project-skip

February 21, 2023

[130]:
```
import numpy as np
import pandas as pd
import math
from numpy.linalg import inv
import plotly.express as px

pd.options.plotting.backend = "plotly"
```

[131]:
```
def rk4_integrator(x_0, t_0, t_n, dt, dx, u):
    x = x_0
    t = t_0

    yield (x, t)

    while t <= t_n:
        # eval x_k+1 using x_k and t_k
        k1 = dx(x, t, u)
        k2 = dx(x + dt * k1 / 2, t + dt / 2, u)
        k3 = dx(x + dt * k2 / 2, t + dt / 2, u)
        k4 = dx(x + dt * k3, t + dt, u)

        x = x + dt * (k1 + 2*k2 + 2*k3 + k4) * (1/6)

        # t increament to t_k+1
        t = t + dt

        yield (x, t)
```

[132]:
```
def plot_state_vs_time(df):
    fig = df.plot()
    fig.update_layout(height=700)
    fig.show()

M = np.array([
    [25.80, 0, 0],
    [0, 33.80, 6.2],
    [0, 6.2, 2.76]
```

```
])

Bt = np.array([
    [1, 0],
    [0, 0],
    [0, 1]
])

Minv = inv(M)

def eta(u, v, sai):
    return np.array([u, v, sai]).T

def nu(x, y, r):
    return np.array([x, y, r]).T

def x(x, y, r, u, v, sai):
    return np.array([x, y, r, u, v, sai]).T

def C(V):
    u = V[0]
    v = V[1]
    r = V[2]
    c13 = -33.8*v -(6.2+6.2)*r/2
    c23 = 25.8*u
    return np.array([
        [0, 0, c13],
        [0, 0, c23],
        [-c13, -c23, 0]
    ])

def D(V):
    u = V[0]
    v = V[1]
    r = V[2]
    return -np.array([
        [-12 - 2.1*abs(u), 0, 0],
        [0, -0.17 -0.45*abs(v) , -0.2],
        [0, -0.5, -0.5 - 0.1*abs(r)]
    ])

def R(sai):
    return np.array([
        [np.cos(sai), -np.sin(sai), 0],
        [np.sin(sai), np.cos(sai), 0],
        [0, 0, 1]
    ])
```

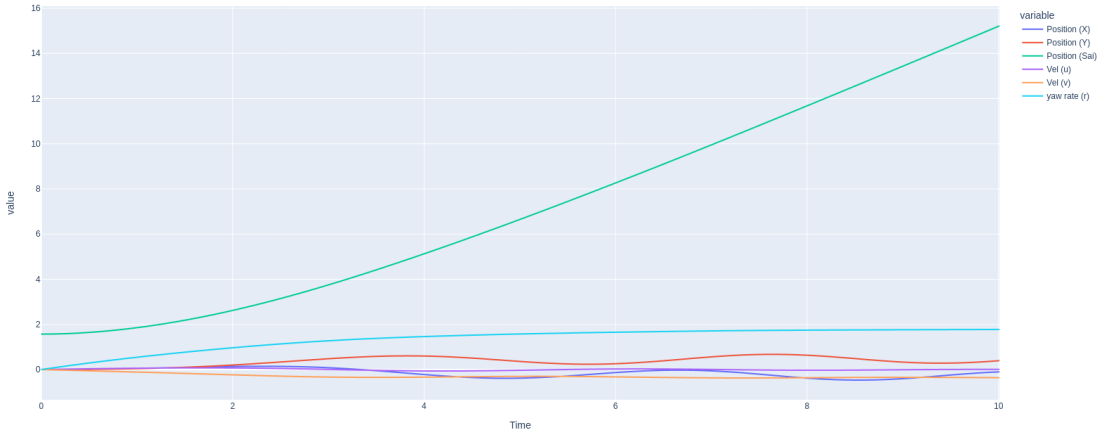
```
def fx(V, sai):
    a = R(sai)@V
    b = -Minv@(C(V) + D(V))@V
    c = np.array([a, b]).reshape(-1)
    return c


def B():
    a = np.array([
        [0, 0],
        [0, 0],
        [0, 0]
    ])
    b = Minv@Bt
    c = np.concatenate((a, b), axis=0)
    return c

# X = [x, y, r, u, v, sai]
def dx(X, t = None, U = np.array([0, 0])):
    V = X[3:]
    sai = X[2]
    return fx(V, sai) + B()@U
```

[133]:
```
# Testing dX

dx(np.array([1, 2, 3, 4, 5, 6]), np.array([0, 0]))
```

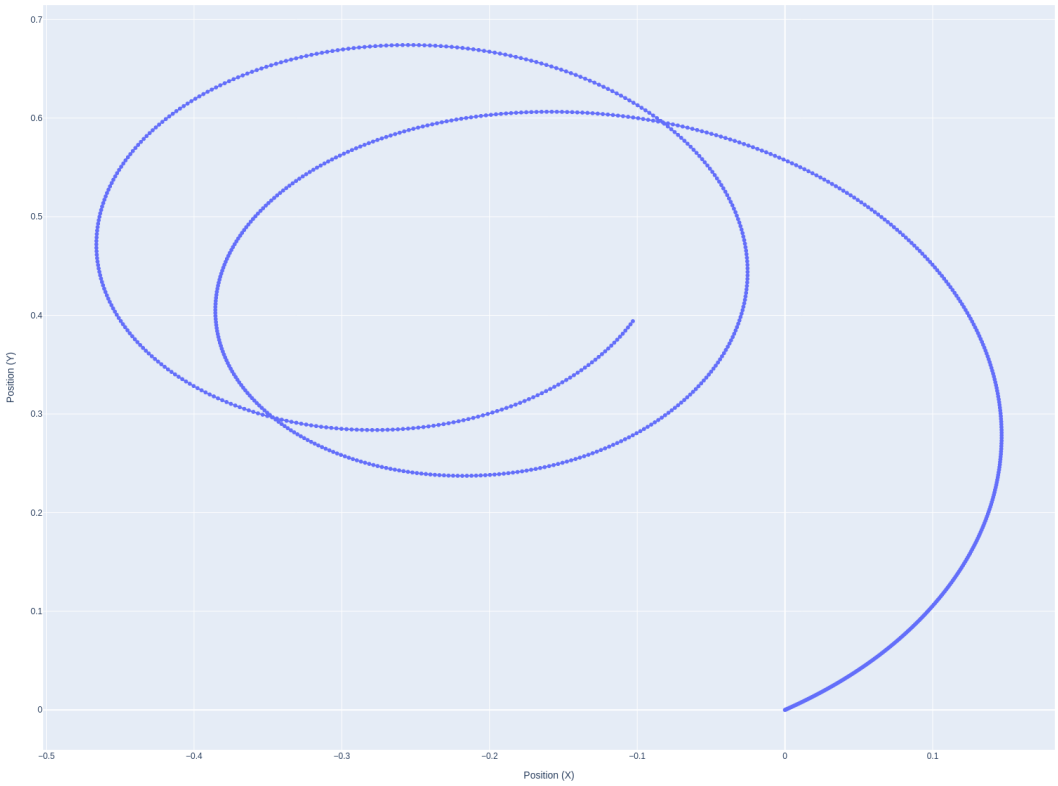
[133]:
```
array([  -4.66557003,   -4.38548245,    6.        ,   44.79069767,
          4.10735123, -124.40781797])
```

[134]:
```
# X_0 = np.array([3.5, 2, np.pi / 2, 1, -2, 1])

'''
initial conditions
x = 0
y = 0
sai = 90deg clockwise with y-axis
u = 0
v = 0
r = 0

F_u = 2
F_v = 1
'''
X_0 = np.array([0, 0, np.pi / 2, 0, 0, 0])
U = [2, 1]
```

```
t_0 = 0
T = 10
dt = 0.01

state_time = []
state_integrator = rk4_integrator(X_0, t_0, T, dt, dx, np.array(U))
for x, t in state_integrator:
    state_time.append({"Position (X)": x[0], "Position (Y)": x[1], "Position␣
 ↪(Sai)": x[2], "Vel (u)": x[3], "Vel (v)": x[4], "yaw rate (r)": x[5], "Time":
 ↪ t})
r_state_time_df = pd.DataFrame(state_time).set_index("Time")

# state_time_df = pd.concat([r_state_time_df, e_state_time_df])
plot_state_vs_time(r_state_time_df)
```



[138]:
```
import plotly.express as px
fig = px.line(r_state_time_df, x='Position (X)', y='Position (Y)',␣
 ↪text="Position (Sai)", height=1200)
fig.show()
```


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Position (Y)
−0.5
−0.4
−0.3
−0.2
−0.1
0
0.1
Position (X)

[ ]: