PID Controller Documentation

# Understanding:

A PID controller works by applying Proportional, Integral and Derivative actions to regulate a process:

* : Controller output with respect to time
* : Error with respect to time
* : Tuning constraints (gains) for Proportional, Integral and Derivative terms.

The PID controller takes a setpoint, then compares it to the actual change from the controlled device (motor, AC unit… etc.), and sends feedback accordingly.

**Proportional Control:** Responds to the error signal and aims to bring the system closer to the setpoint, although it may cause oscillation.

**Integral Control:** Evaluates past errors and eliminates steady-state error.

**Derivative Control:** Estimates the value of the next error according to the previous, making the system more stable by reducing overshoots.

# Simpler Problem implementation

To better understand how a PID controller works, I will be implementing the PID controller for a simpler problem: A vehicle speed PID.



The value of are the result of applying the Zeigler - Nicholas Method. More advanced methods should be used for the required task’s PID controller.

Unfortunately, due to the time constraints implemented and an outside factor. I was unable to implement the complete simulation for the task provided, I was only able to implement the PID and the kinematic model. Sorry, and thank you for your consideration.