ErrorModelling

January 19, 2022

[ ]:
```
import numpy as np
from SandboxSafety.Simulator.Dynamics import update_complex_state
from SandboxSafety.Utils import load_conf
from SandboxSafety.Modes import Modes
```

[ ]:
```
conf = load_conf("forest_kernel")
m = Modes(conf)
resolution = conf.n_dx
phi_range = conf.phi_range
b = 1 / (resolution)
p = conf.phi_range / (conf.n_phi -1)
v = (conf.max_v - conf.min_v) / (conf.nq_velocity - 1)
s = 2 * conf.max_steer / (conf.nq_steer - 1)
time = conf.kernel_time_step

print(f"Limits: b: {b}, p: {p}, v: {v}, s: {s}")
# These Errors are over two when used. This is the full block size, but␣
 ↪theoretically the error will be half this.
```

```
Limits: b: 0.0125, p: 0.078525, v: 0.4, s: 0.08
```

[ ]:
```
b_state = np.array([0, 0, 0, 3.0, 0])
mode_action = np.array([0.4, 2])
# action_id = m.get_mode_id(mode_action[1], mode_action[0])

n_bstate = update_complex_state(b_state, mode_action, time)
dx, dy, phi, vel, steer = n_bstate[0], n_bstate[1], n_bstate[2], n_bstate[3],␣
 ↪n_bstate[4]
bq = m.get_safe_mode_id(vel, steer)
```

[ ]:
```
options = np.array([[b,b,0, 0, 0]
            ,[b,-b,0, 0, 0]
            ,[-b,b,0, 0, 0]
            ,[-b,-b,0, 0, 0]])

for i, opt in enumerate(options):
    state = b_state + options[i]
    new_state = update_complex_state(state, mode_action, time)

    diff = new_state - n_bstate
    print(f"{i} --> State diff: {diff} ")
```

```
0 --> State diff: [0.0125 0.0125 0.     0.     0.    ]
1 --> State diff: [ 0.0125 -0.0125  0.      0.      0.    ]
2 --> State diff: [-0.0125  0.0125  0.      0.      0.    ]
3 --> State diff: [-0.0125 -0.0125  0.      0.      0.    ]
```

[ ]:

```
options = np.array([[0,0,p, 0, 0]
               ,[0,0,-p, 0, 0]])


for i, opt in enumerate(options):
    state = b_state + options[i]
    new_state = update_complex_state(state, mode_action, time)

    diff = new_state - n_bstate
    print(f"{i} --> State diff: {diff} ")
```

```
0 --> State diff: [ 0.02332614 -0.00176401  0.078525    0.          0.        ]
1 --> State diff: [-2.33926346e-02 -7.12277208e-05 -7.85250000e-02
0.00000000e+00
  0.00000000e+00]
```

[ ]:

```
# v = 0.2
# s = 0.04
# s = 0

options = np.array([[0,0,0, v, s]
        ,[0,0,0, -v, s]
        ,[0,0,0, v, -s]
        ,[0,0,0, -v, -s]])

for i, opt in enumerate(options):
    state = b_state + options[i]
    new_state = update_complex_state(state, mode_action, time)

    diff = new_state - n_bstate
    print(f"{i} --> State diff: {diff} ")
    qid = m.get_mode_id(state[3], state[4])
    print(f"bq: {bq}, Qid: {qid} ")
```

```
0 --> State diff: [0.01640173 0.04202359 0.10532685 0.32323983 0.03225118]
bq: None, Qid: 30
1 --> State diff: [ 0.00555213 -0.03490462  0.05054712 -0.16977026  0.03225118]
bq: None, Qid: 24
```

```
2 --> State diff: [-0.00936037  0.04380295 -0.06379811  0.32323983 -0.08      ]
bq: None, Qid: 28
3 --> State diff: [-0.00982992 -0.03407627 -0.08000426 -0.16977026 -0.08      ]
bq: None, Qid: 22
```

```
[ ]:
```

The results show that using a lot more modes (37) leads to significantly better results.