**self.kernel\_f = 0.5\* Matern(length\_scale=0.5, nu=2.5)+WhiteKernel(0.15)**

**self.kernel\_v = 1.5 + np.sqrt(2)\* Matern(length\_scale=0.5, nu=2.5) + WhiteKernel(1e-4)**

**self.beta = 0.8**

For initialization, I define two matern models as the model of the function f and speed v, and set the ratio beta as 0.8.

For the acquisition\_function

**mean, std = self.model\_f.predict([x], return\_std=True)**

**speed = self.model\_v.predict([x])**

predicting the current function value and speed using predict method of the GPR model.

**res = mean[0][0] + self.beta\*np.sqrt(std[0]) \**

**if speed > SAFETY\_THRESHOLD else -10**

Lastly, return **Miu + beta \* Sigma** if the speed is larger than the threshold, otherwise, return -10 as a penalty.

**self.X = np.vstack((self.X, x))**

**self.f = np.vstack((self.f, f))**

**self.v = np.vstack((self.v, v))**

**self.model\_f.fit(self.X, self.f)**

**self.model\_v.fit(self.X, self.v)**

when adding new data f and v, I stack them into the memorizing list and re-fit the model using all of the data I ever know.

For the **next\_recommendation function**, I simply return self.optimize\_acquisition\_function()

**f\_max = -np.Inf**

**for \_, x\_t in enumerate(x\_domain):**

**mean, std = self.model\_f.predict([x\_t], return\_std=True)**

**speed = self.model\_v.predict([x\_t])**

**f\_t = mean + self.beta\* np.sqrt(std)**

**if f\_t > f\_max and speed > SAFETY\_THRESHOLD:**

**f\_max = f\_t**

**x\_opt = x\_t**

**return x\_opt**

After the iteration of updating the models, we can have two models that fit F and V well.

For the solution, I search for the domain of x after partitioning the domain using **self.x\_bins = 4000**, and return the x value that get the biggest **Miu + beta \* Sigma.**

First, I set a temp variable f\_max that equals -infinite and start the looping.

For each x in the partition of the domain, get the mean and variance through the prediction of model\_f and the speed through the prediction of model\_v.

Then calculate f\_t equals Miu + beta \* Sigma, and check if the speed is larger than the threshold as well as f\_t is the max value from now on. If true, update the f\_max and x\_opt.