-0.0012870712402675761
-0.00125909143069652
-0.0012497648275061957
-0.0012404382243158478
-0.0011751520019834118
-0.0011005391764606868
-0.0010539061605089703
-0.0010259263509379143
-0.0009233337158441219
-0.0008114144775599814
-0.000746128255227557
-0.0006621888265144721
-0.0005969026041820597
-0.0005502695882303433
-0.0005036365722786151
-0.0004290237467558547
-0.0003637375244234541
-0.0002984513020910299
-0.00024249168294897727
-0.00022383847656828125
-0.0001958586669972137
-0.00014922565104550904
-0.00012124584147448864
-9.326603190344466e-05
-6.52862223324242e-05
-6.528622233241243e-05
-6.528622233241242e-05
-4.66330159517282e-05
-3.7306412761368425e-05
-5.595961914206442e-05
-7.46128255227604e-05
-7.461282552276043e-05
-6.528622233241243e-05
-6.528622233241242e-05
-5.595961914205263e-05

-6.528622233241242e-05
-6.52862223324242e-05
-6.528622233241243e-05
-9.326603190345641e-05
-8.393942871309664e-05
-6.528622233241243e-05
-6.52862223324242e-05
-7.461282552274864e-05
-6.528622233240064e-05
-6.528622233241242e-05
-6.528622233241242e-05
-5.5959619142076215e-05
-4.6633015951728205e-05
-2.7979809571032215e-05
-1.8653206380672435e-05
-2.7979809571032212e-05
-2.7979809571055773e-05
-1.865320638068421e-05
-2.797980957103221e-05
-2.797980957104399e-05
-3.73064127613802e-05
-3.7306412761368425e-05
-1.8653206380684212e-05
-2.797980957104399e-05
-2.797980957103221e-05
-1.8653206380684212e-05
-3.730641276139199e-05
-3.7306412761380215e-05
-2.7979809571032212e-05
-4.6633015951728205e-05
-5.5959619142052633e-05
-4.6633015951728205e-05
-3.730641276139199e-05
-3.730641276135664e-05
-5.595961914206442e-05
-5.595961914208798e-05
-4.6633015951728205e-05
-2.7979809571020435e-05
1.1779813235968107e-17
0.0
0.0
0.0
9.326603190336214e-06
1.8653206380684212e-05
1.1780940806227491e-17
-1.8653206380695996e-05
-9.326603190347998e-06
-9.326603190336218e-06

-9.32660319035978e-06
-9.326603190348001e-06
-9.326603190324437e-06
1.1781461223270284e-17
9.326603190336218e-06
9.326603190324435e-06
9.326603190336218e-06
0.0
-9.326603190347998e-06
0.0
0.0
0.0
-9.326603190347998e-06
0.0
9.326603190347998e-06
0.0
1.1781461223270284e-17
9.326603190359779e-06
1.1780940806227491e-17
-9.326603190336218e-06
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

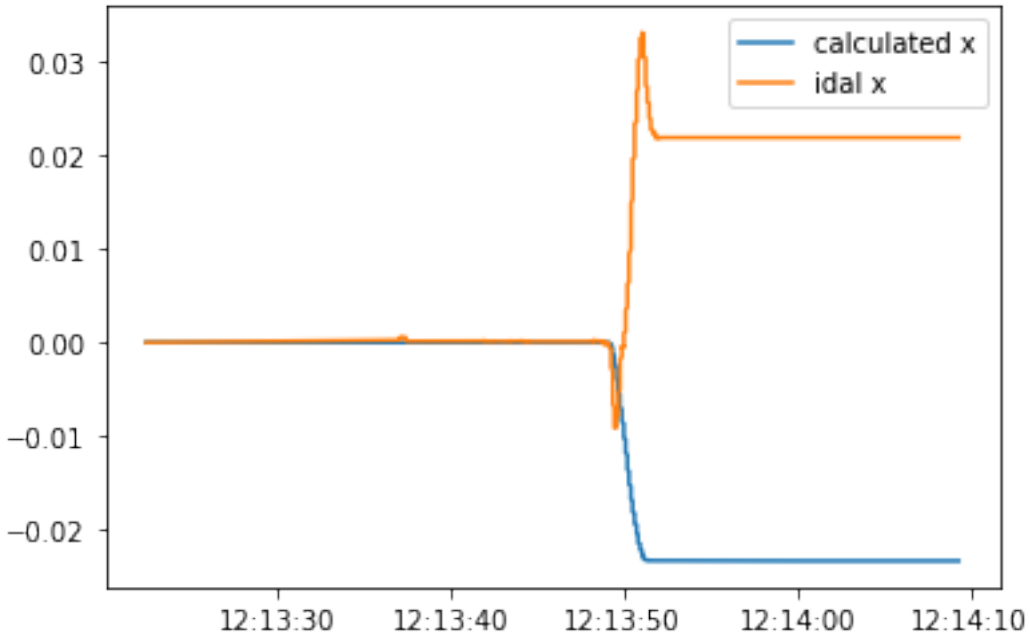
```
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
0.0
```

[ ]:
```
"""calculating displacement for calculated coordinates
```

```
"""
_xval = []
_yval = []
xf_disp = 0
yf_disp = 0
for i in range(len(cart_df["vx"])):
    if i == 0:
        _xval.append(0)
        _yval.append(0)
    else:
        x_disp = 0.5*(cart_df["cal_vx"].iloc[i] + cart_df["cal_vx"].
↪iloc[i-1])*0.01
        y_disp = 0.5*(cart_df["cal_vy"].iloc[i] + cart_df["cal_vy"].
↪iloc[i-1])*0.01
        # print(x_disp)
        xf_disp = xf_disp+x_disp
        yf_disp = yf_disp+y_disp
        _xval.append(xf_disp)
        _yval.append(yf_disp)

cart_df["cal_x"] = _xval
cart_df["cal_y"] = _yval
```

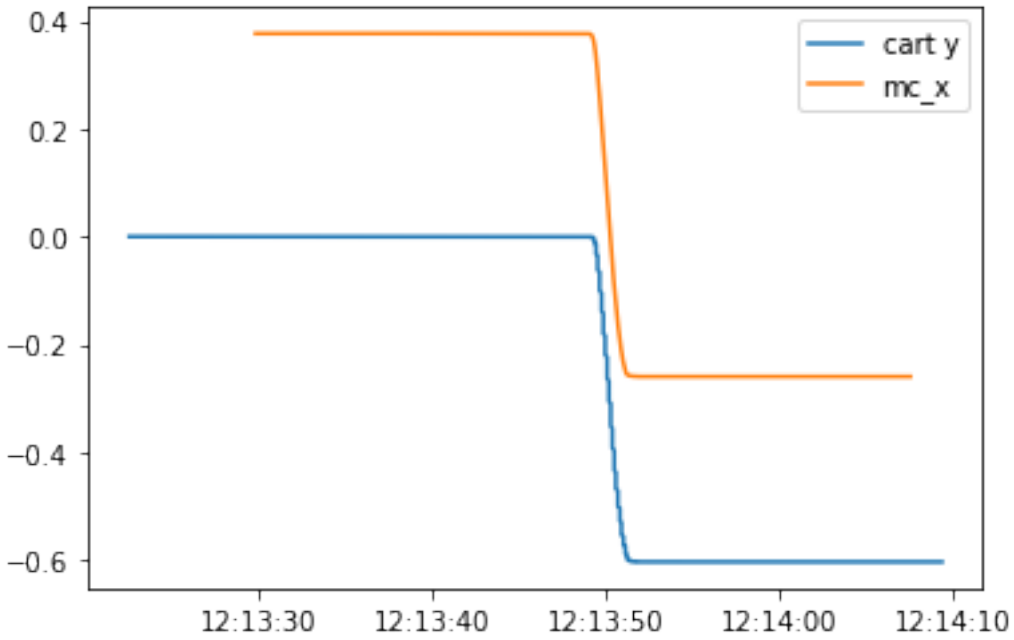
[ ]:
```
# plt.plot(cart_df["sys_time"], cart_df["x_val"], label = "ideal x")
plt.plot(cart_df["sys_time"],cart_df["cal_x"], label = "calculated x")
plt.plot(cart_df["sys_time"], cart_df["x_val"], label = "idal x")
plt.legend()
```

[ ]:
```
<matplotlib.legend.Legend at 0x1e7e14f0a88>
```



[ ]:
```
plt.plot(cart_df["sys_time"], cart_df["y_val"], label = "cart y")
plt.plot(mc["time"], mc["center_x"], label  = "mc_x")
plt.legend()
```

[ ]: <matplotlib.legend.Legend at 0x1e7e58ebb88>

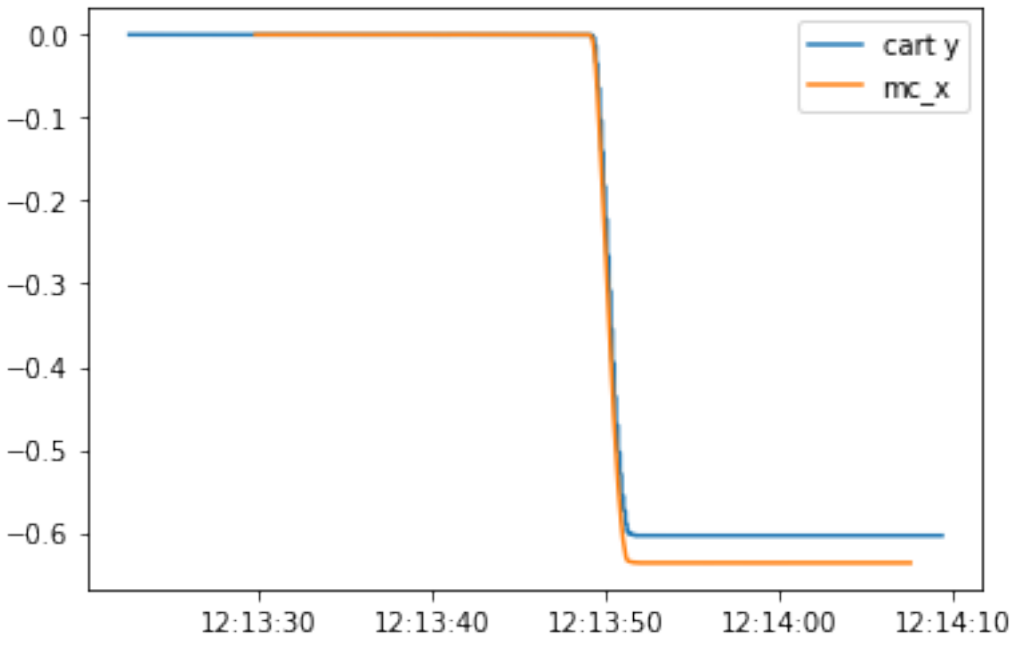


[ ]:
```
_t2 = []
for i in cart_df["sys_time"]:
    _tt = mc["time"][0]-i
    # print(_tt)
    _t2.append(_tt.total_seconds())
```

[ ]:
```
0      2022-06-15 12:13:22.489984
1      2022-06-15 12:13:37.124103
2      2022-06-15 12:13:37.124103
3      2022-06-15 12:13:37.124604
4      2022-06-15 12:13:37.124604
                  …
3529   2022-06-15 12:14:09.344631
3530   2022-06-15 12:14:09.345196
3531   2022-06-15 12:14:09.345196
3532   2022-06-15 12:14:09.345196
3533   2022-06-15 12:14:09.345671
Name: sys_time, Length: 3534, dtype: datetime64[ns]
```

[ ]:
```
offset = cart_df["y_val"].iloc[0] - mc["center_x"].iloc[0]
plt.plot(cart_df["sys_time"], cart_df["y_val"], label = "cart y")
plt.plot(mc["time"], mc["center_x"] +offset, label  = "mc_x")

plt.legend()
print(offset)
```

```
-0.376711
```



[ ]:
```
# print (cart_df["sys_time"].get_loc(mc["time"][0], method='nearest'))
print(cart_df["sys_time"][0])
print(mc["time"][0])

result_index = cart_df['sys_time'].sub(mc["time"][0]).abs().idxmin()

cart_df.iloc[(cart_df['sys_time'] - mc["time"][0]).abs().argsort()[0],:]
# (cart_df['sys_time'] - mc["time"][0]).abs()
result_index
```

```
2022-06-15 12:13:22.489984
2022-06-15 12:13:29.772000
```

[ ]:
```
0
```

[ ]:
```
f = interp1d(mc["time"], mc["center_x"])
a = f(cart_df["sys_time"])
mc["time"]
```

```
---------------------------------------------------------------------------
UFuncTypeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_6956/269749632.py in <module>
      1 f = interp1d(mc["time"], mc["center_x"])
----> 2 a = f(cart_df["sys_time"])
      3 mc["time"]

~\anaconda3\envs\py37t2\lib\site-packages\scipy\interpolate\polyint.py in␣
↪__call__(self, x)
     76         """
     77         x, x_shape = self._prepare_x(x)
---> 78         y = self._evaluate(x)
     79         return self._finish_y(y, x_shape)
     80

~\anaconda3\envs\py37t2\lib\site-packages\scipy\interpolate\interpolate.py in␣
↪_evaluate(self, x_new)
    680         #    The behavior is set by the bounds_error variable.
    681         x_new = asarray(x_new)
--> 682         y_new = self._call(self, x_new)
    683         if not self._extrapolate:
    684             below_bounds, above_bounds = self._check_bounds(x_new)

~\anaconda3\envs\py37t2\lib\site-packages\scipy\interpolate\interpolate.py in␣
↪_call_linear(self, x_new)
```

```
    628         # Note that the following two expressions rely on the specifics
 ↪of the
    629         # broadcasting semantics.
--> 630         slope = (y_hi - y_lo) / (x_hi - x_lo)[:, None]
    631
    632         # 5. Calculate the actual value for each entry in x_new.

UFuncTypeError: ufunc 'true_divide' cannot use operands with types␣
 ↪dtype('float64') and dtype('<m8[ns]')
```

[ ]:
```
# cart_df["diff"] = cart_df["y_val"]- cart_df["center_x"] - 0.34
```

[ ]:
```
rw = []
vr = []
for i in range(len(cart_df["av_rl"])):
    _vy = cart_df["cal_vy"][i]
    _vx = cart_df["cal_vx"][i]
    rw.append(np.arctan2(_vy, _vx))
    vr.append(np.linalg.norm((_vx, _vy)))
cart_df["rw"] = rw
cart_df["vr"] = vr
```

[ ]:
```
cart_df["rw"]
```

[ ]:
```
plt.plot(cart_df["rw"])
```

[ ]:
```
cart_df["rw"]
```