

Software Requirements Specification (SRS)

Adaptive Cruise Control Enhancement for Autonomous Vehicles

1. Functional Requirements

Data Input:

- Simulate ego vehicle following a lead vehicle.
- Parameters: desired distance, vehicle speed, lead acceleration set in code.
- No external data file input needed.

Simulation Logic:

- Compute distance between ego and lead vehicle.
- Enhanced PID controller with predictive behavior compensation.
- Update ego vehicle's speed & position.

Visualization:

- Plot Distance to Lead Vehicle with desired distance line.
- Plot Ego Vehicle Speed profile.
- Display plots with labeled axes and grid lines.

User Output:

- Display simulation plots in a single window.
- Visual performance verification of ACC enhancement.

2. Non-Functional Requirements

Platform:

- OS: Windows, macOS, or Linux
- Environment: Python 3.8+

Performance:

- Efficient simulation up to 1000 time steps.
- Execute visualization in under 5 seconds.

Usability:

- Run as Python script or Jupyter Notebook.
- Readable, modular code structure.

3. Software Requirements

Programming Language:

- Python 3.8+

Required Libraries:

- numpy (pip install numpy)
- matplotlib (pip install matplotlib)

Installation:

pip install numpy matplotlib

4. Execution Instructions:

- Install Python 3.8+

- Install libraries via pip
- Save code as acc_enhancement.py or notebook
- Run script or notebook to visualize results

5. Input Parameters (Configurable in Code):

- Desired Distance: 10 meters
- PID Gains: $k_p = 0.6$, $k_i = 0.08$, $k_d = 0.25$
- Lead Vehicle Prediction Gain: $k_{predict} = 0.3$
- Time Step (dt): 0.1 seconds
- Simulation Duration: 50 seconds

6. Output:

- Distance Tracking Plot with reference line
- Ego Vehicle Speed Profile plot
- Visual confirmation of ACC system enhancement