

### 1. What we learn.

Train three independent GPs (x, y, z) to predict the tracking error

$$e_t = y_t^{\text{obs}} - y_t^{\text{des}}$$

as a function of features you already have:

- commanded pose at time  $t$ :  $(x_t^{\text{cmd}}, y_t^{\text{cmd}}, z_t^{\text{cmd}})$
- desired pose at time  $t$ :  $(x_t^{\text{des}}, y_t^{\text{des}}, z_t^{\text{des}})$
- time  $t$  (helps capture slow drift)
- optional finite-difference velocities of commanded & desired (captures simple dynamics)

Target per axis is  $e_t^x, e_t^y, e_t^z$ .

### 2. How we compensate.

Once the GPs can predict  $\hat{e}_t = \mathbb{E}[e_t \mid \text{features}]$ , we compute a corrected command by fixed-point iteration:

$$x_t^{\text{cmd}(k+1)} = y_t^{\text{des}} - \hat{e}_t(x_t^{\text{cmd}(k)}, y_t^{\text{des}}, t, \dot{x})$$

Initialize with  $x_t^{\text{cmd}(0)} = y_t^{\text{des}}$ . A few iterations (2–5) are usually enough.

### 3. Why this works.

You're learning the *forward* tracking error map under your controller and using it to “pre-distort” the command sequence so your existing low-level controller + mechanics land closer to the desired path — no change to feedback gains needed.

## NOTE about the input

Please have a look at the GCODE\_25\_08\_21\_Repeatability\_BeratExtension.MPF where you will see how I constructed the input for x,y,z

You will see that it is a kind of step function for example for x:

```
_xStart = -134.893
_x1 = -96.839
_x2 = -58.761
_x3 = -20.692
_x4 = 17.378
_x5 = 55.433
_x6 = 93.495
_x7 = 131.553
```

So by looking at the figure you prepared, I see the total simulation time is around 7.5 seconds, it means at the beginning the input is \_xStart until t=1, then switches to \_x1 and so on, you need to create this input list for x, y and z

**NOTE about the target x,y,z and error calculation**

Then you also need to define the target in order to calculate the error, in our case again by looking at your figure, I see that the x is a linear function from 0 to 0.3 then  $y = 0.9$  and  $z = 0.8$  so these are the target x,y,z to be calculated as error with the corresponding data that you used to prepare the figure