

AV Behavior with friction implementation in rainy weather

In [1]:

```
import os
import pandas as pd
pd.set_option('display.max_colwidth', None)
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import scipy.stats
from scipy.stats import norm, binom, poisson
from dtaidistance import dtw
import json
```

Testing Parameters

In [2]:

```
weather_param = [[20,0,0,5,0,0,0,10,0],
                 [20,20,20,5,0,20,5,20,0],
                 [40,40,40,5,0,40,10,30,0],
                 [60,60,60,5,0,60,15,40,0],
                 [80,80,80,5,0,80,20,50,0],
                 [100,100,100,5,0,100,30,70,0]]

def calc_fric(weather_list):
    return np.exp(-0.916*weather_list[5]/100) * (1-weather_list[5]/100)**3 * 0.6 + 0.4 - 0.1*wea

for i in range(len(weather_param)):
    friction_ratio = calc_fric(weather_param[i])
    weather_param[i] += [friction_ratio]

df = pd.DataFrame(weather_param, columns = ["cloudiness","precipitation","precipitation_deposits"])
# df.index.name = 'Scenarios'
df
```

Out[2]:

	cloudiness	precipitation	precipitation_deposits	sun_altitude_angle	sun_azimuth_angle	wetness	fog_density	w
0	20	0	0	5	0	0	0	0
1	20	20	20	5	0	20	5	
2	40	40	40	5	0	40	10	
3	60	60	60	5	0	60	15	
4	80	80	80	5	0	80	20	
5	100	100	100	5	0	100	30	



With reduced friction

In [3]:

```
os.chdir("C:\\\\Users\\\\kilob\\\\Senior Research & Thesis\\\\Simulations")
```

In [4]:

```
rel_path = "./Data/Ghost Cutin/Rain Reduced Friction Exponential"
os.path.isdir(rel_path)
```

```
Out[4]: True
```

```
In [5]: subfolders = os.listdir(path=rel_path)
subfolders
```

```
Out[5]: ['route_highway_epoch24_clear-sunset_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-100_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-20_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-40_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-60_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-80_fi_ghost_cutin']
```

```
In [6]: subfolders.sort()
subfolders
```

```
Out[6]: ['route_highway_epoch24_clear-sunset_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-100_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-20_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-40_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-60_fi_ghost_cutin',
 'route_highway_epoch24_rain-sunset-80_fi_ghost_cutin']
```

```
In [7]: txt_lists_fric = [[] for i in range(6)]

for i in range(len(txt_lists_fric)):
    for root, dirs, files in os.walk(os.path.join(rel_path, subfolders[i])):
        for file in files:
            if file.endswith(".txt"):
                with open(os.path.join(root, file), encoding = 'utf-8') as f:
                    read_string = f.read()
                    json_object = json.loads(read_string)
                    txt_lists_fric[i].append(json_object)
```

```
In [8]: txt_lists_fric[2][50]
# txt_lists_fric[0]
```

```
Out[8]: {'_checkpoint': {'global_record': {},  
    'progress': [0, 1],  
    'records': [{'index': 0,  
        'infractions': {'collisions_layout': [],  
            'collisions_pedestrian': [],  
            'collisions_vehicle': ['Agent collided against object with type=vehicle.tesla.model3 and id  
=82788 at (x=193.154, y=-16.8, z=-0.02)'],  
            'outside_route_lanes': [],  
            'red_light': [],  
            'route_dev': [],  
            'route_timeout': [],  
            'stop_infraction': [],  
            'vehicle_blocked': []},  
        'meta': {'duration_game': 10.82500016130507,  
            'duration_system': 22.345438957214355,  
            'route_length': 131.56012567009267},  
        'route_id': 'RouteScenario_0',  
        'scores': {'score_composed': 31.64542482747425,  
            'score_penalty': 0.6,  
            'score_route': 52.742374712457085},  
        'status': 'Failed'}]},  
    'eligible': False,  
    'entry_status': 'Started',  
    'labels': [],  
    'sensors': ['carla_camera',  
        'carla_camera',  
        'carla_camera',  
        'carla_imu',  
        'carla_gnss',  
        'carla_speedometer'],  
    'values': []}
```

```
In [9]: avg_complete = [ ] * 6  
  
for i in range(len(txt_lists_fric)):  
    count = 0  
    for txt in txt_lists_fric[i]:  
        if txt['_checkpoint']['records'][0]['status'] == 'Completed':  
            count += 1  
    avg_complete[i] = count/100  
  
df = pd.DataFrame({'Scenarios':subfolders, 'Average complete ratio':avg_complete})  
# df.index.name = 'Scenarios'  
  
df
```

	Scenarios	Average complete ratio
0	route_highway_epoch24_clear-sunset_fi_ghost_cutin	1.00
1	route_highway_epoch24_rain-sunset-100_fi_ghost_cutin	1.00
2	route_highway_epoch24_rain-sunset-20_fi_ghost_cutin	0.00
3	route_highway_epoch24_rain-sunset-40_fi_ghost_cutin	0.00
4	route_highway_epoch24_rain-sunset-60_fi_ghost_cutin	0.99
5	route_highway_epoch24_rain-sunset-80_fi_ghost_cutin	0.02

```
In [10]: count = 0  
  
avg_score = [ ] * 6  
  
for i in range(len(txt_lists_fric)):
```

```

count = 0
for txt in txt_lists_fric[i]:
    count += txt['_checkpoint']['records'][0]['scores']['score_route']
avg_score[i] = count/100

df = pd.DataFrame({'Scenarios':subfolders, 'Average score route':avg_complete})
df

```

Out[10]:

	Scenarios	Average score route
0	route_highway_epoch24_clear-sunset_fi_ghost_cutin	1.00
1	route_highway_epoch24_rain-sunset-100_fi_ghost_cutin	1.00
2	route_highway_epoch24_rain-sunset-20_fi_ghost_cutin	0.00
3	route_highway_epoch24_rain-sunset-40_fi_ghost_cutin	0.00
4	route_highway_epoch24_rain-sunset-60_fi_ghost_cutin	0.99
5	route_highway_epoch24_rain-sunset-80_fi_ghost_cutin	0.02

Setting up into list of DataFrames

In [11]:

```
os.chdir("C:\\Users\\kilob\\Senior Research & Thesis\\Simulations")
```

In [12]:

```
os.path.isdir("./Data/Ghost Cutin/Rain Original/Simulations Rain PC/route_highway_epoch24_clear-
```

Out[12]:

```
True
```

In [13]:

```
df_array_fric = np.empty((6,100), dtype=object)

for i in range(len(df_array_fric)):
    for j in range((len(df_array_fric[i]))):
        df_array_fric[i][j] = pd.DataFrame()
```

In [14]:

```
for i in range(len(txt_lists_fric)):
    dir_path = os.path.join(rel_path, subfolders[i])
    print(dir_path)
    # List to store files
    res = []

    # Iterate directory
    run_index = 0
    for path in os.listdir(dir_path):
        # check if current path is a file
        if not os.path.isfile(os.path.join(dir_path, path)):
            df = pd.DataFrame()
            for file in os.listdir(os.path.join(dir_path, path)):
                file_path = os.path.join(dir_path, os.path.join(path, file))
                # print("file_path", file_path)
                if "_ctl.csv" in file_path:
                    df = pd.read_csv(file_path)
                elif "_cvip.csv" in file_path:
                    temp = pd.read_csv(file_path)
                    df = pd.concat([df, temp], axis=1)
                elif "_traj.csv" in file_path:
                    temp = pd.read_csv(file_path)
                    df = pd.concat([df, temp], axis=1)

            df_array_fric[i][run_index] = df
            run_index += 1
```

```
./Data/Ghost Cutin/Rain Reduced Friction Exponential\route_highway_epoch24_clear-sunset_fi_ghost_cutin  
./Data/Ghost Cutin/Rain Reduced Friction Exponential\route_highway_epoch24_rain-sunset-100_fi_ghost_cutin  
./Data/Ghost Cutin/Rain Reduced Friction Exponential\route_highway_epoch24_rain-sunset-20_fi_ghost_cutin  
./Data/Ghost Cutin/Rain Reduced Friction Exponential\route_highway_epoch24_rain-sunset-40_fi_ghost_cutin  
./Data/Ghost Cutin/Rain Reduced Friction Exponential\route_highway_epoch24_rain-sunset-60_fi_ghost_cutin  
./Data/Ghost Cutin/Rain Reduced Friction Exponential\route_highway_epoch24_rain-sunset-80_fi_ghost_cutin
```

In [15]: `df_array_fric[5][99]`

Out[15]:

	ts	agent_id	throttle	steer	brake	ts	agent_id	cvip	cvip_x	cvip_y	ci
0	2425249	0	0.9	-0.002883	0.0	2425249	0	500.497261	198.767441	-95.832657	-499.78
1	2425250	0	0.9	0.000204	0.0	2425250	0	5.598742	195.567444	-90.832657	0.10
2	2425251	0	0.9	-0.013590	0.0	2425251	0	5.595580	195.567444	-90.832657	0.09
3	2425252	0	0.9	-0.005940	0.0	2425252	0	5.592744	195.567444	-90.832657	0.08
4	2425253	0	0.9	-0.003160	0.0	2425253	0	5.590235	195.567444	-90.832657	0.06
...
422	2425671	0	0.0	0.015676	1.0	2425671	0	3.268278	195.070969	-18.427528	-0.06
423	2425672	0	0.0	0.011617	1.0	2425672	0	3.289026	194.964508	-18.154963	-0.06
424	2425673	0	0.0	0.009099	1.0	2425673	0	3.316099	194.855225	-17.883526	-0.06
425	2425674	0	0.0	0.005824	1.0	2425674	0	3.350513	194.743454	-17.612896	-0.06
426	2425675	0	0.0	0.002597	1.0	2425675	0	3.393804	194.630203	-17.342436	-0.06

427 rows × 17 columns

In [16]: `df_array_fric[0][0]['x']`

Out[16]:

```
0    192.362411
1    192.362411
2    192.362411
3    192.362411
4    192.362411
      ...
752   193.186905
753   193.184387
754   193.181885
755   193.179367
756   193.176895
Name: x, Length: 757, dtype: float64
```

Without friction reduction

In [17]: `os.chdir("C:\\\\Users\\\\kilob\\\\Senior Research & Thesis\\\\Simulations")`

In [18]: `rel_path = "./Data/Ghost Cutin/Rain Original/Simulations Rain PC"`
`os.path.isdir(rel_path)`

```
Out[18]: True
```

```
In [19]: subfolders = os.listdir(path=rel_path)  
subfolders
```

```
Out[19]: ['route_highway_epoch24_clear-sunset_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-100_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-20_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-40_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-60_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-80_fi_ghost_cutin']
```

```
In [20]: subfolders.sort()  
subfolders
```

```
Out[20]: ['route_highway_epoch24_clear-sunset_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-100_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-20_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-40_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-60_fi_ghost_cutin',  
          'route_highway_epoch24_rain-sunset-80_fi_ghost_cutin']
```

```
In [21]: txt_lists_fric = [[] for i in range(6)]  
  
for i in range(len(txt_lists_fric)):  
    for root, dirs, files in os.walk(os.path.join(rel_path, subfolders[i])):  
        for file in files:  
            if file.endswith(".txt"):  
                with open(os.path.join(root, file), encoding = 'utf-8') as f:  
                    read_string = f.read()  
                    json_object = json.loads(read_string)  
                    txt_lists_fric[i].append(json_object)
```

```
In [22]: txt_lists_fric[2][50]  
# txt_lists_fric[0]
```

```
Out[22]: {'_checkpoint': {'global_record': {},  
    'progress': [0, 1],  
    'records': [{index: 0,  
        'infractions': {'collisions_layout': [],  
            'collisions_pedestrian': [],  
            'collisions_vehicle': [],  
            'outside_route_lanes': []},  
        'red_light': [],  
        'route_dev': [],  
        'route_timeout': [],  
        'stop_infraction': [],  
        'vehicle_blocked': []}],  
    'meta': {'duration_game': 19.05000028386712,  
        'duration_system': 36.68422293663025,  
        'route_length': 131.56012567009267},  
    'route_id': 'RouteScenario_0',  
    'scores': {'score_composed': 100.0,  
        'score_penalty': 1.0,  
        'score_route': 100.0},  
    'status': 'Completed'}]},  
    'eligible': False,  
    'entry_status': 'Started',  
    'labels': [],  
    'sensors': ['carla_camera',  
        'carla_camera',  
        'carla_camera',  
        'carla_imu',  
        'carla_gnss',  
        'carla_speedometer'],  
    'values': []}
```

```
In [23]: avg_complete = [ ] * 6  
  
for i in range(len(txt_lists_fric)):  
    count = 0  
    for txt in txt_lists_fric[i]:  
        if txt['_checkpoint']['records'][0]['status'] == 'Completed':  
            count += 1  
    avg_complete[i] = count/100  
  
df = pd.DataFrame({'Scenarios':subfolders, 'Average complete ratio':avg_complete})  
# df.index.name = 'Scenarios'  
  
df
```

```
Out[23]:
```

	Scenarios	Average complete ratio
0	route_highway_epoch24_clear-sunset_fi_ghost_cutin	1.0
1	route_highway_epoch24_rain-sunset-100_fi_ghost_cutin	1.0
2	route_highway_epoch24_rain-sunset-20_fi_ghost_cutin	1.0
3	route_highway_epoch24_rain-sunset-40_fi_ghost_cutin	1.0
4	route_highway_epoch24_rain-sunset-60_fi_ghost_cutin	1.0
5	route_highway_epoch24_rain-sunset-80_fi_ghost_cutin	1.0

```
In [24]: count = 0  
  
avg_score = [ ] * 6  
  
for i in range(len(txt_lists_fric)):  
    count = 0
```

```

for txt in txt_lists_fric[i]:
    count += txt['_checkpoint']['records'][0]['scores']['score_route']
avg_score[i] = count/100

df = pd.DataFrame({'Scenarios':subfolders, 'Average score route':avg_complete})
df

```

Out[24]:

	Scenarios	Average score route
0	route_highway_epoch24_clear-sunset_fi_ghost_cutin	1.0
1	route_highway_epoch24_rain-sunset-100_fi_ghost_cutin	1.0
2	route_highway_epoch24_rain-sunset-20_fi_ghost_cutin	1.0
3	route_highway_epoch24_rain-sunset-40_fi_ghost_cutin	1.0
4	route_highway_epoch24_rain-sunset-60_fi_ghost_cutin	1.0
5	route_highway_epoch24_rain-sunset-80_fi_ghost_cutin	1.0

Setting up into list of DataFrames

```
In [25]: os.chdir("C:\\Users\\kilob\\Senior Research & Thesis\\Simulations")
```

```
In [26]: os.path.isdir("./Data/Ghost Cutin/Rain Original/Simulations Rain PC/route_highway_epoch24_clear-")
```

Out[26]: True

```
In [27]: df_array_orig = np.empty((6,100), dtype=object)

for i in range(len(df_array_orig)):
    for j in range(len(df_array_orig[i])):
        df_array_orig[i][j] = pd.DataFrame()
```

```
In [28]: rel path = "./Data/Ghost Cutin/Rain Original/Simulations Rain PC/"
```

```
for i in range(len(txt_lists_fric)):
    dir_path = os.path.join(rel_path, subfolders[i])
    print(dir_path)

# List to store files
res = []

# Iterate directory
run_index = 0
for path in os.listdir(dir_path):
    # check if current path is a file
    if not os.path.isfile(os.path.join(dir_path, path)):
        df = pd.DataFrame()
        for file in os.listdir(os.path.join(dir_path, path)):
            file_path = os.path.join(dir_path, os.path.join(path, file))
            print("file_path", file_path)
            if "_ctl.csv" in file_path:
                df = pd.read_csv(file_path)
            elif "_cvip.csv" in file_path:
                temp = pd.read_csv(file_path)
                df = pd.concat([df, temp], axis=1)
            elif "_traj.csv" in file_path:
                temp = pd.read_csv(file_path)
                df = pd.concat([df, temp], axis=1)
```

```

df_array_orig[i][run_index] = df
run_index += 1

./Data/Ghost Cutin/Rain Original/Simulations Rain PC/route_highway_epoch24_clear-sunset_fi_ghost_cutin
./Data/Ghost Cutin/Rain Original/Simulations Rain PC/route_highway_epoch24_rain-sunset-100_fi_ghost_cutin
./Data/Ghost Cutin/Rain Original/Simulations Rain PC/route_highway_epoch24_rain-sunset-20_fi_ghost_cutin
./Data/Ghost Cutin/Rain Original/Simulations Rain PC/route_highway_epoch24_rain-sunset-40_fi_ghost_cutin
./Data/Ghost Cutin/Rain Original/Simulations Rain PC/route_highway_epoch24_rain-sunset-60_fi_ghost_cutin
./Data/Ghost Cutin/Rain Original/Simulations Rain PC/route_highway_epoch24_rain-sunset-80_fi_ghost_cutin

```

In [29]: `df_array_orig[5][99]`

Out[29]:

	ts	agent_id	throttle	steer	brake	ts	agent_id	cvip	cvip_x	cvip_y	v
0	2174716	0	0.900000	-0.007014	0.0	2174716	0	500.491189	198.767441	-95.832657	-499.8
1	2174717	0	0.900000	-0.003541	0.0	2174717	0	5.595580	195.567444	-90.832657	0.1
2	2174718	0	0.900000	-0.002026	0.0	2174718	0	5.592365	195.567444	-90.832657	0.0
3	2174719	0	0.900000	-0.004804	0.0	2174719	0	5.589578	195.567444	-90.832657	0.0
4	2174720	0	0.900000	-0.003671	0.0	2174720	0	5.587154	195.567444	-90.832657	0.0
...
781	2175497	0	0.669983	-0.001719	0.0	2175497	0	64.363119	192.148407	107.512459	0.0
782	2175498	0	0.515898	-0.001412	0.0	2175498	0	64.471228	192.118759	107.842316	0.0
783	2175499	0	0.591930	-0.000968	0.0	2175499	0	64.578560	192.087906	108.172234	0.0
784	2175500	0	0.604719	-0.000167	0.0	2175500	0	64.685191	192.055908	108.502243	0.0
785	2175501	0	0.593757	-0.000461	0.0	2175501	0	64.791258	192.022827	108.832397	0.0

786 rows × 17 columns

In [30]: `df_array_orig[2][1]['x']`

Out[30]:

0	192.362411
1	192.362411
2	192.362411
3	192.362411
4	192.362411
...	...
757	193.180420
758	193.177841
759	193.175354
760	193.172867
761	193.170441

Name: x, Length: 762, dtype: float64

Comparison

In [31]: `attribute_order = ["x", "y", "v", "cvip", "steer", "brake", "throttle"]`

```
In [32]: friction_legend = [" with original implementation", " with friction implementation"]
full_legend = []
for i in range(len(subfolders)):
    for j in range(len(friction_legend)):
        full_legend.append(subfolders[i]+friction_legend[j])
```

```
Out[32]: ['route_highway_epoch24_clear-sunset_fi_ghost_cutin with original implementation',
          'route_highway_epoch24_clear-sunset_fi_ghost_cutin with friction implementation',
          'route_highway_epoch24_rain-sunset-100_fi_ghost_cutin with original implementation',
          'route_highway_epoch24_rain-sunset-100_fi_ghost_cutin with friction implementation',
          'route_highway_epoch24_rain-sunset-20_fi_ghost_cutin with original implementation',
          'route_highway_epoch24_rain-sunset-20_fi_ghost_cutin with friction implementation',
          'route_highway_epoch24_rain-sunset-40_fi_ghost_cutin with original implementation',
          'route_highway_epoch24_rain-sunset-40_fi_ghost_cutin with friction implementation',
          'route_highway_epoch24_rain-sunset-60_fi_ghost_cutin with original implementation',
          'route_highway_epoch24_rain-sunset-60_fi_ghost_cutin with friction implementation',
          'route_highway_epoch24_rain-sunset-80_fi_ghost_cutin with original implementation',
          'route_highway_epoch24_rain-sunset-80_fi_ghost_cutin with friction implementation']
```

x,y traces

```
In [59]: for weather in range(len(txt_lists_fric)):
    plt.figure(figsize=(15,8))
    df_orig_x = pd.DataFrame()
    df_orig_y = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['x'].dtypes == 'object' or df_array_orig[weather][i]['y'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_orig[weather][i]['x'].isnull().values.any() and not df_array_orig[i]['y'].isnull().values.any():
                df_orig_x["Run"+str(i)] = df_array_orig[weather][i]['x']
                df_orig_y["Run"+str(i)] = df_array_orig[weather][i]['y']

    median_orig_x = df_orig_x.median(axis=1)
    median_orig_y = df_orig_y.median(axis=1)

    plt.plot(median_orig_y, median_orig_x, alpha=0.7)

df_fric_x = pd.DataFrame()
df_fric_y = pd.DataFrame()

for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['x'].dtypes == 'object' or df_array_fric[weather][i]['y'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['x'].isnull().values.any() and not df_array_fric[weather][i]['y'].isnull().values.any():
            df_fric_x["Run"+str(i)] = df_array_fric[weather][i]['x']
            df_fric_y["Run"+str(i)] = df_array_fric[weather][i]['y']
```

```

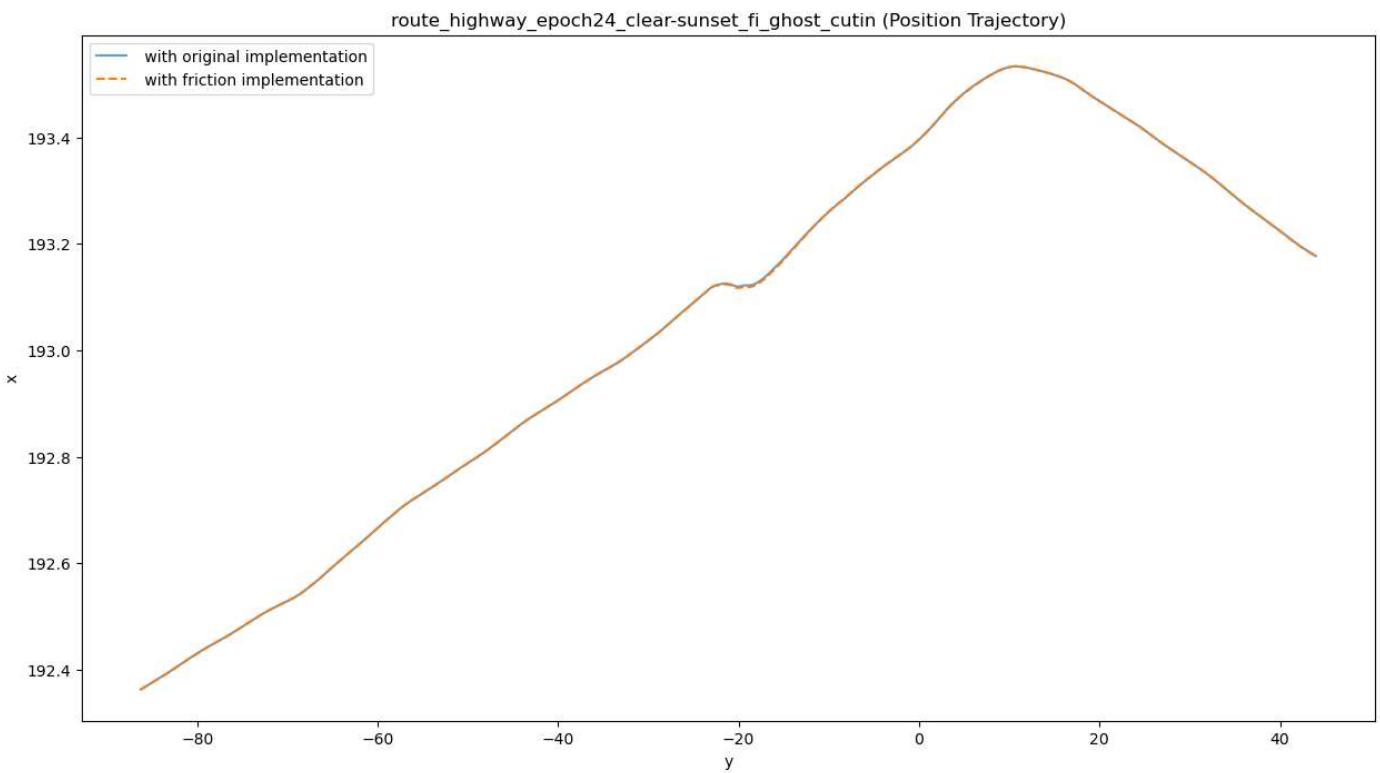
median_fric_x = df_fric_x.median(axis=1)
median_fric_y = df_fric_y.median(axis=1)

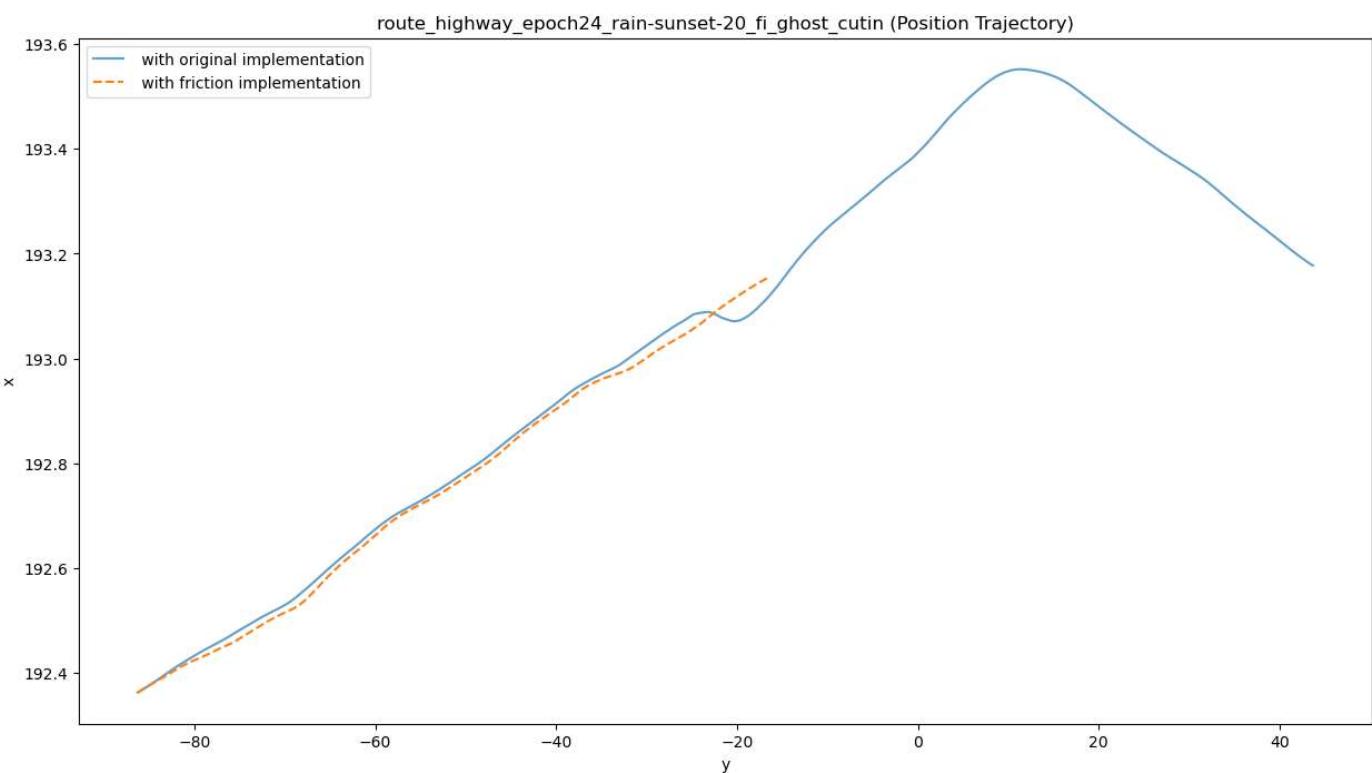
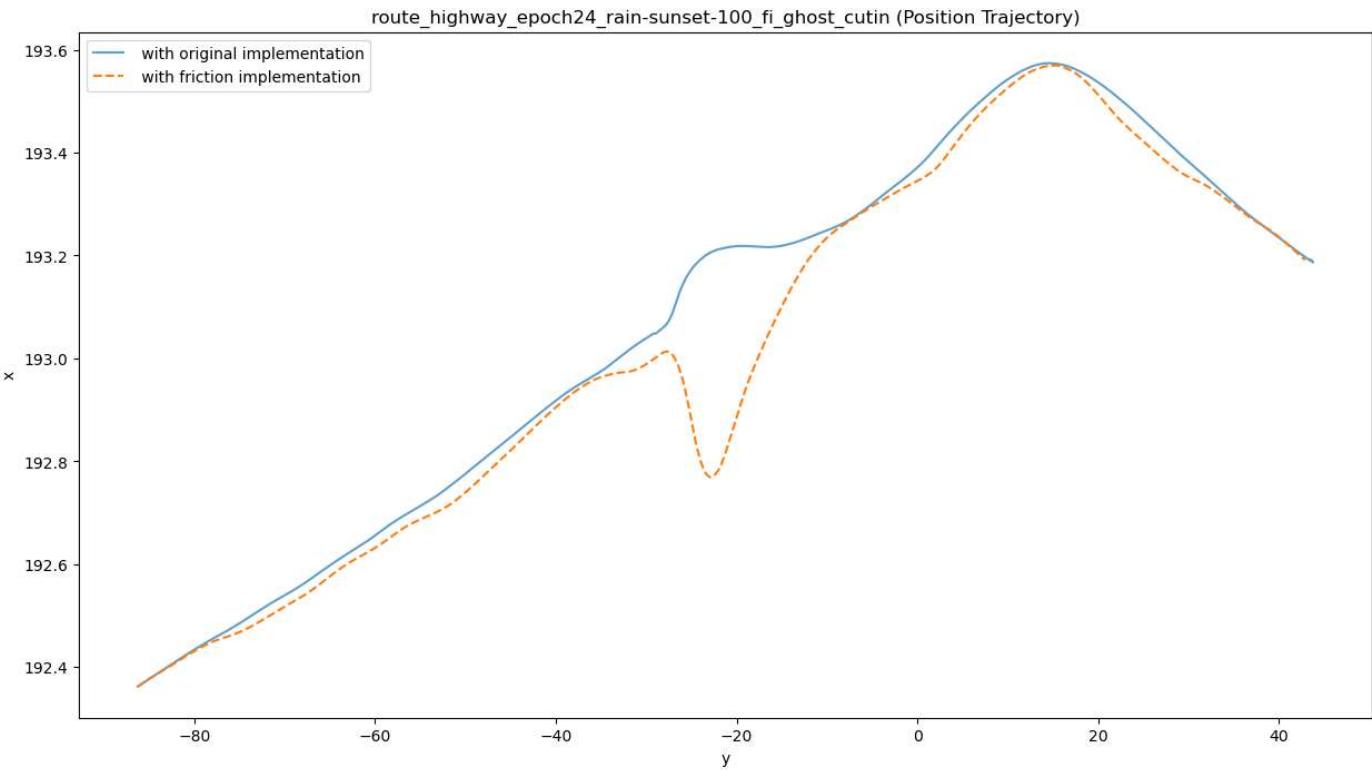
all_medians[weather][0].append(median_orig_x)
all_medians[weather][0].append(median_fric_x)
all_medians[weather][1].append(median_orig_y)
all_medians[weather][1].append(median_fric_y)

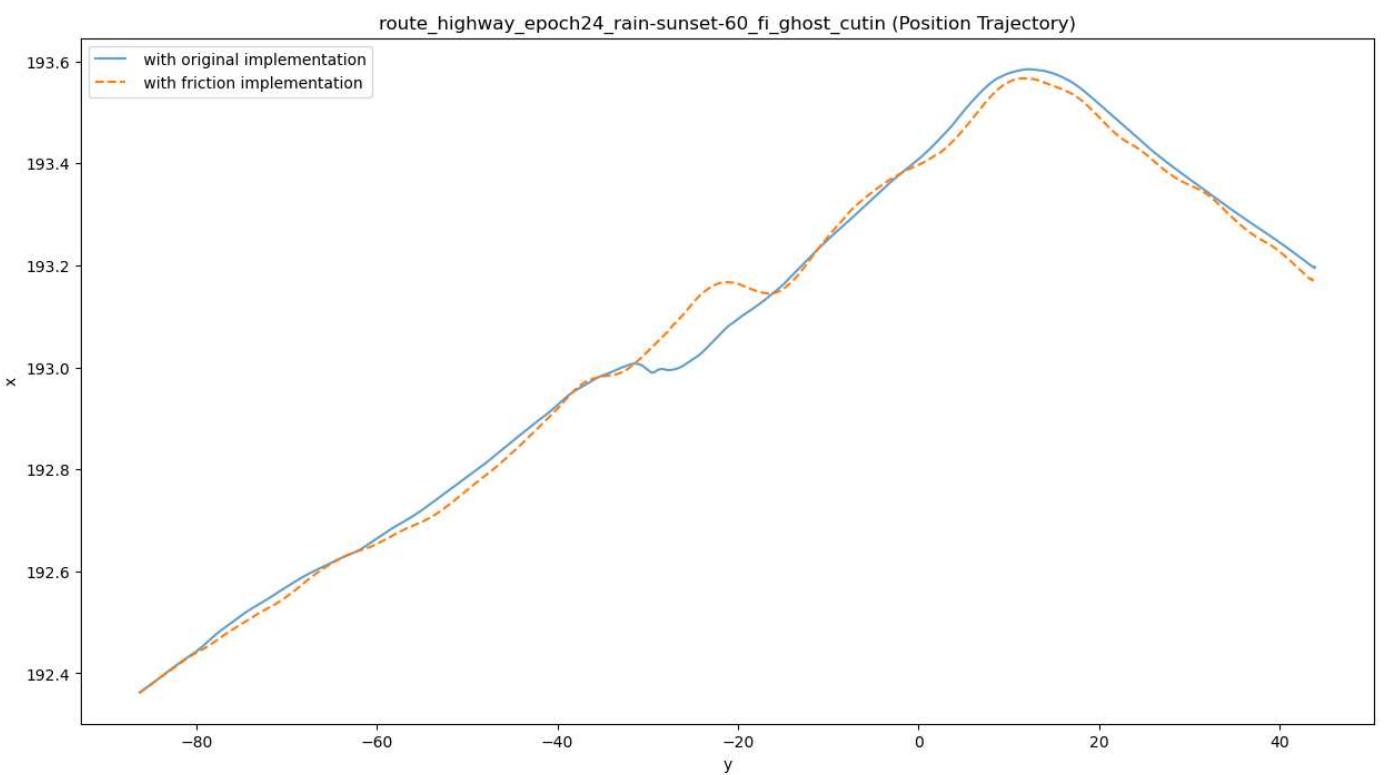
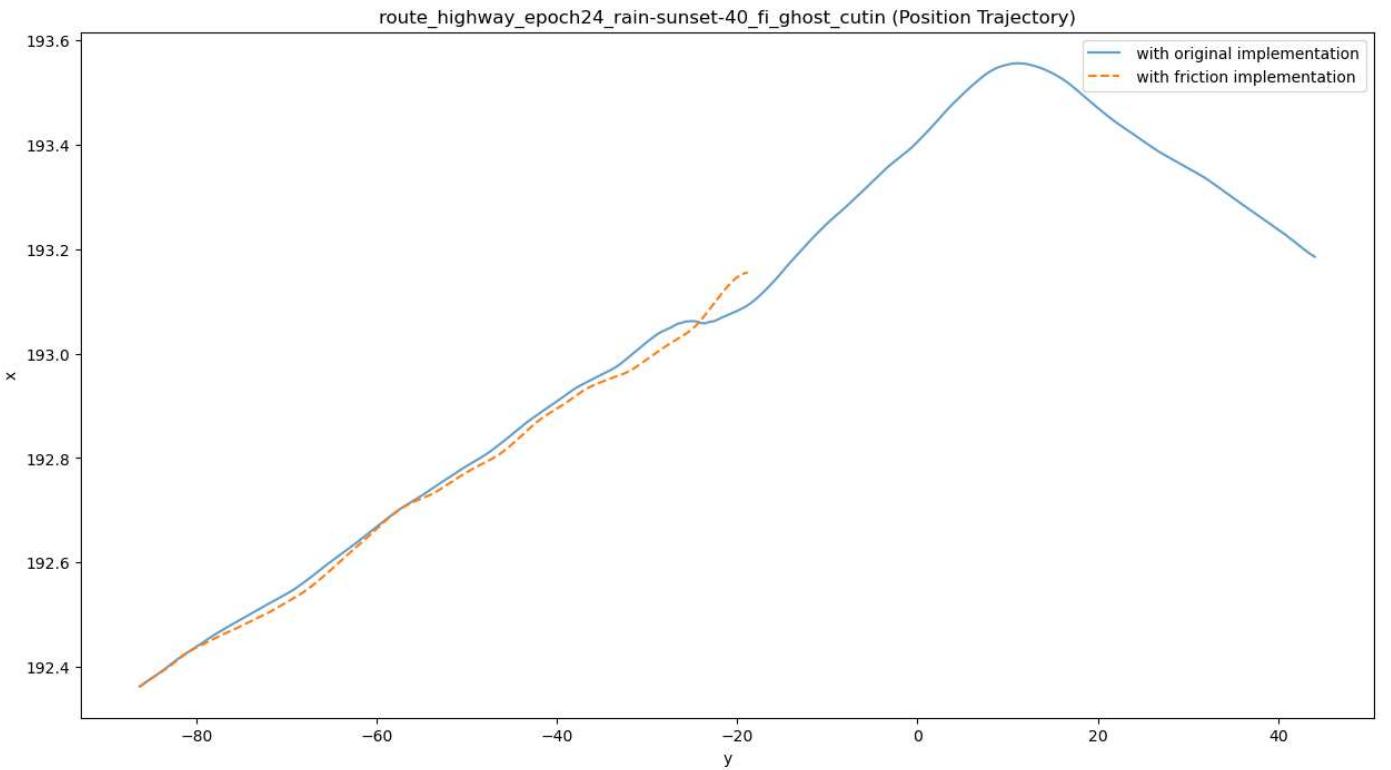
plt.plot(median_fric_y, median_fric_x, '--')
plt.xlabel("y")
plt.ylabel("x")

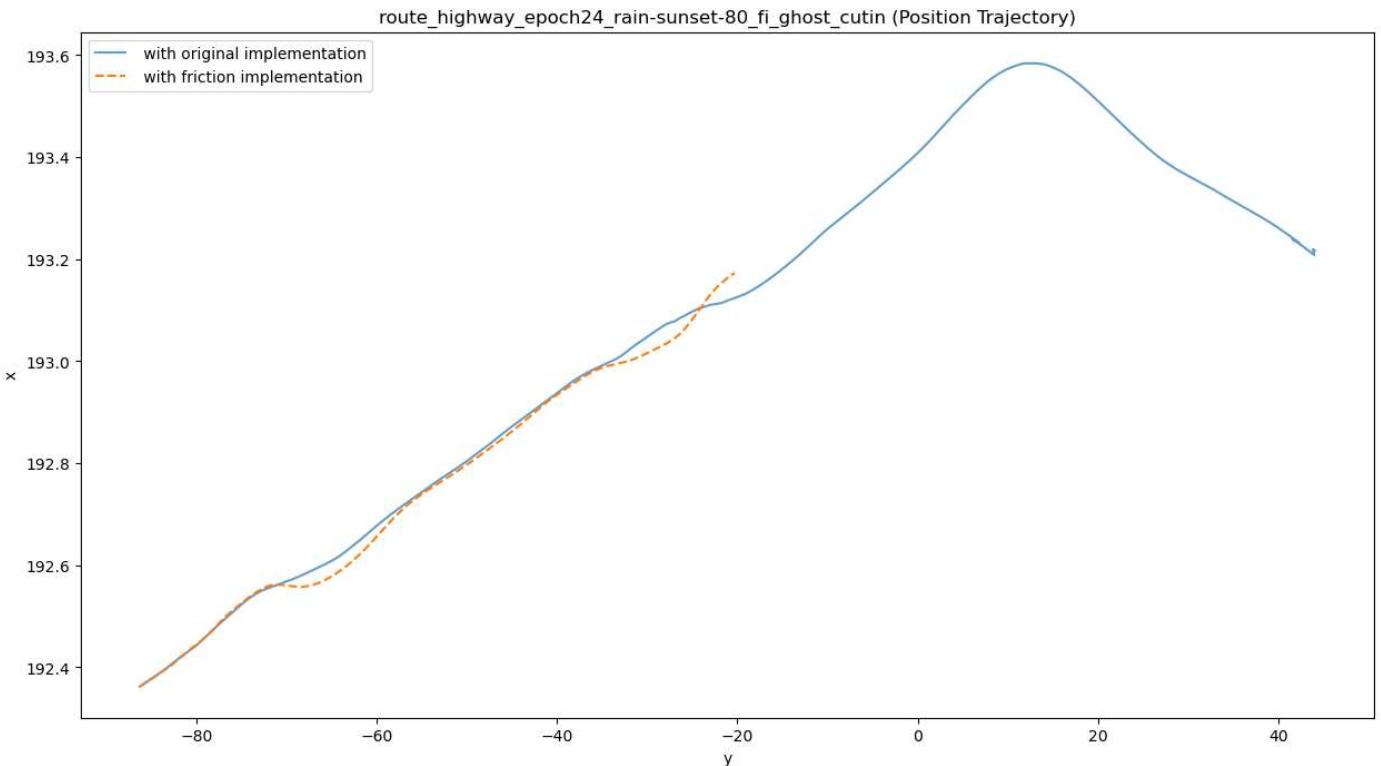
plt.title(subfolders[weather] + " (Position Trajectory)")
plt.legend(friction_legend)
plt.show()

```









```
In [34]: plt.figure(figsize=(20,10))

for weather in range(len(txt_lists_fric)):
    df_orig_x = pd.DataFrame()
    df_orig_y = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['x'].dtypes == 'object' or df_array_orig[weather][i]['y'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_orig[weather][i]['x'].isnull().values.any() and not df_array_orig[i]['y'].isnull().values.any():
                df_orig_x["Run"+str(i)] = df_array_orig[weather][i]['x']
                df_orig_y["Run"+str(i)] = df_array_orig[weather][i]['y']

    median_orig_x = df_orig_x.median(axis=1)
    median_orig_y = df_orig_y.median(axis=1)

    plt.plot(median_orig_y, median_orig_x, alpha=0.7)

df_fric_x = pd.DataFrame()
df_fric_y = pd.DataFrame()

for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['x'].dtypes == 'object' or df_array_fric[weather][i]['y'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['x'].isnull().values.any() and not df_array_fric[i]['y'].isnull().values.any():
            df_fric_x["Run"+str(i)] = df_array_fric[weather][i]['x']
            df_fric_y["Run"+str(i)] = df_array_fric[weather][i]['y']

    median_fric_x = df_fric_x.median(axis=1)
    median_fric_y = df_fric_y.median(axis=1)

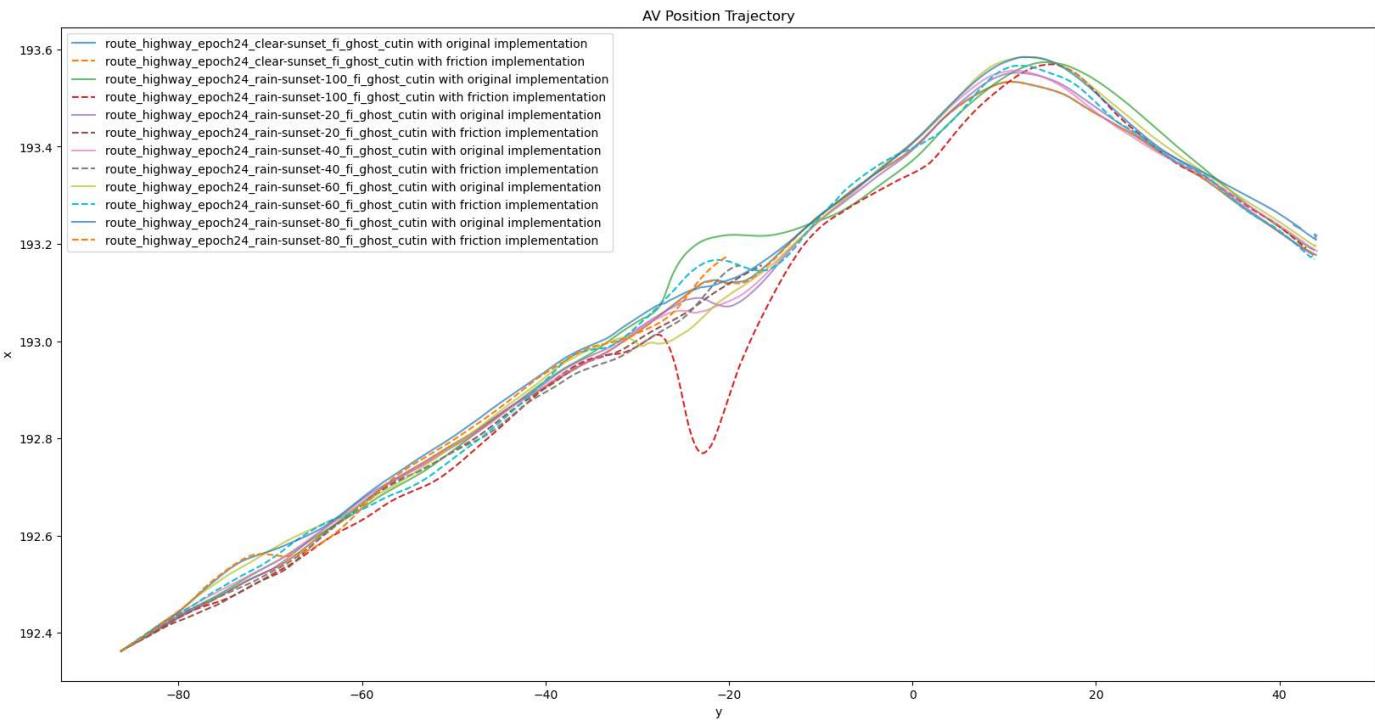
    plt.plot(median_fric_y, median_fric_x, '--')

# plt.axis('equal')
plt.xlabel("y")
plt.ylabel("x")
```

```

plt.title("AV Position Trajectory")
plt.legend(full_legend)
plt.show()

```



V

```

In [61]: for weather in range(len(txt_lists_fric)):
    plt.figure(figsize=(15,8))
    df_orig_v = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['v'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_orig[weather][i]['v'].isnull().values.any():
                df_orig_v["Run"+str(i)] = df_array_orig[weather][i]['v']

    median_orig_v = df_orig_v.median(axis=1)

    plt.plot(median_orig_v, alpha=0.7)

    df_fric_v = pd.DataFrame()

    for i in range(0, len(df_array_fric)):
        if(df_array_fric[weather][i]['v'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_fric[weather][i]['v'].isnull().values.any():
                df_fric_v["Run"+str(i)] = df_array_fric[weather][i]['v']

    median_fric_v = df_fric_v.median(axis=1)

    all_medians[weather][2].append(median_orig_v)
    all_medians[weather][2].append(median_fric_v)

    plt.plot(median_fric_v, '--')

    plt.title(subfolders[weather])
    plt.legend(friction_legend)

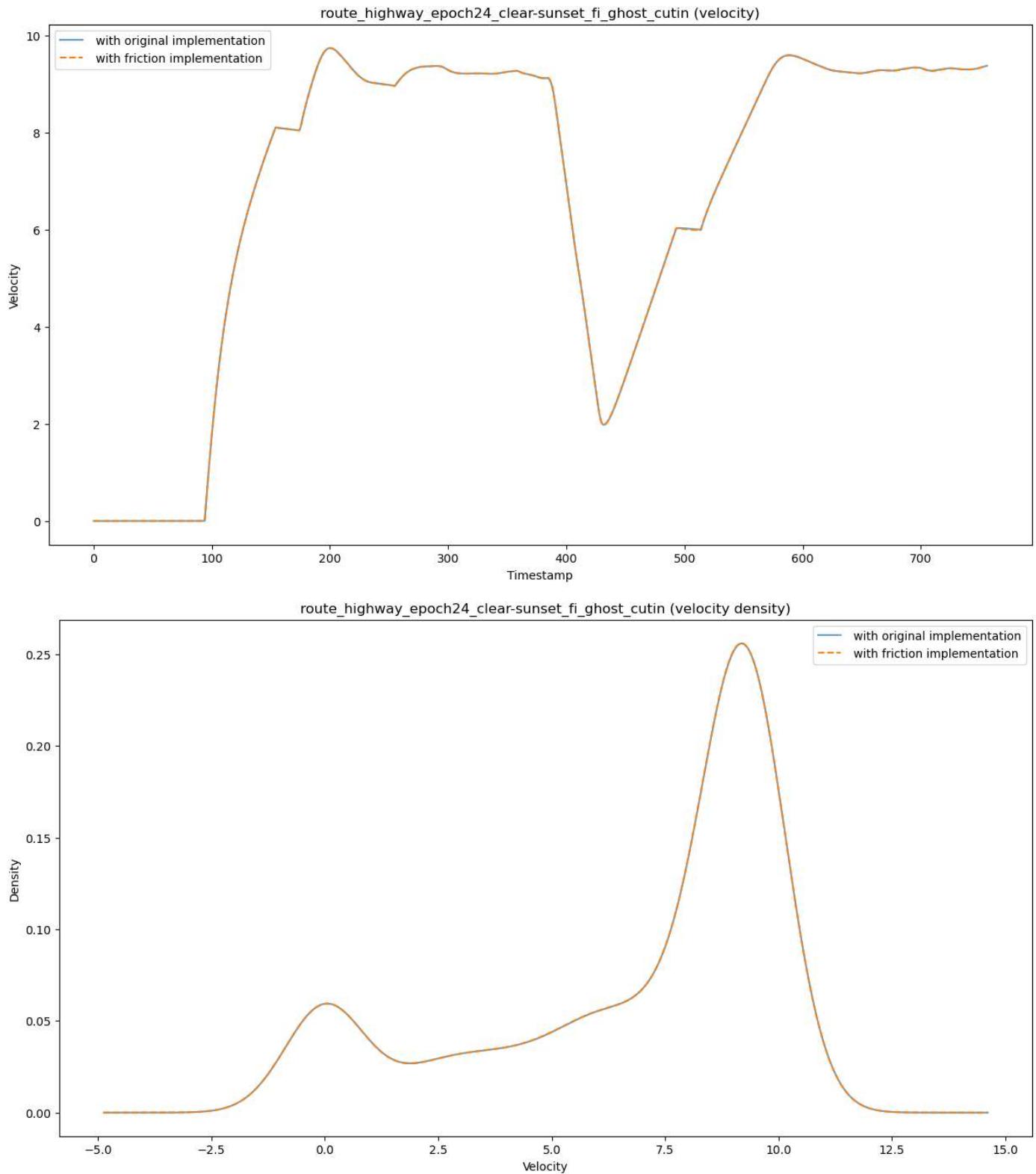
```

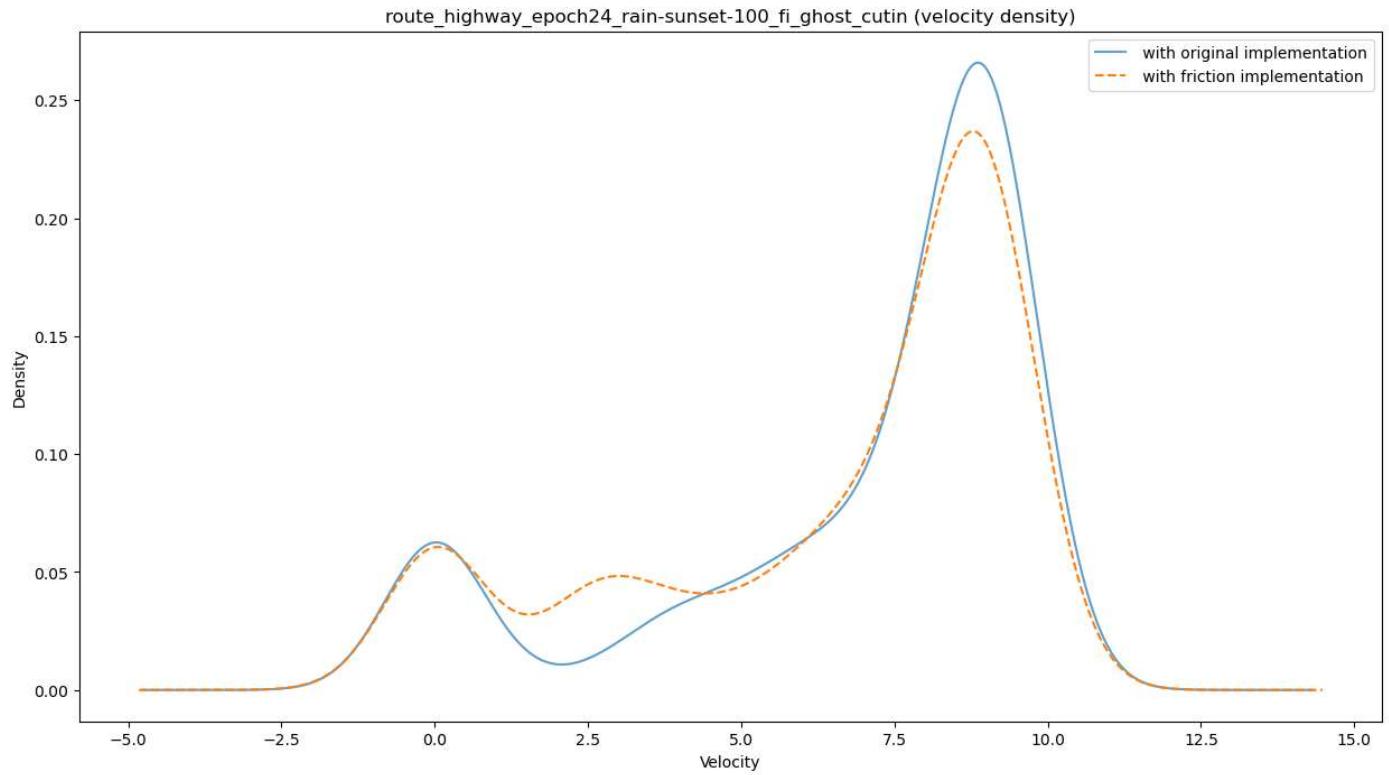
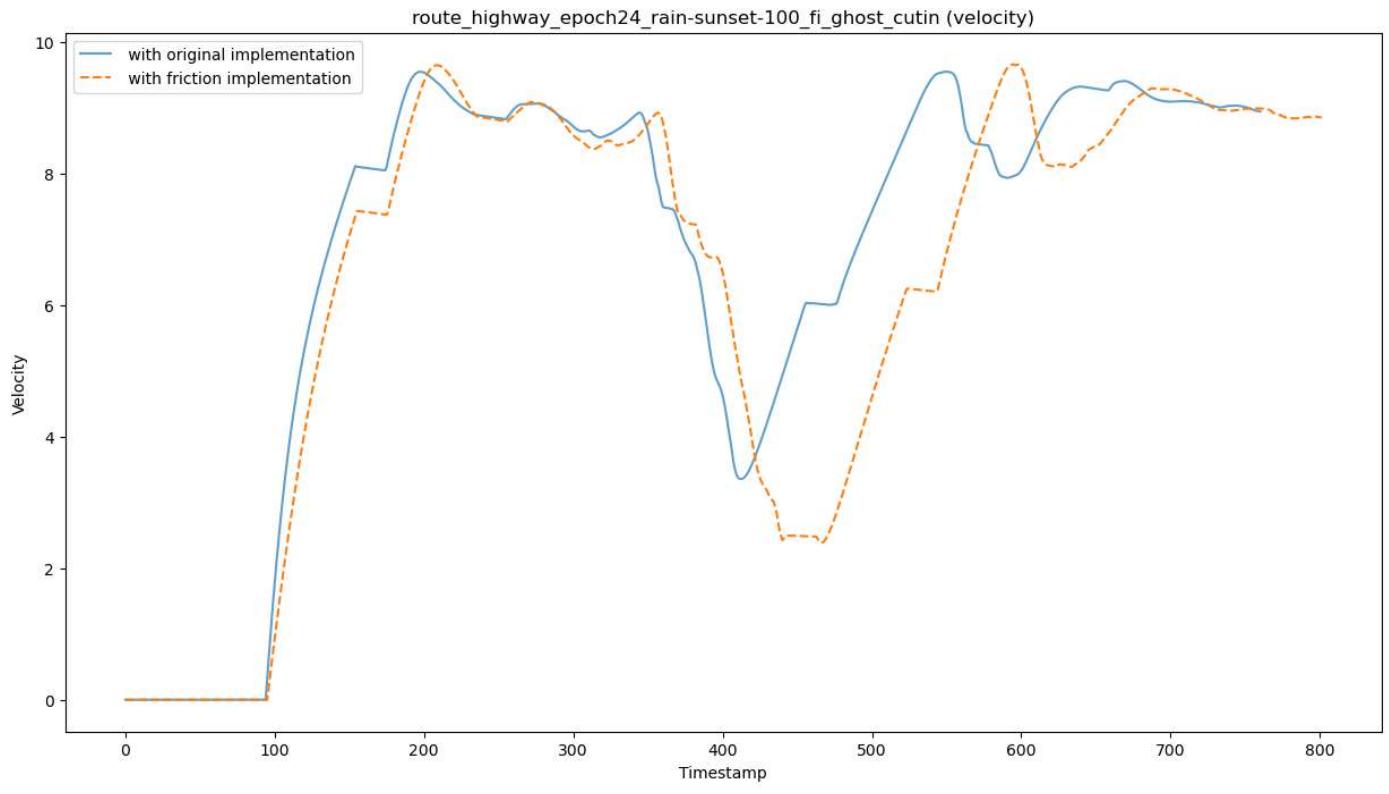
```

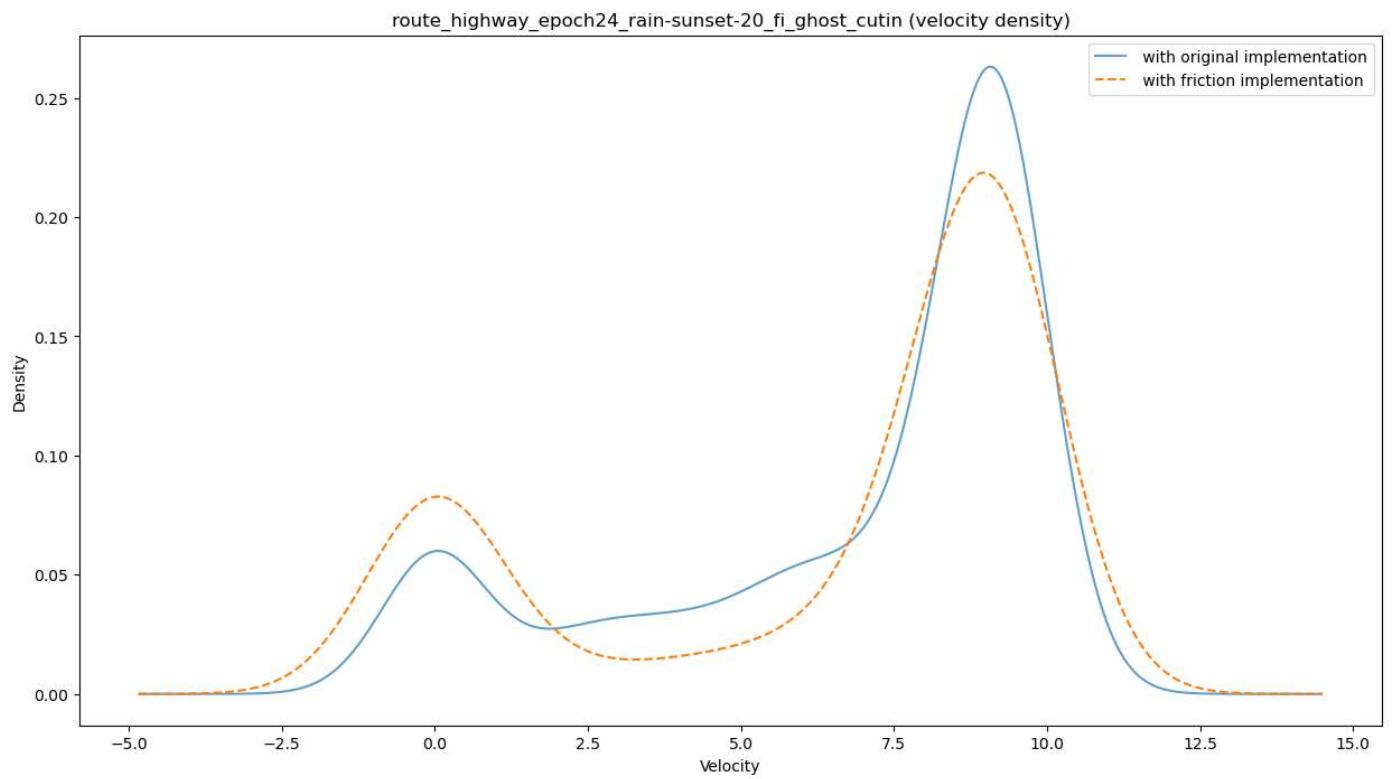
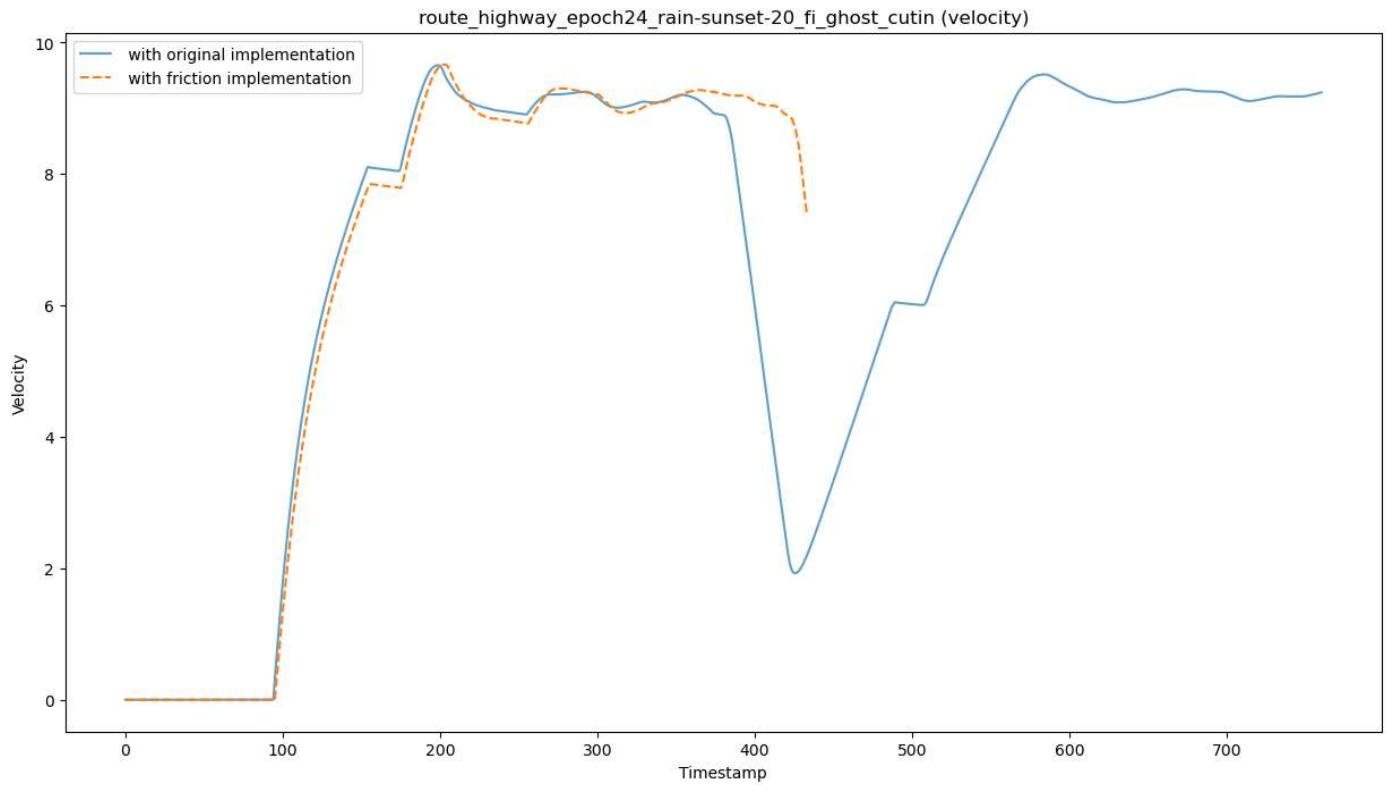
plt.xlabel("Timestamp")
plt.ylabel("Velocity")
plt.title(subfolders[weather] + " (velocity)")
plt.show()

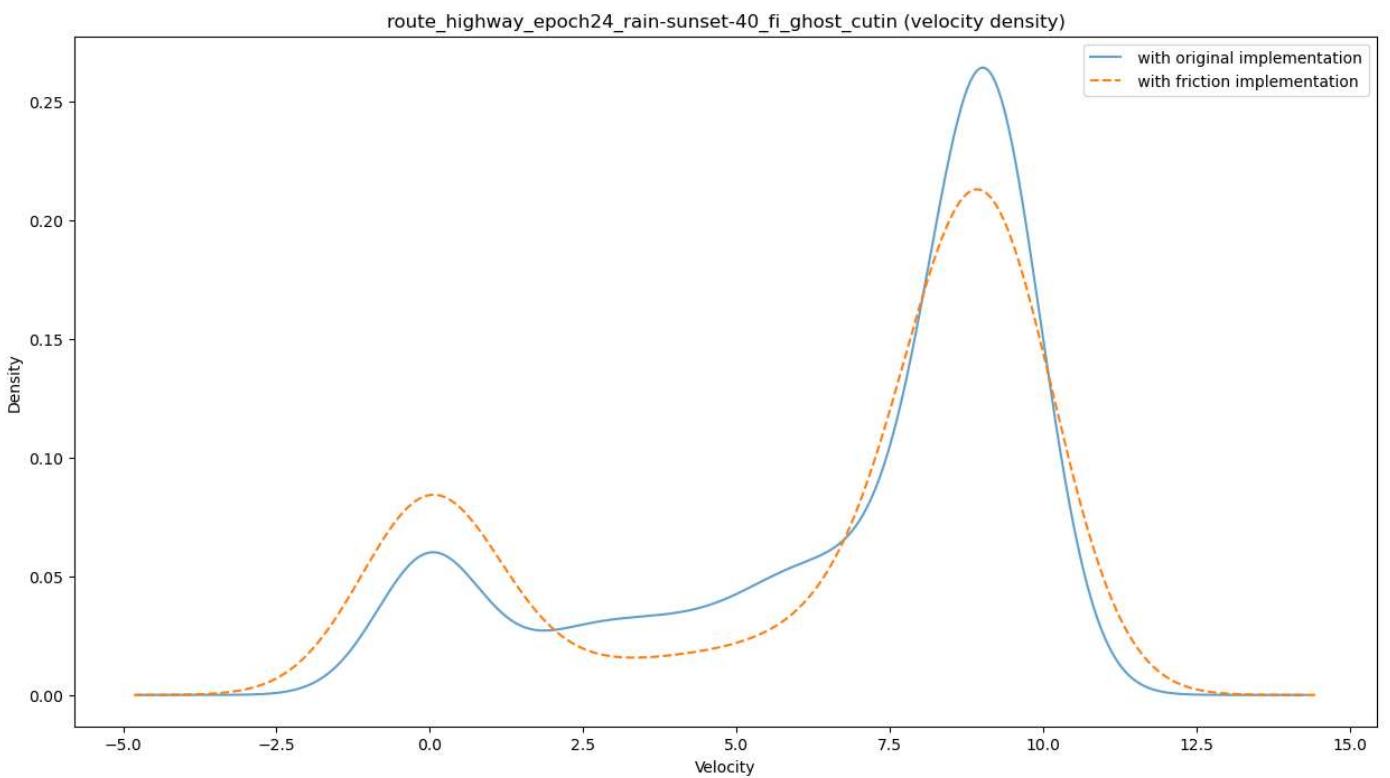
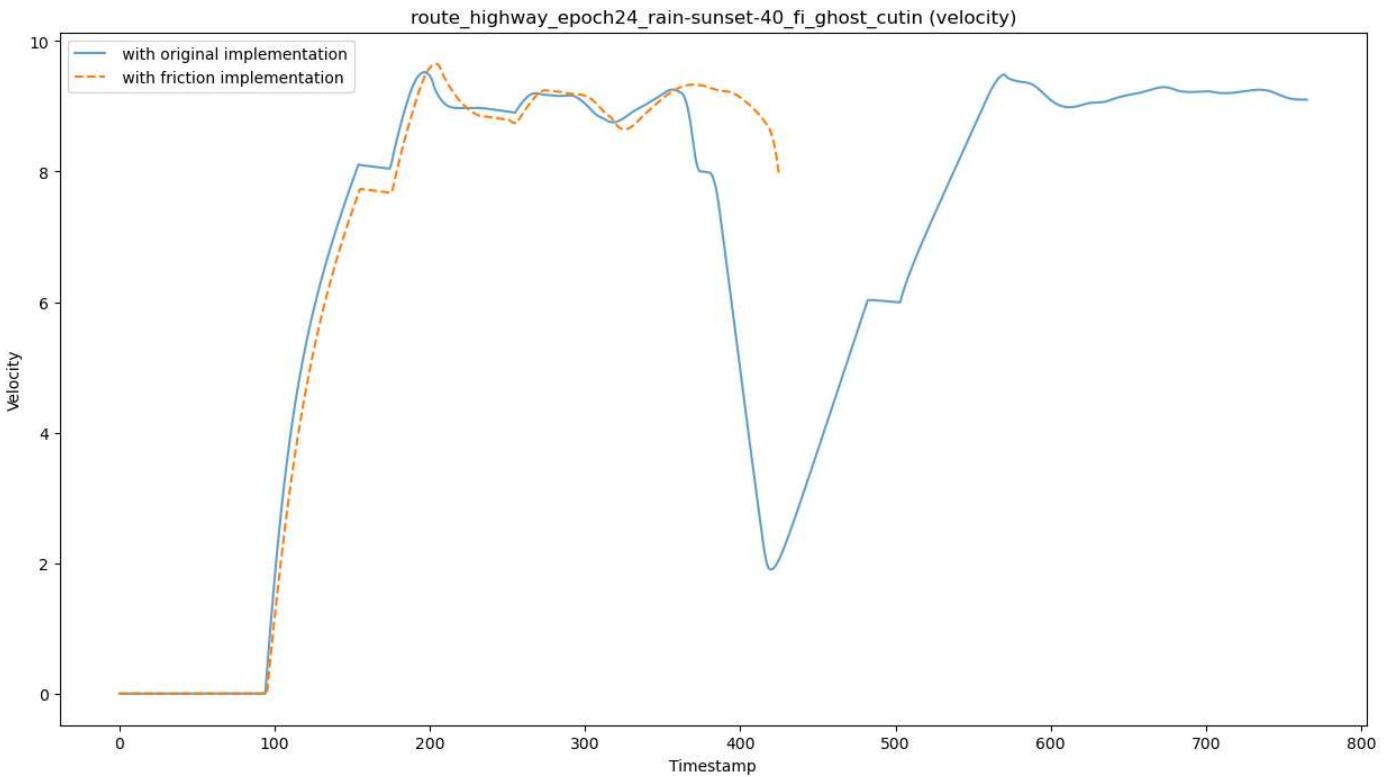
plt.figure(figsize=(15,8))
plt.xlabel("Velocity")
plt.ylabel("Density")
median_orig_v.plot.density(alpha=0.7)
median_fric_v.plot.density(style='--')
plt.title(subfolders[weather] + " (velocity density)")
plt.legend(friction_legend)
plt.show()

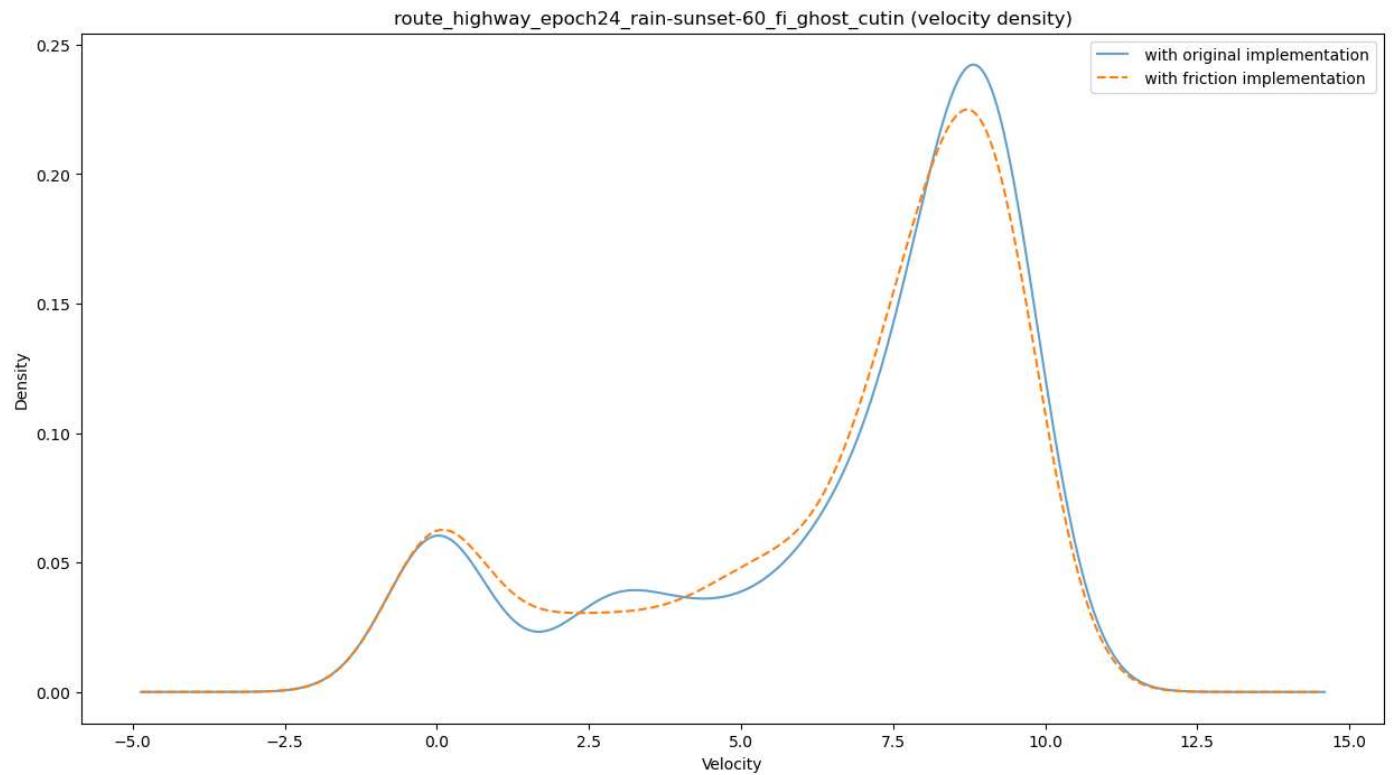
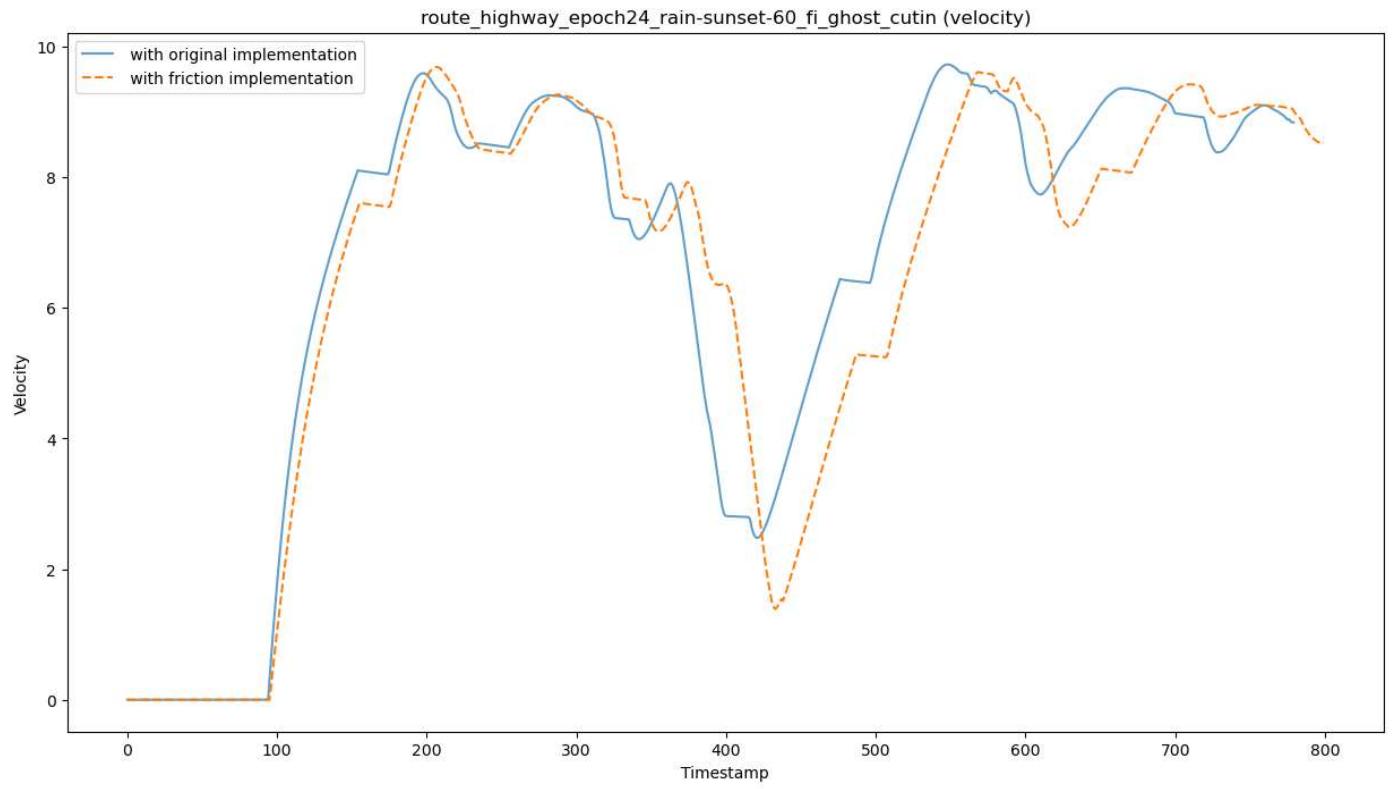
```



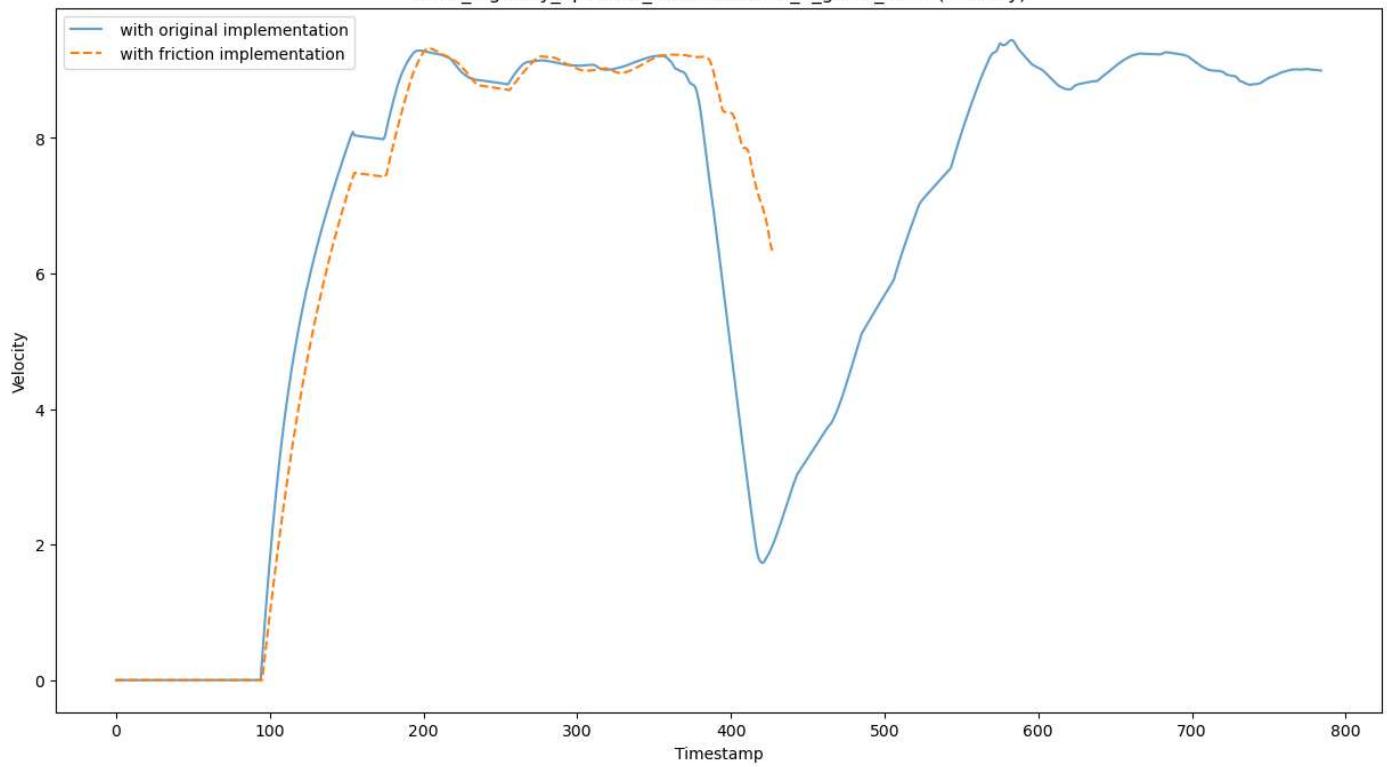




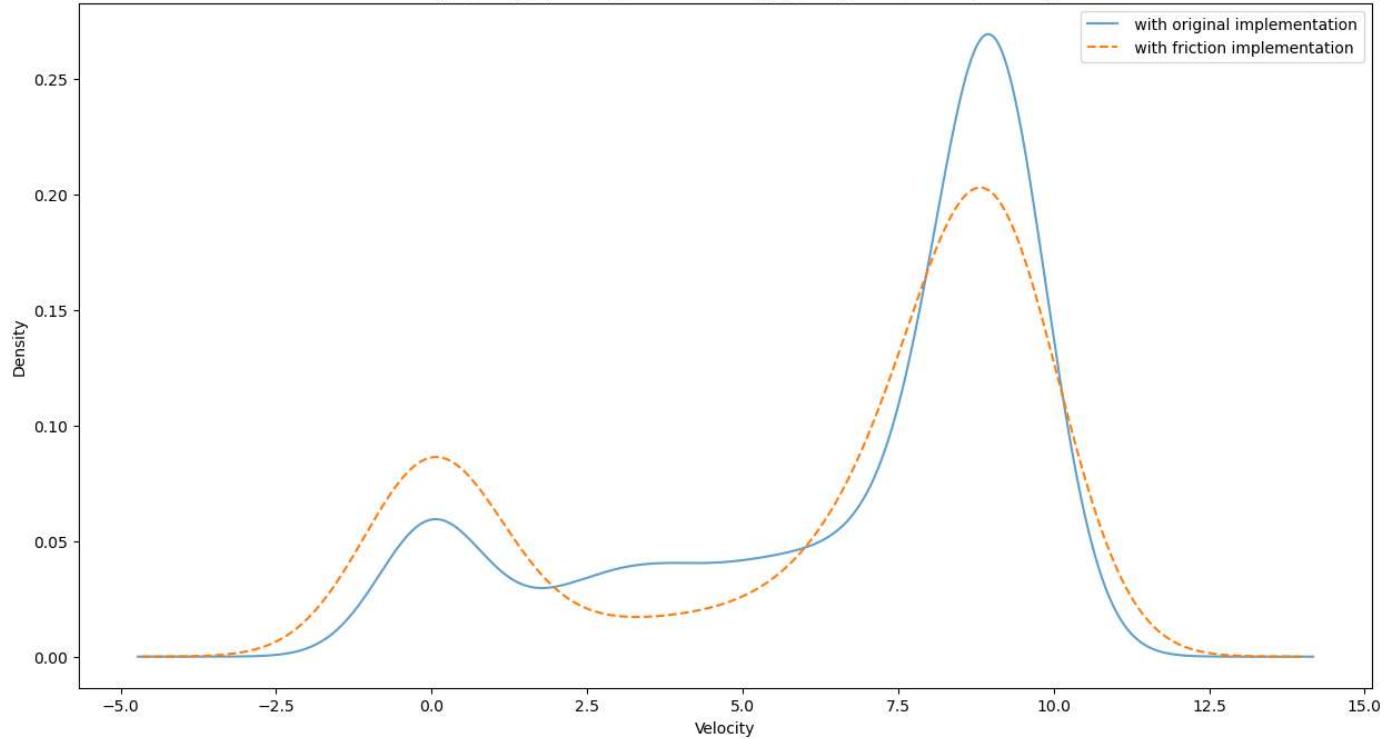




route_highway_epoch24_rain-sunset-80_fi_ghost_cutin (velocity)



route_highway_epoch24_rain-sunset-80_fi_ghost_cutin (velocity density)



```
In [36]: plt.figure(figsize=(20,10))
```

```
for weather in range(len(txt_lists_fric)):

    df_orig_v = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['v'].dtypes == 'v'):
            print(i)
        else:
            if not df_array_orig[weather][i]['v'].isnull().values.any():
                df_orig_v["Run"+str(i)] = df_array_orig[weather][i]['v']

    median_orig_v = df_orig_v.median(axis=1)
```

```

plt.plot(median_orig_v[200:], alpha=0.7)

df_fric_v = pd.DataFrame()

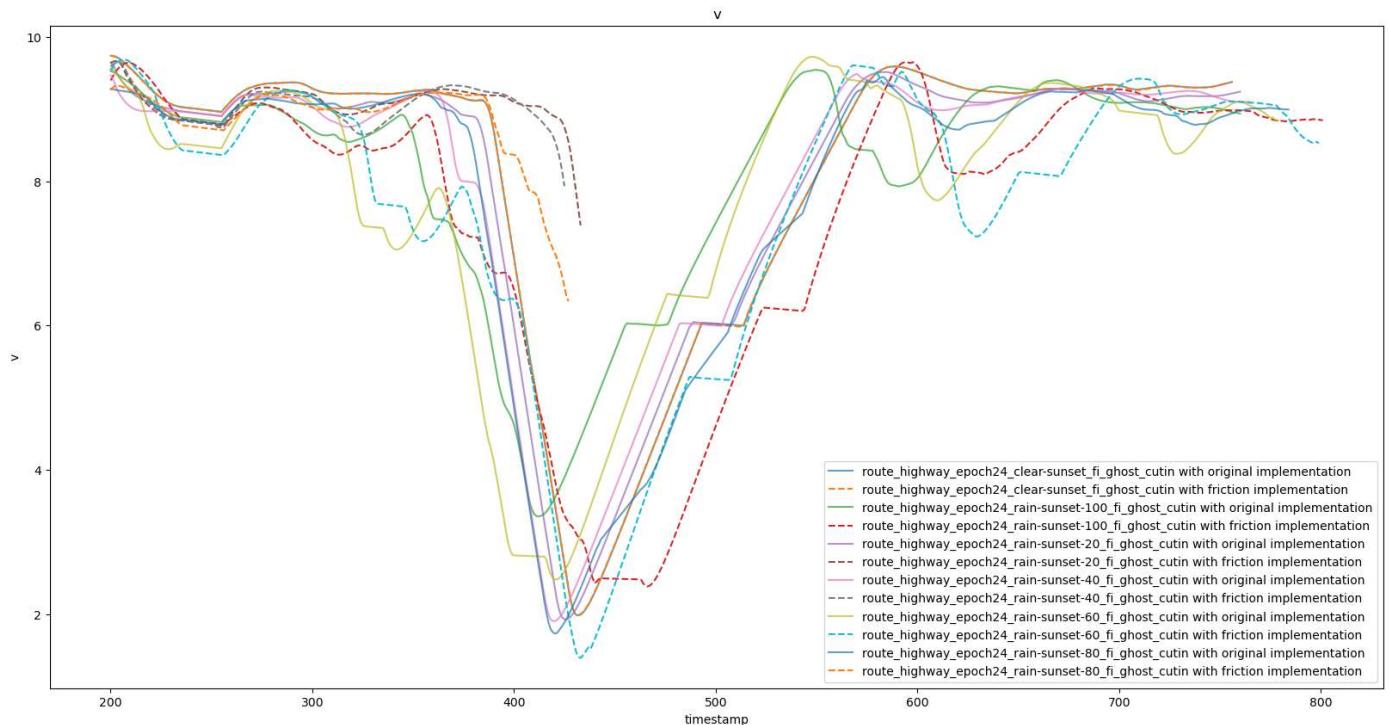
for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['v'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['v'].isnull().values.any():
            df_fric_v["Run"+str(i)] = df_array_fric[weather][i]['v']

median_fric_v = df_fric_v.median(axis=1)

plt.plot(median_fric_v[200:], '--')

# plt.axis('equal')
plt.xlabel("timestamp")
plt.ylabel("v")
plt.title("v")
plt.legend(full_legend)
plt.show()

```



cvip

```

In [67]: for weather in range(len(txt_lists_fric)):
    plt.figure(figsize=(15,8))
    df_orig_cvip = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['cvip'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_orig[weather][i]['cvip'].isnull().values.any():
                df_orig_cvip["Run"+str(i)] = df_array_orig[weather][i]['cvip']

    median_orig_cvip = df_orig_cvip.median(axis=1)

    plt.plot(median_orig_cvip.iloc[200:600], alpha=0.7)

```

```

df_fric_cvip = pd.DataFrame()

for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['cvip'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['cvip'].isnull().values.any():
            df_fric_cvip["Run"+str(i)] = df_array_fric[weather][i]['cvip']

median_fric_cvip = df_fric_cvip.median(axis=1)

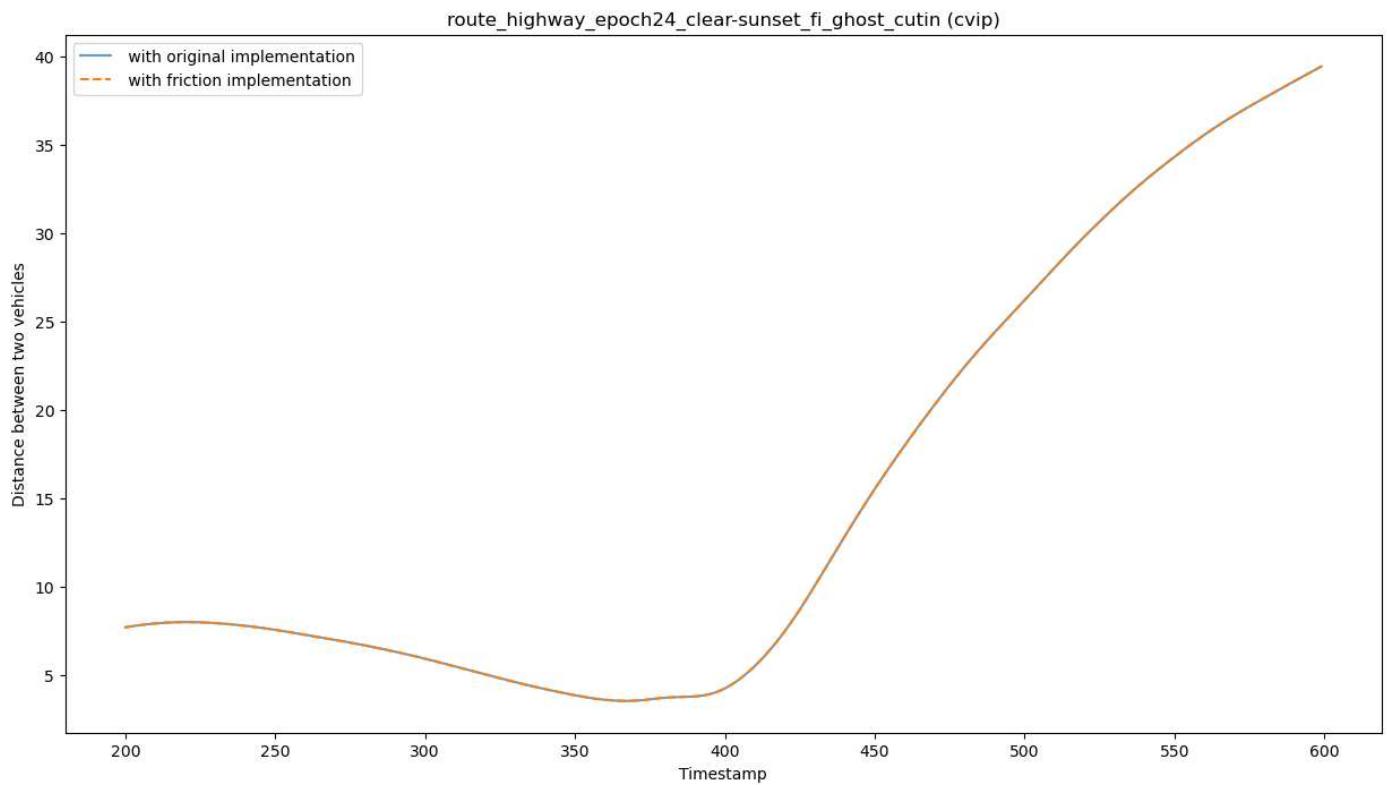
all_medians[weather][2].append(median_orig_cvip)
all_medians[weather][2].append(median_fric_cvip)

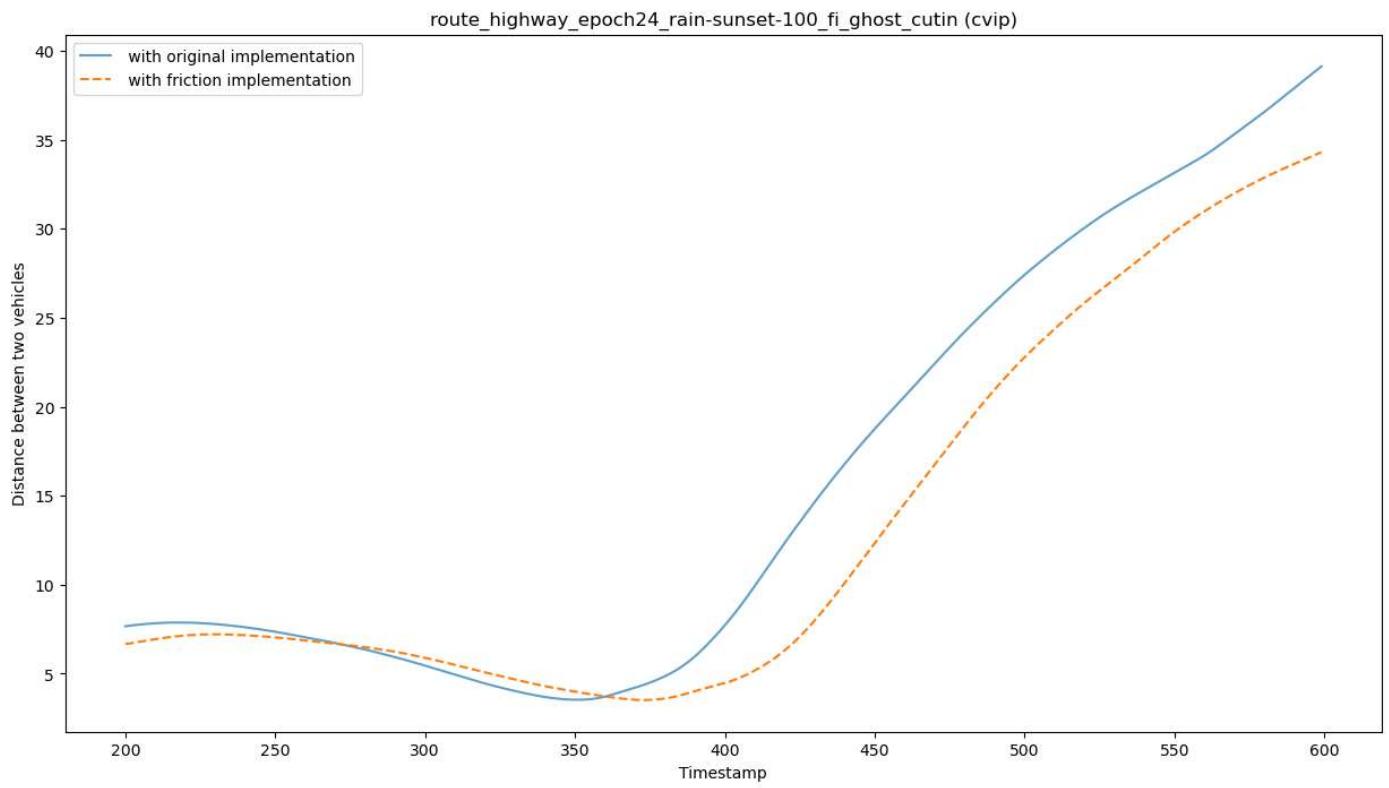
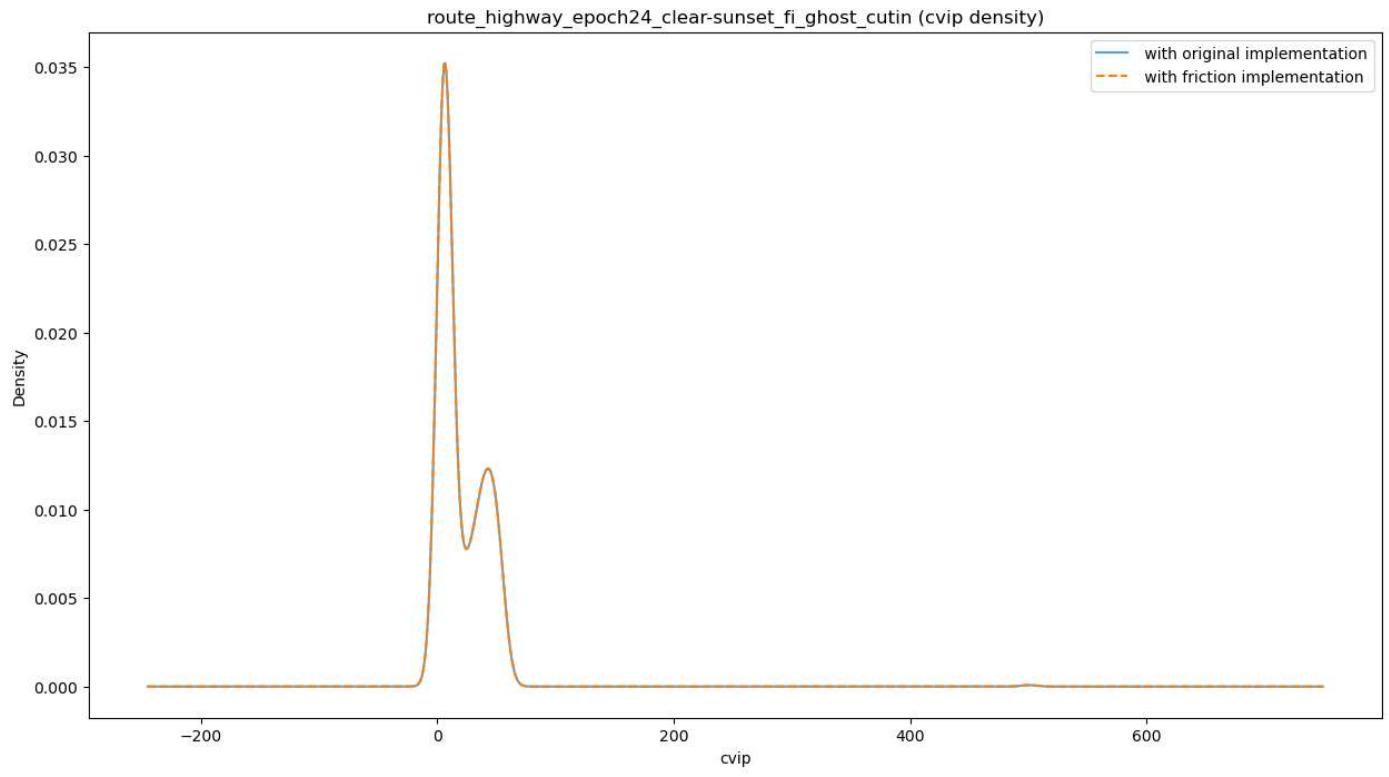
plt.plot(median_fric_cvip.iloc[200:600], '--')

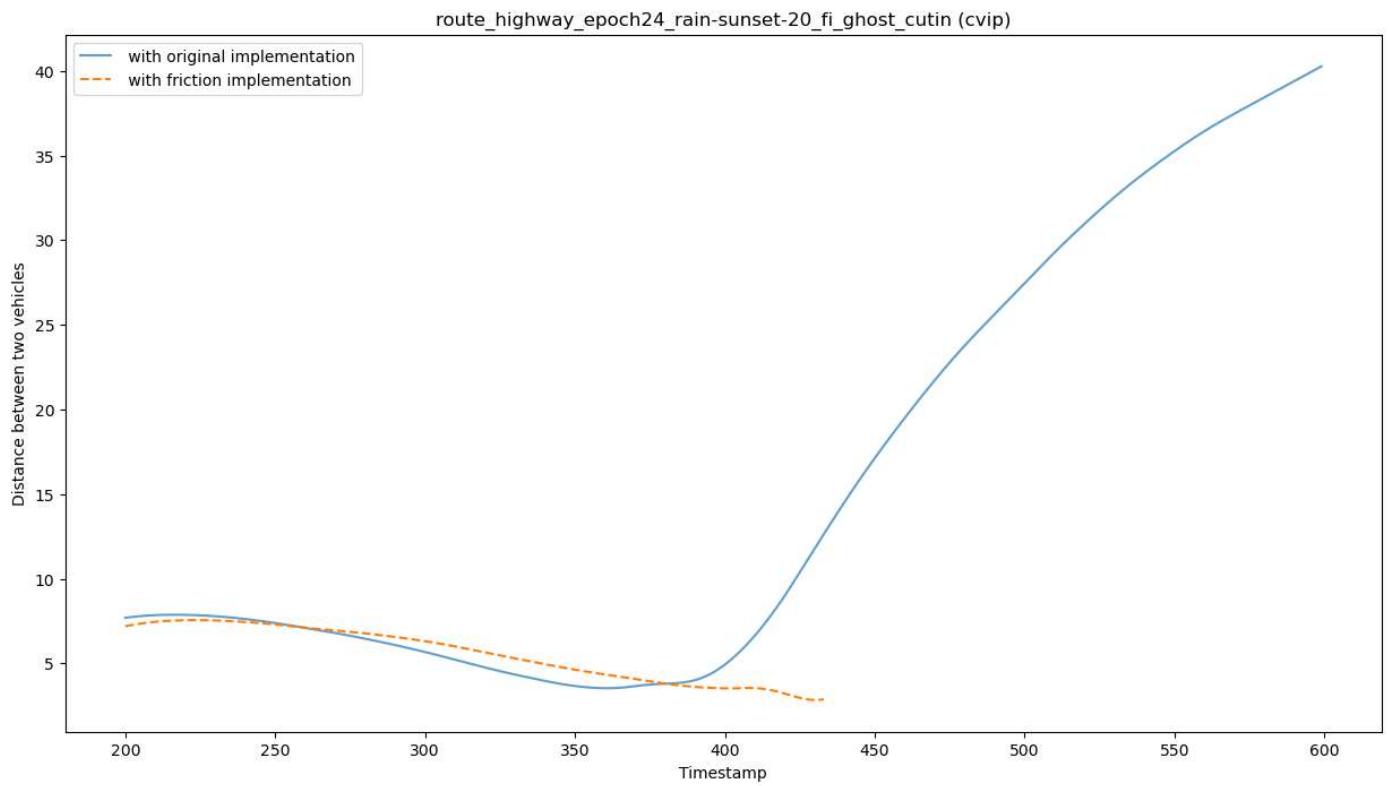
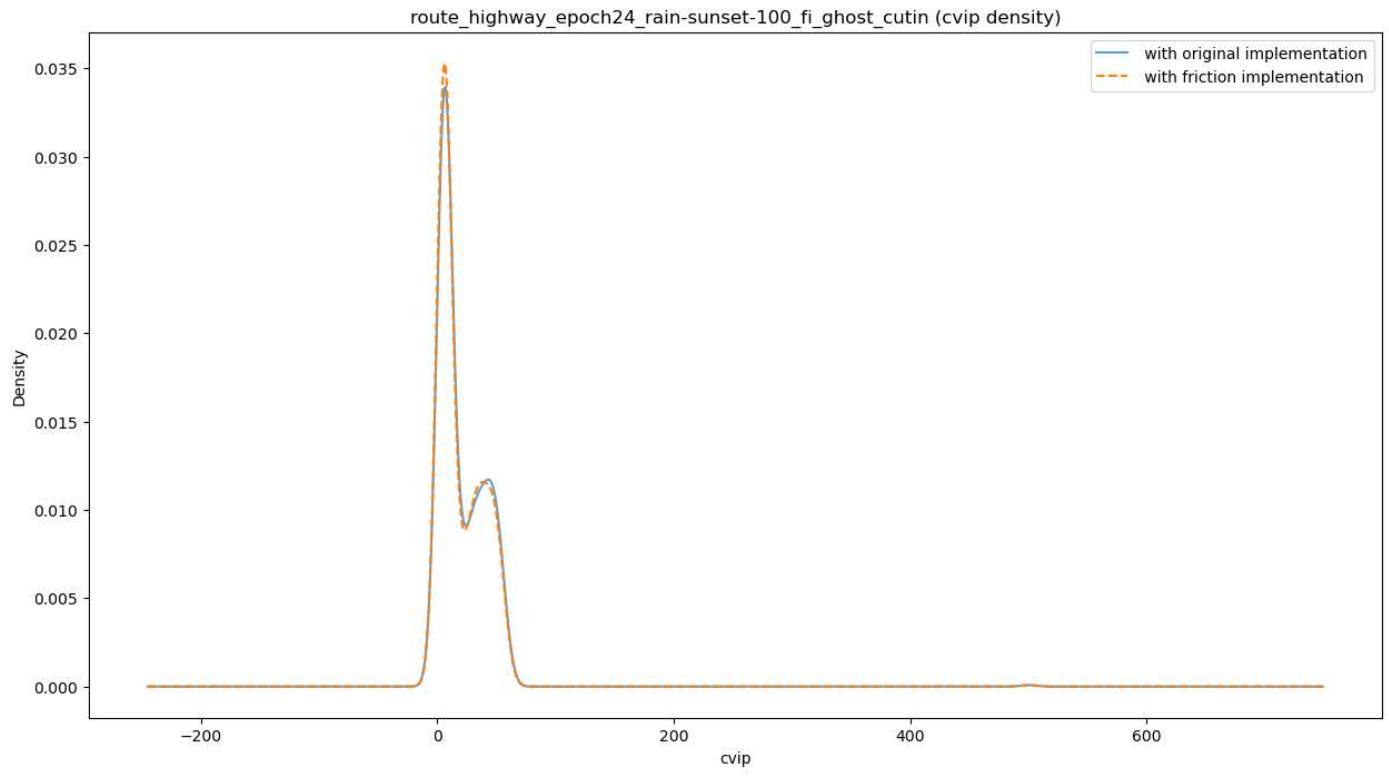
plt.title(subfolders[weather])
plt.legend(friction_legend)
plt.xlabel("Timestamp")
plt.ylabel("Distance between two vehicles")
plt.title(subfolders[weather] + " (cvip)")
plt.show()

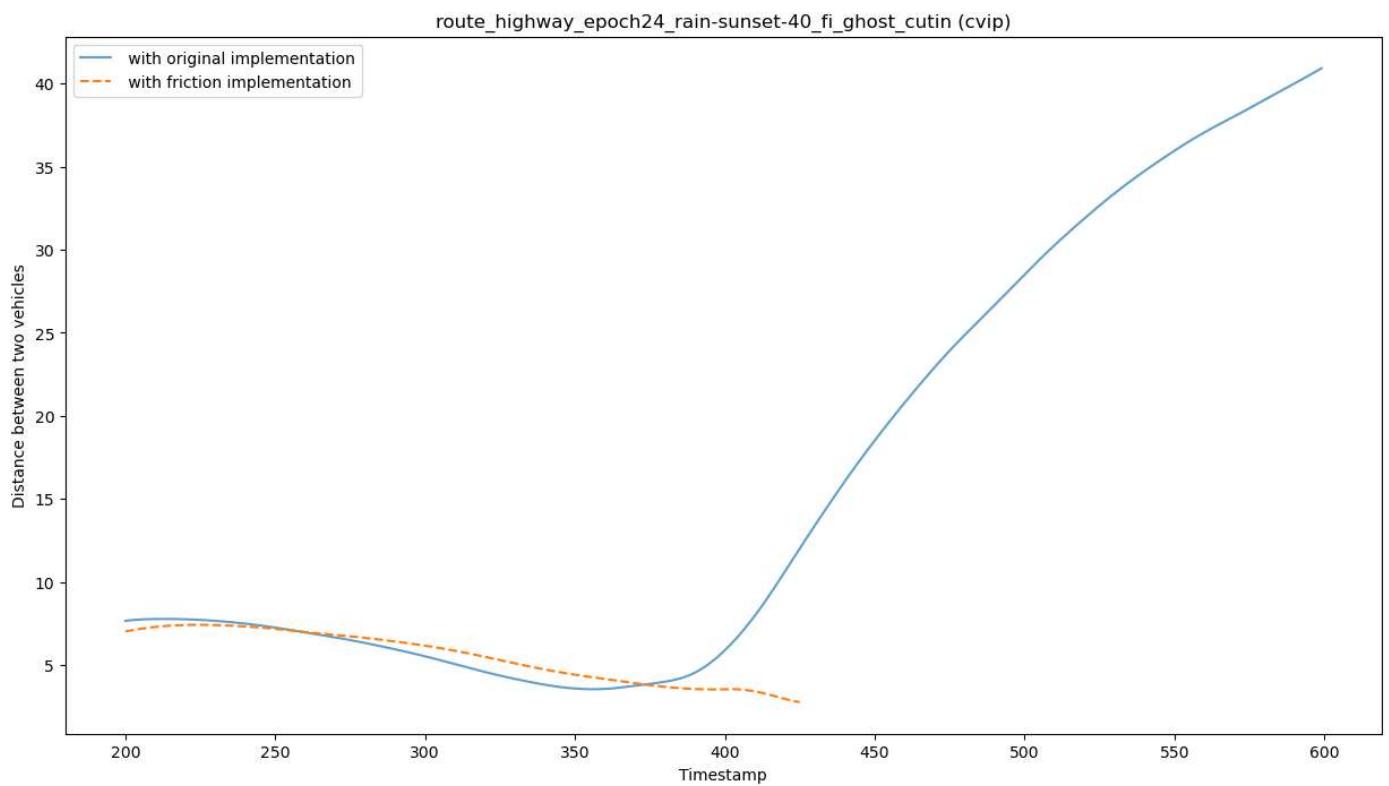
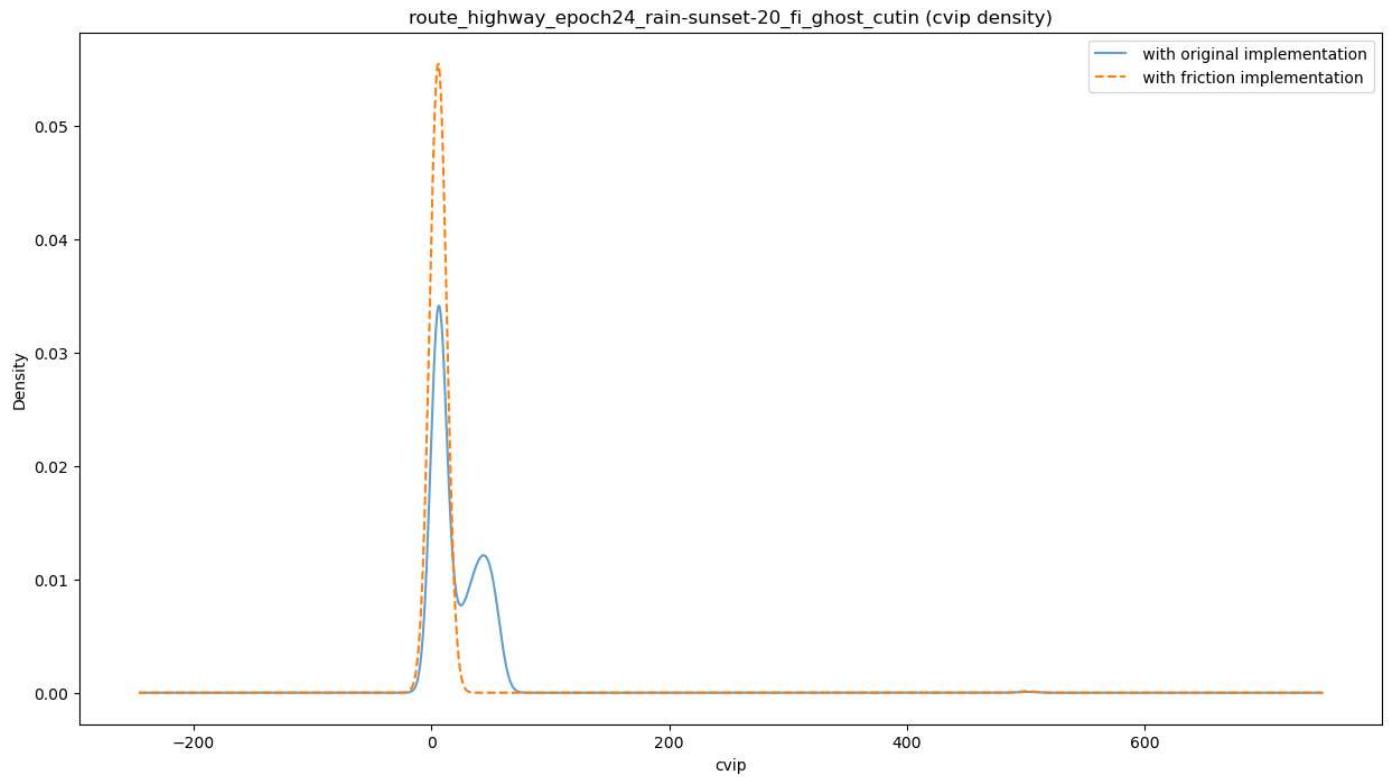
plt.figure(figsize=(15,8))
plt.xlabel("cvip")
plt.ylabel("Density")
median_orig_cvip.plot.density(alpha=0.7)
median_fric_cvip.plot.density(style='--')
plt.title(subfolders[weather] + " (cvip density)")
plt.legend(friction_legend)
plt.show()

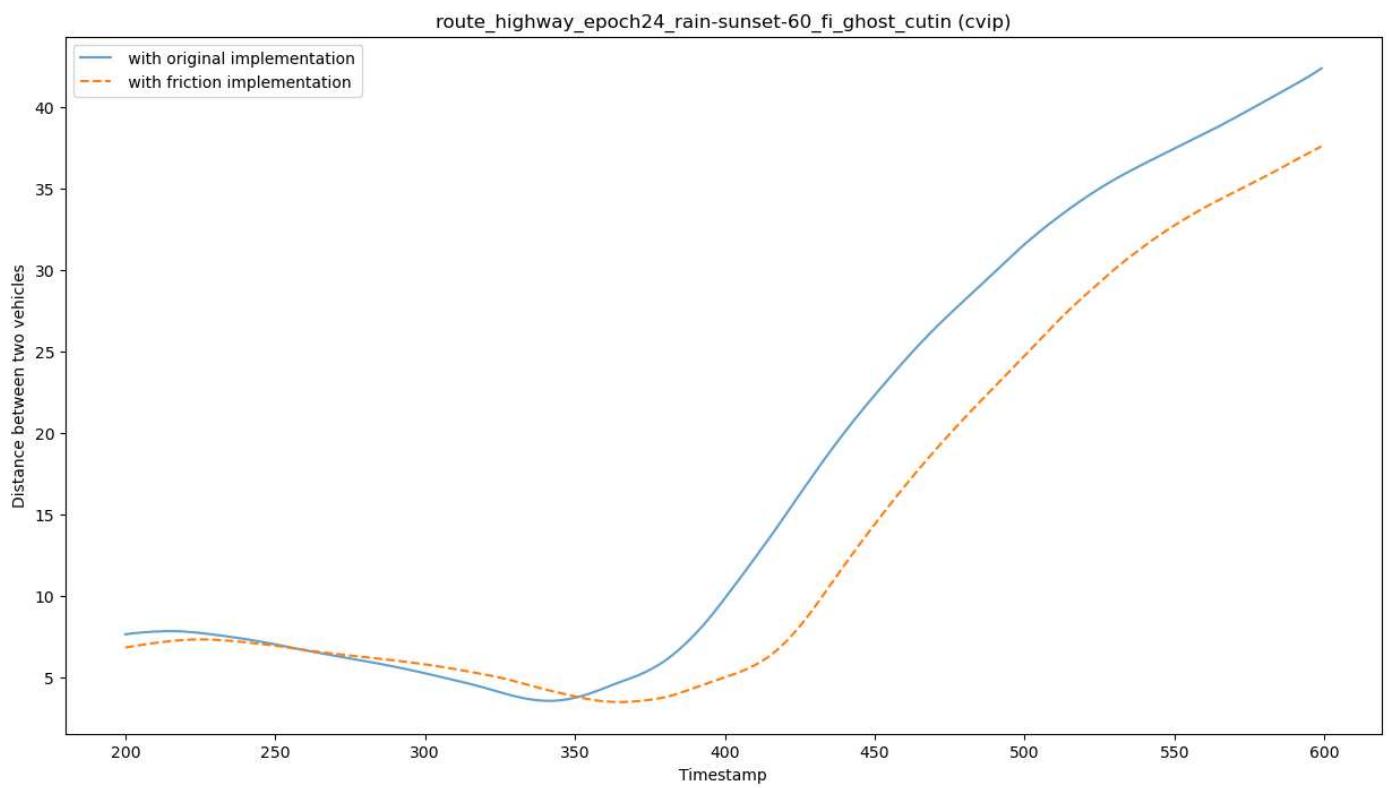
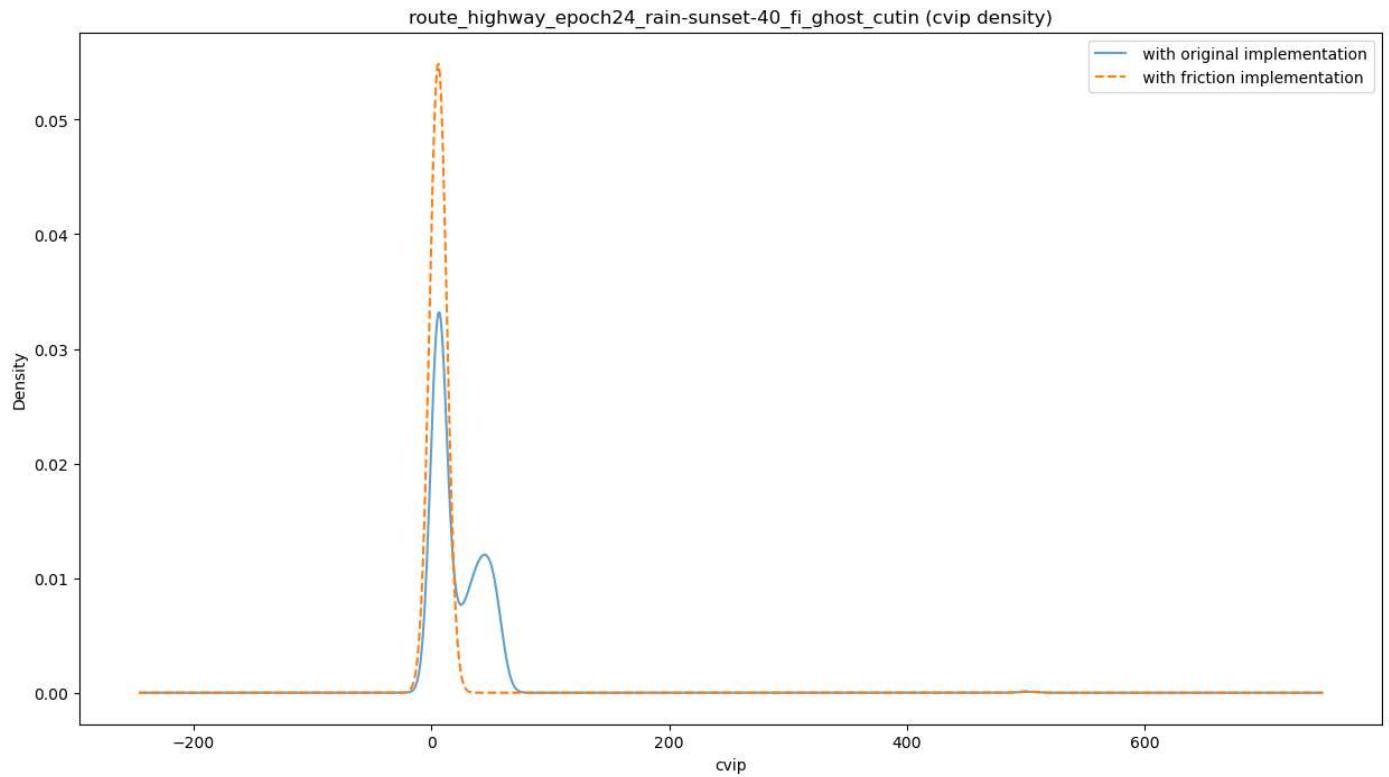
```

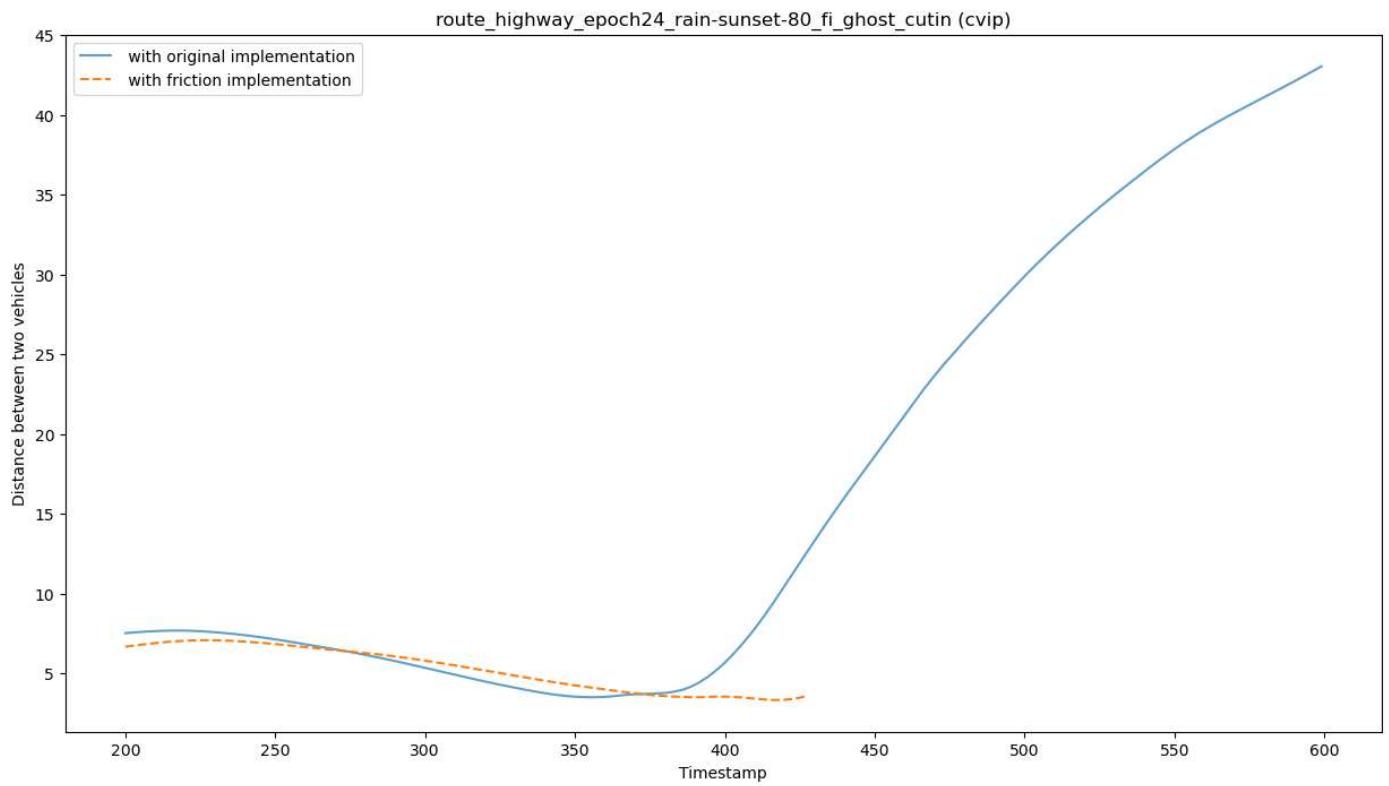
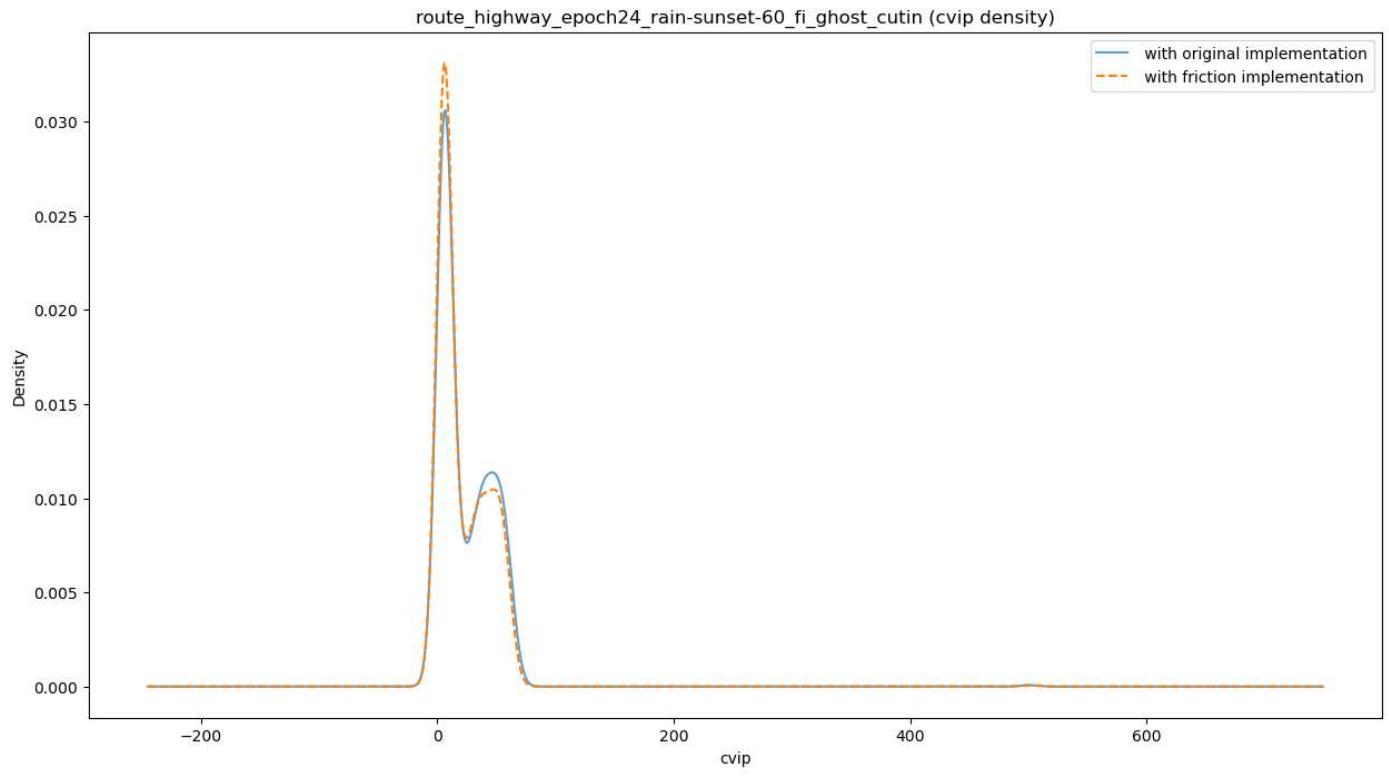


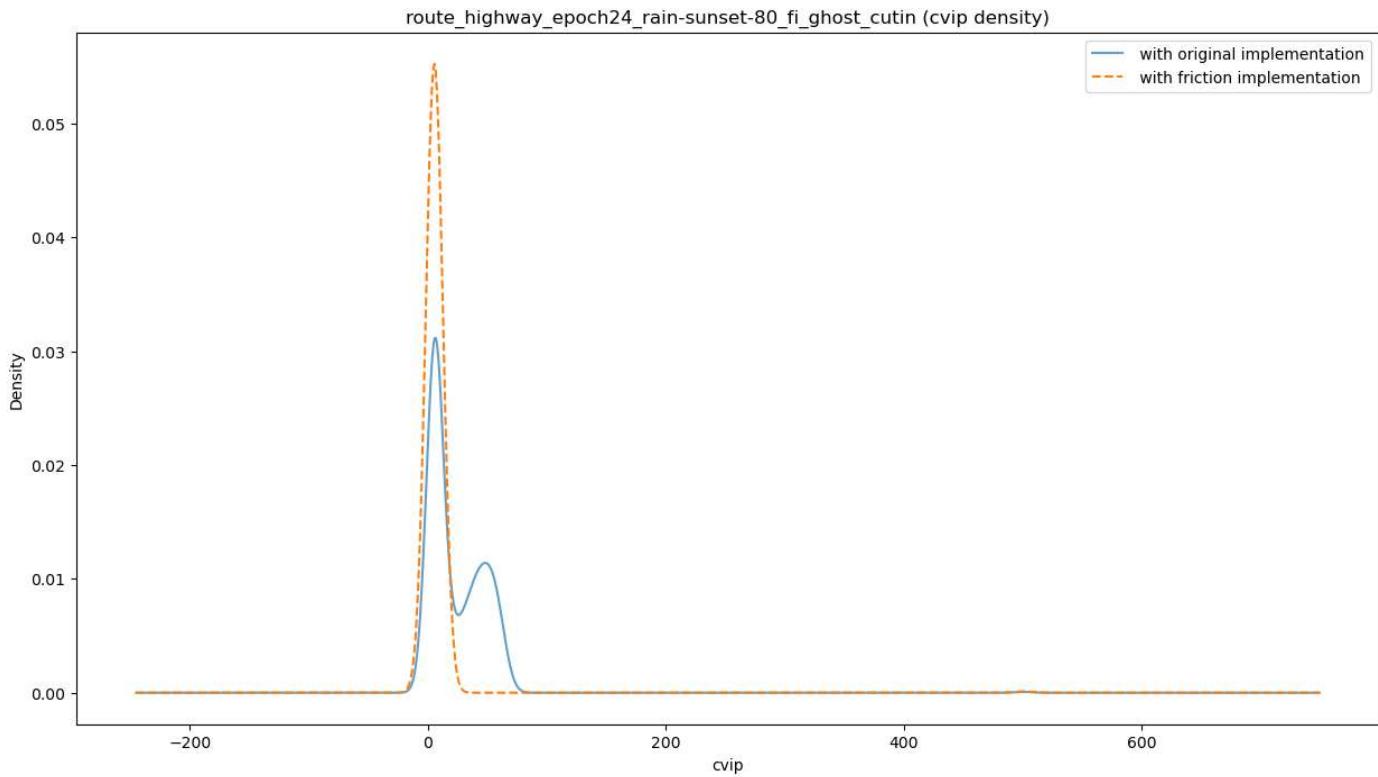












```
In [38]: plt.figure(figsize=(20,10))

for weather in range(len(txt_lists_fric)):

    df_orig_cvip = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['cvip'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_orig[weather][i]['cvip'].isnull().values.any():
                df_orig_cvip["Run"+str(i)] = df_array_orig[weather][i]['cvip']

    median_orig_cvip = df_orig_cvip.median(axis=1)

    plt.plot(median_orig_cvip[200:], alpha=0.7)

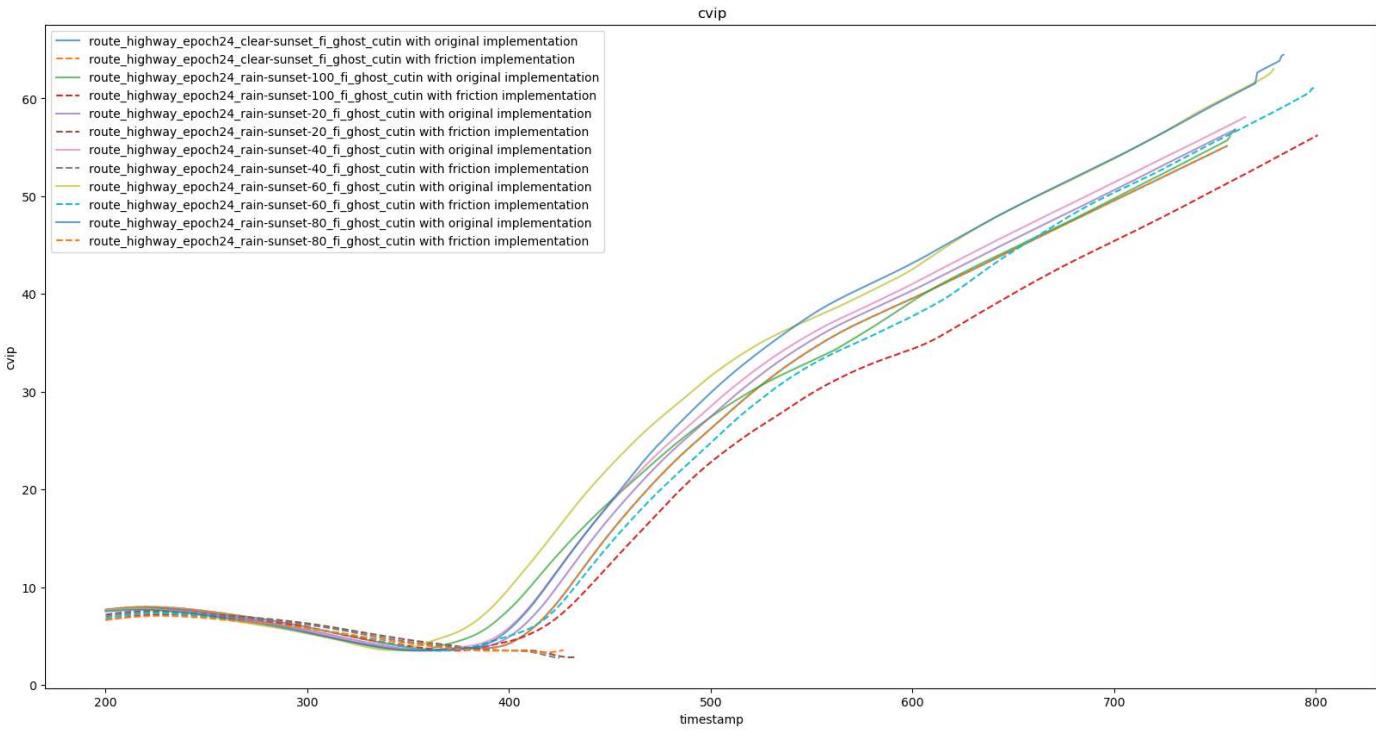
    df_fric_cvip = pd.DataFrame()

    for i in range(0, len(df_array_fric)):
        if(df_array_fric[weather][i]['cvip'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_fric[weather][i]['cvip'].isnull().values.any() :
                df_fric_cvip["Run"+str(i)] = df_array_fric[weather][i]['cvip']

    median_fric_cvip = df_fric_cvip.median(axis=1)

    plt.plot(median_fric_cvip[200:], '--')

    # plt.axis('equal')
    plt.xlabel("timestamp")
    plt.ylabel("cvip")
    plt.title("cvip")
    plt.legend(full_legend)
    plt.show()
```



steer

```
In [63]: for weather in range(len(txt_lists_fric)):
    plt.figure(figsize=(15,8))
    df_orig_steer = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['steer'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_orig[weather][i]['steer'].isnull().values.any():
                df_orig_steer["Run"+str(i)] = df_array_orig[weather][i]['steer']

    median_orig_steer = df_orig_steer.median(axis=1)

    plt.plot(median_orig_steer, alpha=0.7)

df_fric_steer = pd.DataFrame()

for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['steer'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['steer'].isnull().values.any():
            df_fric_steer["Run"+str(i)] = df_array_fric[weather][i]['steer']

median_fric_steer = df_fric_steer.median(axis=1)

all_medians[weather][2].append(median_orig_steer)
all_medians[weather][2].append(median_fric_steer)

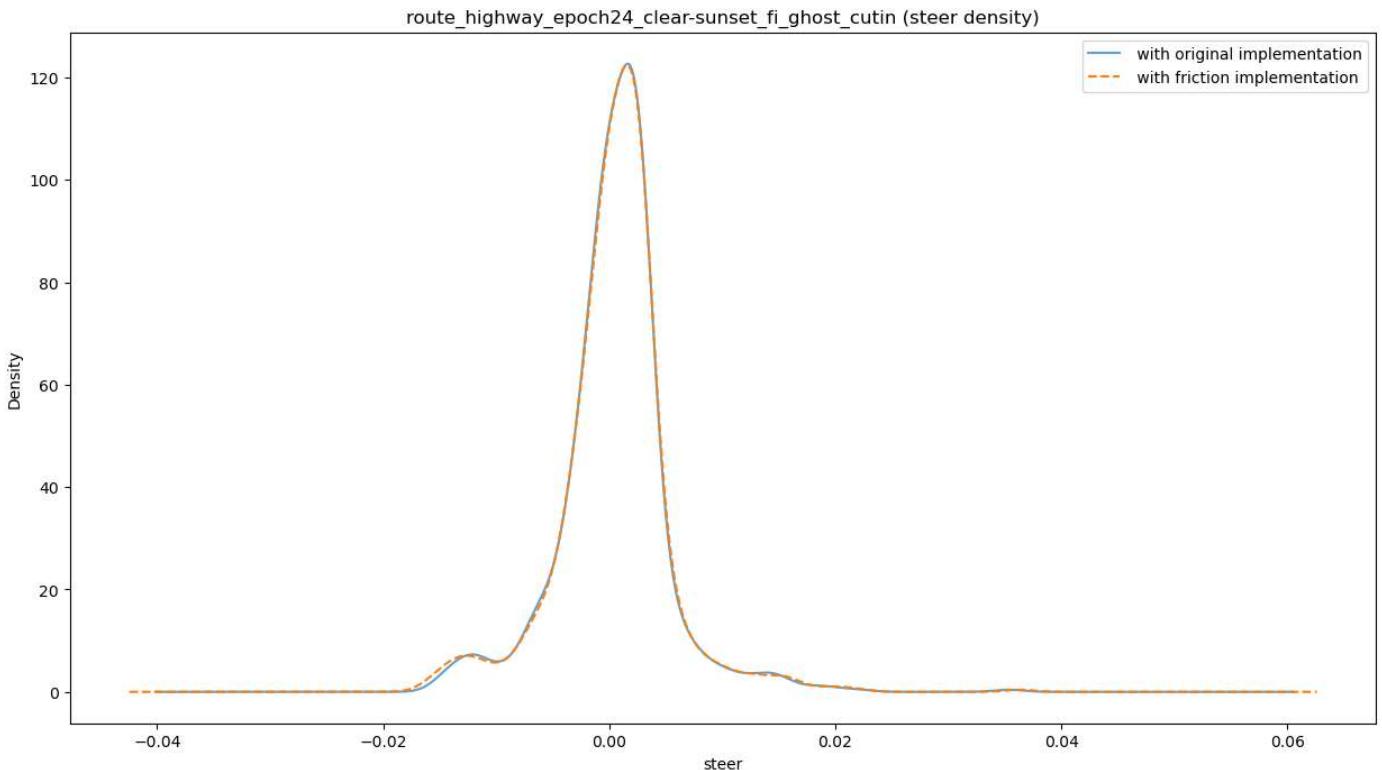
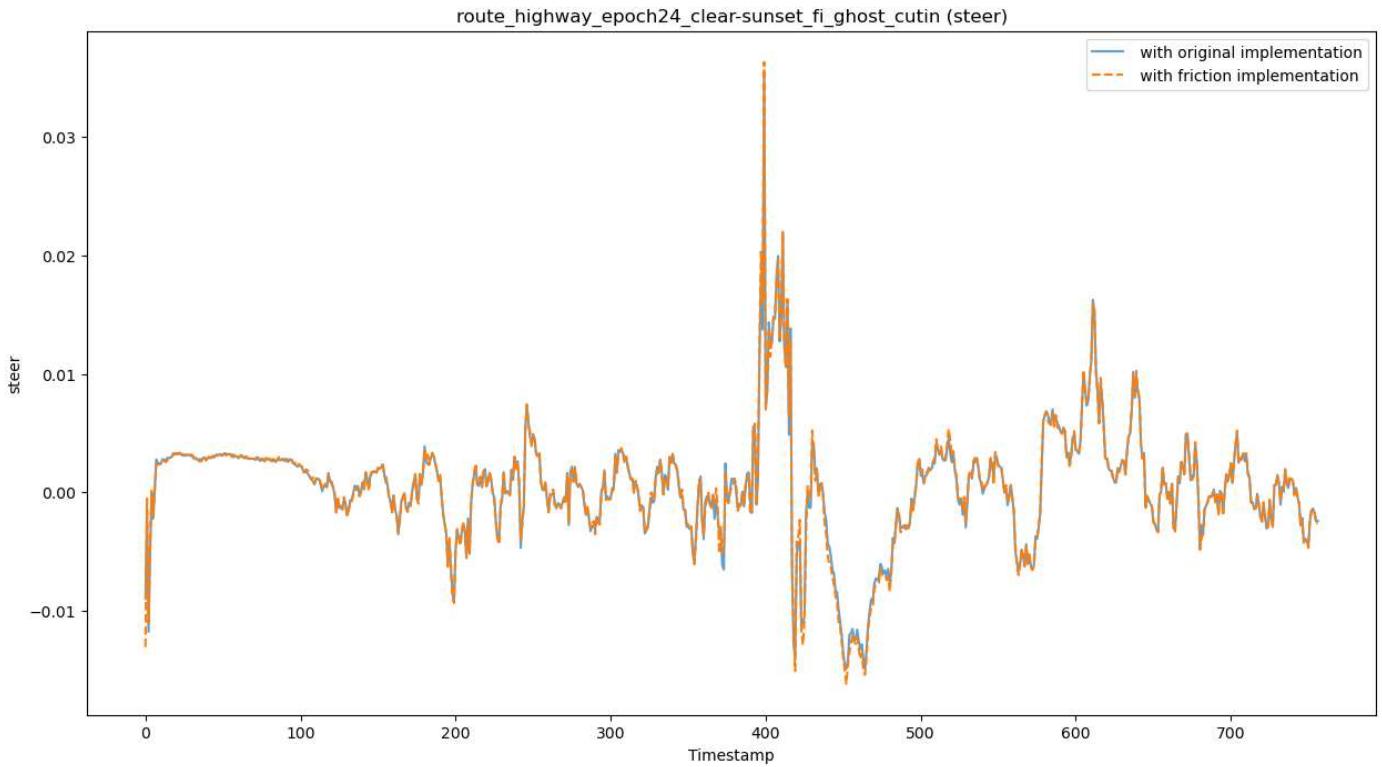
plt.plot(median_fric_steer, '--')

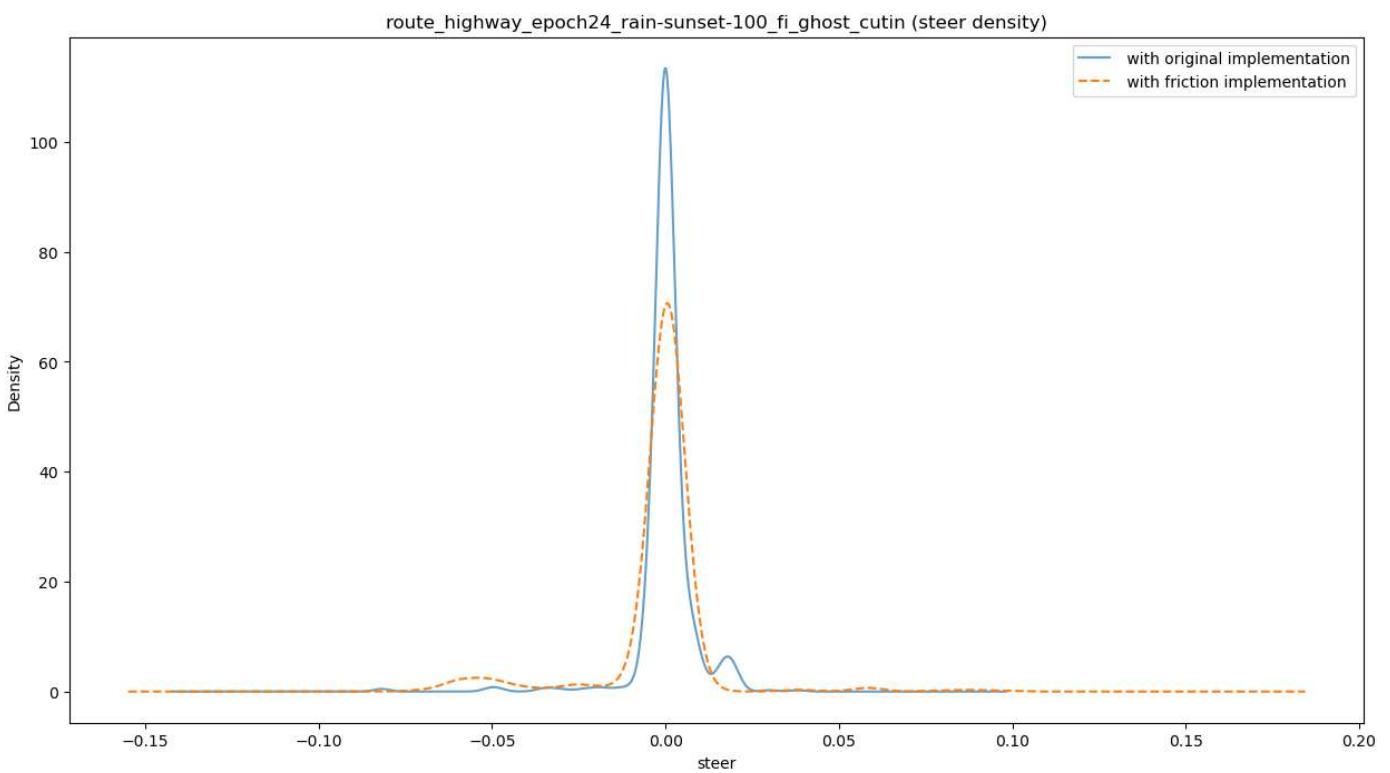
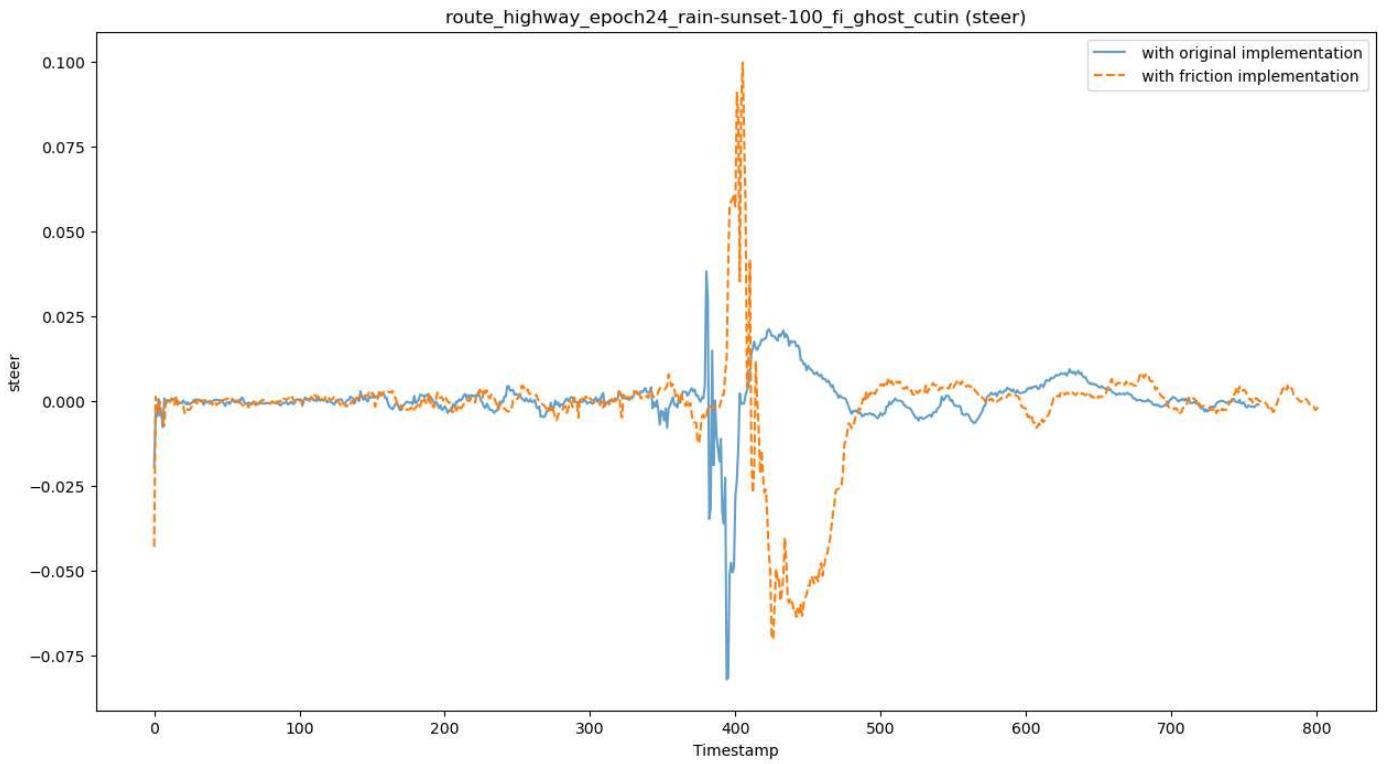
plt.title(subfolders[weather])
plt.legend(friction_legend)
plt.xlabel("Timestamp")
plt.ylabel("steer")
plt.title(subfolders[weather] + " (steer)")
plt.show()
```

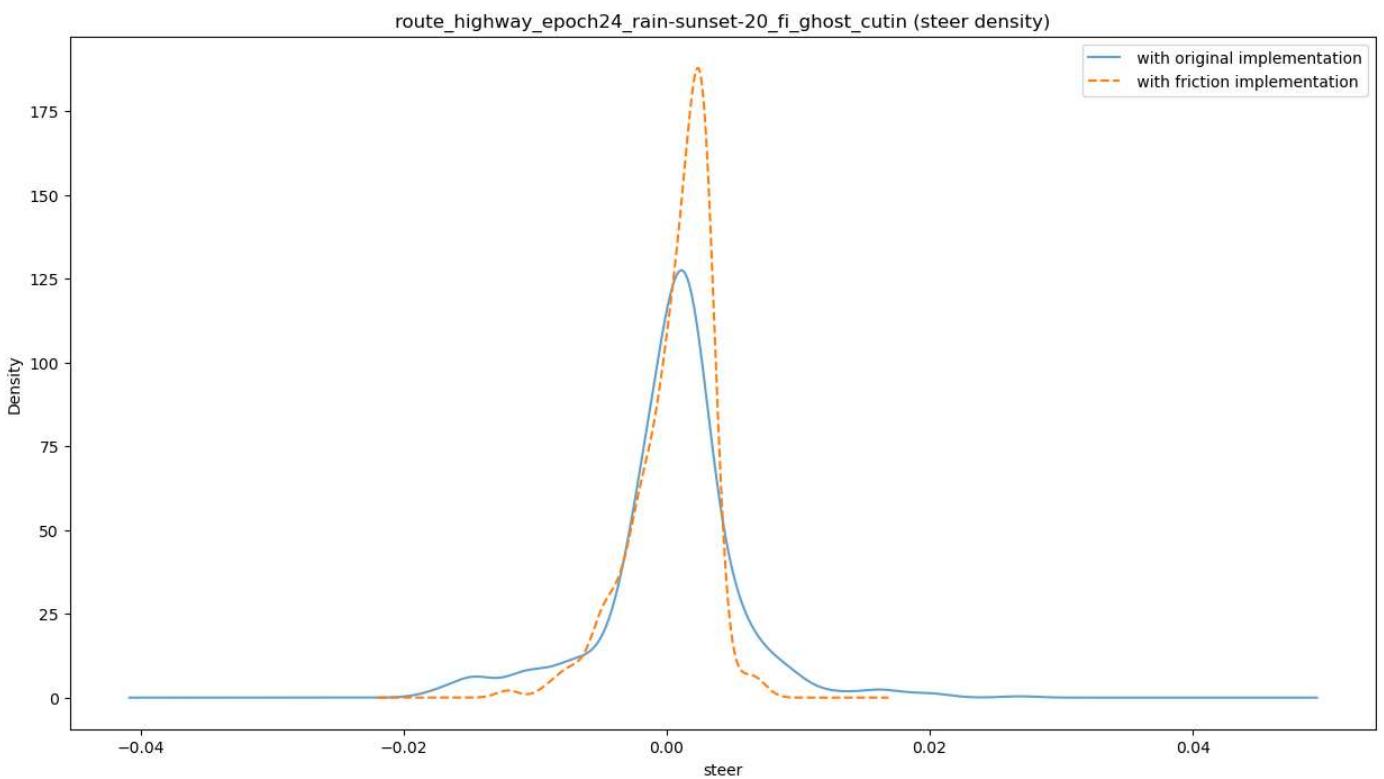
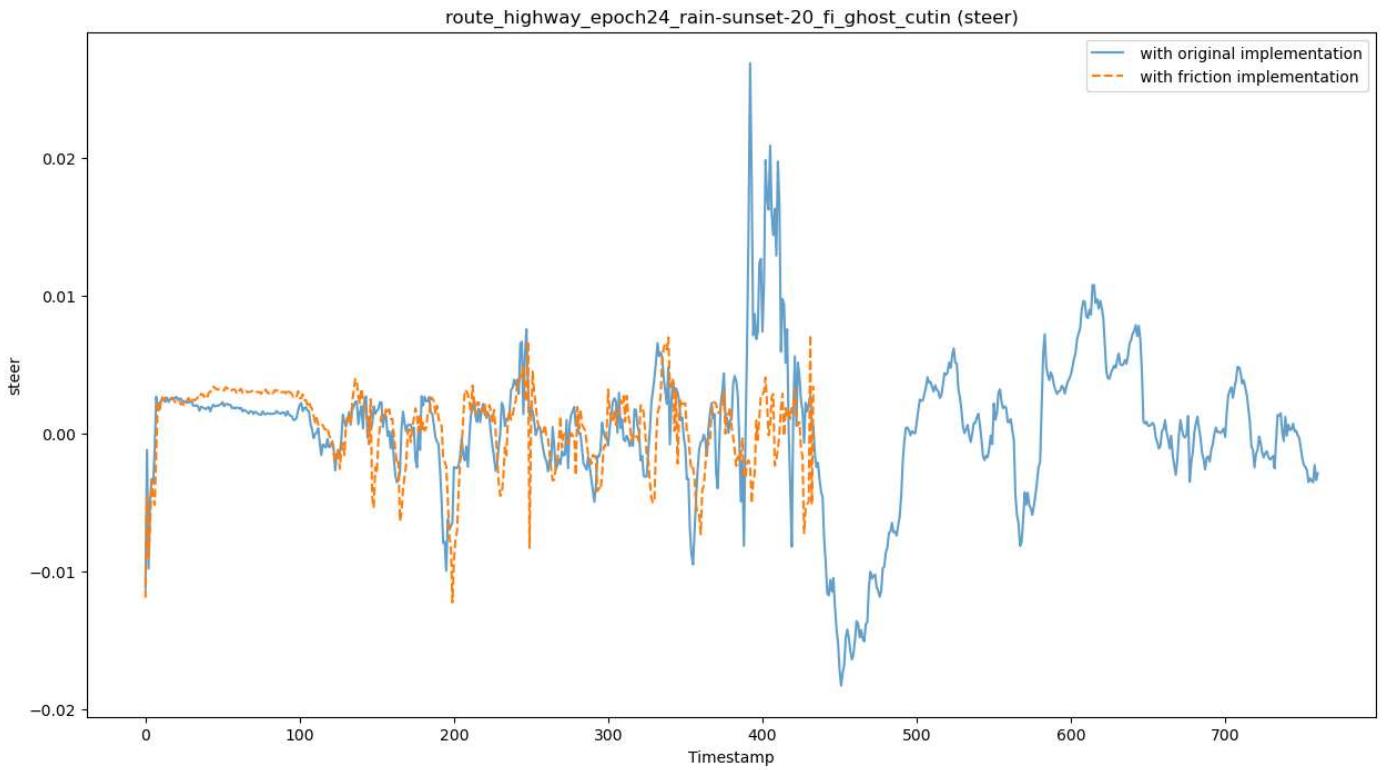
```

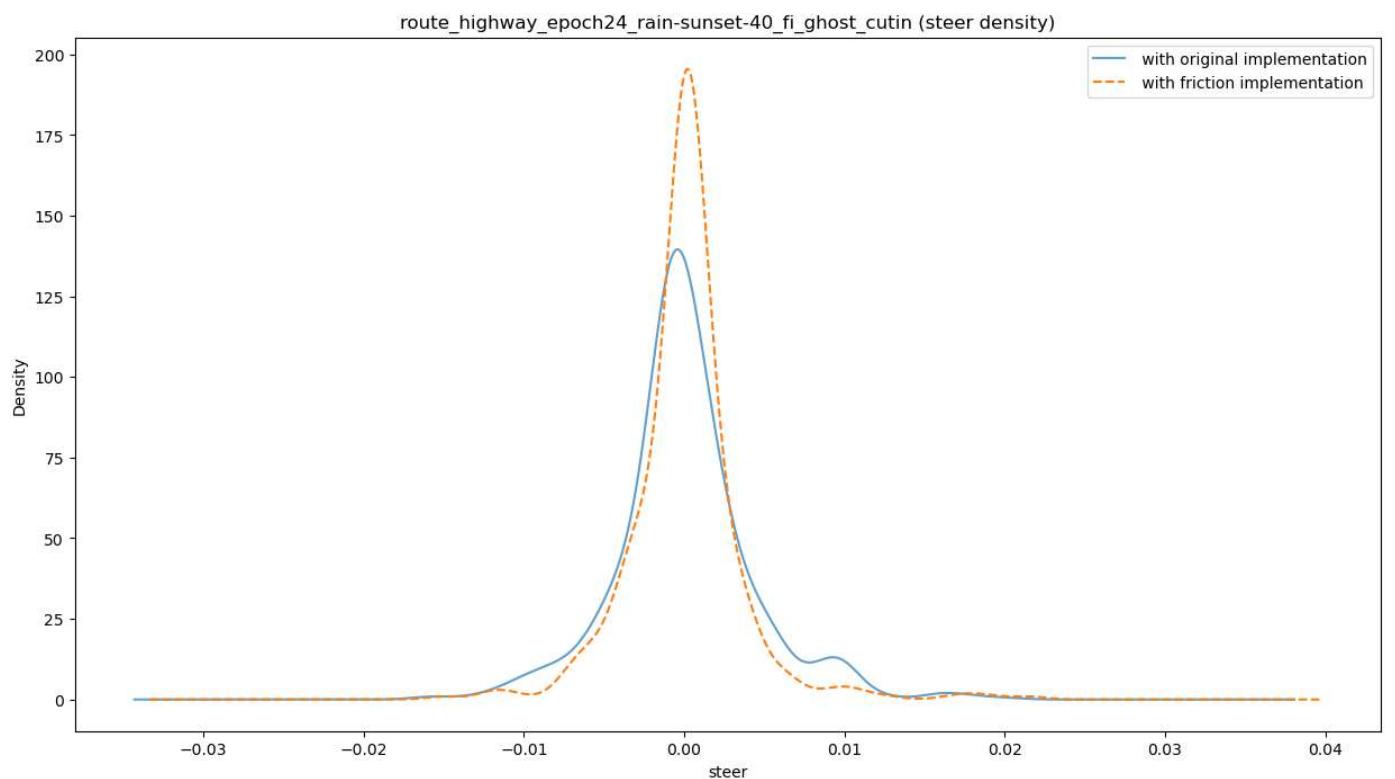
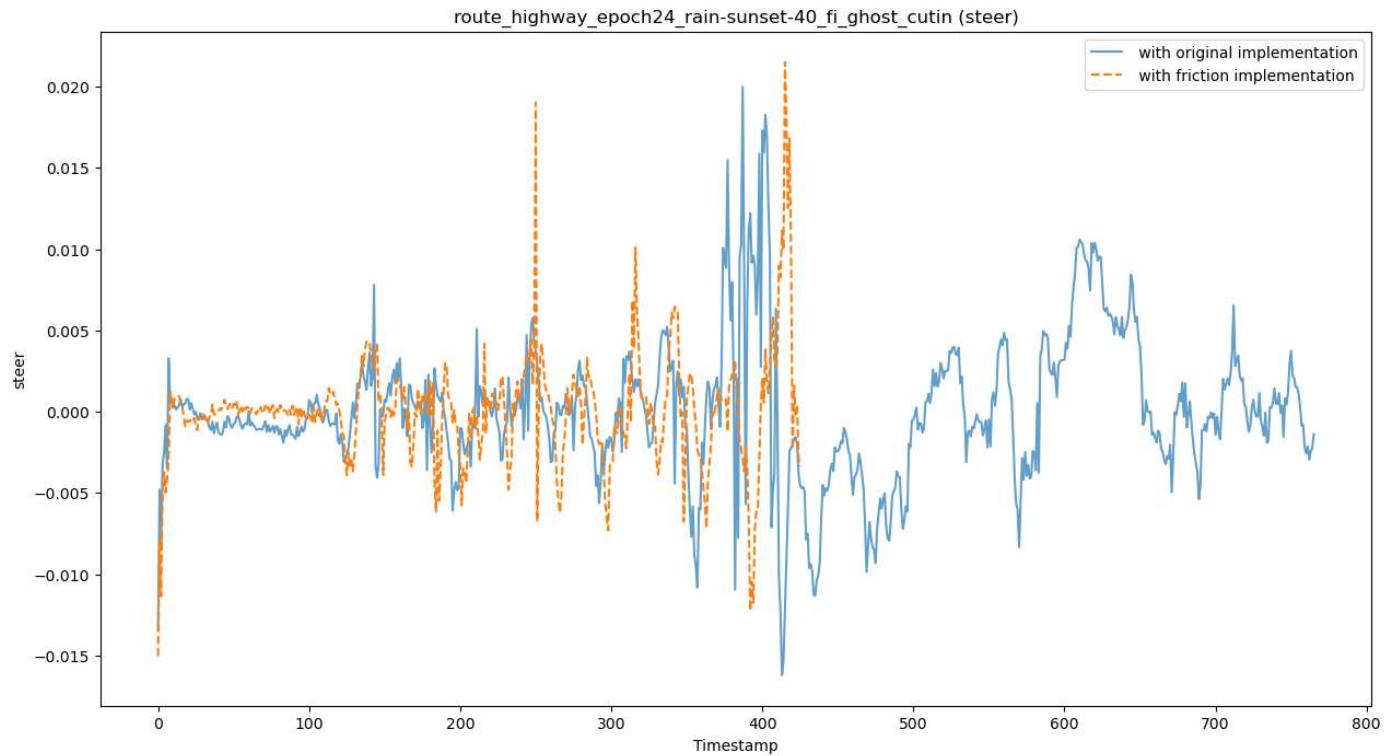
plt.figure(figsize=(15,8))
plt.xlabel("steer")
plt.ylabel("Density")
median_orig_steer.plot.density(alpha=0.7)
median_fric_steer.plot.density(style='--')
plt.title(subfolders[weather] + " (steer density)")
plt.legend(friction_legend)
plt.show()

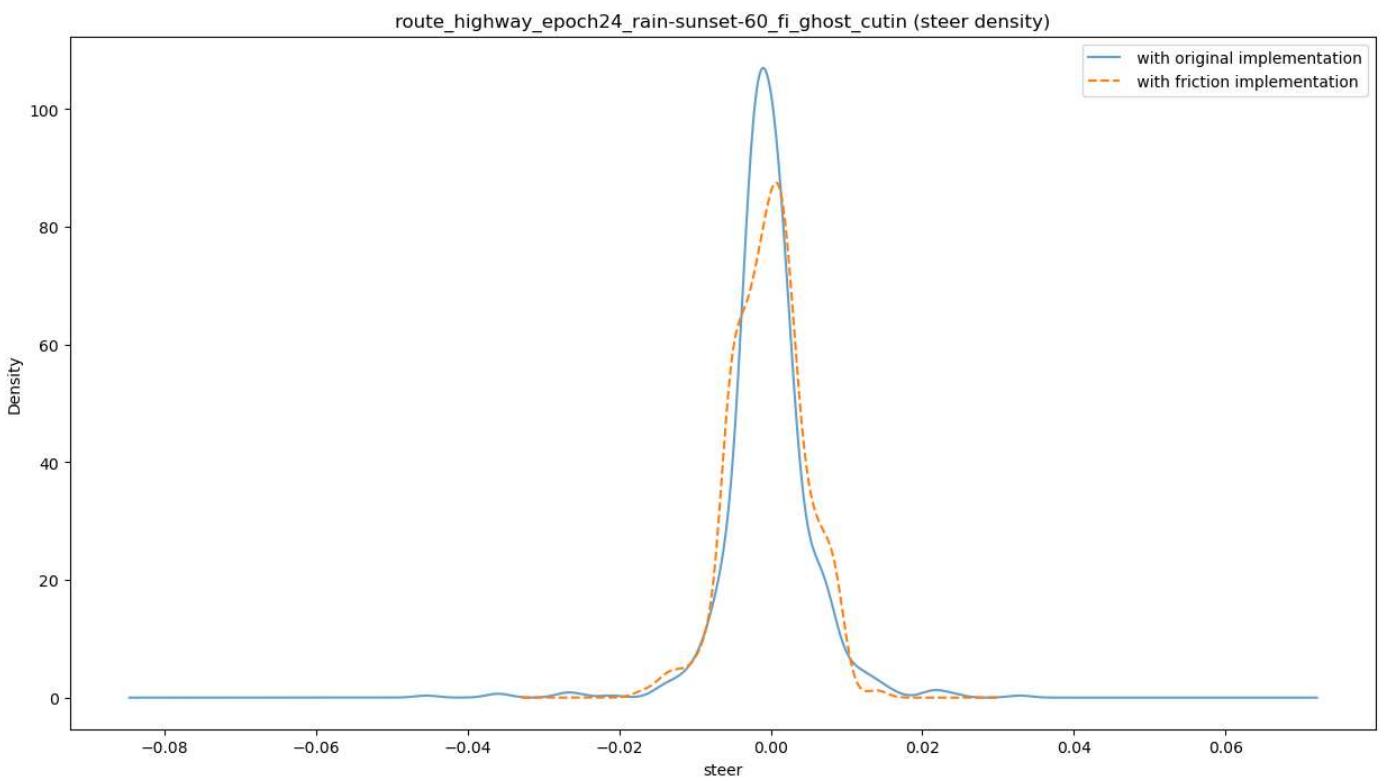
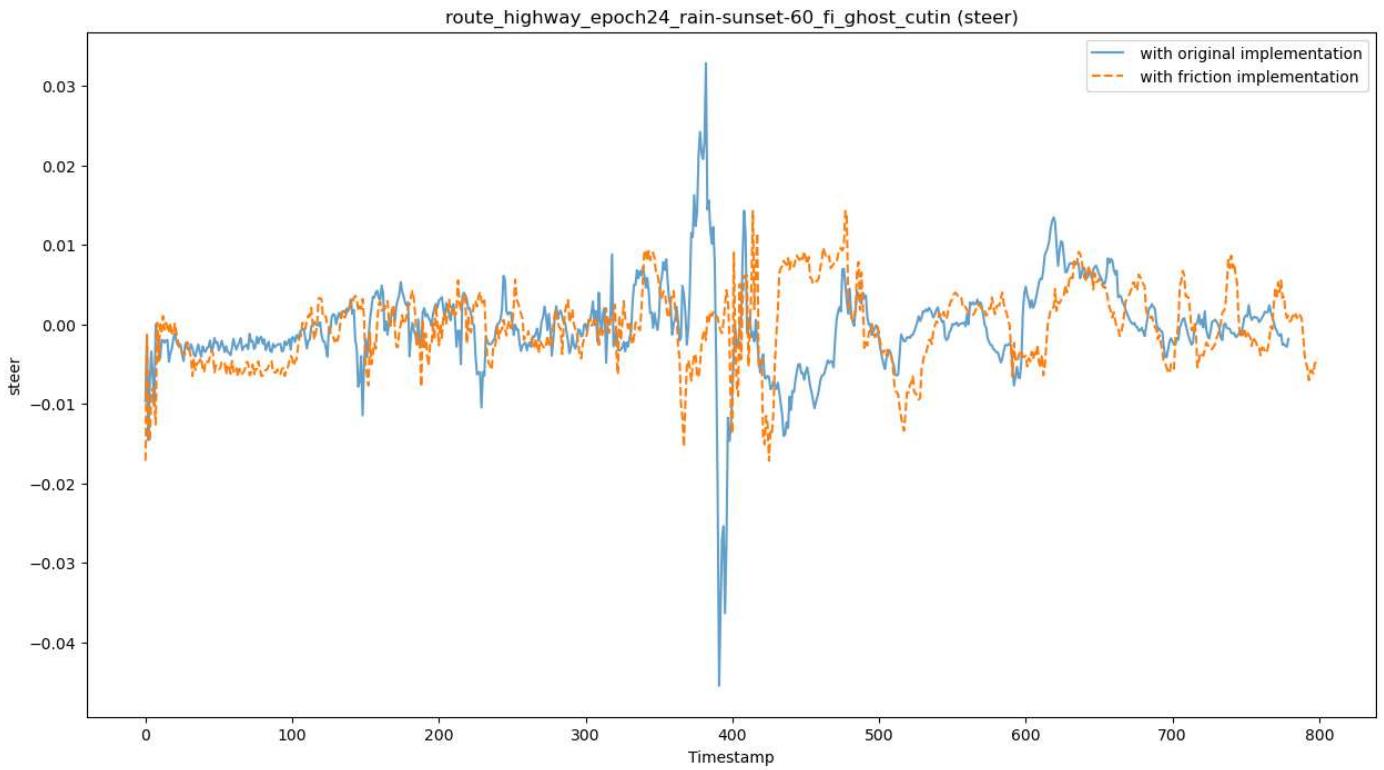
```

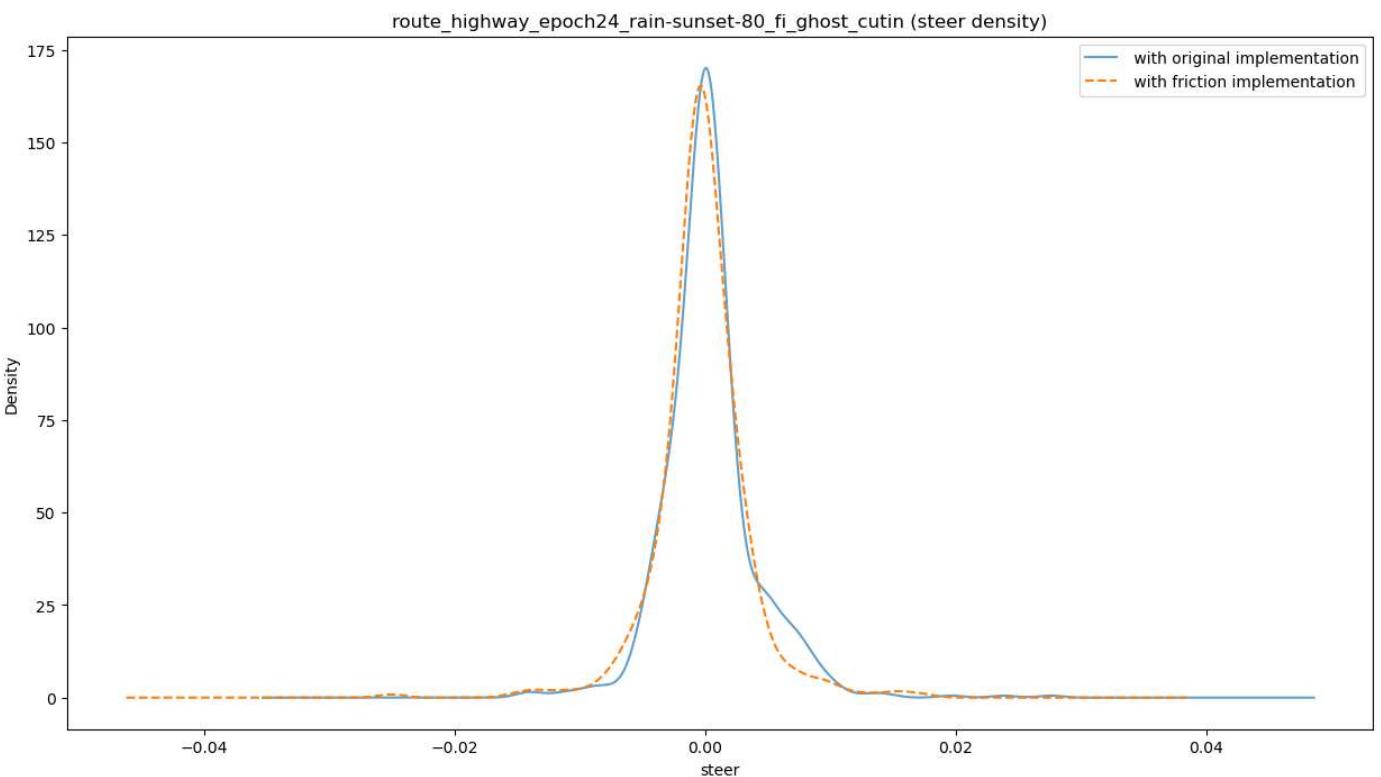
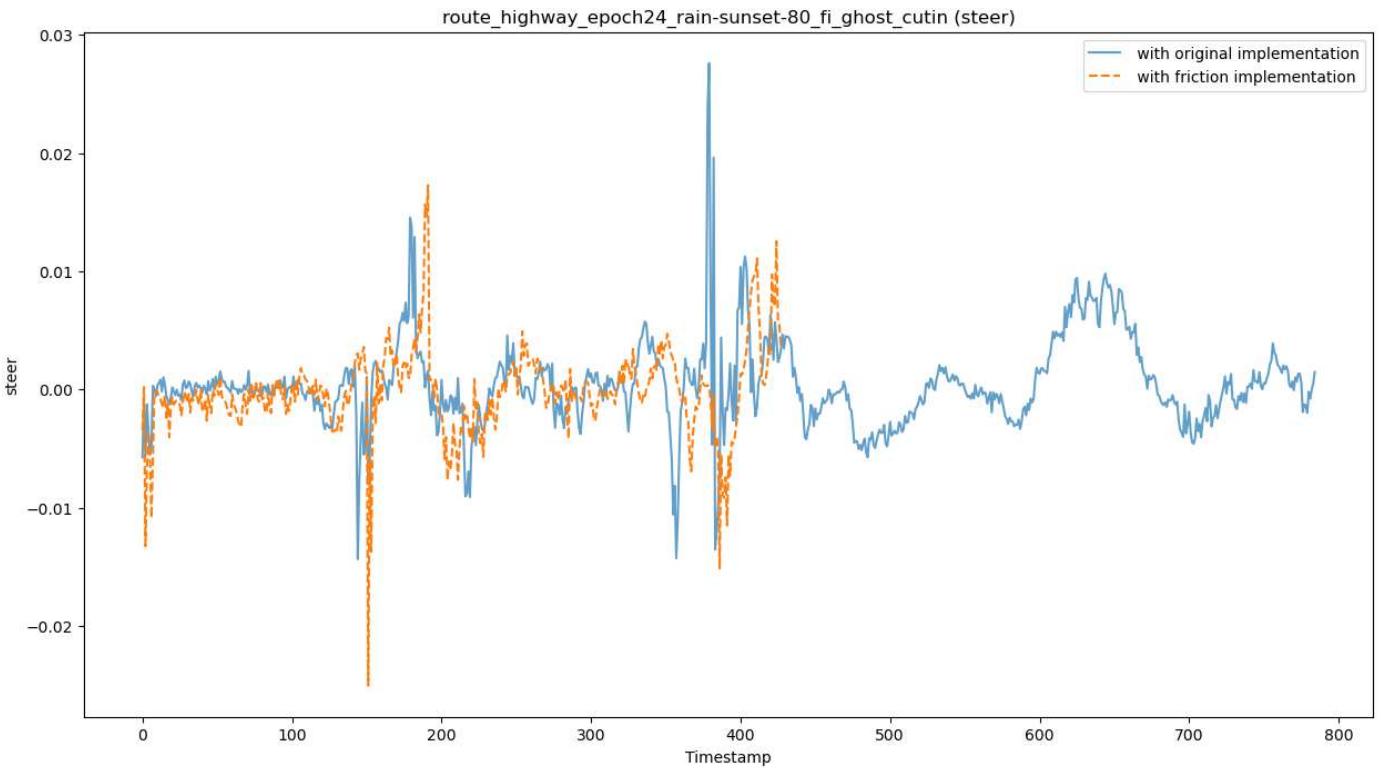












```
In [40]: plt.figure(figsize=(20,10))

for weather in range(len(txt_lists_fric)):

    df_orig_steer = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['steer'].dtypes == 'steer'):
            print(i)
        else:
            if not df_array_orig[weather][i]['steer'].isnull().values.any():
                df_orig_steer["Run"+str(i)] = df_array_orig[weather][i]['steer']

    median_orig_steer = df_orig_steer.median(axis=1)
```

```

plt.plot(median_orig_steer[200:], alpha=0.7)

df_fric_steer = pd.DataFrame()

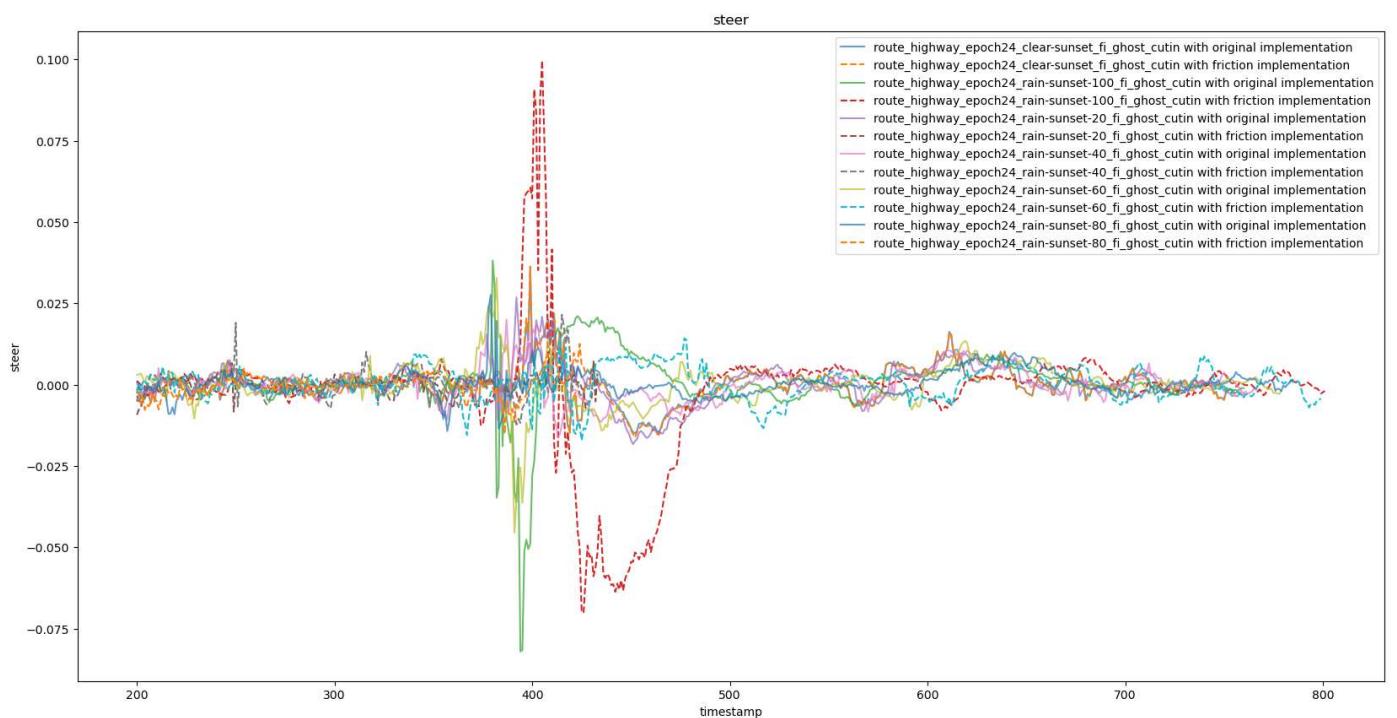
for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['steer'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['steer'].isnull().values.any():
            df_fric_steer["Run"+str(i)] = df_array_fric[weather][i]['steer']

median_fric_steer = df_fric_steer.median(axis=1)

plt.plot(median_fric_steer[200:], '--')

# plt.axis('equal')
plt.xlabel("timestamp")
plt.ylabel("steer")
plt.title("steer")
plt.legend(full_legend)
plt.show()

```



brake

```

In [64]: for weather in range(len(txt_lists_fric)):
    plt.figure(figsize=(15,8))
    df_orig_brake = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['brake'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_orig[weather][i]['brake'].isnull().values.any():
                df_orig_brake["Run"+str(i)] = df_array_orig[weather][i]['brake']

    median_orig_brake = df_orig_brake.median(axis=1)

    plt.plot(median_orig_brake, alpha=0.7)

```

```

df_fric_brake = pd.DataFrame()

for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['brake'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['brake'].isnull().values.any():
            df_fric_brake["Run"+str(i)] = df_array_fric[weather][i]['brake']

median_fric_brake = df_fric_brake.median(axis=1)

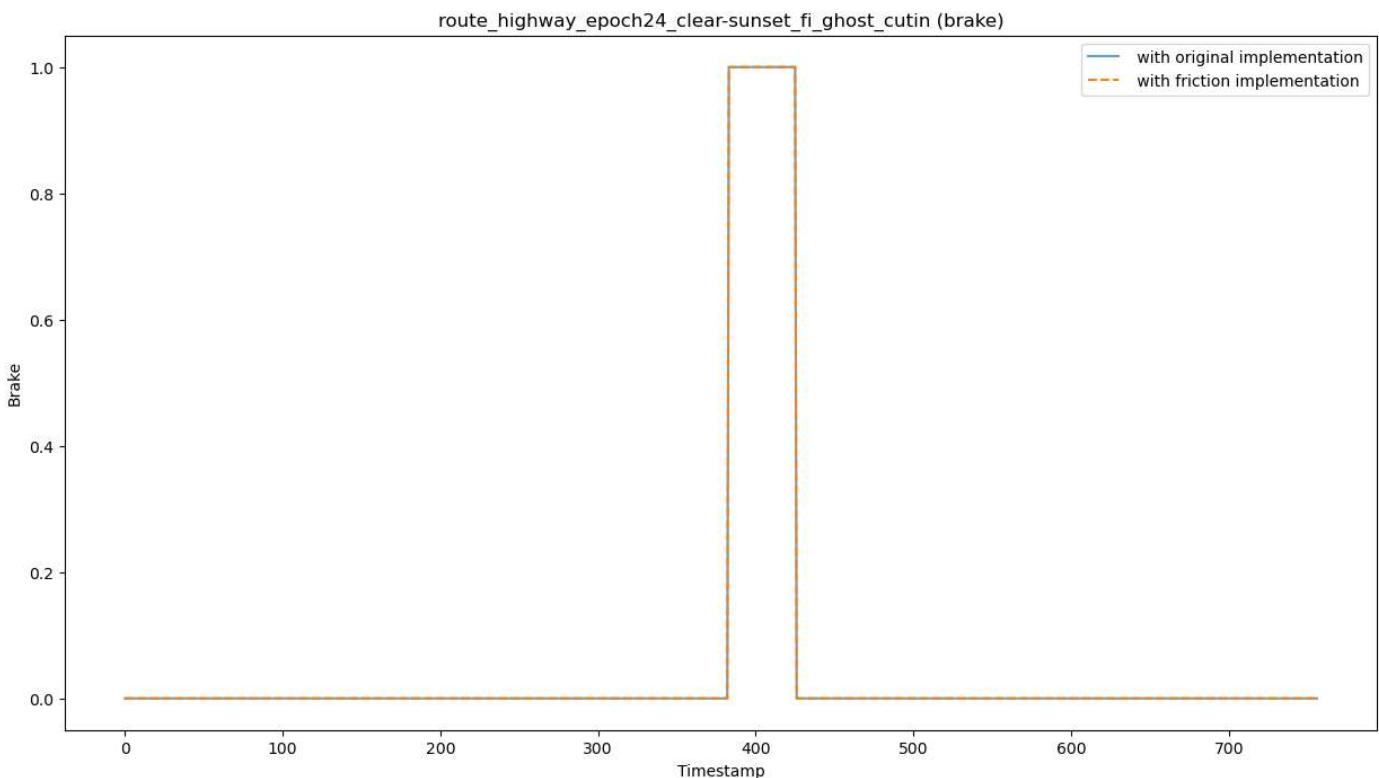
all_medians[weather][2].append(median_orig_brake)
all_medians[weather][2].append(median_fric_brake)

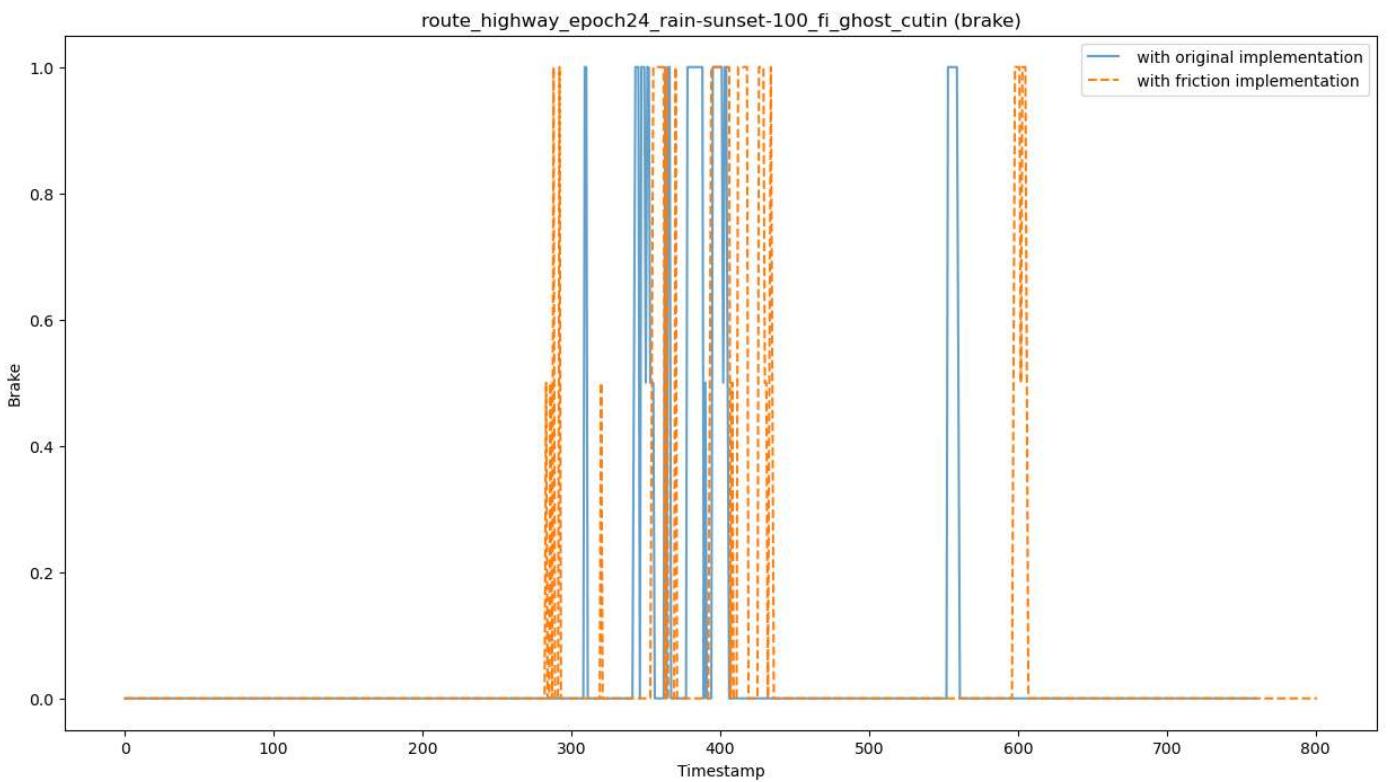
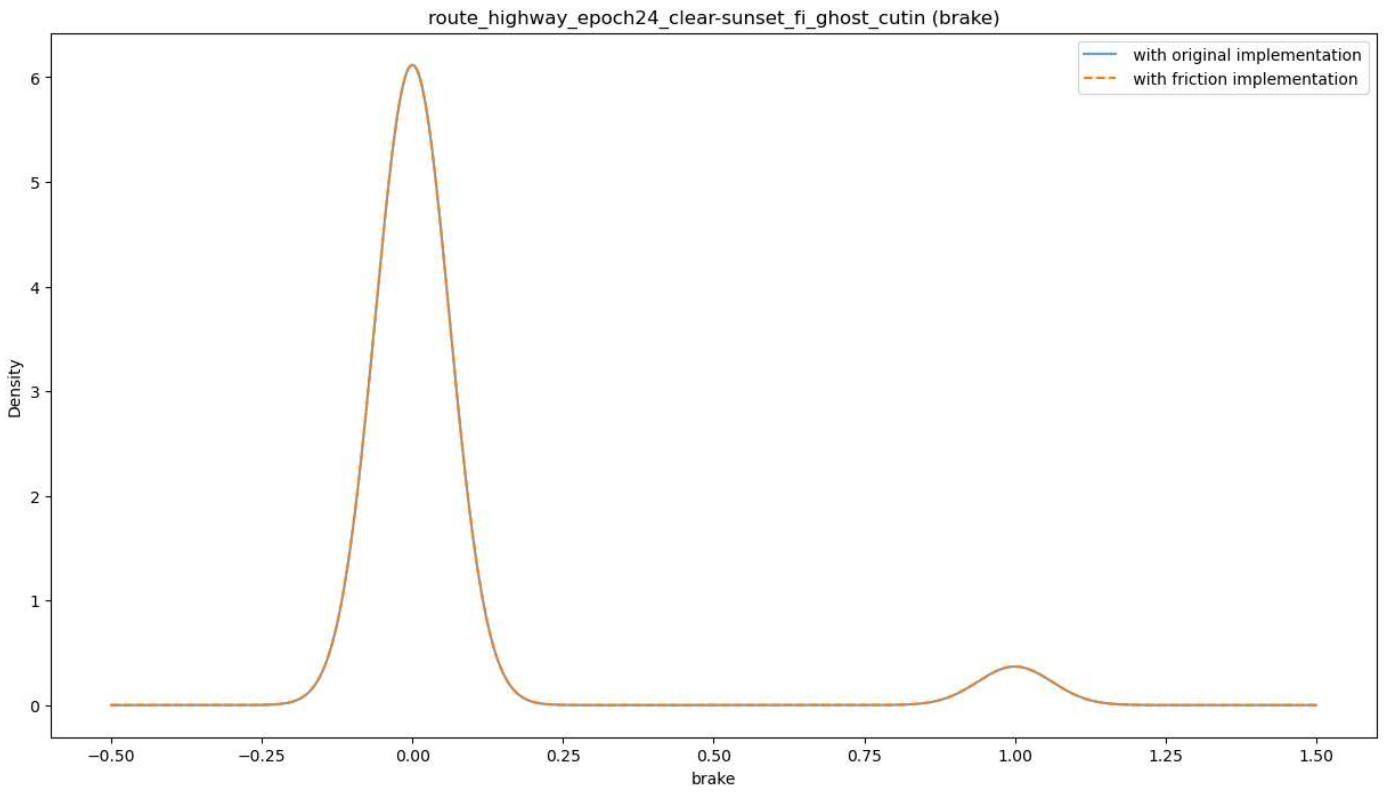
plt.plot(median_fric_brake, '--')

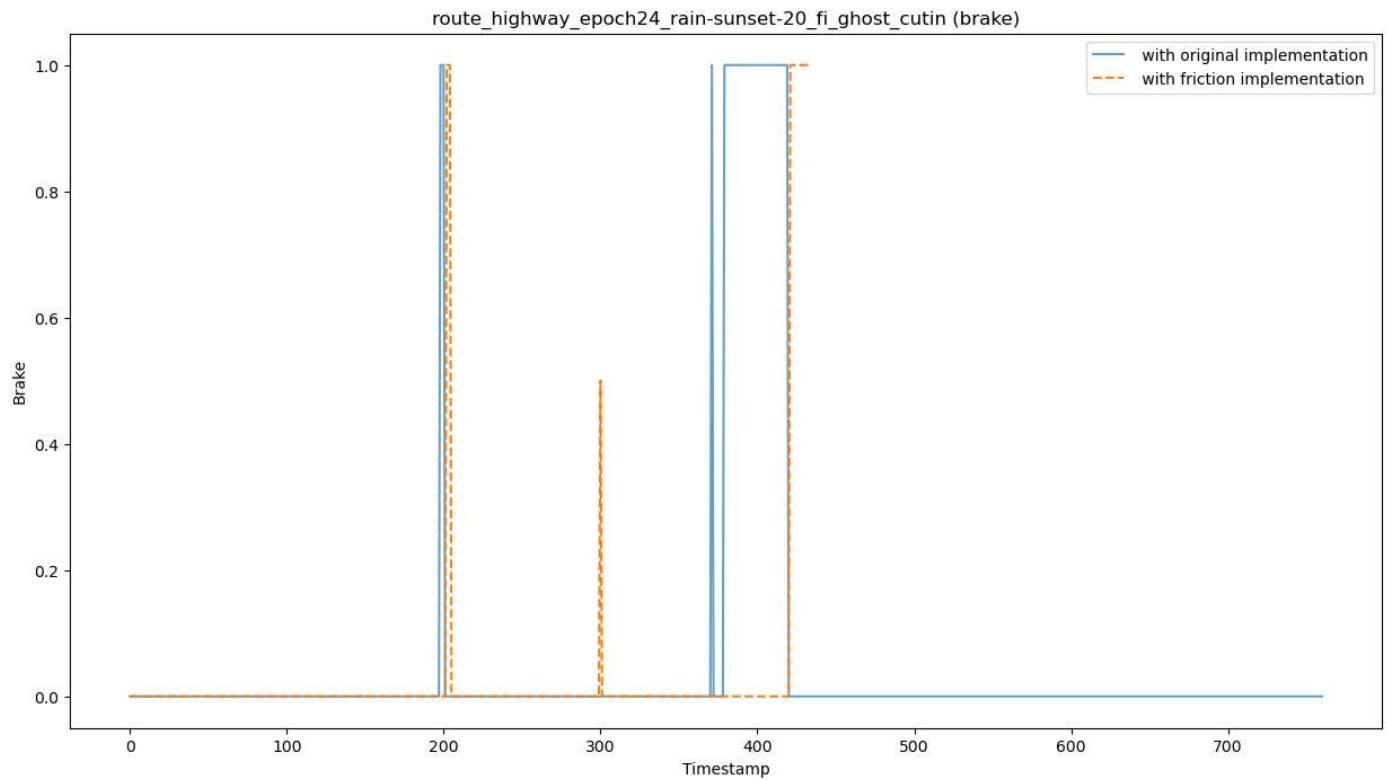
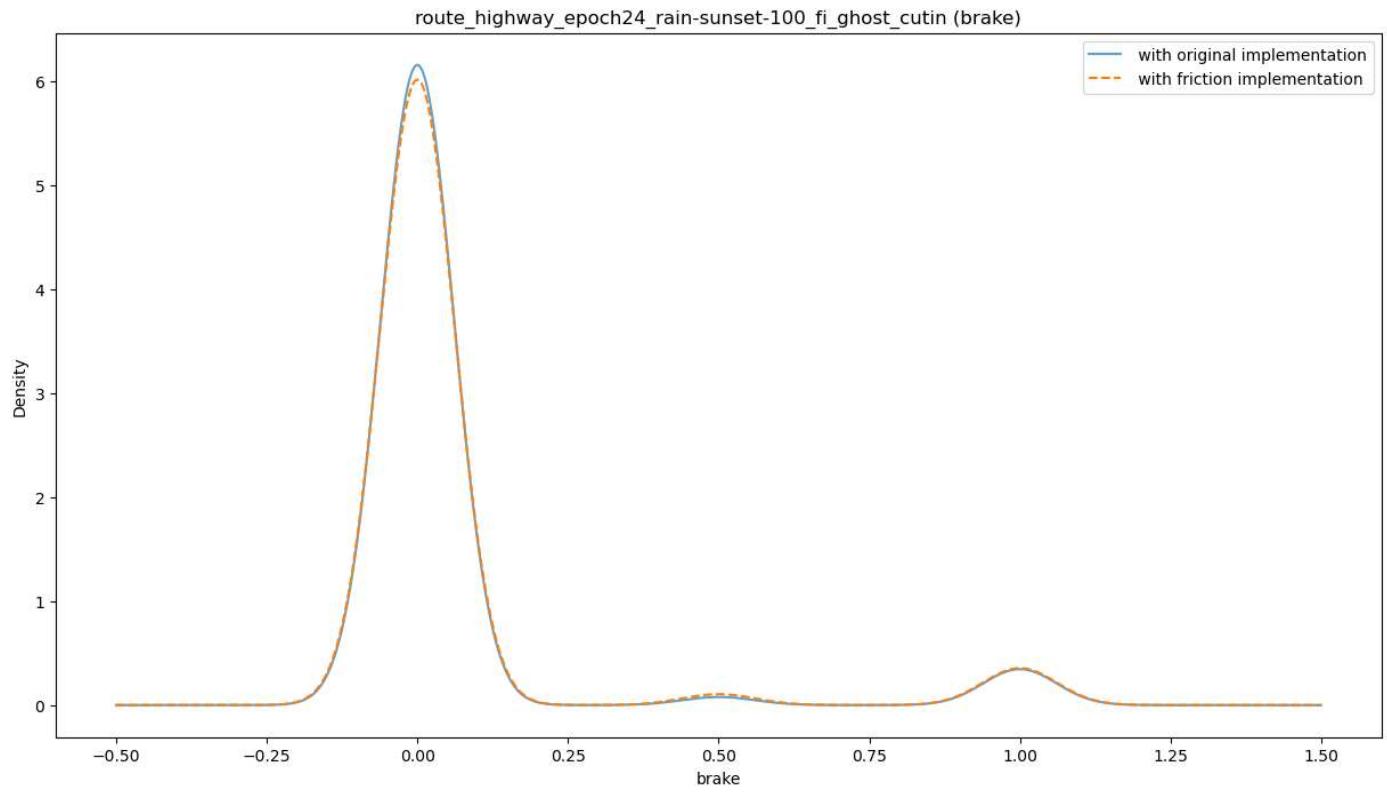
plt.title(subfolders[weather])
plt.legend(friction_legend)
plt.xlabel("Timestamp")
plt.ylabel("Brake")
plt.title(subfolders[weather] + " (brake)")
plt.show()

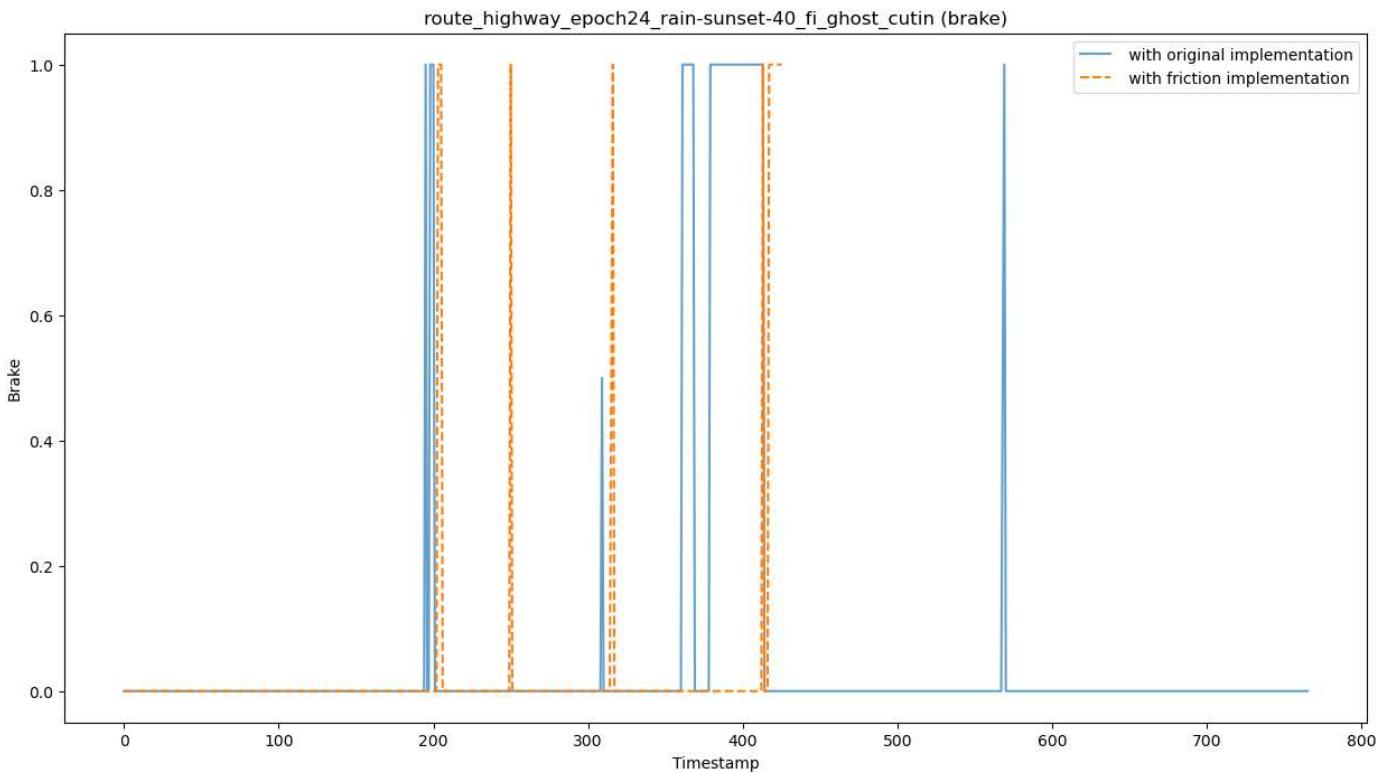
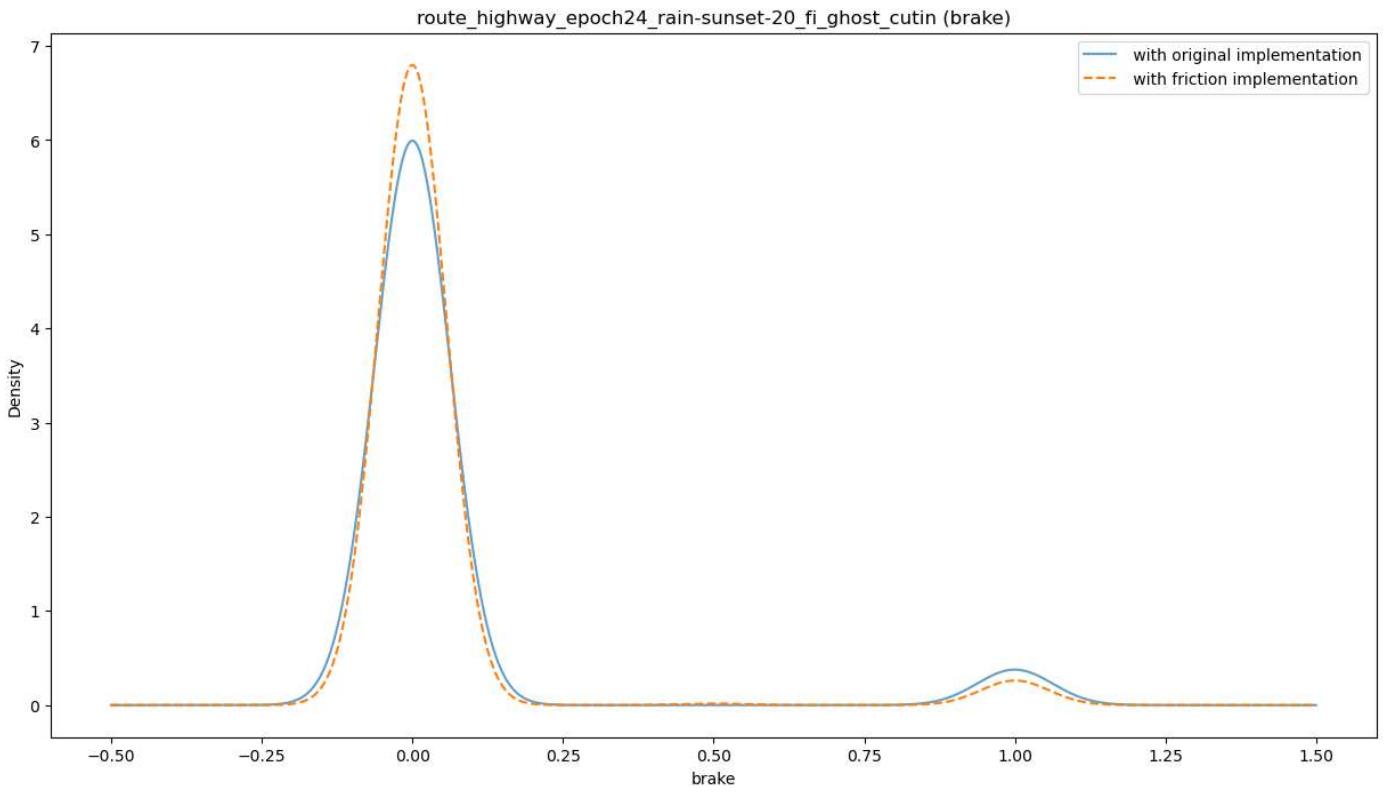
plt.figure(figsize=(15,8))
plt.xlabel("brake")
plt.ylabel("Density")
median_orig_brake.plot.density(alpha=0.7)
median_fric_brake.plot.density(style='--')
plt.title(subfolders[weather] + " (brake)")
plt.legend(friction_legend)
plt.show()

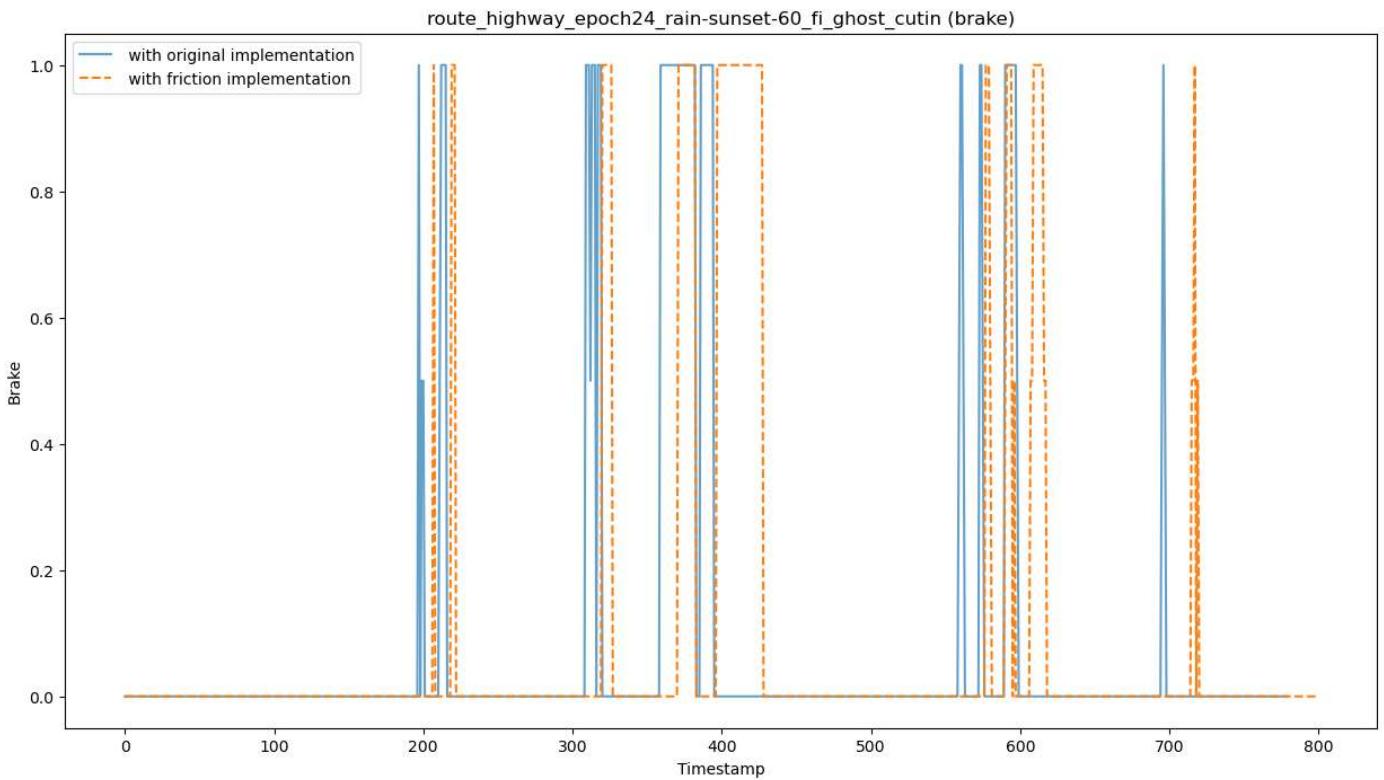
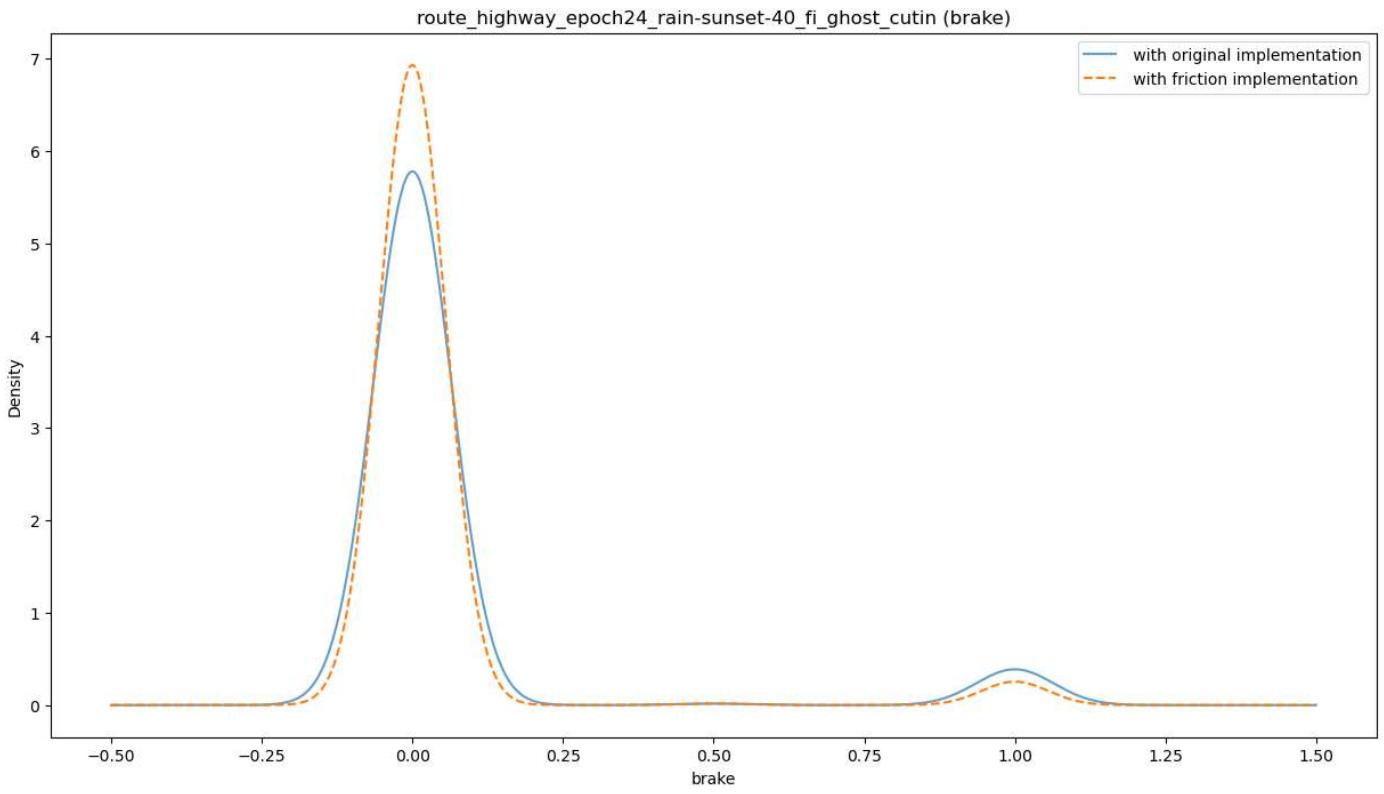
```

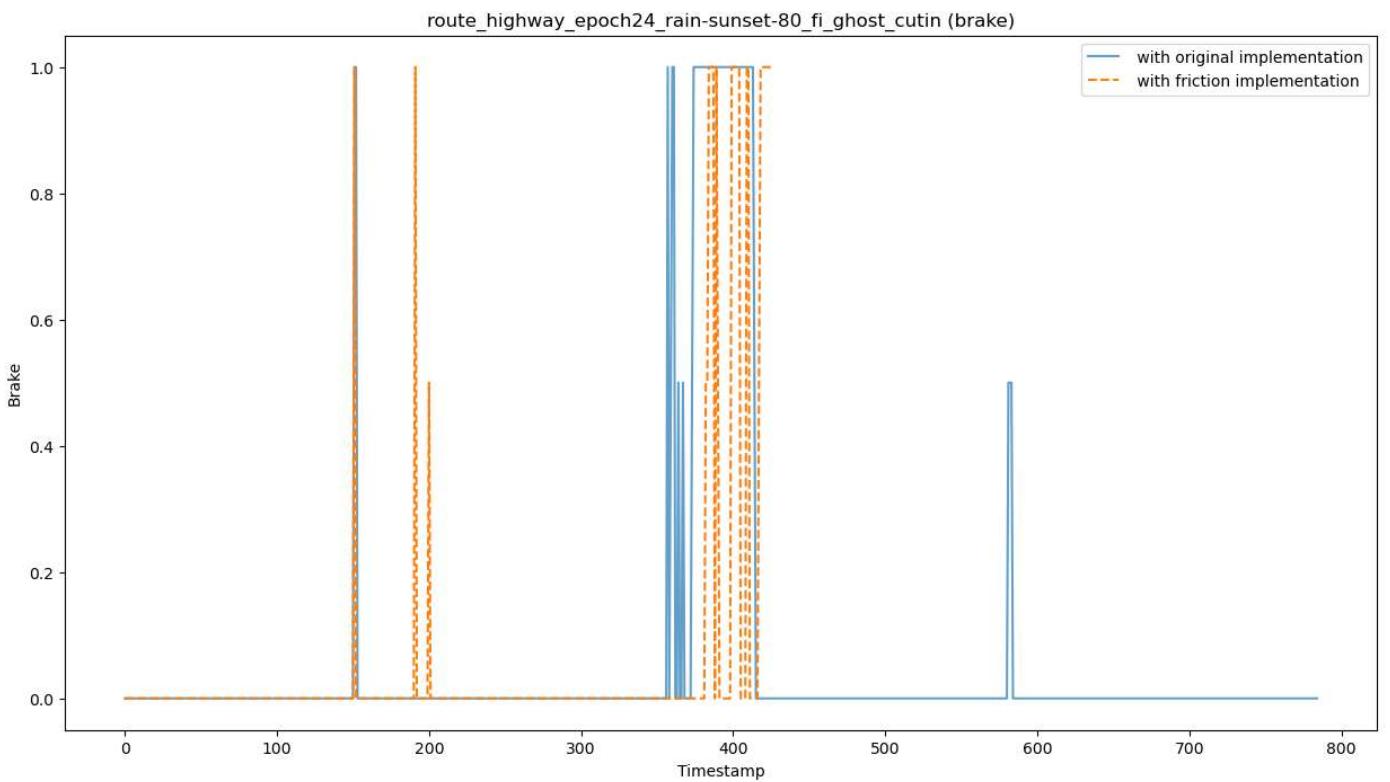
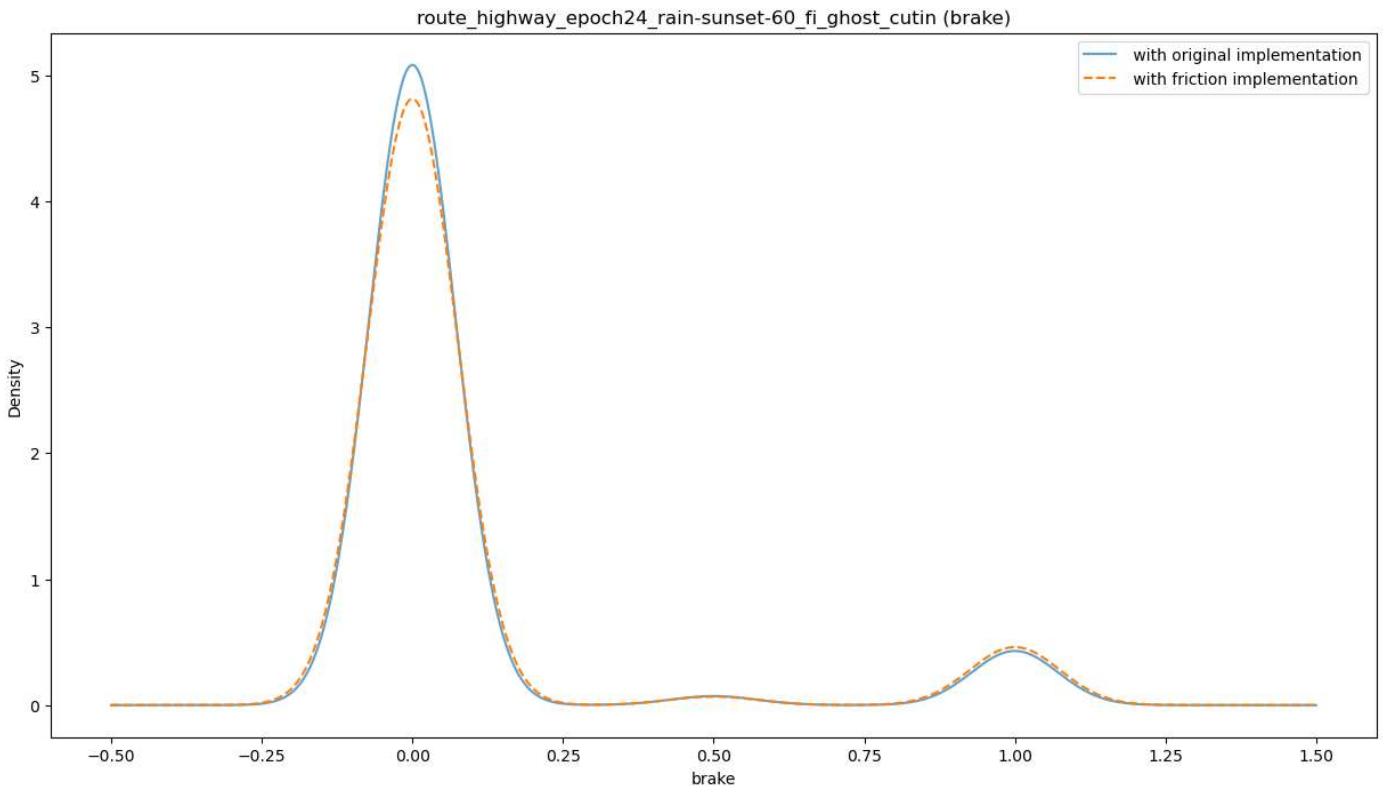


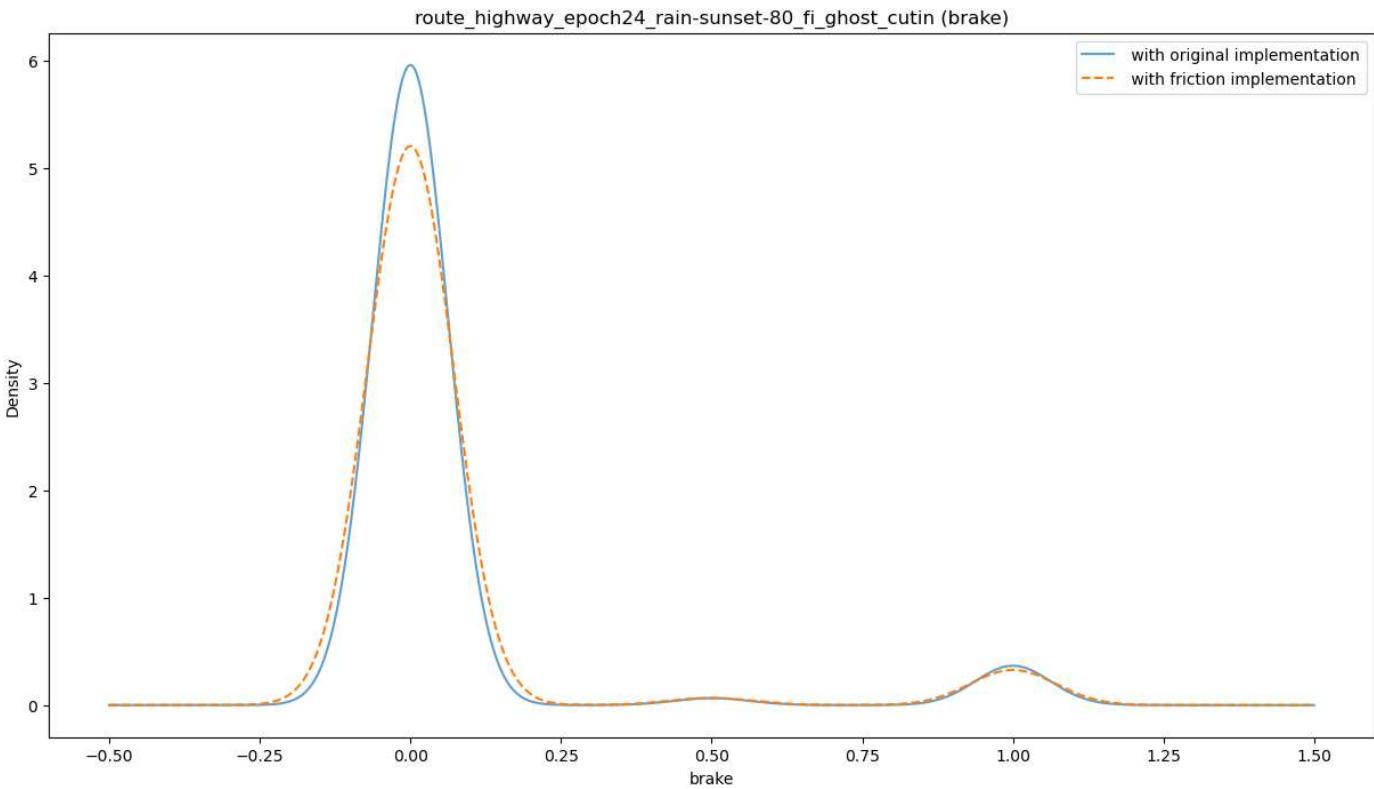












```
In [42]: plt.figure(figsize=(20,10))

for weather in range(len(txt_lists_fric)):

    df_orig_brake = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['brake'].dtypes == 'brake'):
            print(i)
        else:
            if not df_array_orig[weather][i]['brake'].isnull().values.any():
                df_orig_brake["Run"+str(i)] = df_array_orig[weather][i]['brake']

    median_orig_brake = df_orig_brake.median(axis=1)

    plt.plot(median_orig_brake[200:], alpha=0.7)

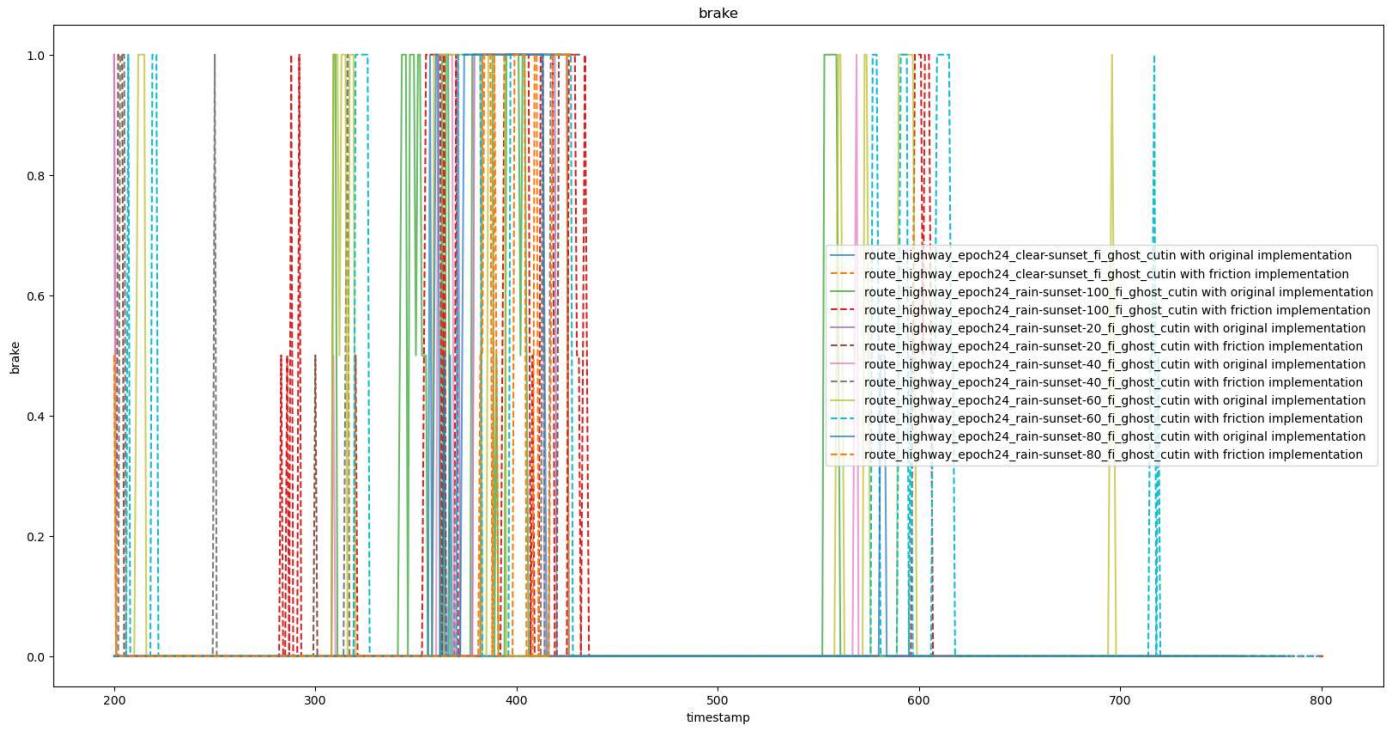
    df_fric_brake = pd.DataFrame()

    for i in range(0, len(df_array_fric)):
        if(df_array_fric[weather][i]['brake'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_fric[weather][i]['brake'].isnull().values.any() :
                df_fric_brake["Run"+str(i)] = df_array_fric[weather][i]['brake']

    median_fric_brake = df_fric_brake.median(axis=1)

    plt.plot(median_fric_brake[200:], '--')

    # plt.axis('equal')
    plt.xlabel("timestamp")
    plt.ylabel("brake")
    plt.title("brake")
    plt.legend(full_legend)
    plt.show()
```



```
In [65]: for weather in range(len(txt_lists_fric)):
    plt.figure(figsize=(15,8))
    df_orig_throttle = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['throttle'].dtypes == 'object'):
            print(i)
        else:
            if not df_array_orig[weather][i]['throttle'].isnull().values.any():
                df_orig_throttle["Run"+str(i)] = df_array_orig[weather][i]['throttle']

    median_orig_throttle = df_orig_throttle.median(axis=1)

    plt.plot(median_orig_throttle, alpha=0.7)

df_fric_throttle = pd.DataFrame()

for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['throttle'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['throttle'].isnull().values.any():
            df_fric_throttle["Run"+str(i)] = df_array_fric[weather][i]['throttle']

    median_fric_throttle = df_fric_throttle.median(axis=1)

all_medians[weather][2].append(median_orig_throttle)
all_medians[weather][2].append(median_fric_throttle)

plt.plot(median_fric_throttle, '--')

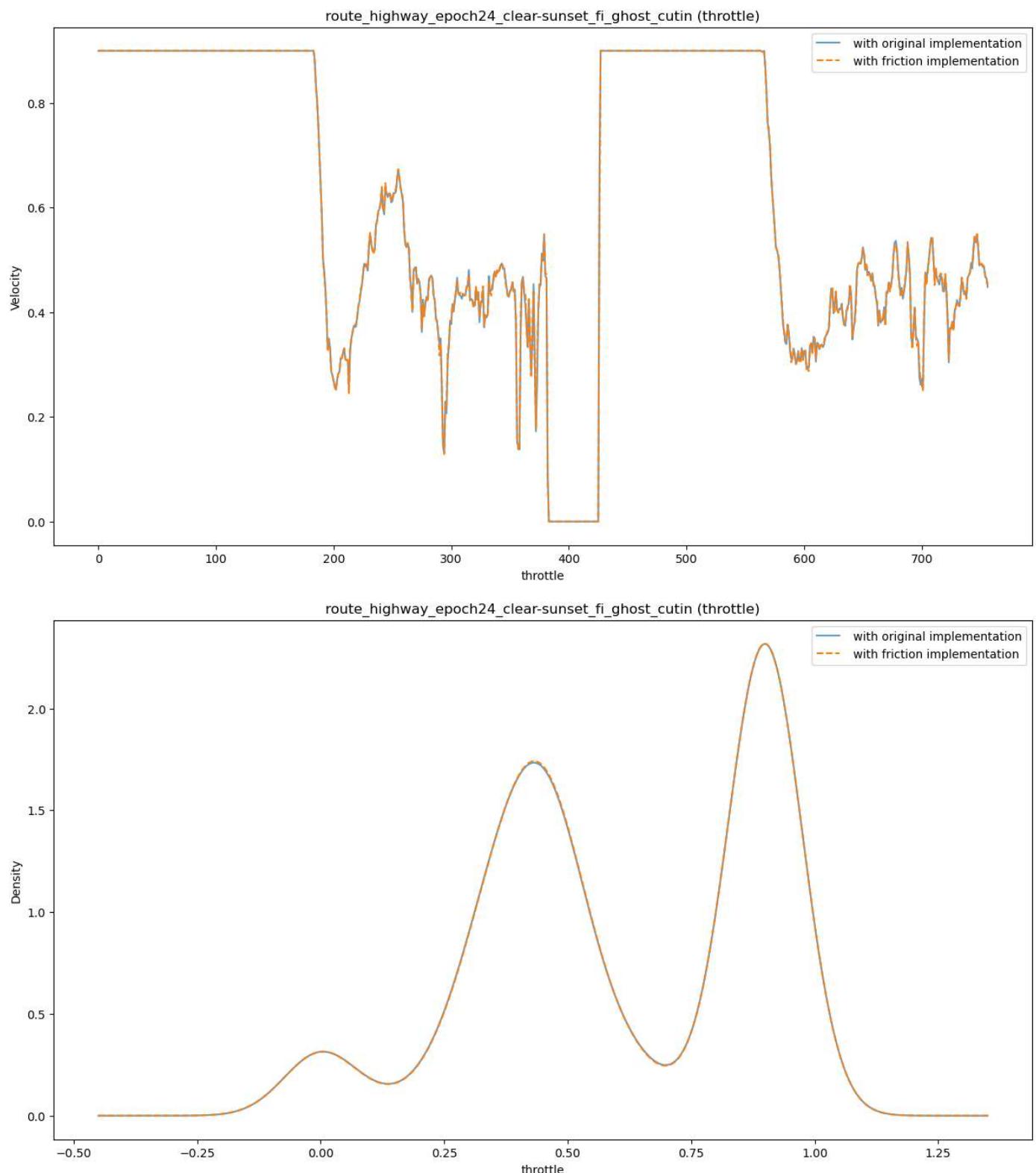
plt.title(subfolders[weather])
plt.legend(friction_legend)
plt.xlabel("throttle")
plt.ylabel("Velocity")
plt.title(subfolders[weather] + " (throttle)")
plt.show()

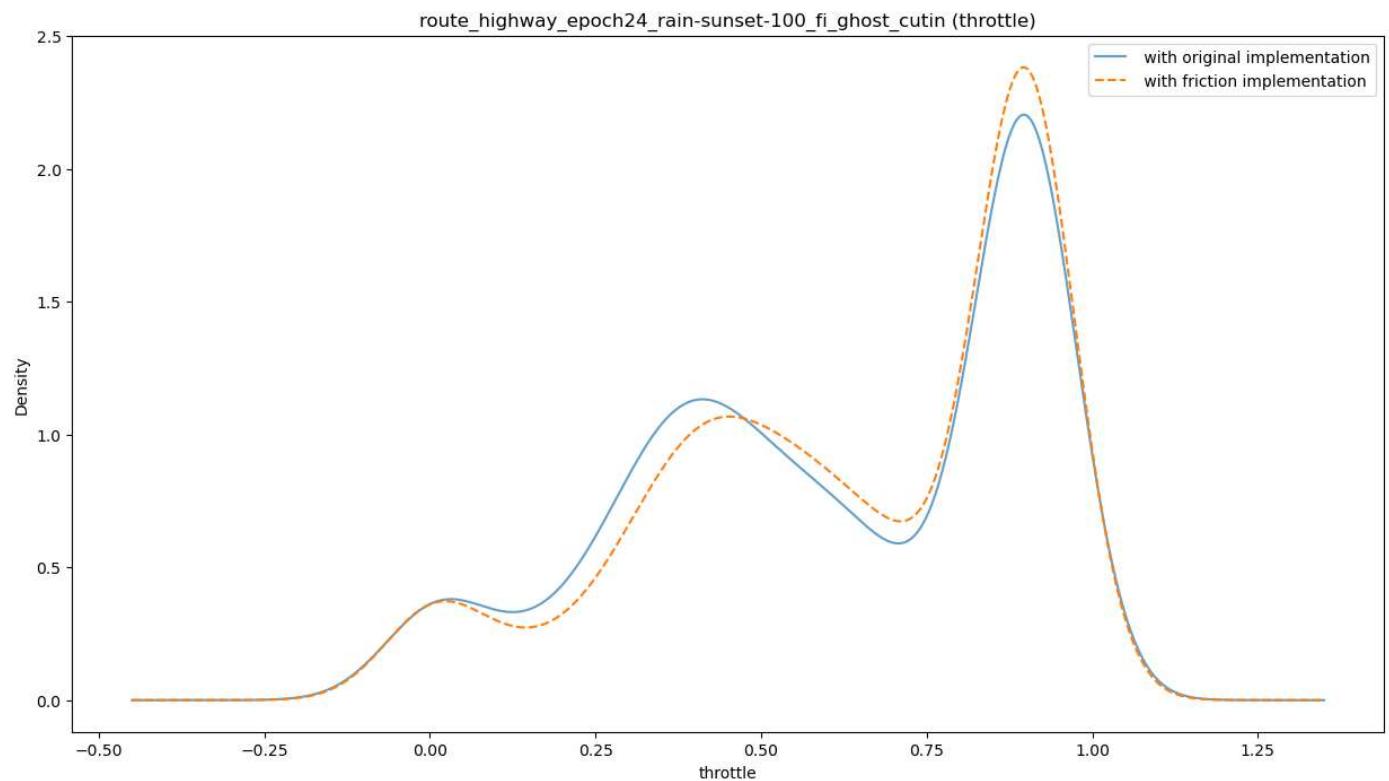
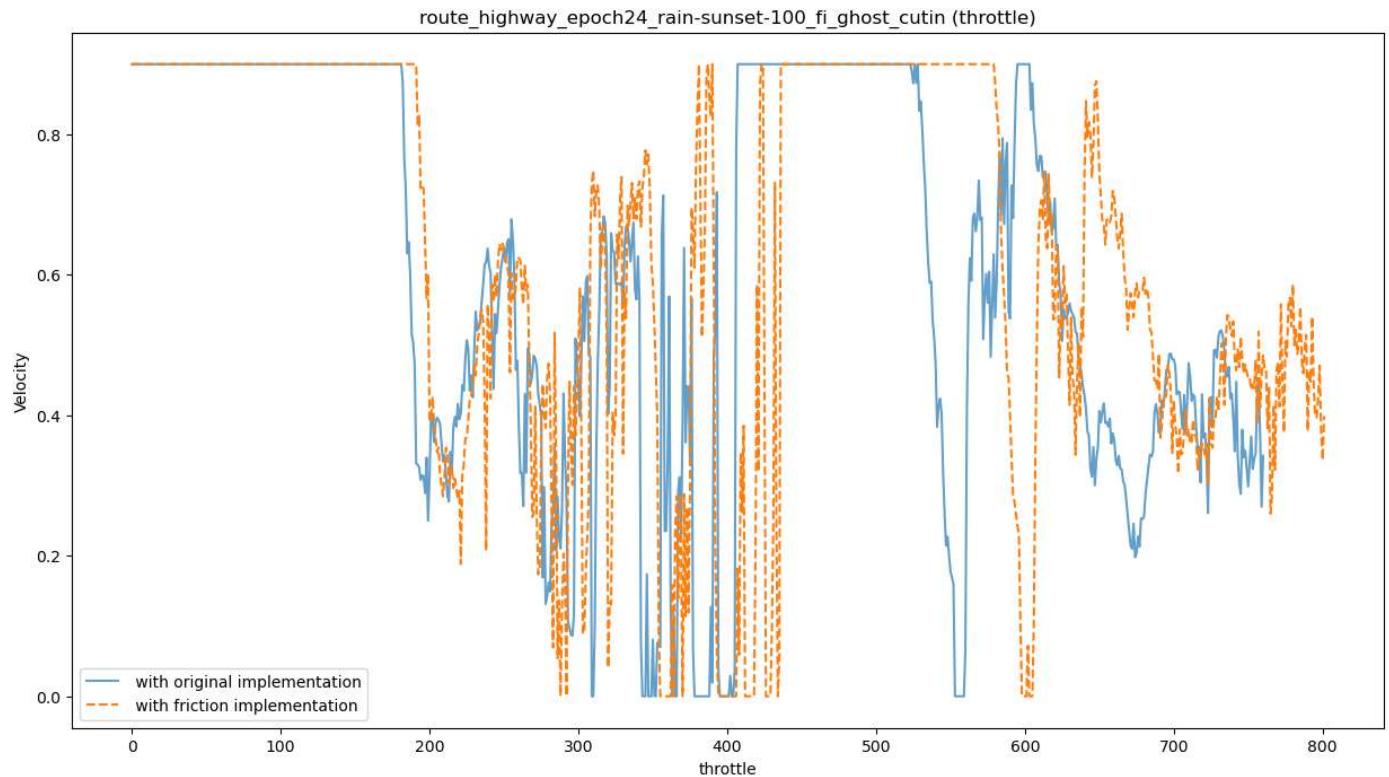
plt.figure(figsize=(15,8))
plt.xlabel("throttle")
```

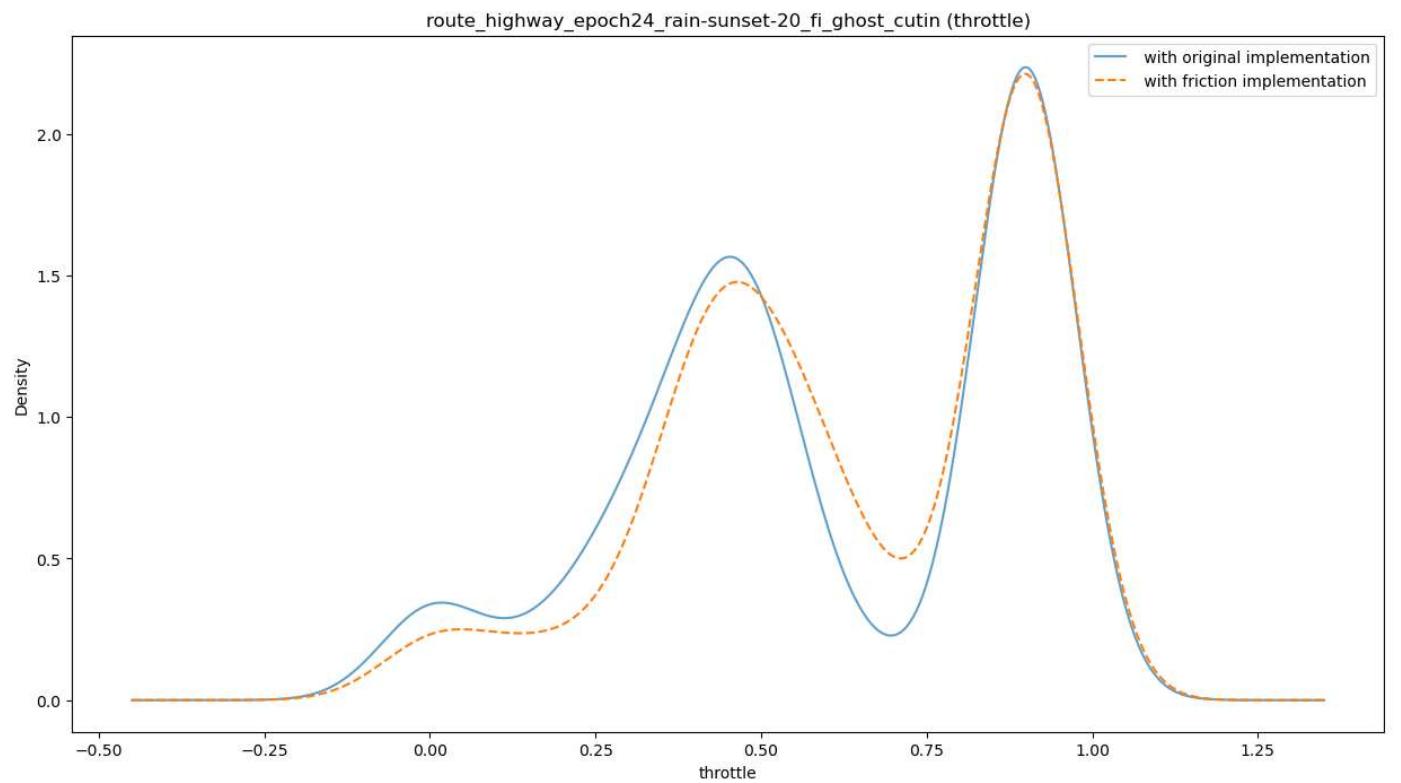
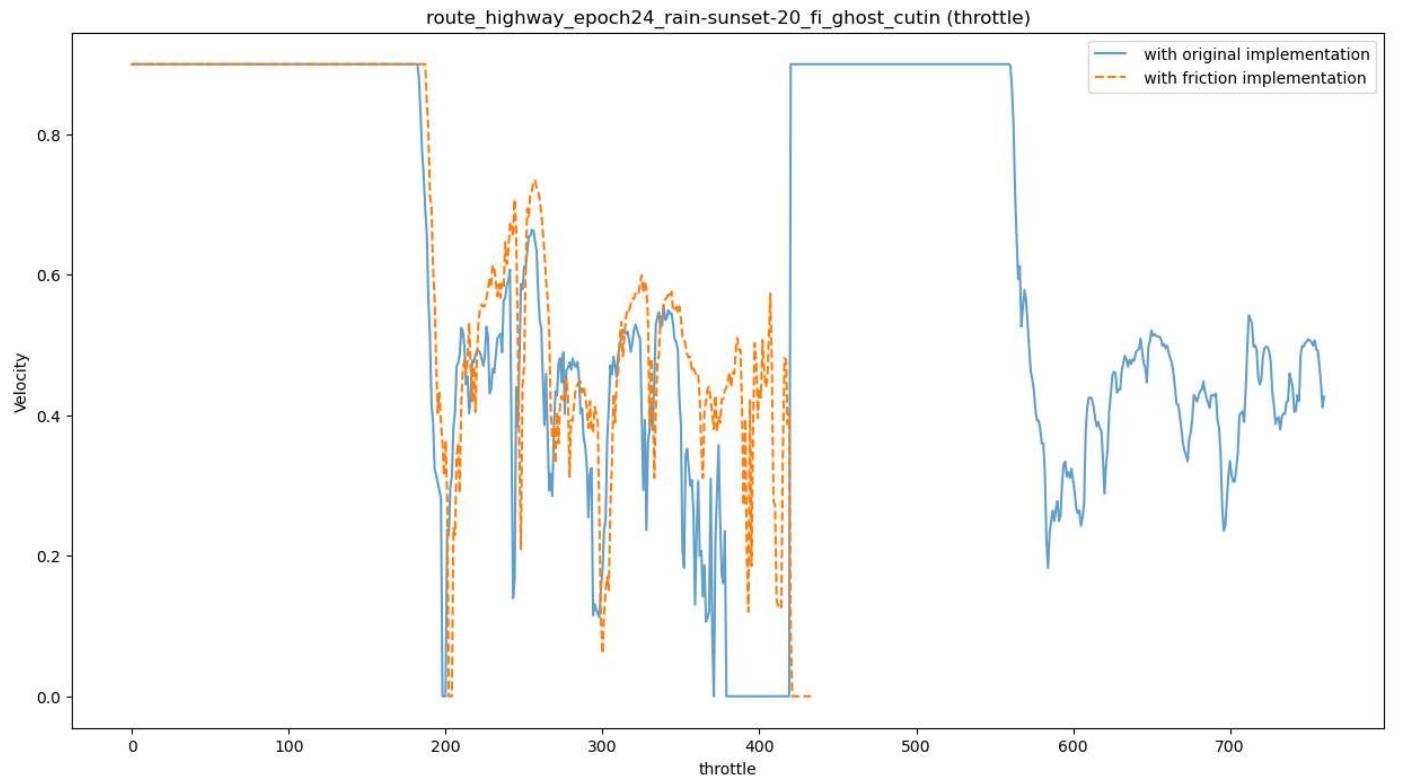
```

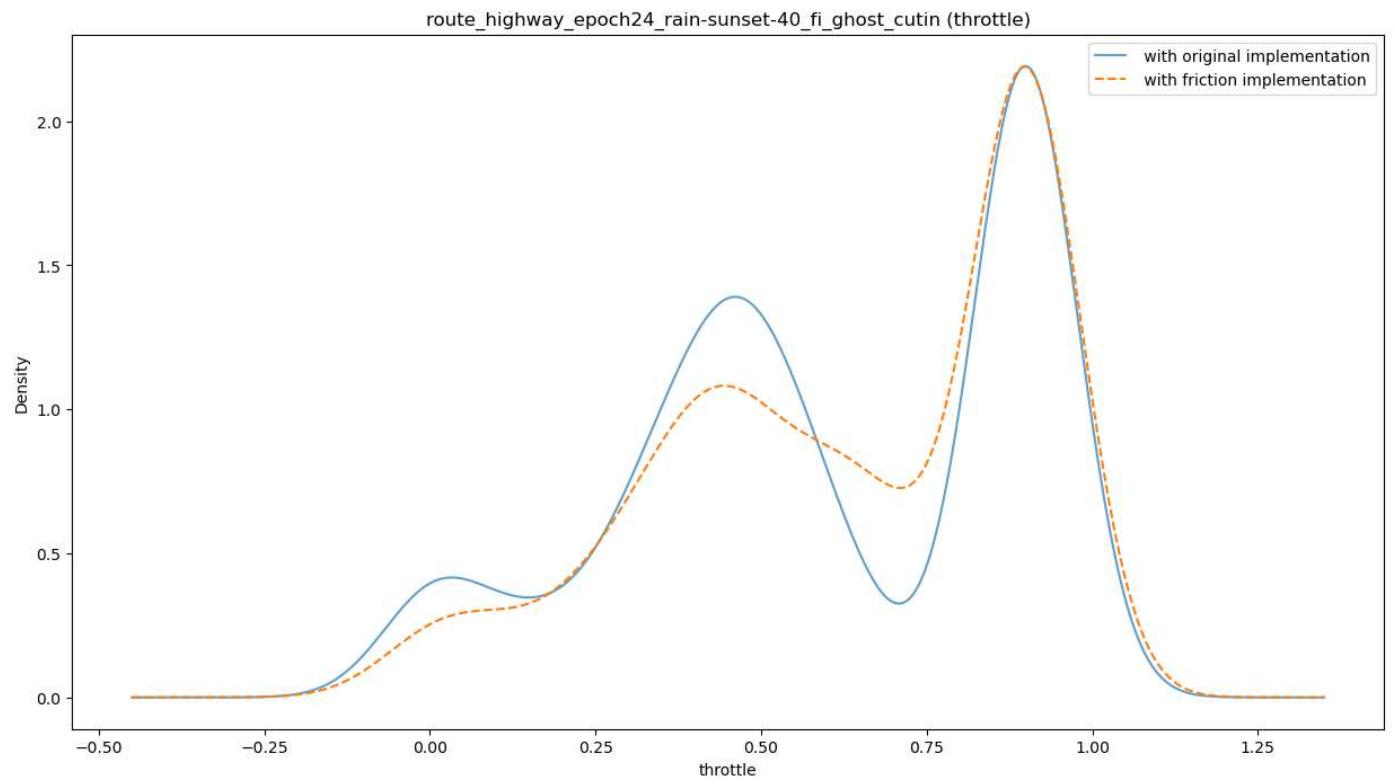
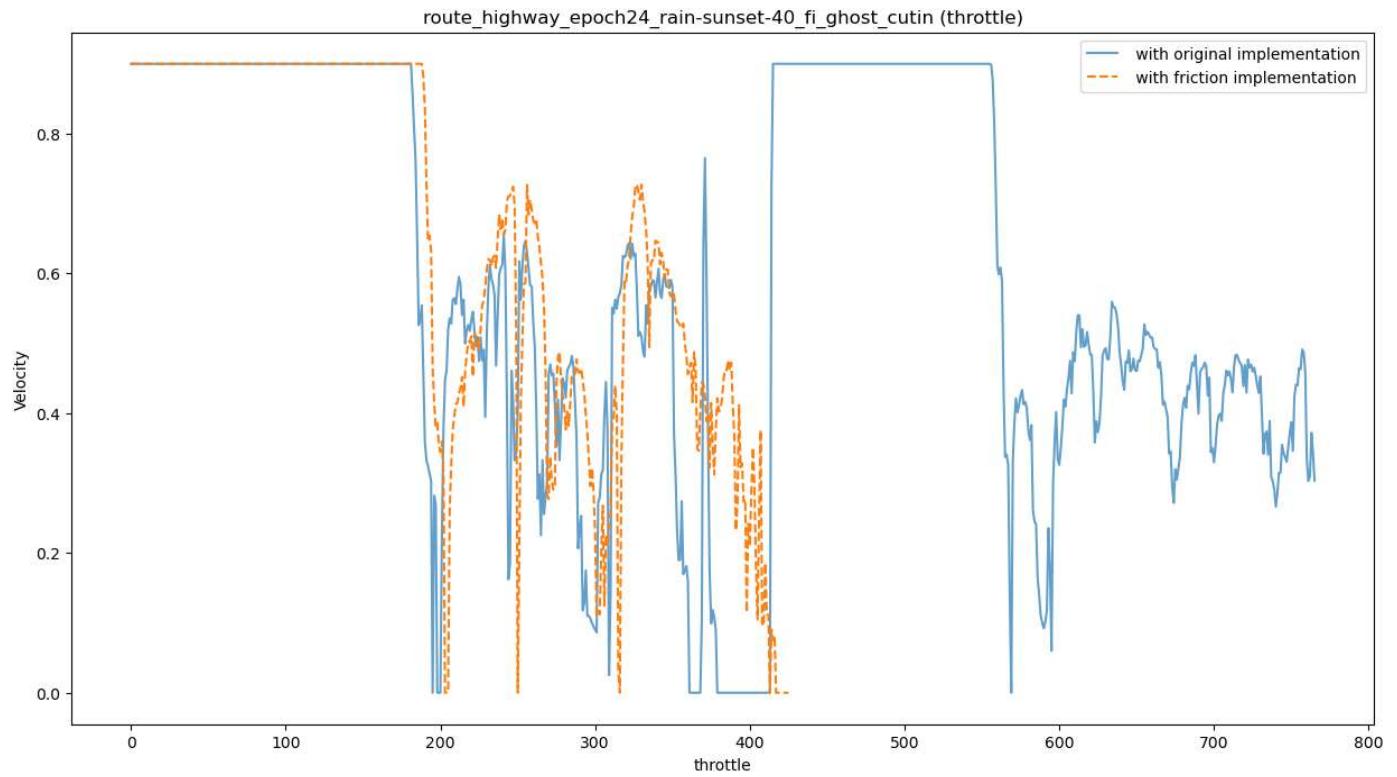
plt.ylabel("Density")
median_orig_throttle.plot.density(alpha=0.7)
median_fric_throttle.plot.density(style='--')
plt.title(subfolders[weather] + " (throttle)")
plt.legend(friction_legend)
plt.show()

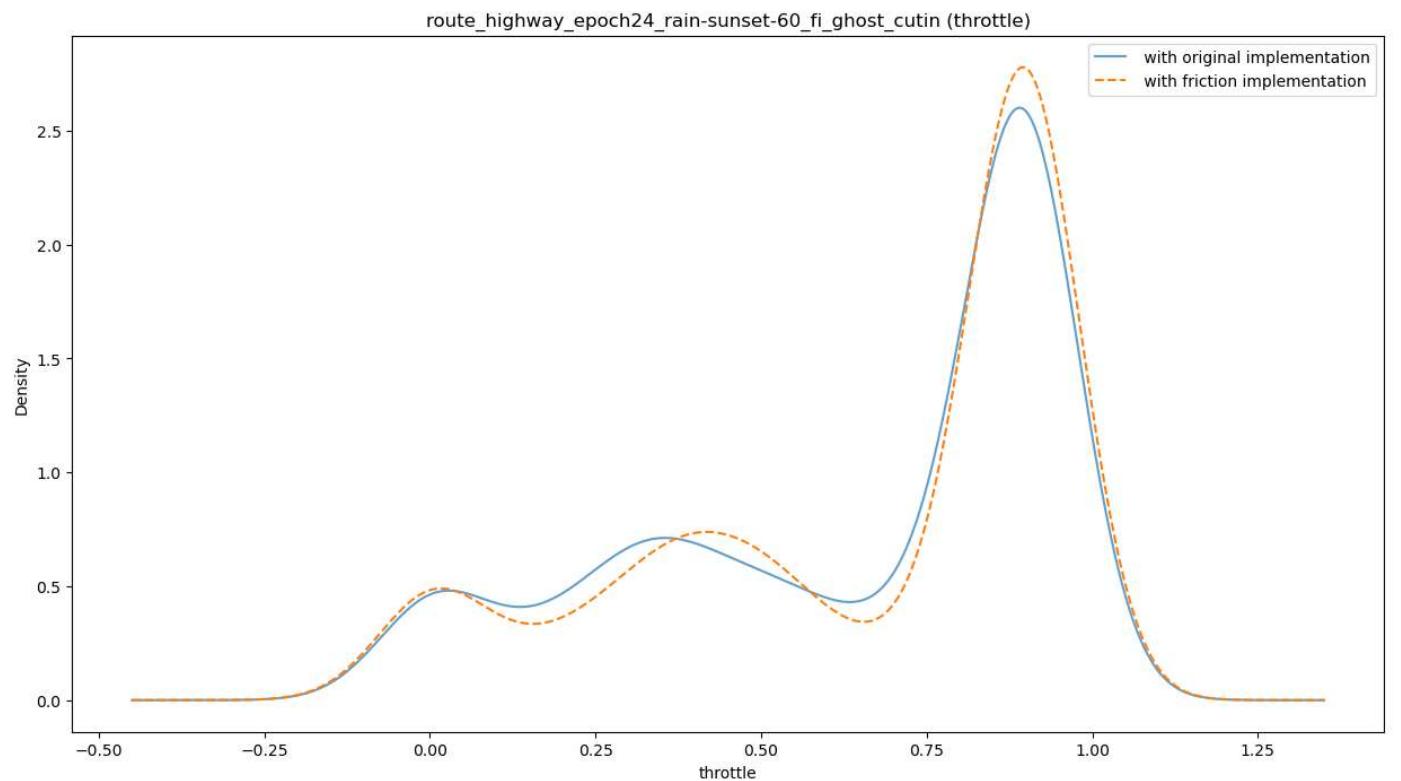
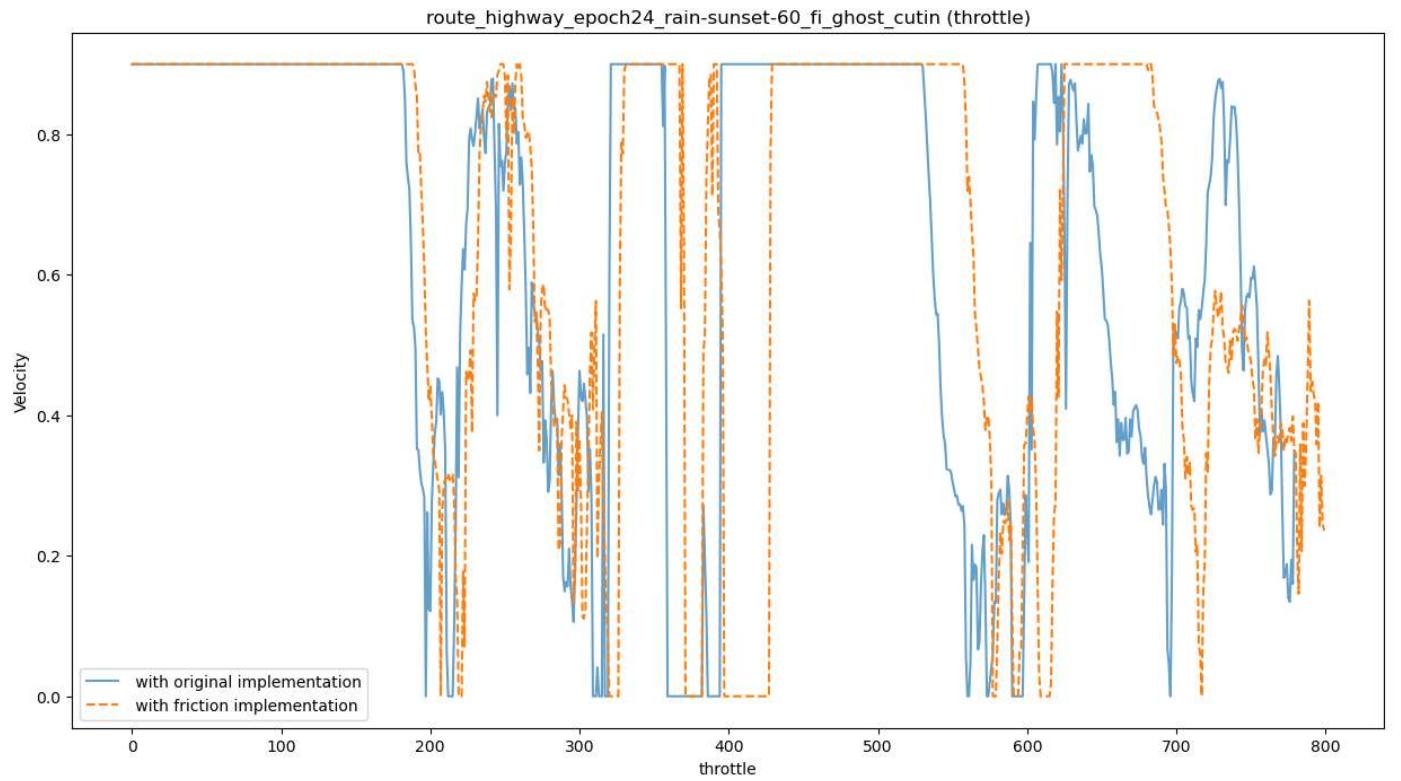
```

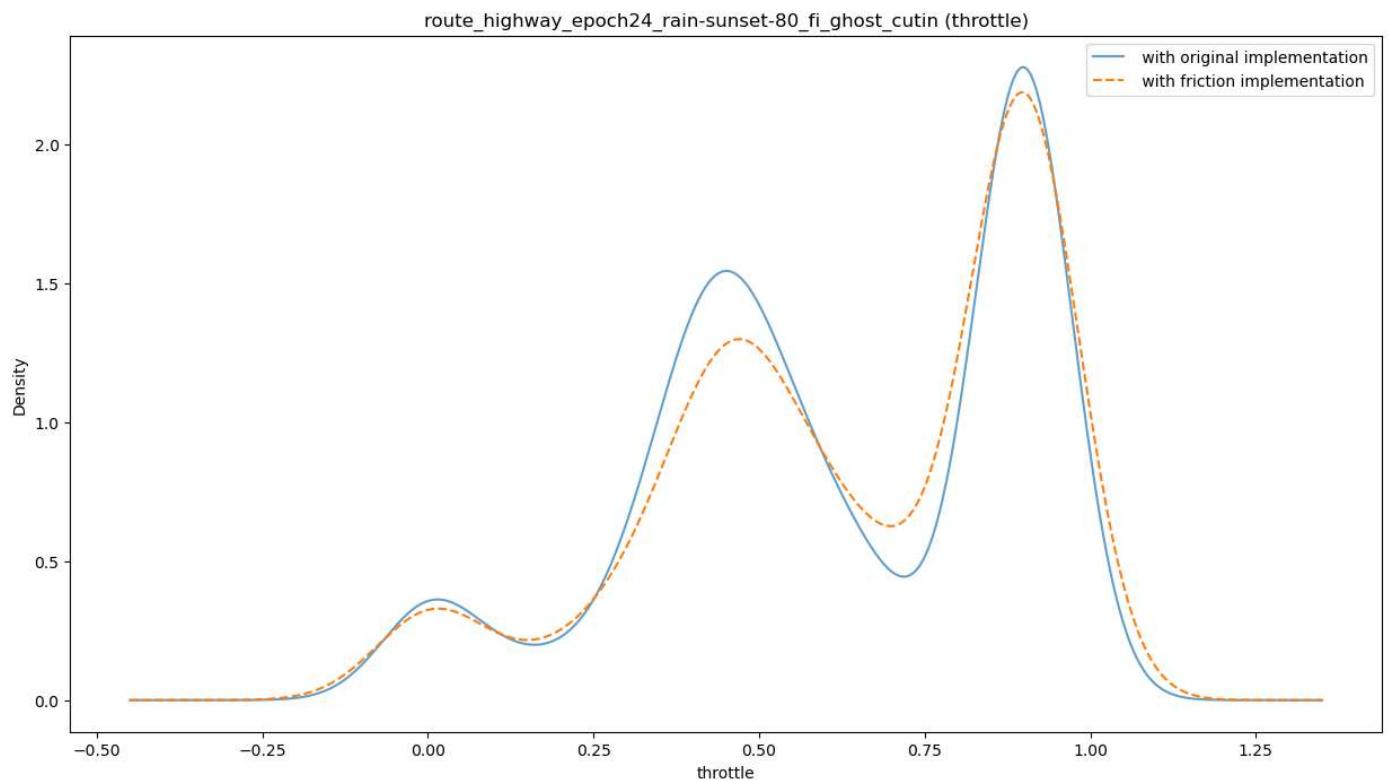
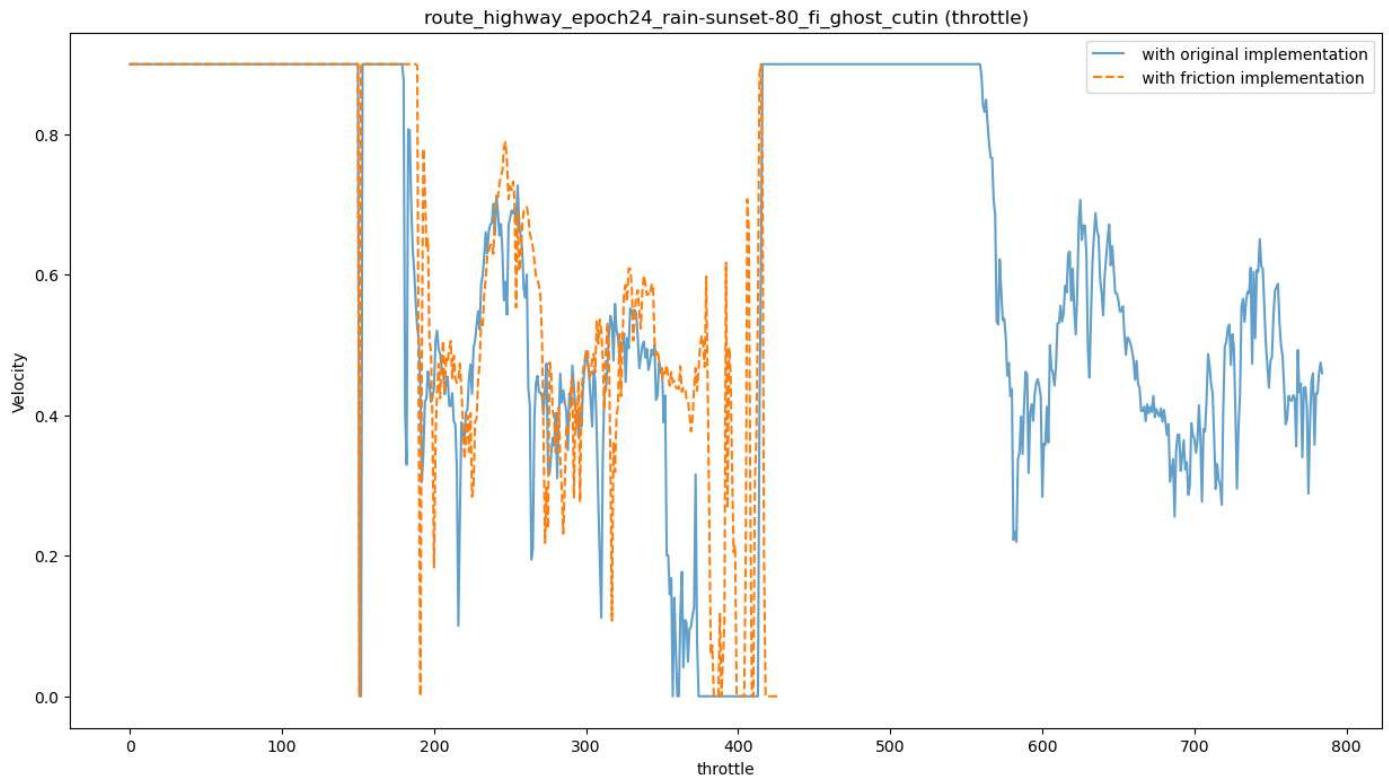












```
In [44]: plt.figure(figsize=(20,10))

for weather in range(len(txt_lists_fric)):

    df_orig_throttle = pd.DataFrame()

    for i in range(0, len(df_array_orig)):
        if(df_array_orig[weather][i]['throttle'].dtypes == 'throttle'):
            print(i)
        else:
            if not df_array_orig[weather][i]['throttle'].isnull().values.any():
                df_orig_throttle["Run"+str(i)] = df_array_orig[weather][i]['throttle']

    median_orig_throttle = df_orig_throttle.median(axis=1)
```

```

plt.plot(median_orig_throttle[200:], alpha=0.7)

df_fric_throttle = pd.DataFrame()

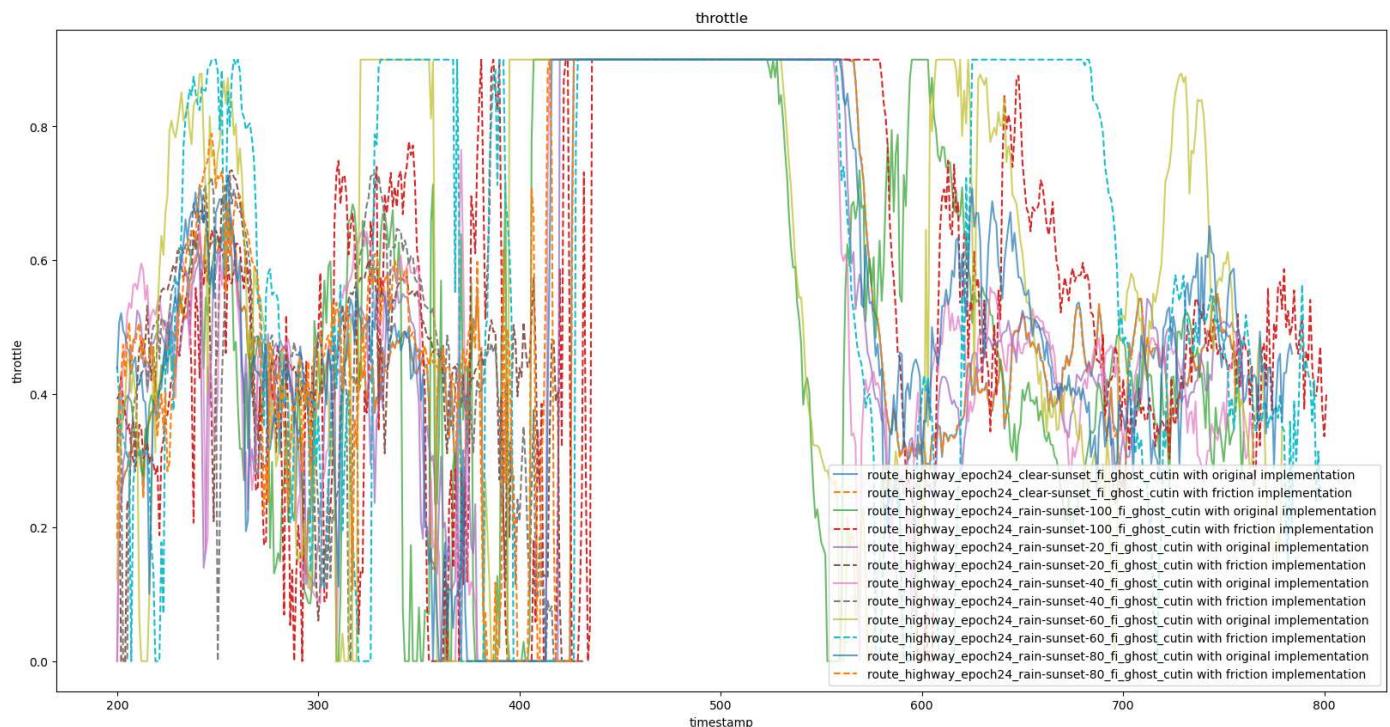
for i in range(0, len(df_array_fric)):
    if(df_array_fric[weather][i]['throttle'].dtypes == 'object'):
        print(i)
    else:
        if not df_array_fric[weather][i]['throttle'].isnull().values.any() :
            df_fric_throttle["Run"+str(i)] = df_array_fric[weather][i]['throttle']

median_fric_throttle = df_fric_throttle.median(axis=1)

plt.plot(median_fric_throttle[200:], '--')

# plt.axis('equal')
plt.xlabel("timestamp")
plt.ylabel("throttle")
plt.title("throttle")
plt.legend(full_legend)
plt.show()

```



Dynamic time warping (DTW) for cvip

measuring similarity between two temporal sequences

```
In [45]: all_dtw = [[0] * len(attribute_order) for i in range(len(txt_lists_fric))]

for i in range(len(txt_lists_fric)):
    for j in range(len(attribute_order)):
        all_dtw[i][j] = dtw.distance(all_medians[i][j][0], all_medians[i][j][1])
```

```
In [46]: df = pd.DataFrame(all_dtw, columns = attribute_order, dtype = float)
df.index = subfolders
df
```

Out[46]:

	x	y	v	cvip	steer	brake	throttle
route_highway_epoch24_clear-sunset_fi_ghost_cutin	0.011677	0.139985	0.011677	0.139985	0.011677	0.139985	0.011677
route_highway_epoch24_rain-sunset-100_fi_ghost_cutin	1.076657	2.284300	1.076657	2.284300	1.076657	2.284300	1.076657
route_highway_epoch24_rain-sunset-20_fi_ghost_cutin	4.020098	562.956380	4.020098	562.956380	4.020098	562.956380	4.020098
route_highway_epoch24_rain-sunset-40_fi_ghost_cutin	4.063439	603.335770	4.063439	603.335770	4.063439	603.335770	4.063439
route_highway_epoch24_rain-sunset-60_fi_ghost_cutin	0.120522	1.368785	0.120522	1.368785	0.120522	1.368785	0.120522
route_highway_epoch24_rain-sunset-80_fi_ghost_cutin	4.119787	650.236137	4.119787	650.236137	4.119787	650.236137	4.119787

KS Test for cvip

In [47]:

```
import scipy as sp
```

In [50]:

```
all_ks = [[0] * len(attribute_order) for i in range(len(txt_lists_fric))]

for i in range(len(txt_lists_fric)):
    for j in range(len(attribute_order)):
        ks_test = sp.stats.ks_2samp(all_medians[i][j][0], all_medians[i][j][1])
        all_ks[i][j] = (ks_test.pvalue, ks_test.statistic)
```

C:\Users\kilob\AppData\Local\Temp\ipykernel_15632\2404621166.py:5: RuntimeWarning: ks_2samp: Exact calculation unsuccessful. Switching to method=asymp.
 ks_test = sp.stats.ks_2samp(all_medians[i][j][0], all_medians[i][j][1])

In [51]:

```
df = pd.DataFrame(all_ks, columns = attribute_order, dtype = float)
df.index = subfolders
df
```

C:\Users\kilob\AppData\Local\Temp\ipykernel_15632\955596543.py:1: FutureWarning: Could not cast to float64, falling back to object. This behavior is deprecated. In a future version, when a dtype is passed to 'DataFrame', either all columns will be cast to that dtype, or a TypeError will be raised.

```
df = pd.DataFrame(all_ks, columns = attribute_order, dtype = float)
```

Out[51]:

	x	y	v
route_highway_epoch24_clear-sunset_fi_ghost_cutin	(0.6354086611071141, 0.03830911492734478)	(1.0, 0.001321003963011889)	(0.6354086611071141, 0.03830911492734478)
route_highway_epoch24_rain-sunset-100_fi_ghost_cutin	(9.209229400339801e-10, 0.16500634091512348)	(0.014993377078013686, 0.07855361596009976)	(9.209229400339801e-10, 0.16500634091512348)
route_highway_epoch24_rain-sunset-20_fi_ghost_cutin	(3.682660381154921e-43, 0.4147556271459455)	(6.94631497741255e-40, 0.39900506851886614)	(3.682660381154921e-43, 0.4147556271459455)
route_highway_epoch24_rain-sunset-40_fi_ghost_cutin	(2.1814333028290965e-47, 0.43610487993233554)	(9.551058731386906e-43, 0.4146900550386742)	(2.1814333028290965e-47, 0.43610487993233554)
route_highway_epoch24_rain-sunset-60_fi_ghost_cutin	(0.0002769209183479568, 0.10541666666666667)	(0.013850797048139457, 0.07875)	(0.0002769209183479568, 0.10541666666666667)
route_highway_epoch24_rain-sunset_80_fi_ghost_cutin	(8.549055067354e-52, 0.1528204547204482)	(4.5435626120388053e-48, 0.4528204547204482)	(8.549055067354e-52, 0.1528204547204482)

You reject the null hypothesis that the two samples were drawn from the same distribution if the p-value is less than your significance level. **pvalue=4.2177381979173086e-06**