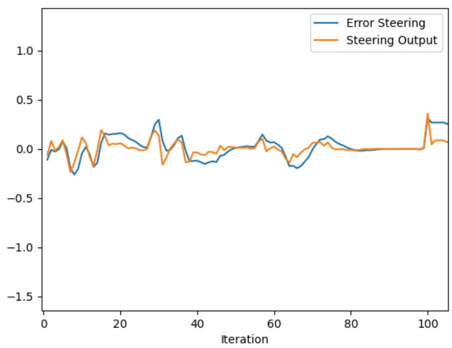
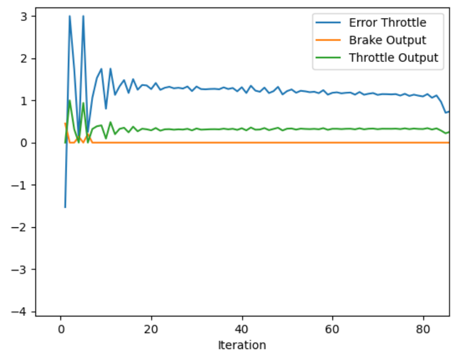
// Answer the following questions:

// Add the plots to your report and explain them (describe what you see)

Answer:





- The first plot shows Steering error and Steering output vs Iterations. We can see that the output closely follows the error and also error stays close to zero. PID parameters can be further tuned to get perfectly overlapping error and output.
- The second plot shows Throttle error and Throttle output vs Iterations. We can see that the output curve follows the error curve but there is a bias in the error and it does not go to zero. This is mainly due to limitations in lateral acceleration. The PID parameters mainly integral parameter can be further tuned to bring the bias close to zero.

// What is the effect of the PID according to the plots, how each part of the PID affects the control command?

Answer:

- Proportional parameter: As seen in the first plot this part helps us to bring the error down to zero based on how big the error is. The bigger the error the higher is the output of proportional part of PID.
- Derivative Parameter: As seen in the second plot this part helps us to respond fast to changing errors. The bigger the rate of change of error the higher is the output of derivative part of PID
- Integral Parameter: This part helps us to bring the bias in error down to zero. The integral part of throttle can be increased to bring down the bias closer to zero.

// How would you design a way to automatically tune the PID parameters?

Answer:

- The first method would to use twiddle algorithm that we learnt in our course but the complexity of the simulation makes it difficult to implement for the given project.
- The second method could be by creating a simple mathematical model of the simulation for example bicycle model of the car and collecting error data and input data such as steering input, velocity and timestamp from the real simulation. Now the simple model can be run for many iterations and an algorithm like twiddle can be implemented to find the best PID parameters.

// PID controller is a model free controller, i.e. it does not use a model of the car. Could you explain the pros and cons of this type of controller?

Pros of model free controller:

- The system(car) can be controlled without any knowledge of the inner workings of the system. The throttle can be controlled without the need of knowing how the motor works.
- Less computation is required.
- Much cheaper and easier to implement.

Cons:

- A model-based controller can provide a much robust control when compared to a model free PID controlled.
- PID controller's parameters have to be changed and computed again when there is a small change in the system. But in a model-based controller only the value of the variable associated with the change have to be tweaked to get the controller working. For example, if a new steering wheel is installed then only some of the parameters of the steering model have to be changed based on the new steering wheel, but in case of PID controller new PID parameters have to be computed which can be challenging.
- Tuning and finding the best control parameters is always a challenging task for a model free controller.