

# Performance\_prediction

August 17, 2023

```
[1]: import P3_seenovate as p3
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('data/2019_data.csv', delimiter=",")
df["date"] = pd.to_datetime(df["date"], format = "%Y-%m-%d")
df["athlete"] = df["athlete"] - 1
np.random.seed(123)

# fix an outlier:
df.loc[(df["athlete"] == 3) & (df["date"] == "2019-09-06"), "performance"] = 0

mods = []
for i in range(9):
    mods.append({})
    mods[i]["Fitness"] = p3.Fitness (df.loc[df["athlete"] == i,:].copy(),
                                     objective = "min",
                                     HL_fitness = 30)
    mods[i]["Banister"] = p3.Banister(df.loc[df["athlete"] == i,:].copy(),
                                     objective = "min",
                                     HL_fitness = 30,
                                     HL_fatigue = 2)

athletes_id = [0,1,2,3,4,7,8] # 6 and 7 removed: not enough data for cross val

[ ]: n_runs = 10
for i in athletes_id:
    mods[i]["Fitness"].fitFreq(n_runs = n_runs,
                              bounds = {"a0": (-1e4,1e4),
                                         "k" : (-1,0),
                                         "tauG": (0,100)},
                              n_augment = 0,
                              fix_tau = True,
                              options = {"disp": False})
    mods[i]["Banister"].fitFreq(n_runs = n_runs,
                                bounds = {"a0": (-1e4,1e4),
                                           "k" : (-1,0),
```

```

        "tauG": (0,100),
        "tauH": (0,100)},
    n_augment = 0,
    fix_tau = True,
    options = {"disp": False})

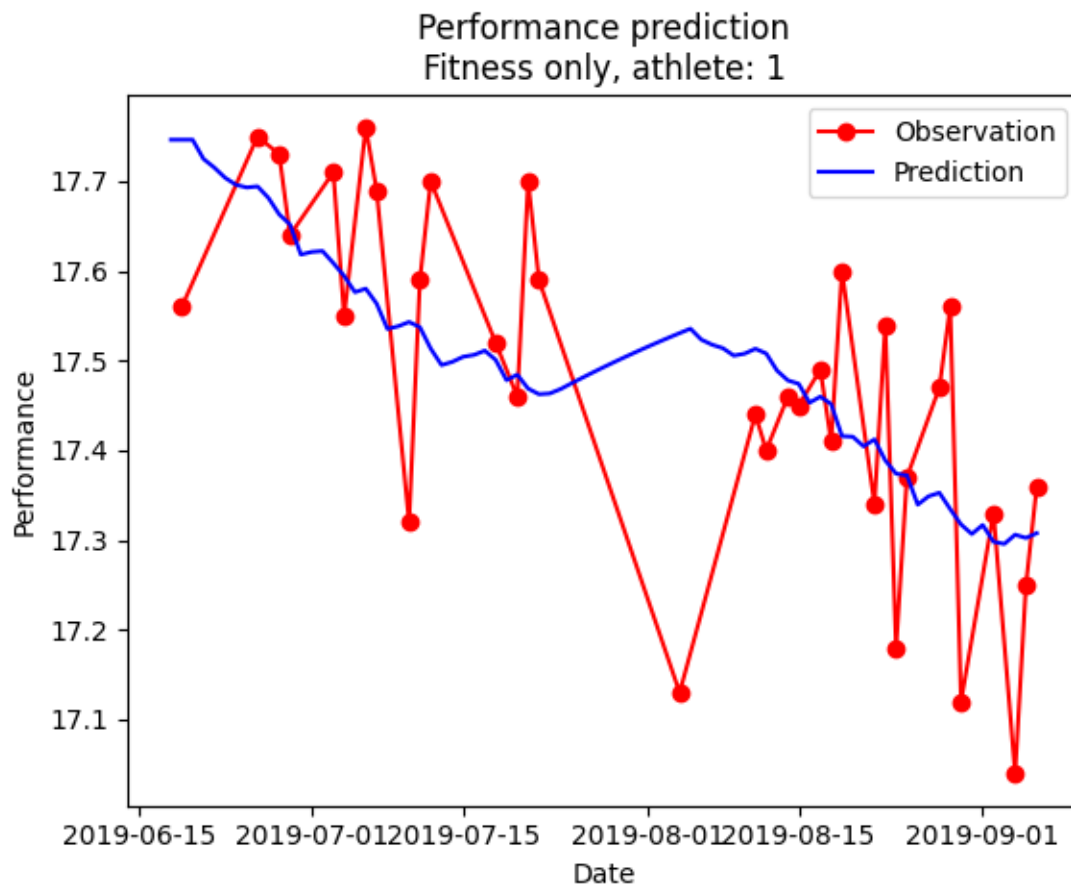
```

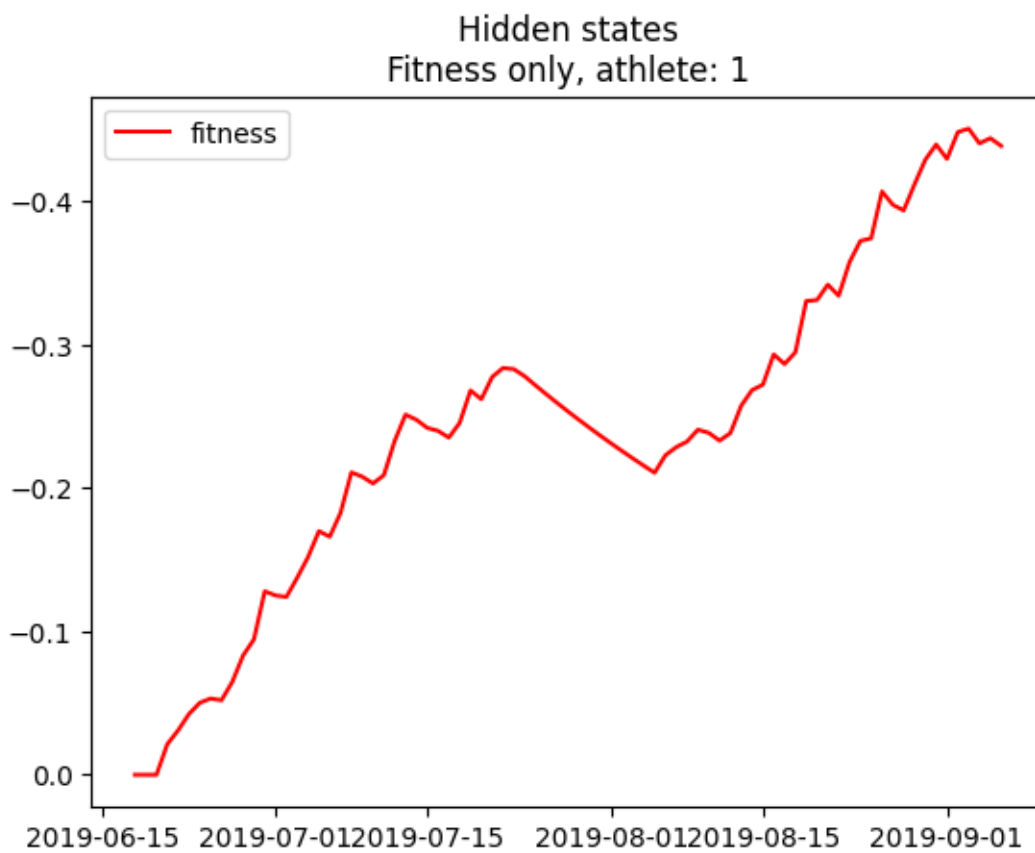
```

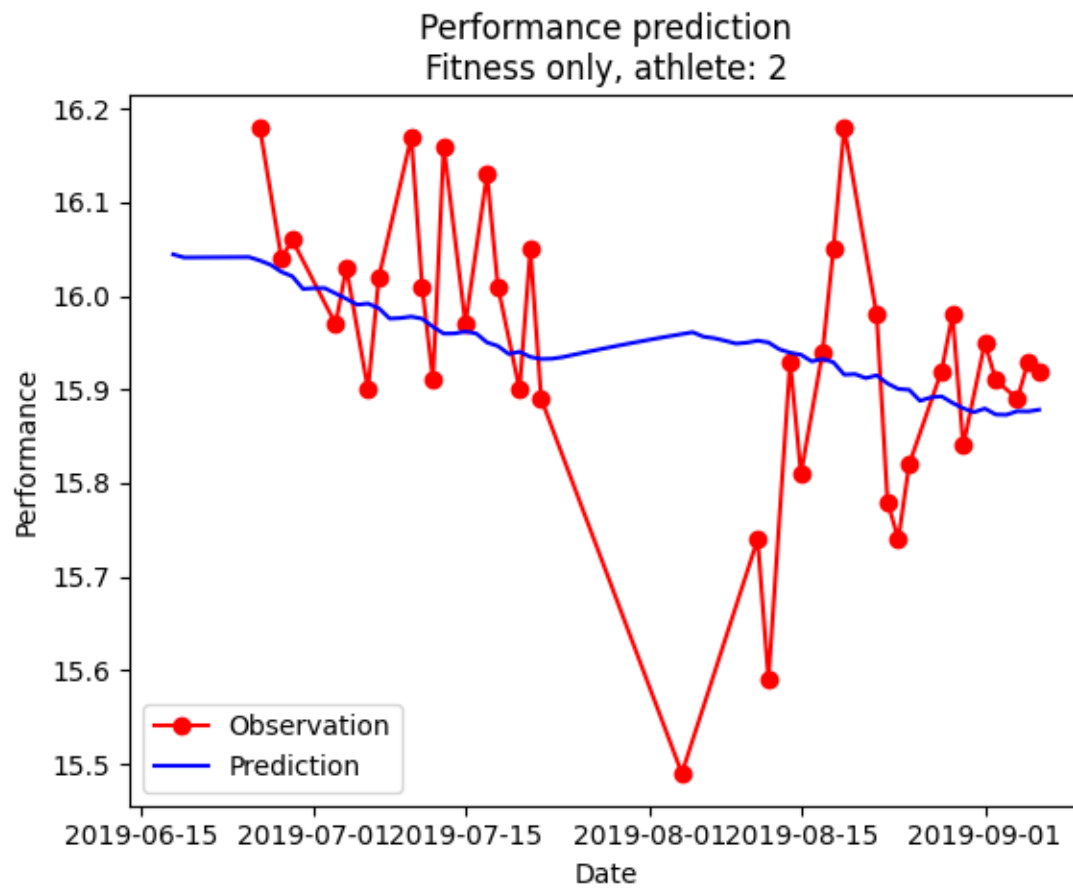
[3]: for i in athletes_id:
    mods[i]["Fitness"].train.plot_perf(show = False)
    plt.title("Performance prediction\nFitness only, athlete: "+str(i+1))
    plt.legend()
    plt.show()

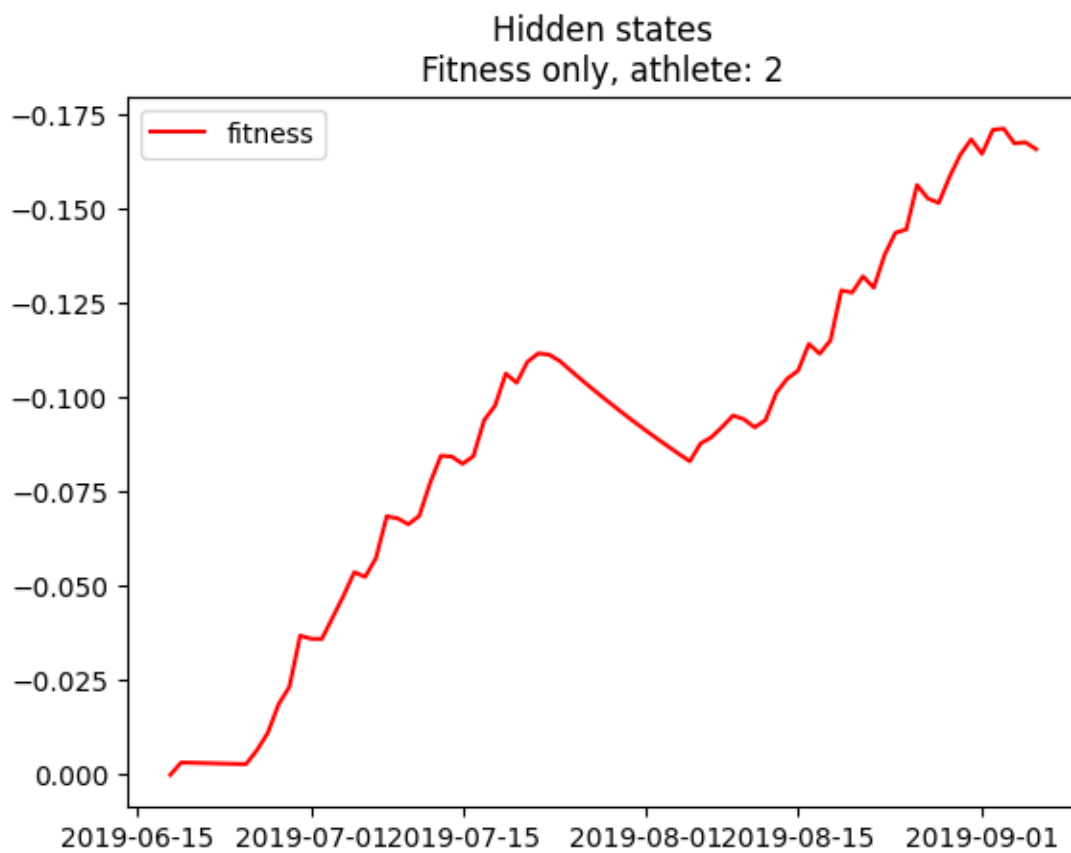
    mods[i]["Fitness"].train.plot_state(show = False)
    plt.title("Hidden states\nFitness only, athlete: "+str(i+1))
    plt.legend()
    plt.show()

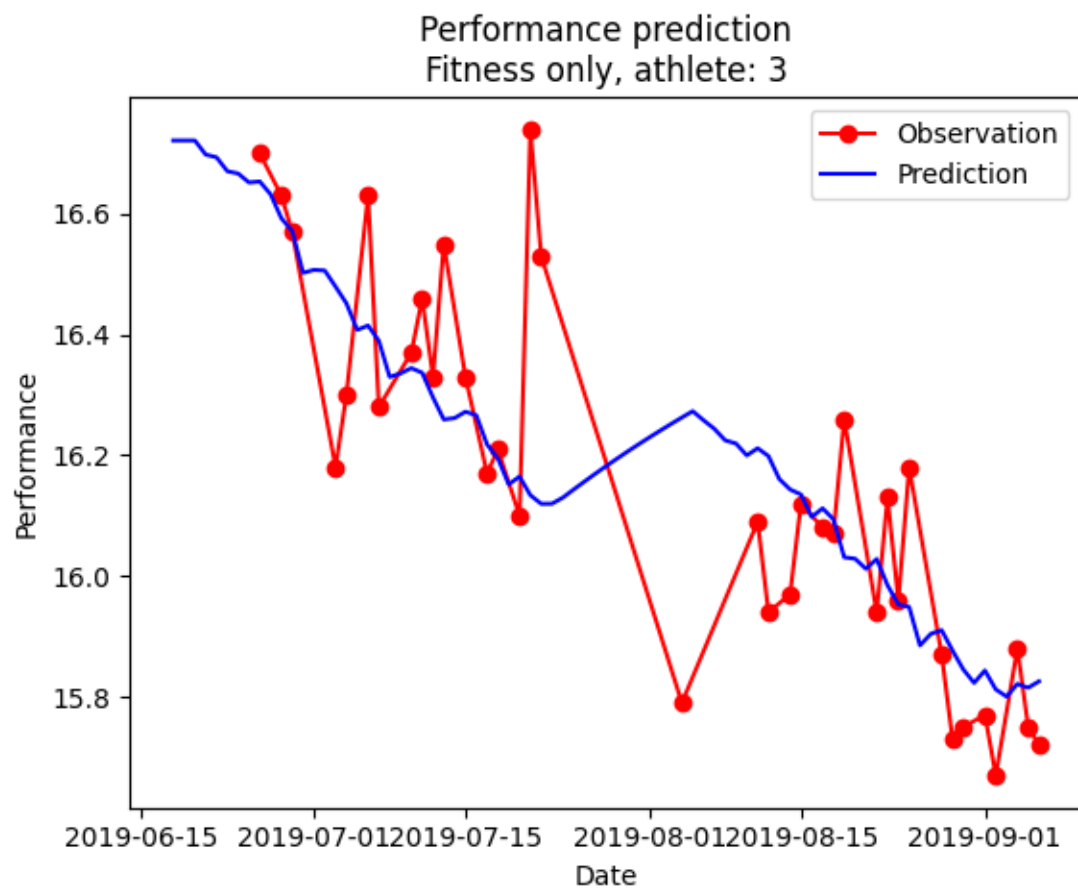
```



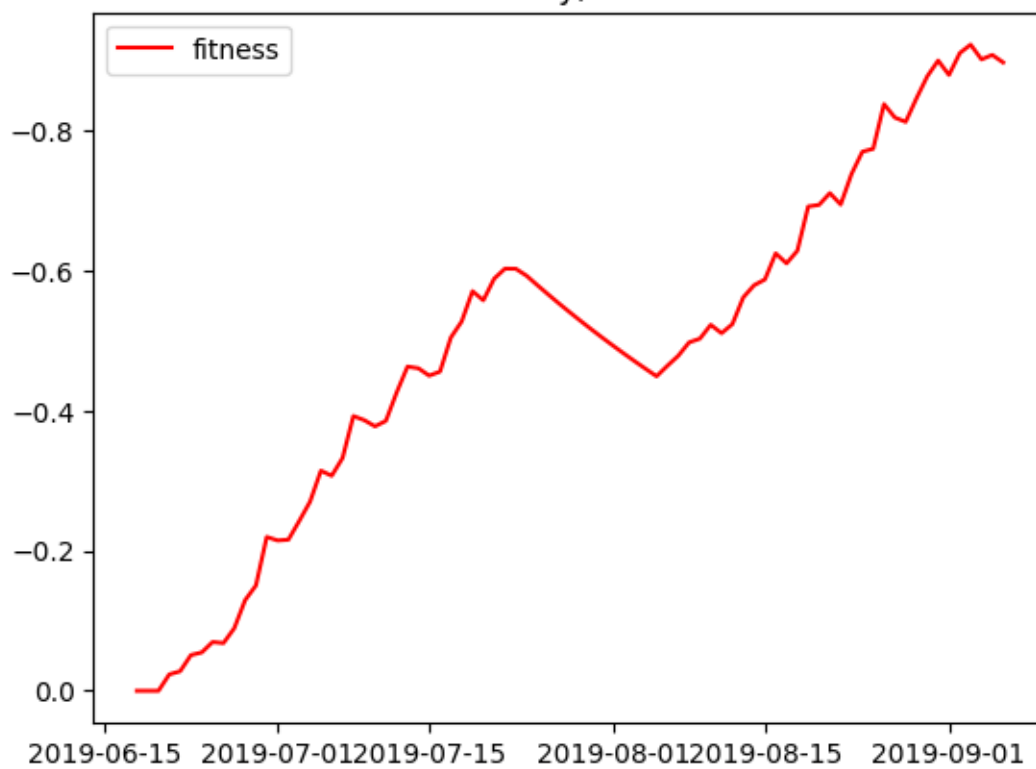


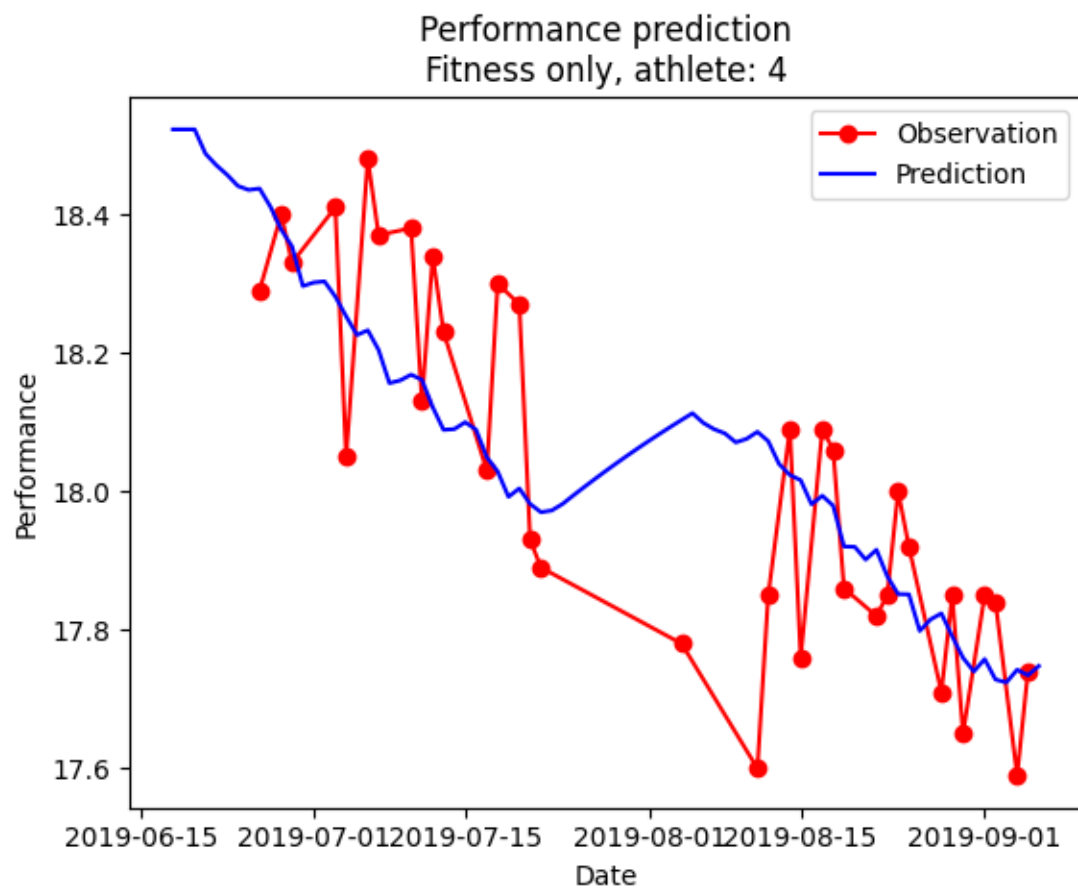




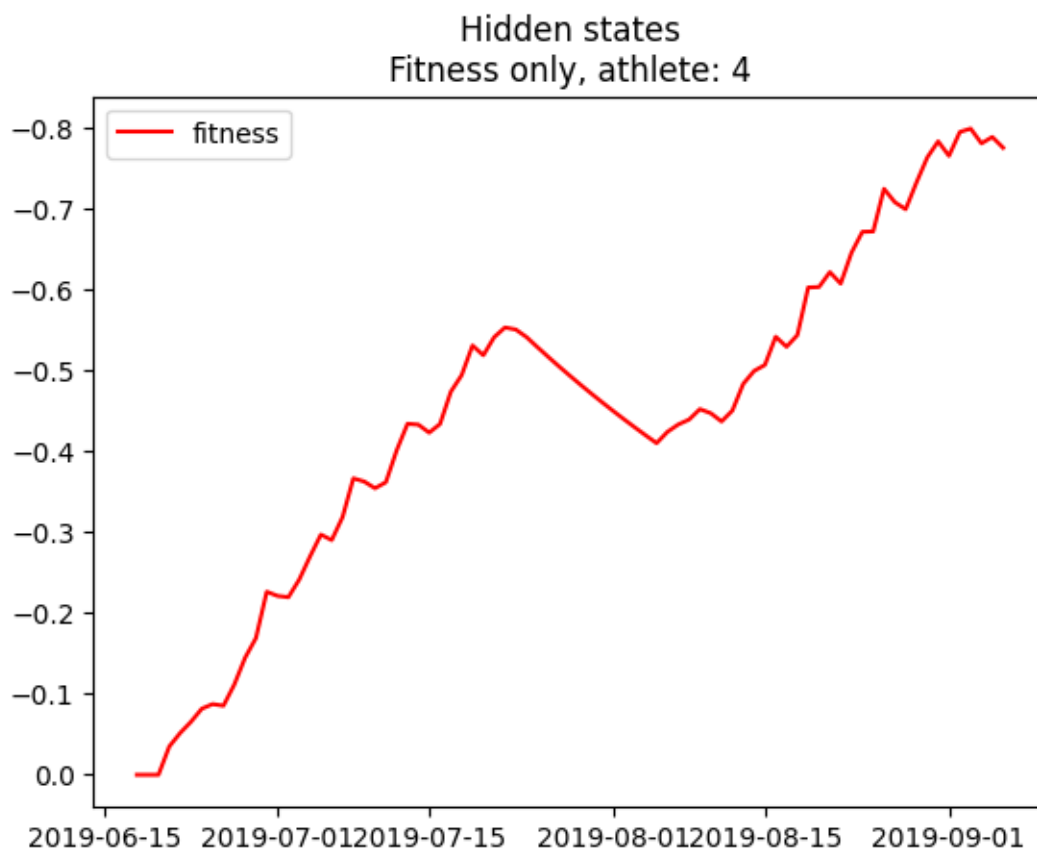


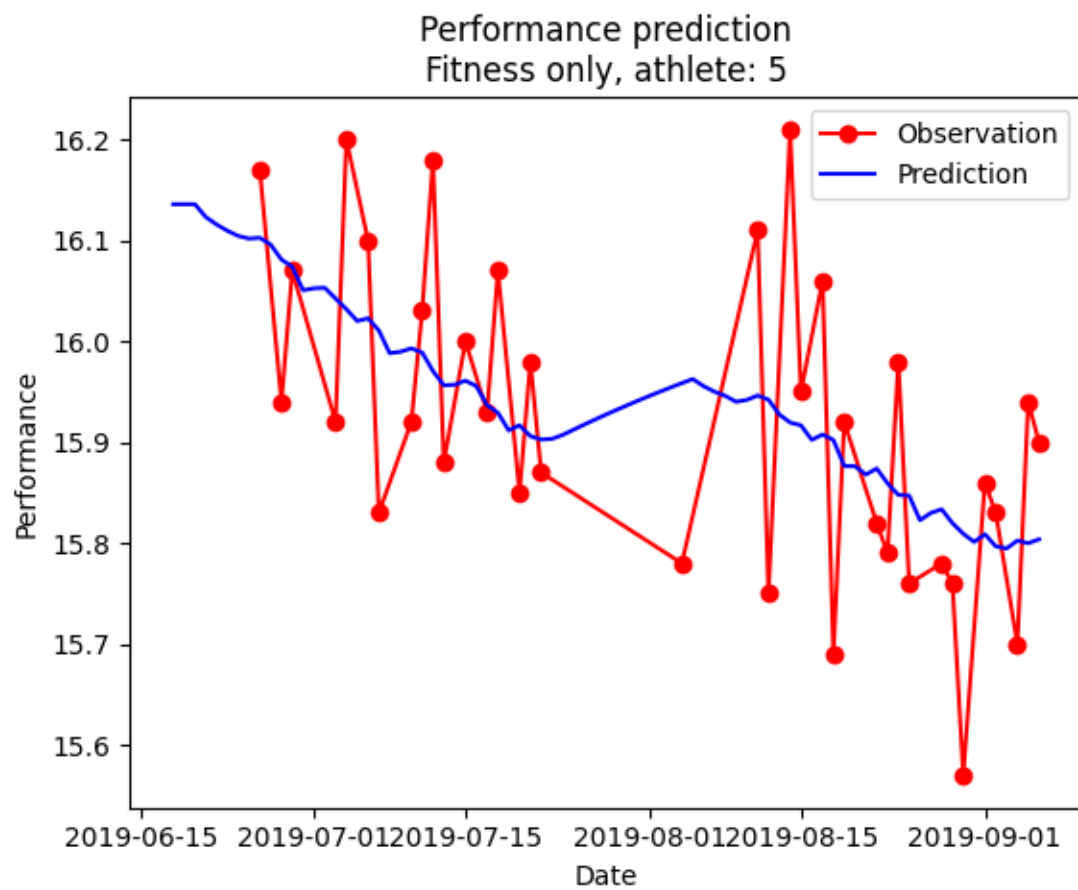
Hidden states  
Fitness only, athlete: 3

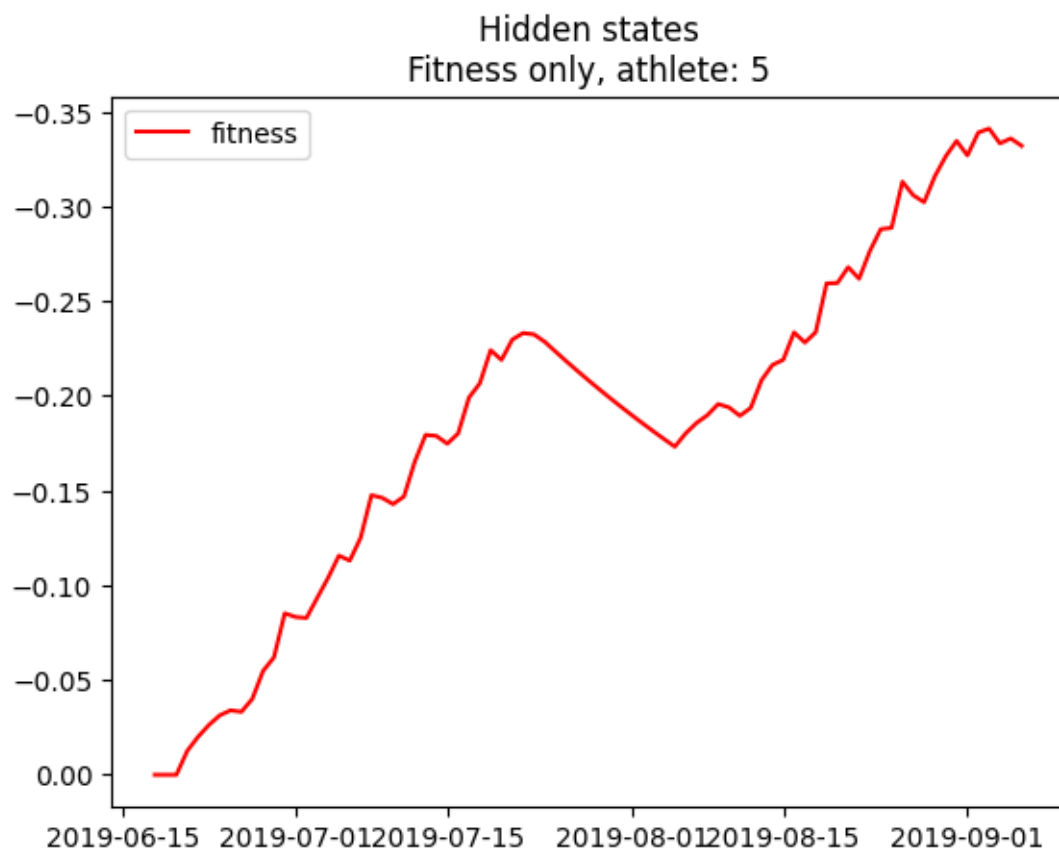


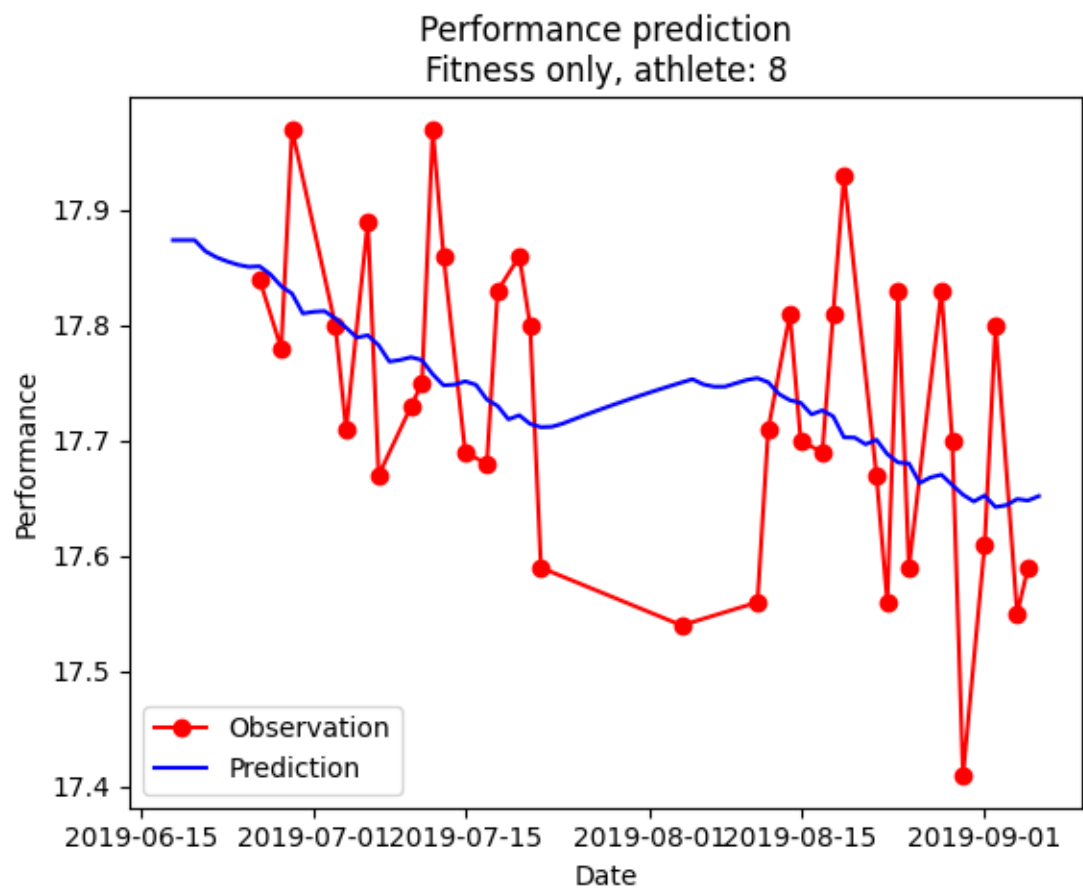


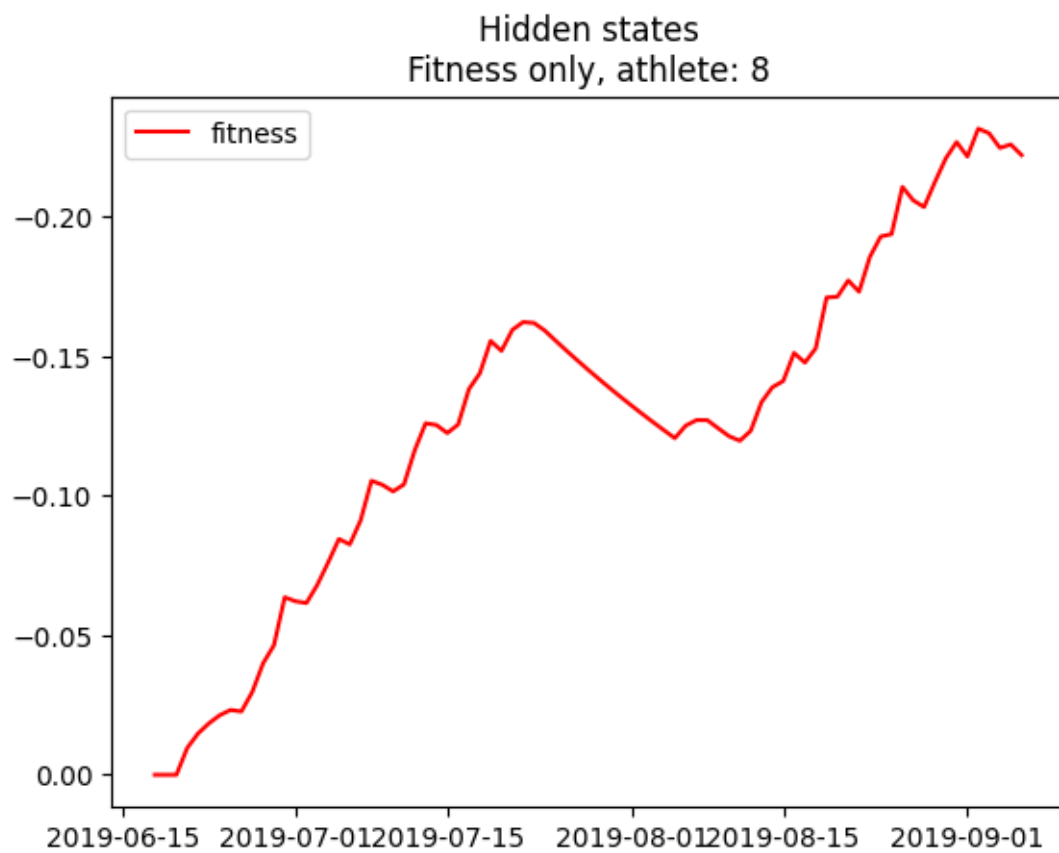


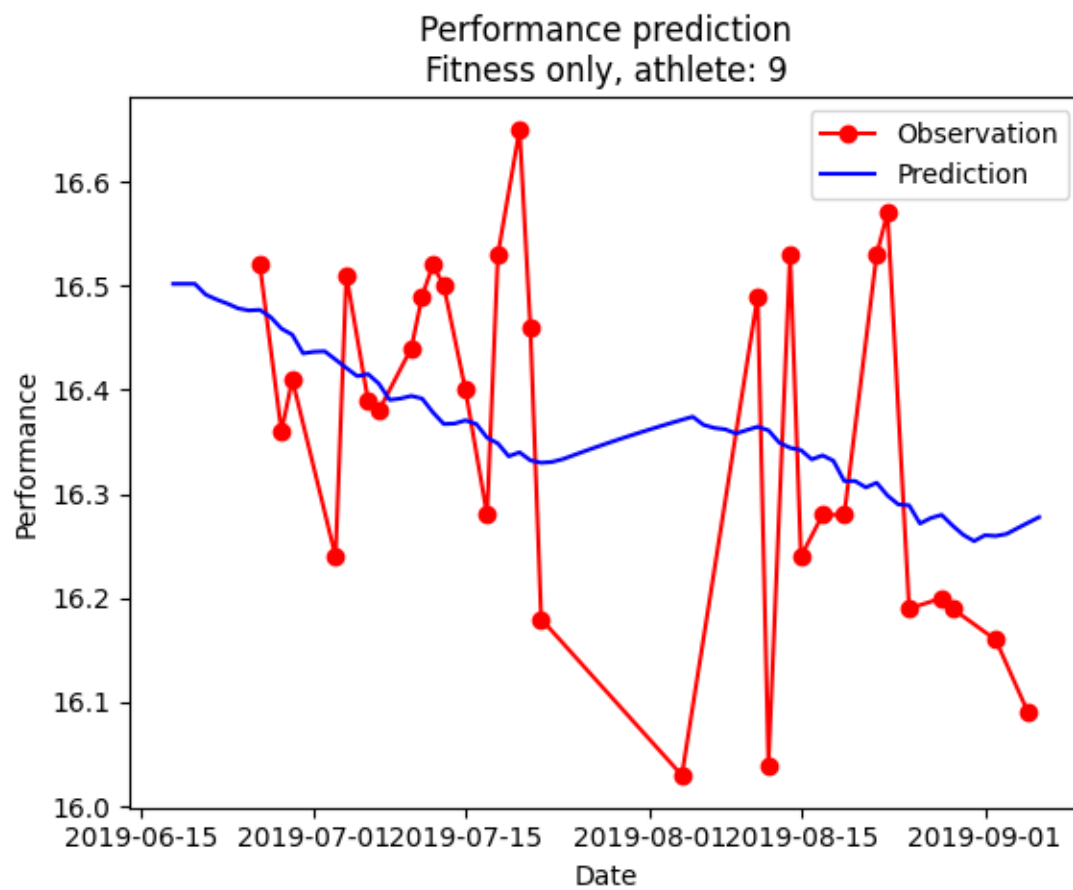


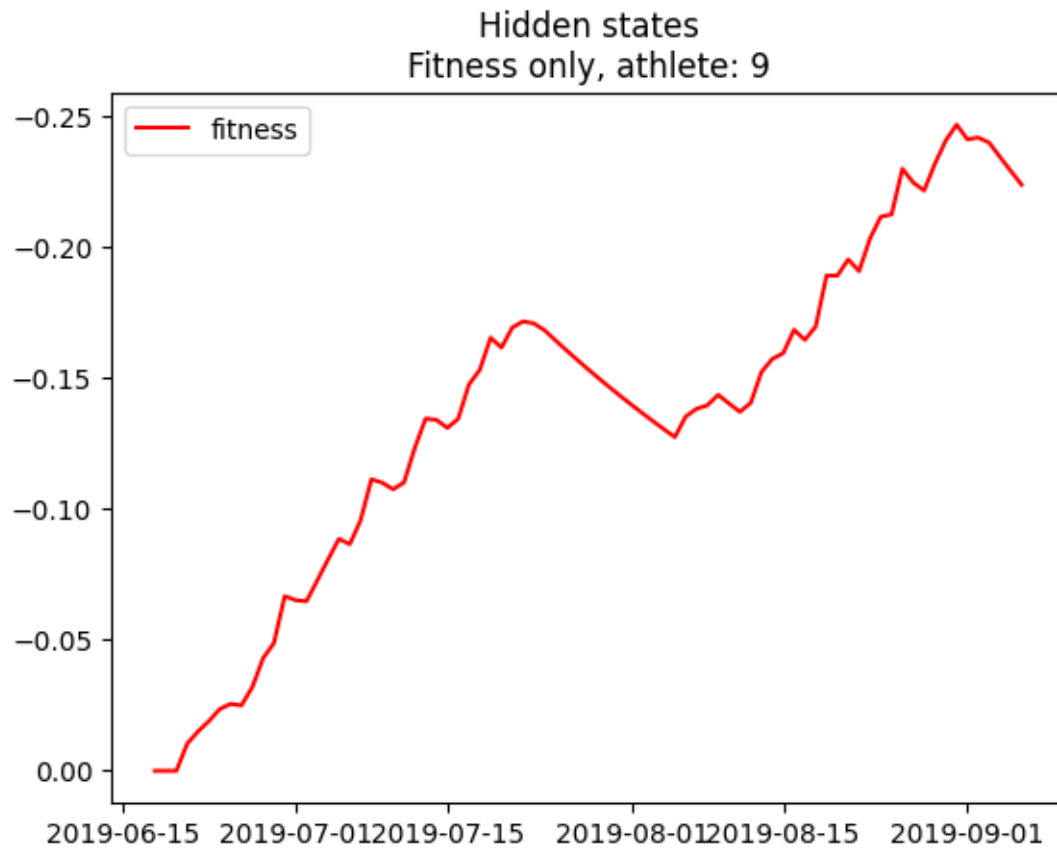






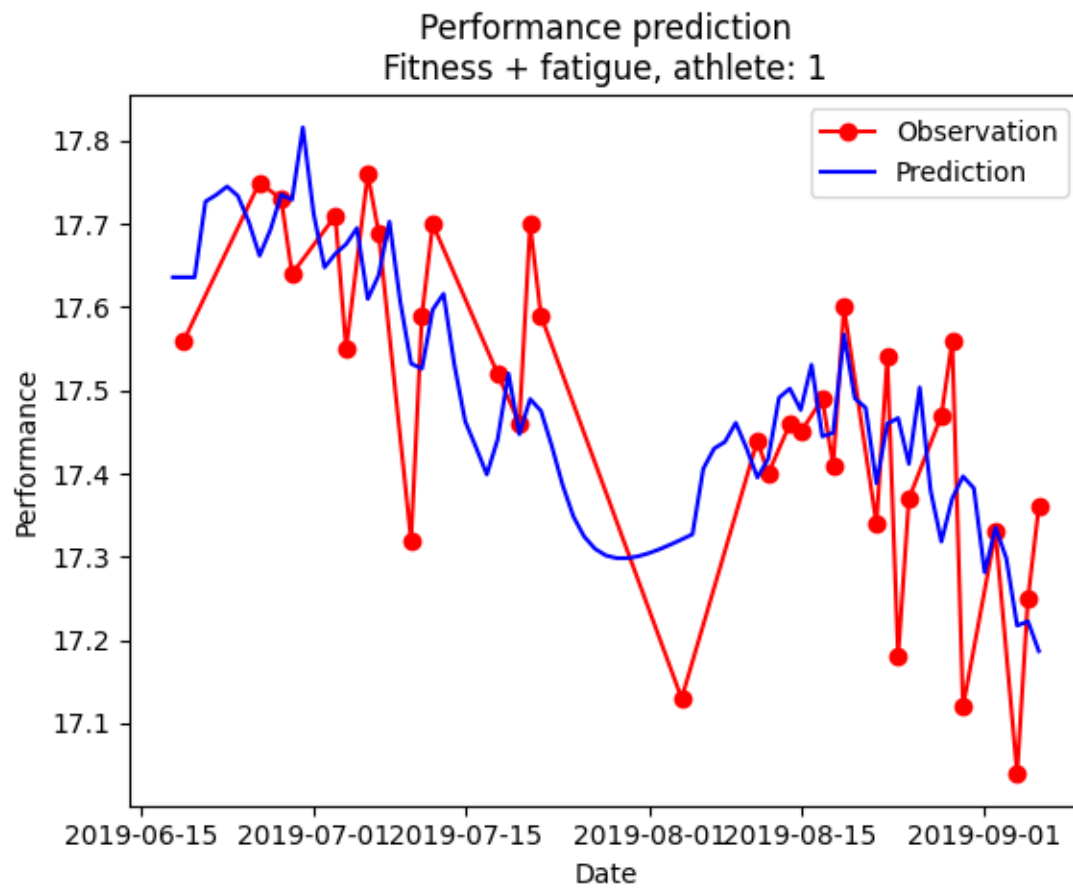






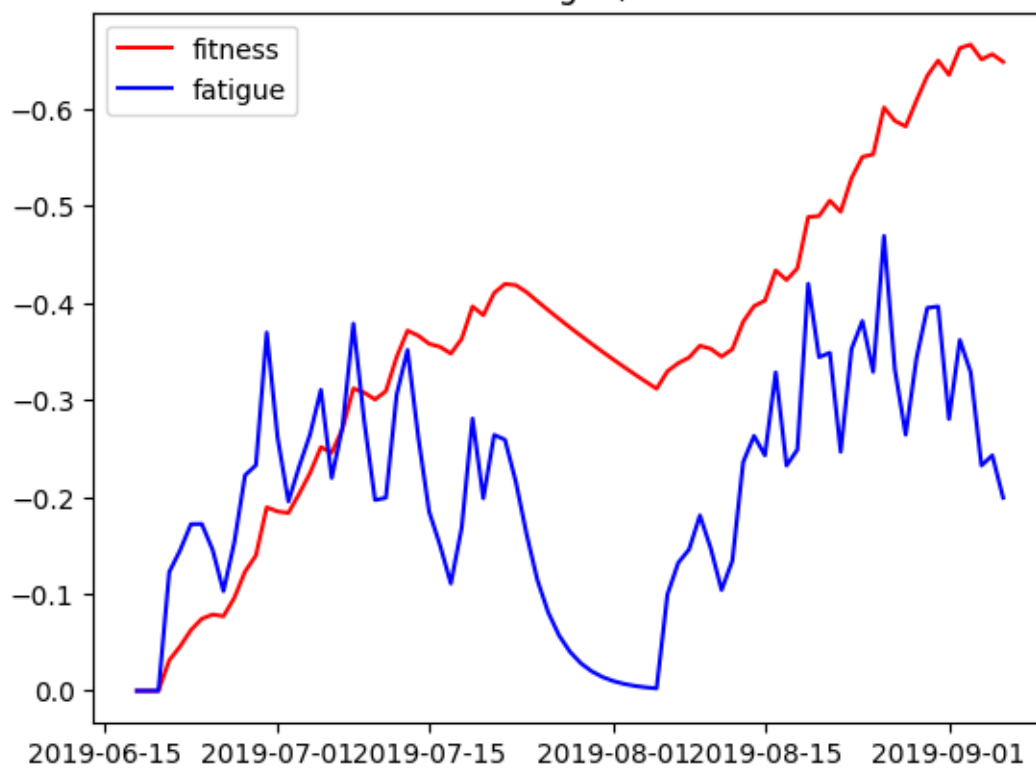
```
[4]: for i in athletes_id:
      mods[i]["Banister"].train.plot_perf(show = False)
      plt.title("Performance prediction\nFitness + fatigue, athlete: "+str(i+1))
      plt.legend()
      plt.show()

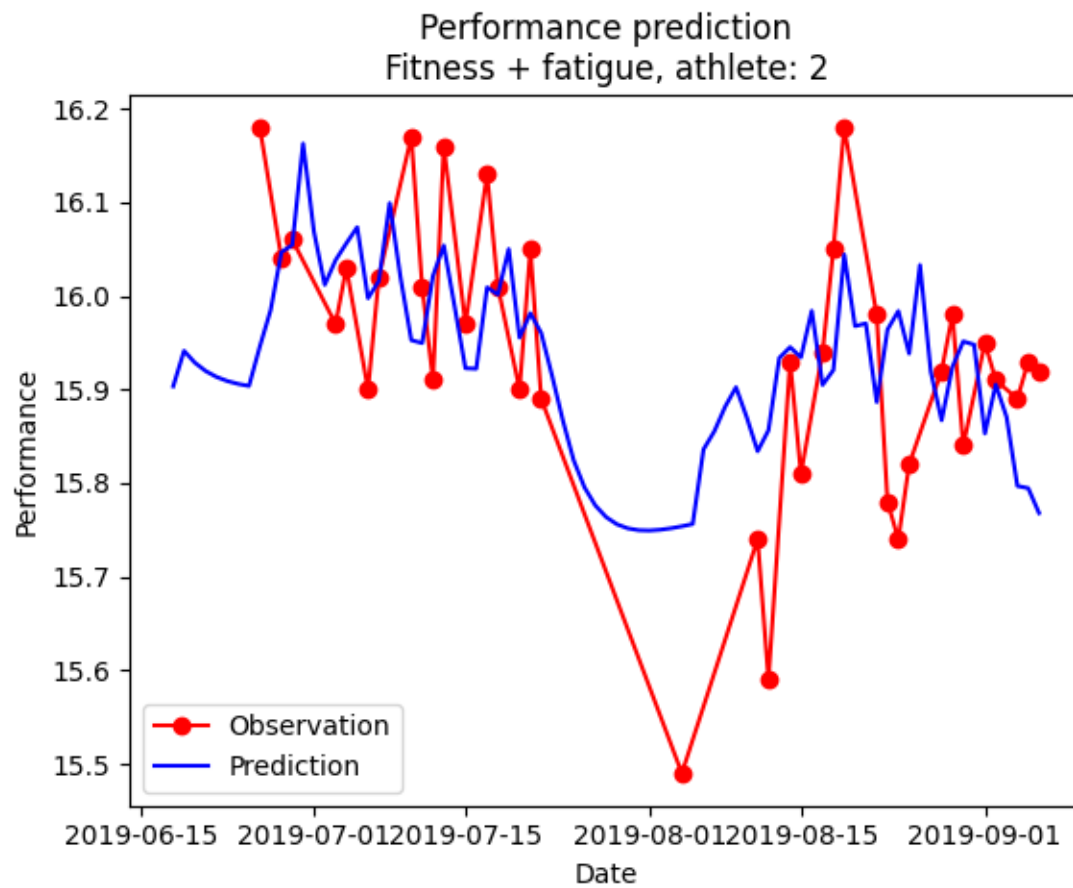
      mods[i]["Banister"].train.plot_state(show = False)
      plt.title("Hidden states\nFitness + fatigue, athlete: "+str(i+1))
      plt.legend()
      plt.show()
```

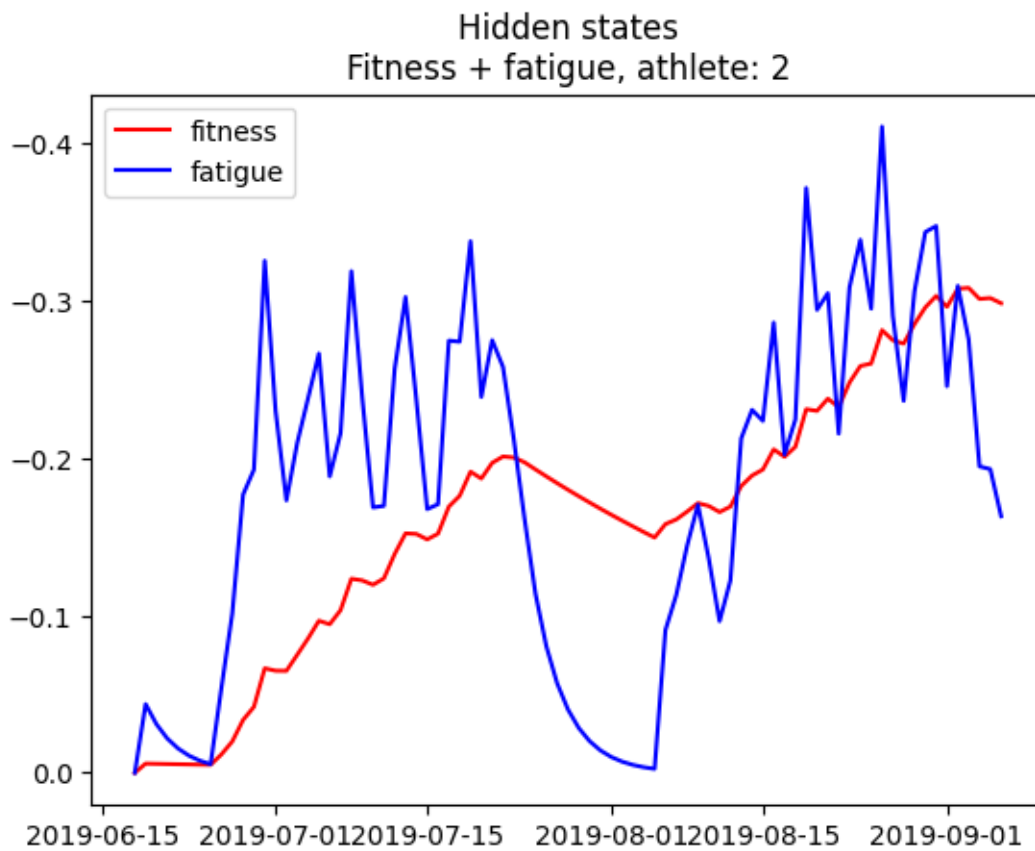


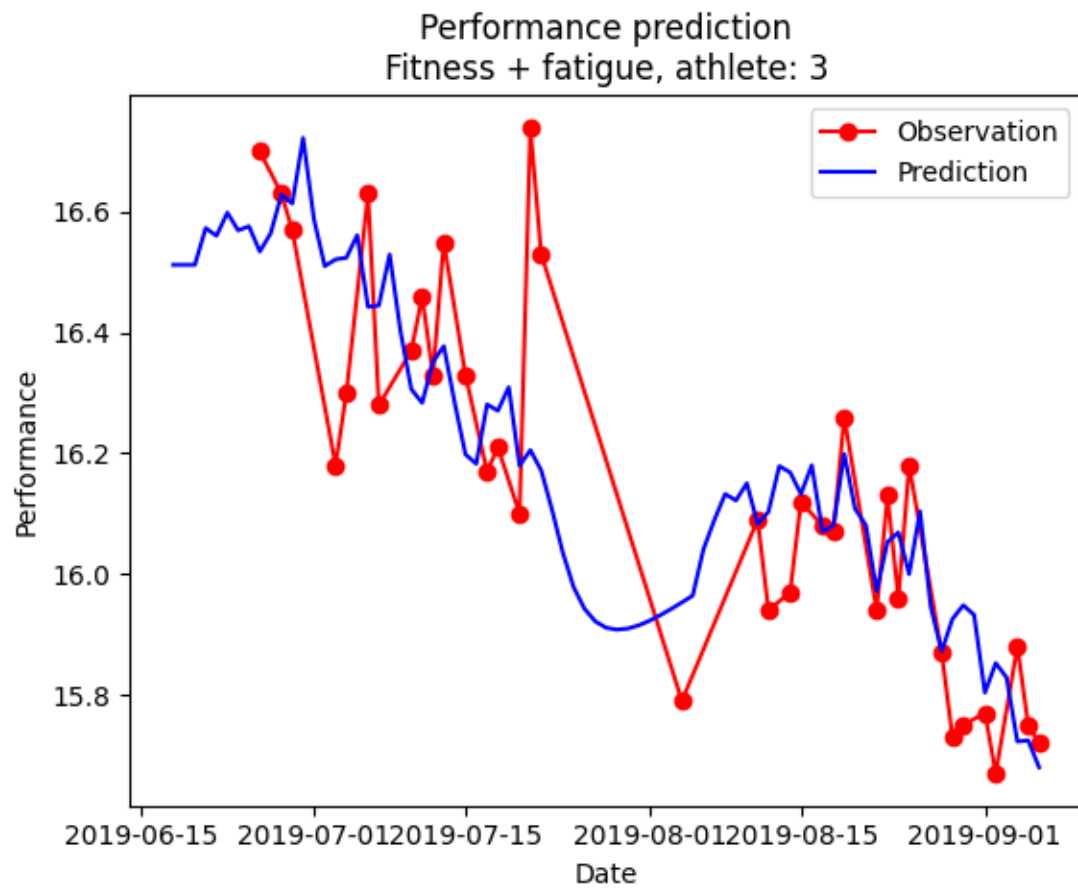


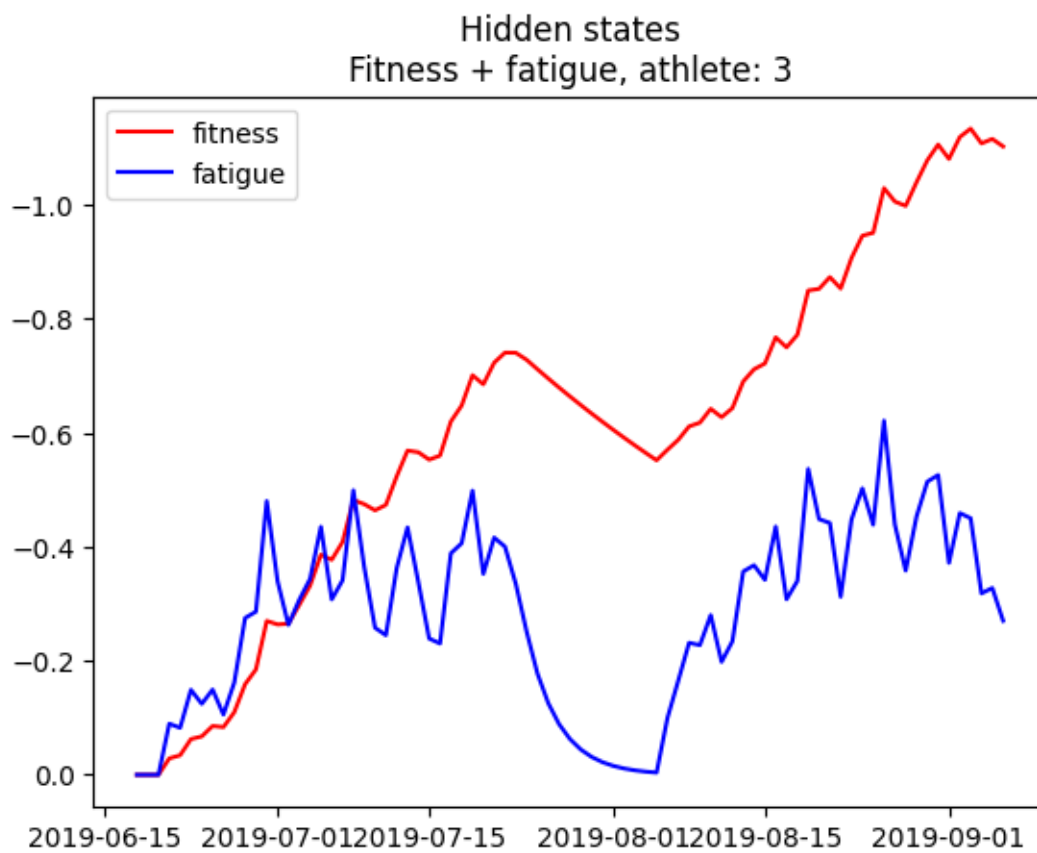
Hidden states  
Fitness + fatigue, athlete: 1



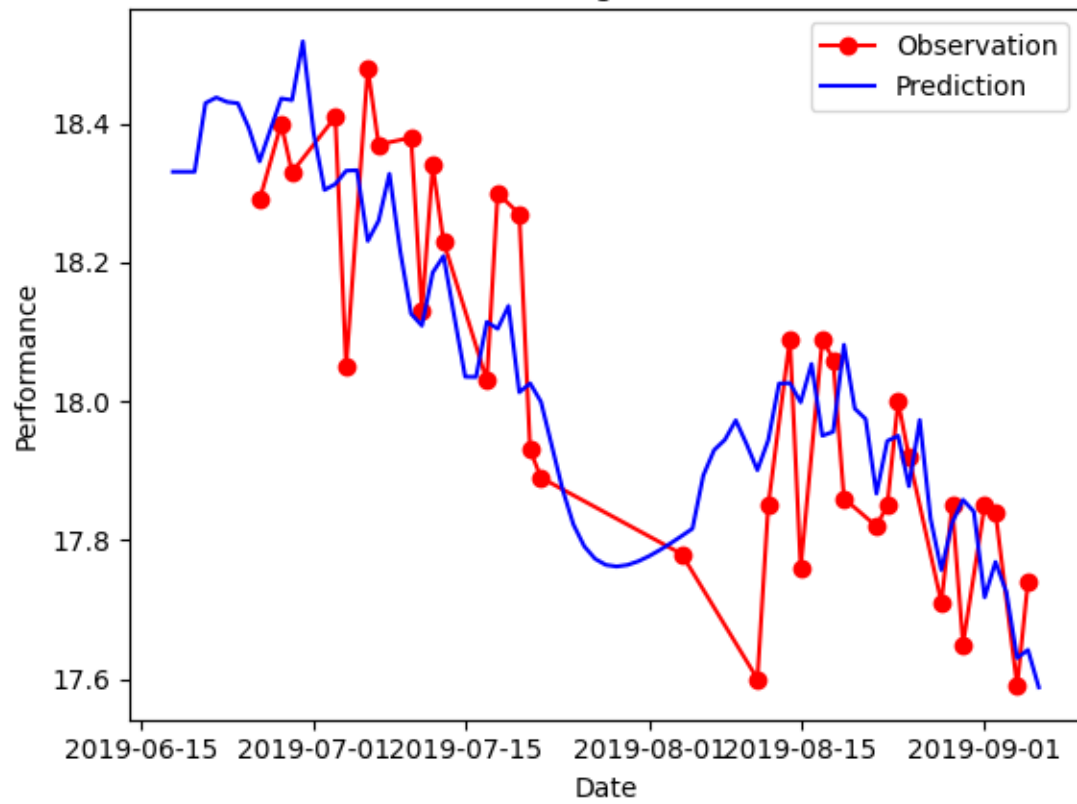


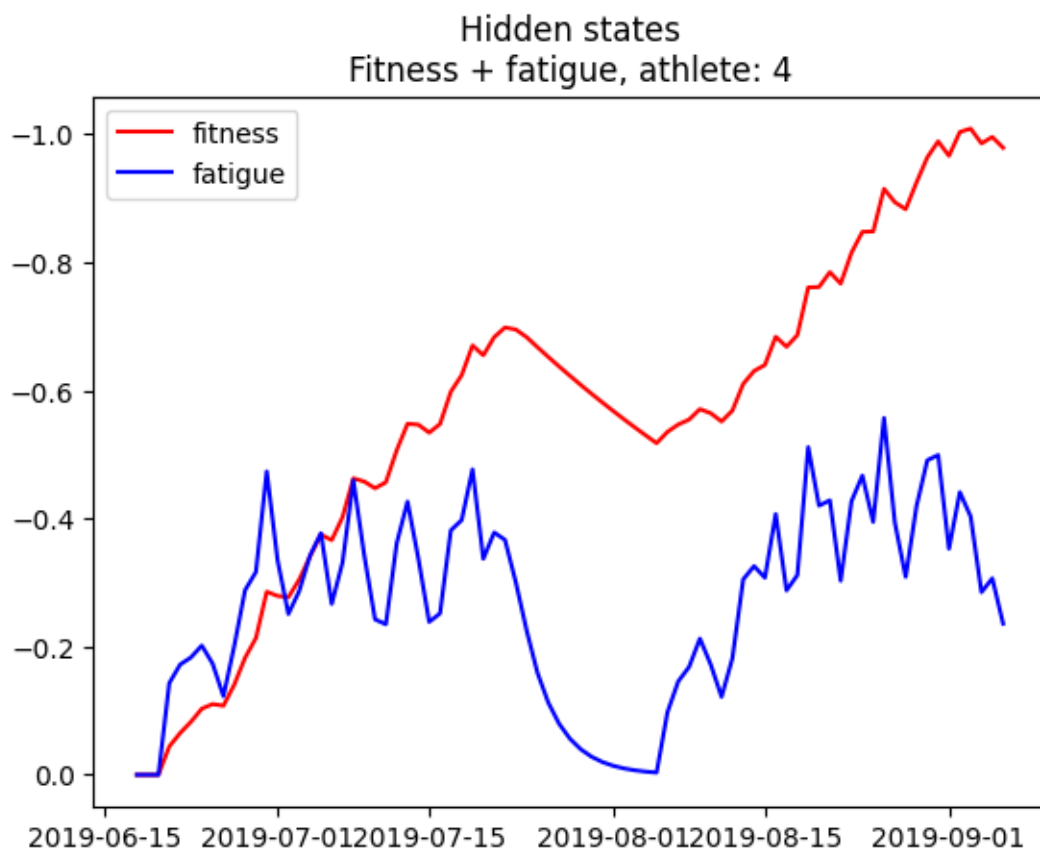


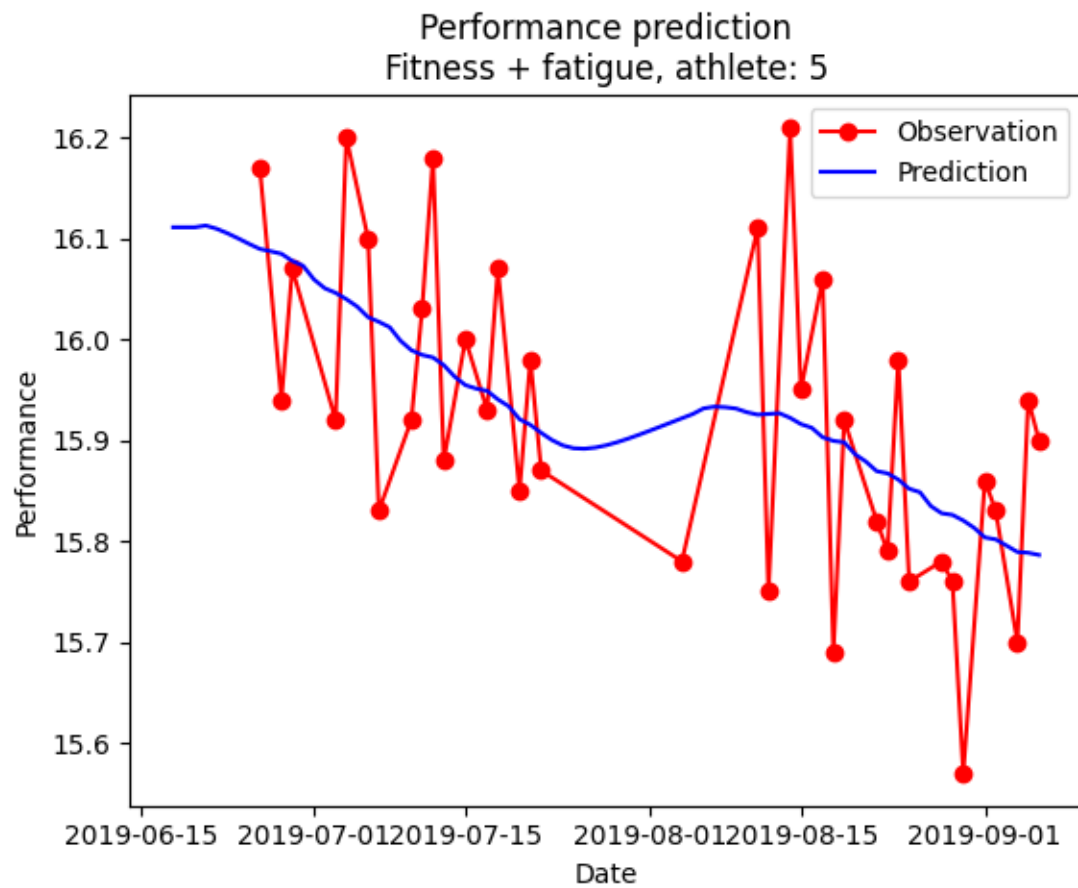




Performance prediction  
Fitness + fatigue, athlete: 4

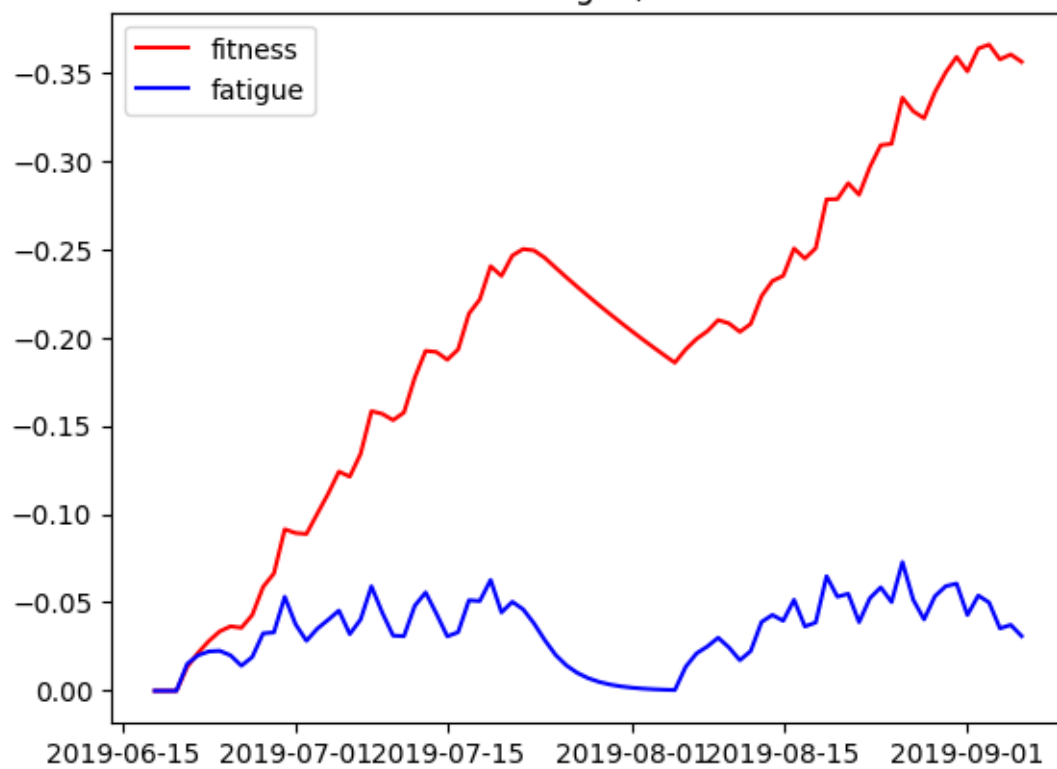


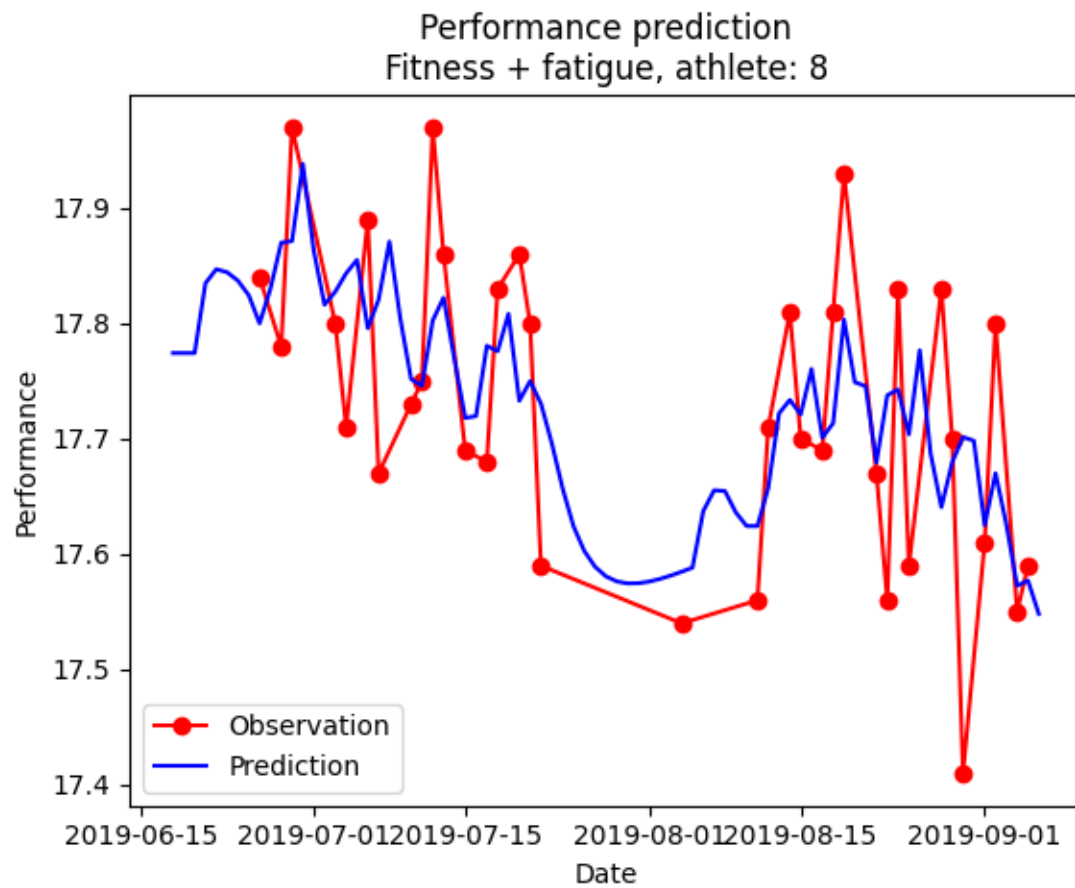




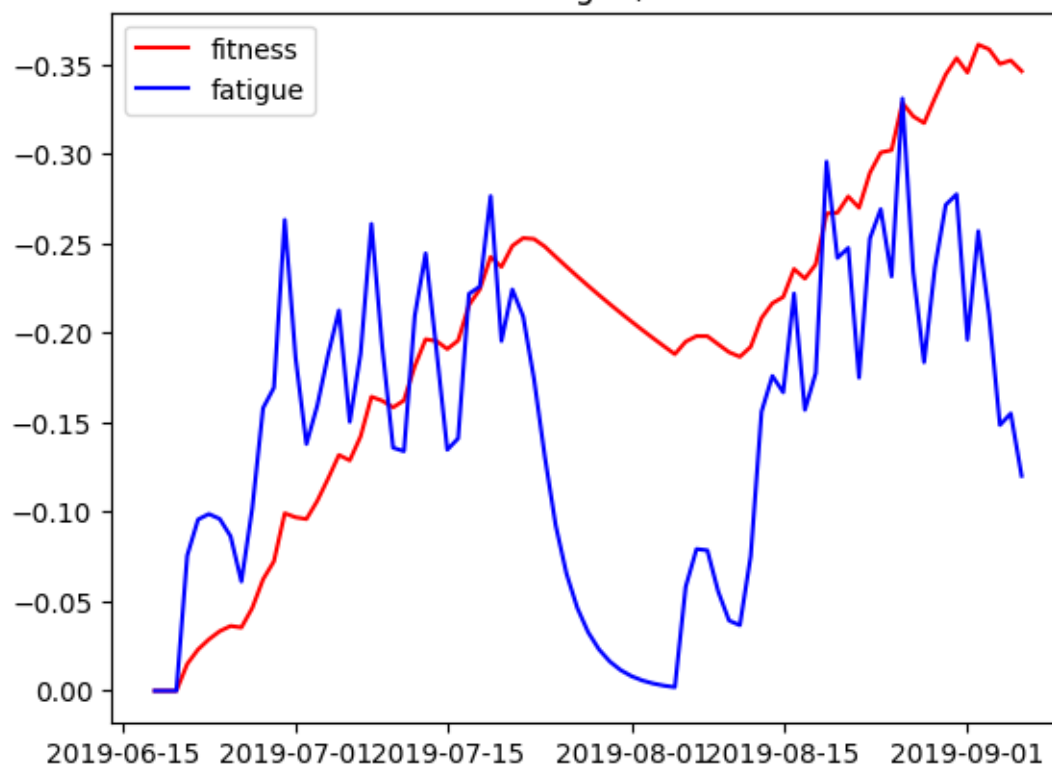


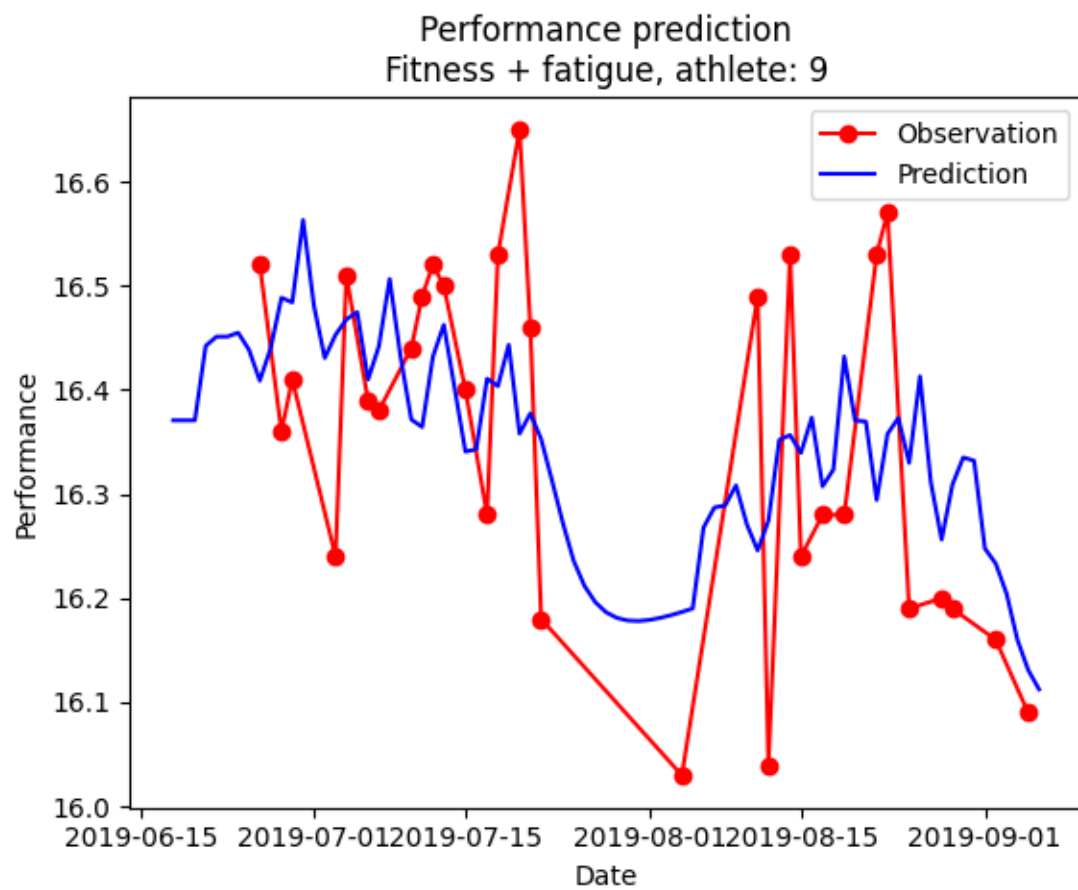
Hidden states  
Fitness + fatigue, athlete: 5

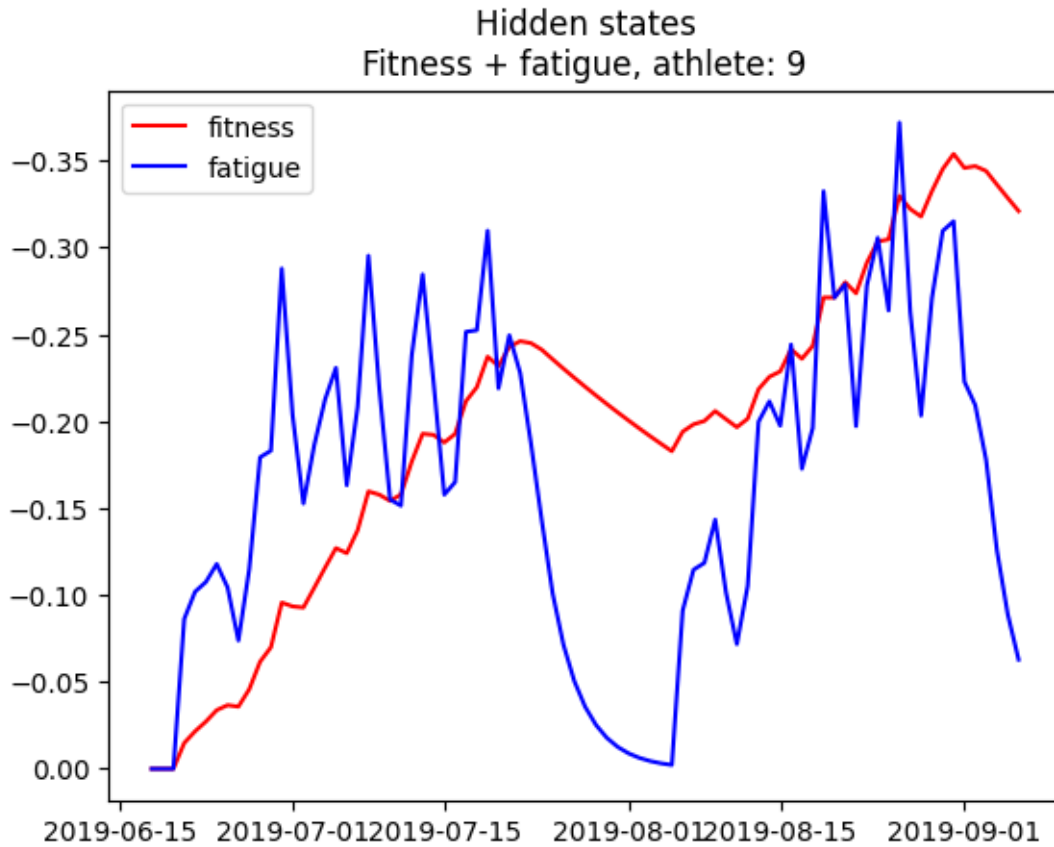




Hidden states  
Fitness + fatigue, athlete: 8







```
[5]: df = pd.DataFrame.from_dict({"athlete": [i+1 for i in athletes_id] +
    ↳ athletes_id],
                                "model": ["Fitness" for i in athletes_id] +
    ↳ ["Banister" for i in athletes_id],
                                "RMSE": [mods[i]["Fitness"].train.RMSE for i in
    ↳ athletes_id] +
    ↳ [mods[i]["Banister"].train.RMSE for i in
    ↳ athletes_id],
                                "a0": [mods[i]["Fitness"].train.params["a0"]
    ↳ for i in athletes_id] +
    ↳ [mods[i]["Banister"].train.params["a0"]
    ↳ for i in athletes_id]})
df.to_csv("Performances_in_training.csv")
```