Fig.1 Image of a still car after completing the tasks in pid_controllers.cpp and pid_controllers.h

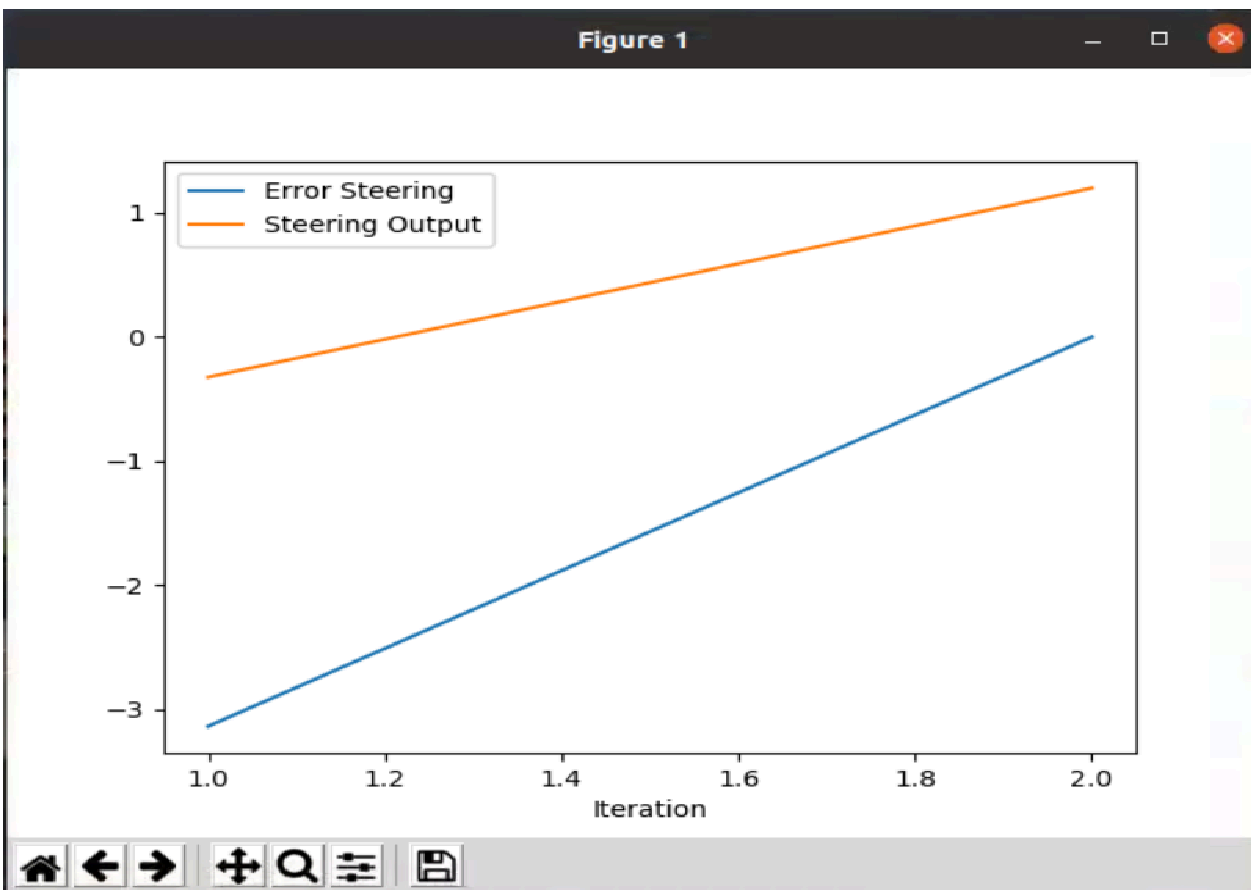

Fig.2 Plot displaying the steering error and steering output

Analysis of the Plot in fig.2

1) Add the plots to your report and explain them (describe what you see) 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? (Optional) What would you do to improve the PID controller?

Description: The plot displays two lines representing "Error Steering" and "Steering Output" against the number of iterations.

1. **Error Steering (Blue Line):**
   - The error starts at a negative value and decreases further as the iterations progress.
   - This indicates that the steering error is becoming more negative over time, which could suggest that the vehicle is increasingly deviating from its intended path.
2. **Steering Output (Orange Line):**
   - The steering output starts near zero and increases positively as the iterations progress.
   - This output represents the corrective action applied by the PID controller to adjust the steering.

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

Effect of the PID Controller:

- The PID (Proportional-Integral-Derivative) controller adjusts the steering output based on the error. In this plot:
  - **Proportional (P):** The proportional term is likely causing an immediate response to the error. As the error grows more negative, the steering output increases to counteract this error.
  - **Integral (I):** If implemented, the integral term would accumulate the error over time, potentially leading to larger steering corrections if the error persists.
  - **Derivative (D):** This term would respond to the rate of change of the error. However, the plots suggest that either the derivative term is not dominant or that the error rate is relatively constant.

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

Tuning PID Parameters: To automatically tune PID parameters, methods such as the Ziegler–Nichols method, manual tuning, or optimization algorithms (e.g., genetic algorithms, particle swarm optimization) can be used. An automated tuning method might involve:

- Running a set of initial tests to gather data on system response.
- Using an optimization algorithm to minimize a cost function, such as overshoot, settling time, or steady-state error, based on the system's performance.

- Iteratively adjusting the PID parameters and re-evaluating the system's response until the desired performance is achieved.

4) 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 and Cons of Model-Free PID Controllers:

Pros:

- **Simplicity:** PID controllers are straightforward to implement and tune, especially for linear systems or systems with minimal dynamics.
- **No Model Required:** They do not require a detailed mathematical model of the system, which is advantageous in complex or poorly understood systems.
- **Wide Applicability:** Effective for a broad range of control problems in industrial and practical applications.

Cons:

- **Limited in Handling Nonlinearities:** They may not perform well in systems with strong nonlinear characteristics or where dynamics change significantly over time.
- **Tuning Challenges:** Finding the right parameters can be challenging, especially without a model, and may not achieve optimal performance.
- **No Predictive Capability:** Lacks the ability to anticipate future errors based on current trends, unlike model-based controllers.

5) (Optional) What would you do to improve the PID controller?

Improvements for the PID Controller:

- **Adaptive Tuning:** Implementing adaptive tuning mechanisms that adjust PID parameters in real-time based on system feedback.
- **Feedforward Control:** Adding a feedforward control component to improve the controller's response to known disturbances or changes in setpoints.
- **Hybrid Controllers:** Combining PID with other control strategies, such as model predictive control (MPC) or neural networks, to handle complex or nonlinear system behaviors better.
- **Gain Scheduling:** Implementing gain scheduling, where PID parameters are adjusted based on operating conditions or system state, to improve performance across a range of conditions.