Control Systems Engineering (ME5659) Project Report
Project Topic: Cruise Control System
Team Members: Aryaman Shardul, Sachidanand Halhalli, Jaykumar Goswami

1. Problem Statement and Scope of the Project

- We aim to design an automatic cruise control system for a car (Toyota Camry XLE) to maintain a desired constant velocity despite resistive forces like aerodynamic drag and rolling resistance.
- The project focuses on modeling vehicle dynamics, implementing controllers such as PID, LQR, and Lead-Lag, and comparing their performances.
- We focused on speed control, and assumed flat roads with winds opposing the car's trajectory.

2. Background of the problem

- Cruise control is a very important aspect of autonomous driving.
- Maintaining constant vehicle speed improves fuel efficiency, reduces driver fatigue, and ensures a safer driving process.
- Some of the techniques that people have used before to solve this problem include using the PID controller, MPC for constraint handling, and optimal control methods like LQR for performance balancing.

3. Contributions and Approach:

In this project, our approach was structured as follows:

- **Plant Modeling**

  We first modeled the vehicle dynamics considering aerodynamic drag, rolling resistance, and external wind disturbances. Both a linear approximation and a full nonlinear drag model were developed. After that, we designed the linear and non-linear plants in Simulink.

- **Plant Model Equations:**

  For the linear plant, the system dynamics are approximated as:

$$\dot{v}(t) = -\frac{b}{m}v(t) + \frac{1}{m}u(t)$$

where:

- $v(t)$ = vehicle velocity,
- $u(t)$ = control force input,
- b = total damping coefficient,
- m = mass of the vehicle.

  For the nonlinear plant, the true aerodynamic drag dependence on velocity is captured as:

$$\dot{v}(t) = \frac{1}{m}\left(u(t) - F_r - dv(t)^2\right)$$

where:

- F_r = rolling resistance (constant),
- d = aerodynamic drag coefficient,
- $v(t)$ = vehicle velocity.

- **Controller Design**

  We implemented multiple control strategies:

  - **Lead-Lag Controller**: Designed to improve phase margin and reduce steady-state error.

- **PID Controller**: Designed and initially tuned manually for acceptable performance. Performed pole-placement later to find suitable gain values for a certain desired performance criteria.
- **LQR Controller**: Designed optimal state-feedback gains minimizing a weighted cost of speed error and control effort.

- **Performance Evaluation**

  Each controller was simulated on both the linear and nonlinear plants. Metrics such as overshoot, settling time, and steady-state error were calculated.

- **Comparison and Analysis**

  We compared the results between linear and nonlinear models. Special attention was given to the impact of plant nonlinearity on controller performance.

4. Controller Design

- **Lead-Lag Controller:**
  The lead-lag transfer function is given by:

$$C(s) = K\frac{s+z}{s+p}$$

  Where K is the Gain Value, z are the zeros and p are the poles.

  The lead component of the controller helps to reach near the desired velocity very quickly. But due to its aggressive behavior, it might often overshoot or undershoot the target. The lag component of the controller helps to counter this problem by slowing down the approach towards the desired velocity.

- **PID Controller:**

  The PID law is given by:

$$u(t) = K_p\, e(t) + K_i \int e(t)\, dt + K_d \frac{de(t)}{dt}$$

Where Kp, Ki, and Kd are the proportional, integral, and the derivative gains. The proportional component reacts to the current error e(t) and tries to reduce it immediately, providing a direct response where as the integral component accumulates the past error over time and corrects this steady-state error by ensuring that any small, persistent errors are gradually eliminated and the derivative component predicts the future trend of the error by looking at its rate of change, hence reducing overshoot and oscillations, improving system stability.

- **LQR Controller:**

  The LQR cost function is given by:

$$J = \int_0^{\infty} \left(Qx(t)^2 + Ru(t)^2 \right) dt$$

Penalize deviation from target speed (x(t)) heavily and penalize huge forces (u(t)) mildly (when R is small), or more strictly (if R is large).

The control law is given by:

$$u(t) = -Kx(t) + N_{\text{bar}} r$$

Where K is found by solving the Riccati equation.

5. Results

---

a) Performance Metrics:

- **Percent Overshoot (% OS):** The maximum amount by which the system output exceeds the desired setpoint (expressed as a percentage).
- **Settling Time (Ts):** The time taken for the system output to enter and remain within a 2% specified tolerance band around the desired setpoint.
- **Steady-state error (final velocity error):** The difference between the final value of the system output and the desired setpoint after all transient effects have died out.

b) Simulation Results:

We ran the simulation for 5 minutes (300 seconds) for every case.

Controller	Plant Type	Desired Velocity (m/s)	Percent Overshoot (% OS)	Settling Time (sec)	Final Speed (m/s)
Lead-Lag	Linear	27.78	83.41	268.03	27.77
Lead-Lag	Non-Linear	27.78	55.79	97.35	27.63
Lead-Lag	Linear	70	76.76	265.93	69.97
Lead-Lag	Non-Linear	70	53.82	73.67	69.42
Lead-Lag	Linear	600	83.30	269.10	599.80
Lead-Lag	Non-Linear	600	17.01	Didn’t settle	563.80
PID	Linear	27.78	1.59	42.00	27.78
PID	Non-Linear	27.78	2.57	72.00	27.78
PID	Linear	600	1.59	42.00	599.99
PID	Non-Linear	600	-2.23	Didn’t settle	586.59
LQR	Linear	27.78	0.00	136.74	27.78
LQR	Non-Linear	27.78	5.09	Didn’t settle	29.20
LQR	Linear	600	0.00	131.48	599.90
LQR	Non-Linear	600	17.01	Didn’t settle	273.70

6. Analyzing our Results

a) Major Design Decisions:

- Decided to model a Toyota Camry XLE using parameters from the car's specification sheet.
- Including wind disturbances in modeling.
- Keeping the same gain values for each controller in both linear and non-linear systems to ensure fair comparison.

b) Challenges:

- Nonlinear drag caused unexpected behavior, settling issues.
- Trying to understand the effect of the aerodynamic drag on the linear and non-linear systems.
- Settling time detection was tricky because signals wobble slightly.

c) Our Learning:

- Nonlinearities matter a lot even when linear models seem reasonable.
- LQR gives smooth control but depends on accurate modeling. The same LQR gain values that worked for the linear system won't work well for the non-linear system as compared to the other two controllers.
- Visualization through plots helps to understand hidden behaviors such as the aerodynamic resistance for non-linear systems actually being less of an opposition to the car as long as it's below the desired value.

d) Next Steps:

- Design adaptive controllers to handle nonlinearities better.
- Model road slopes and vehicle mass changes (e.g., loading/unloading).

7. Plots and Figures:

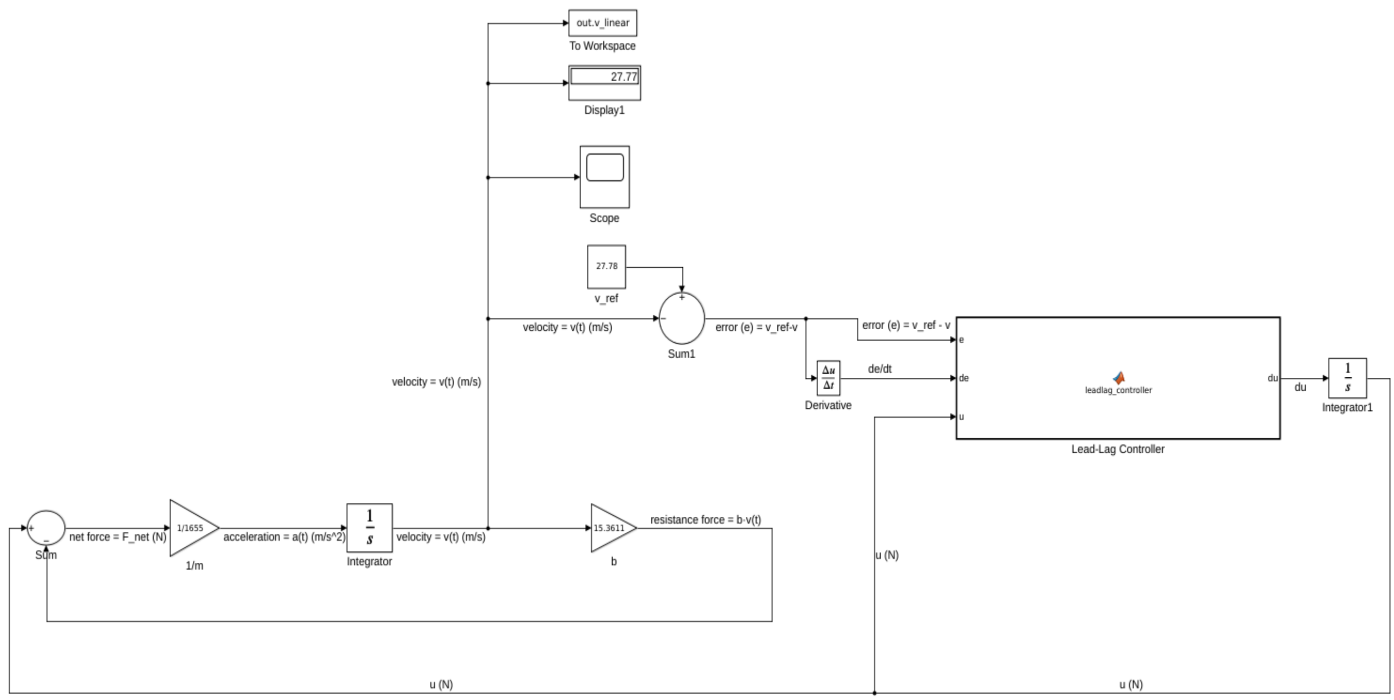

Figure 1: Linear system plant using the Lead-Lag controller.

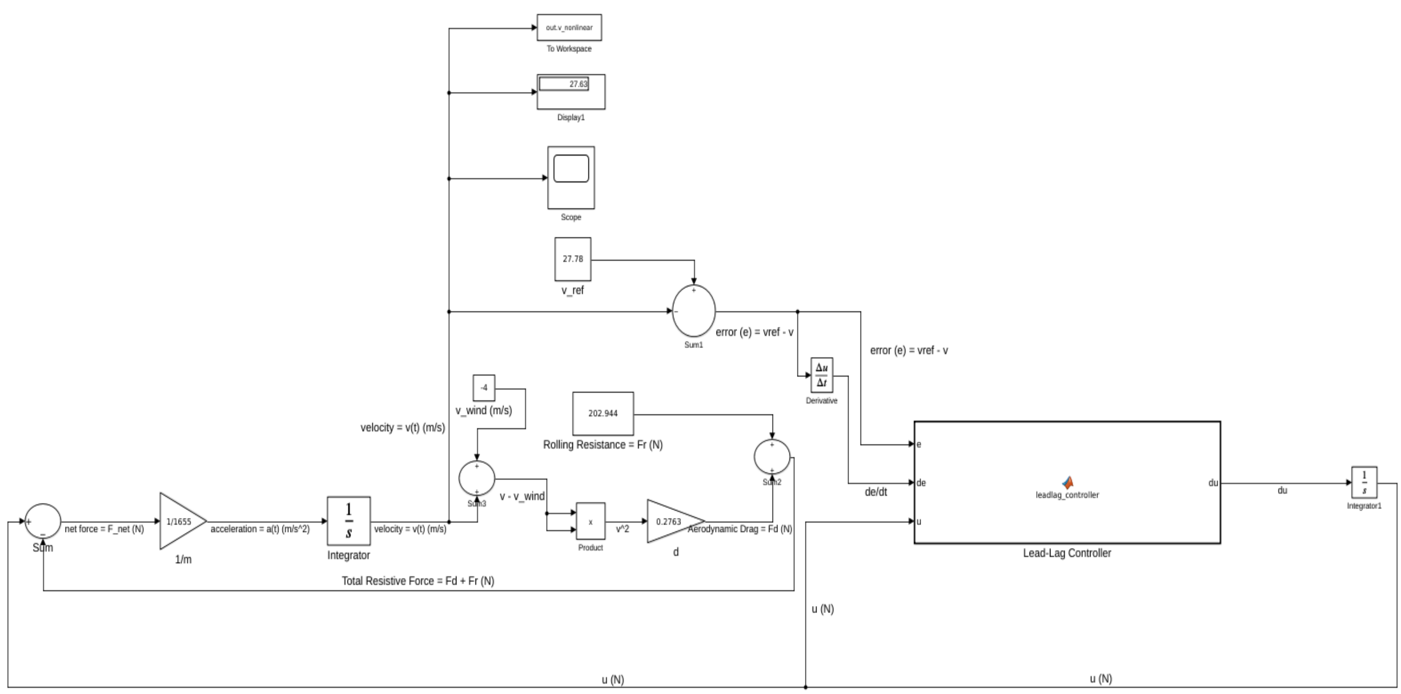

Figure 2: Non-Linear system plant using the Lead-Lag controller.

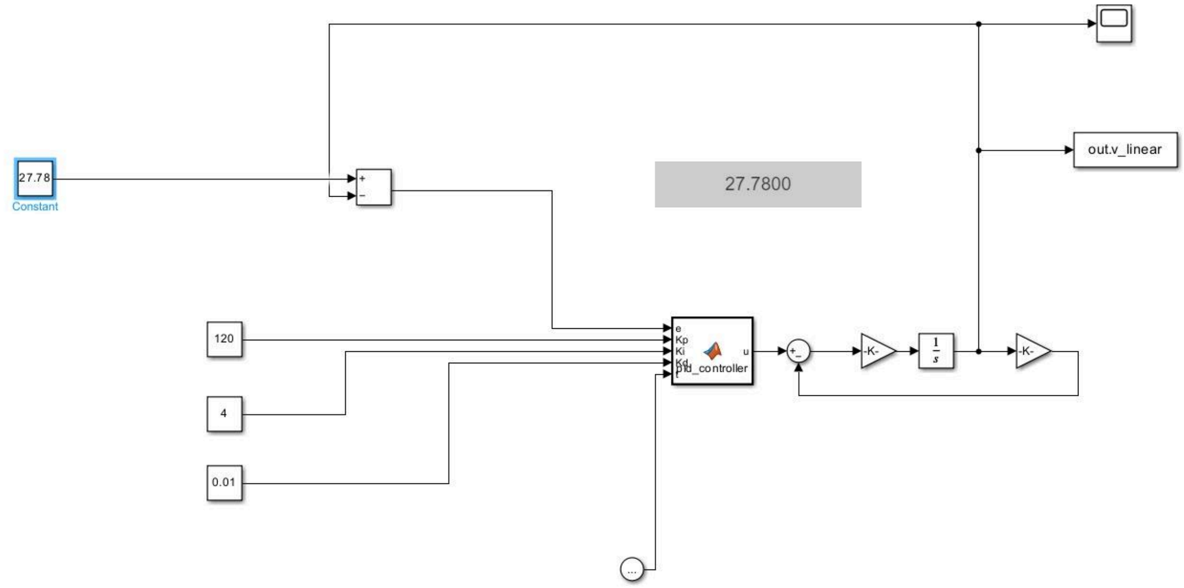

Figure 3: Linear system plant using the PID controller.

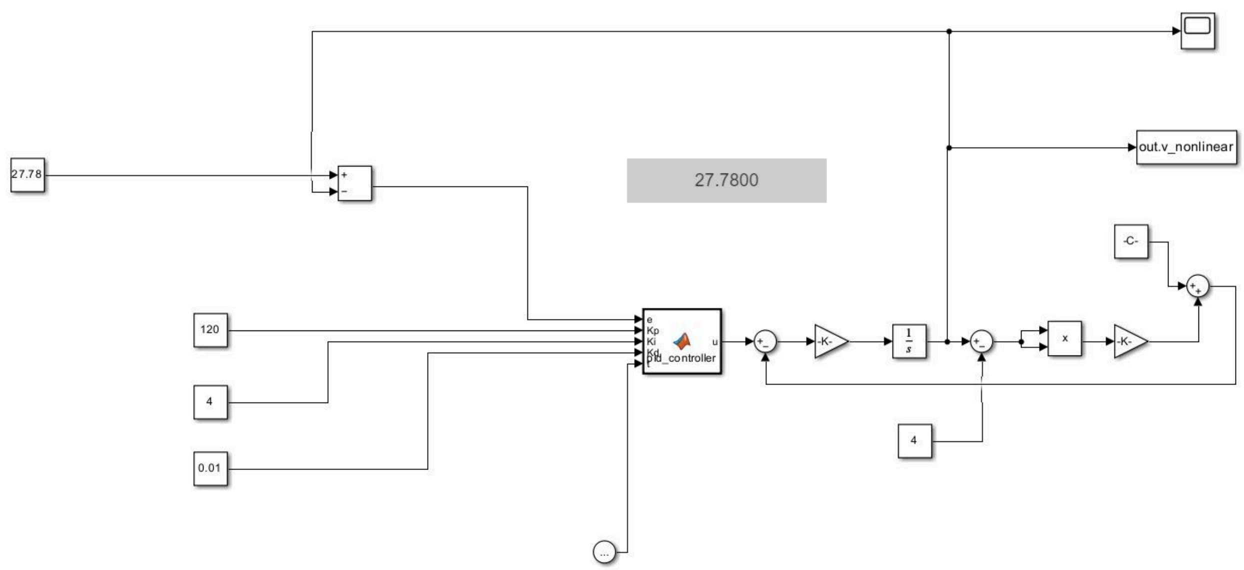

Figure 4: Non-Linear system plant using the PID controller.

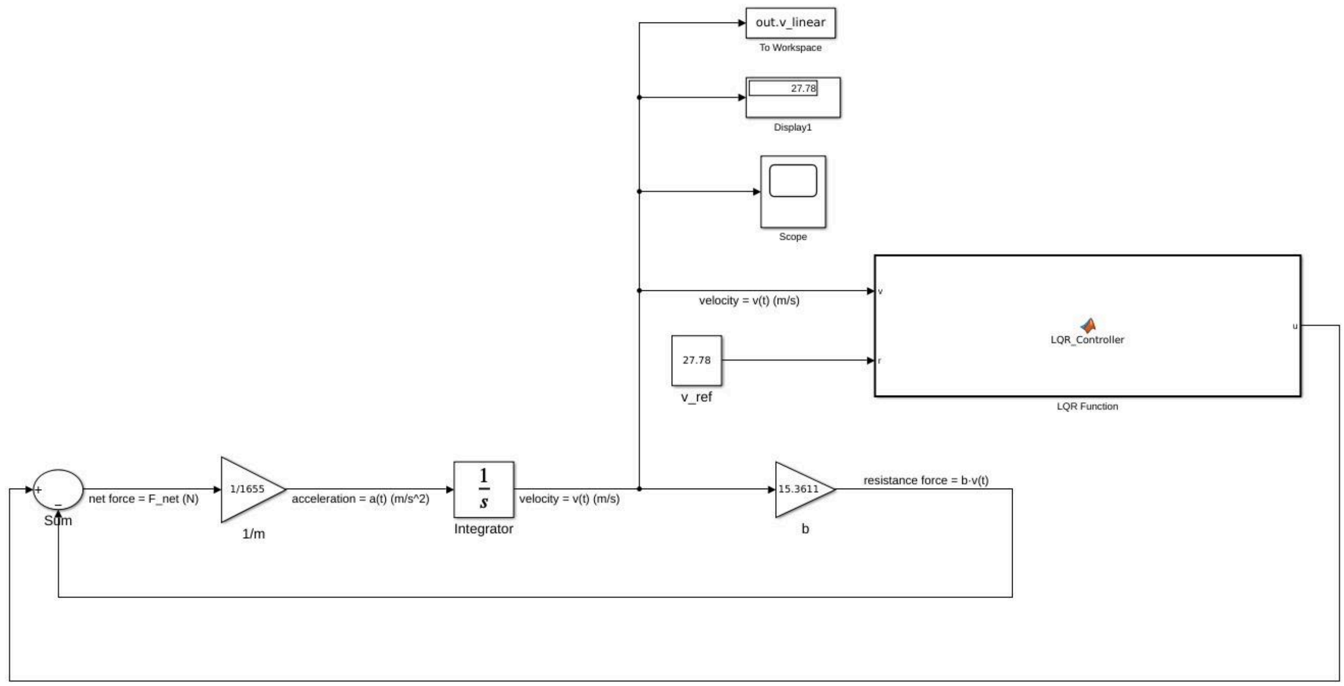

Figure 5: Linear system plant using the LQR controller.

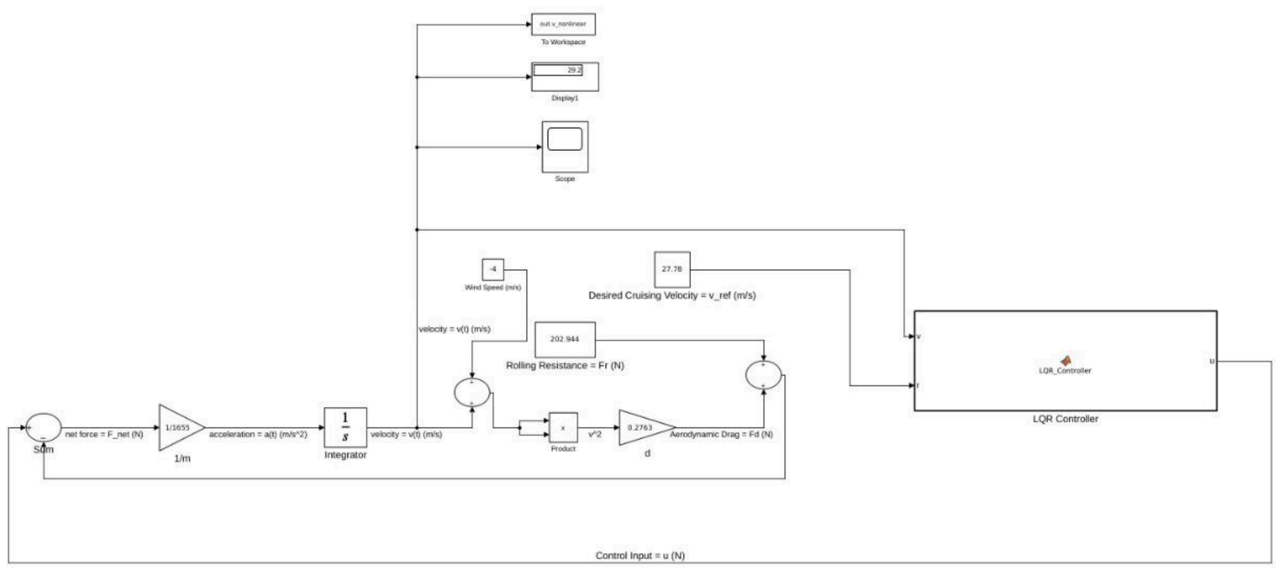

Figure 6: Non-Linear system plant using the LQR controller.

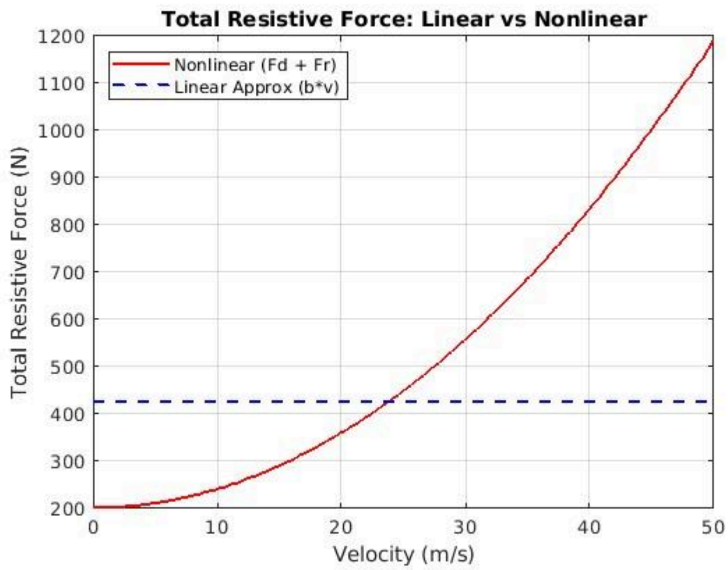

Figure 7: Comparison between the total resistive force faced by the linear and non-linear systems for a desired velocity of 27.78 m/s.

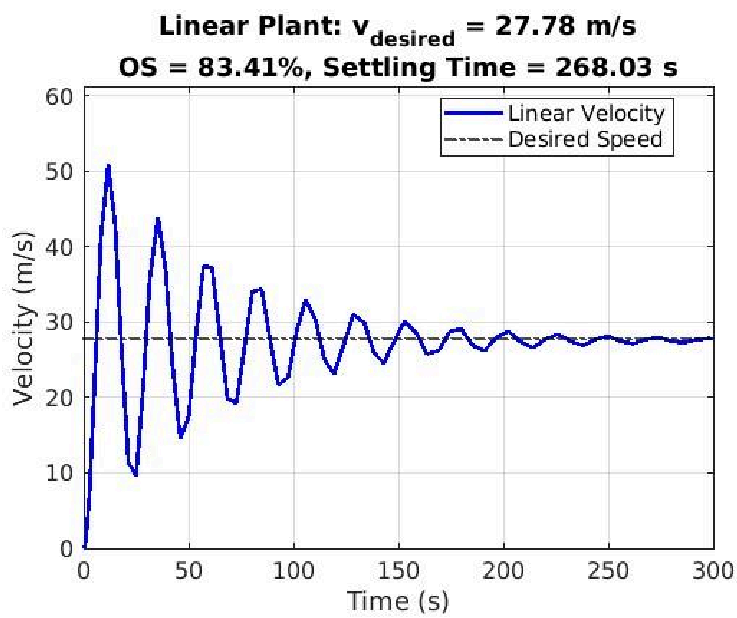

Figure 8: Linear system converging at a value of 27.77 m/s using a Lead-Lag controller when the desired velocity value is 27.78 m/s.

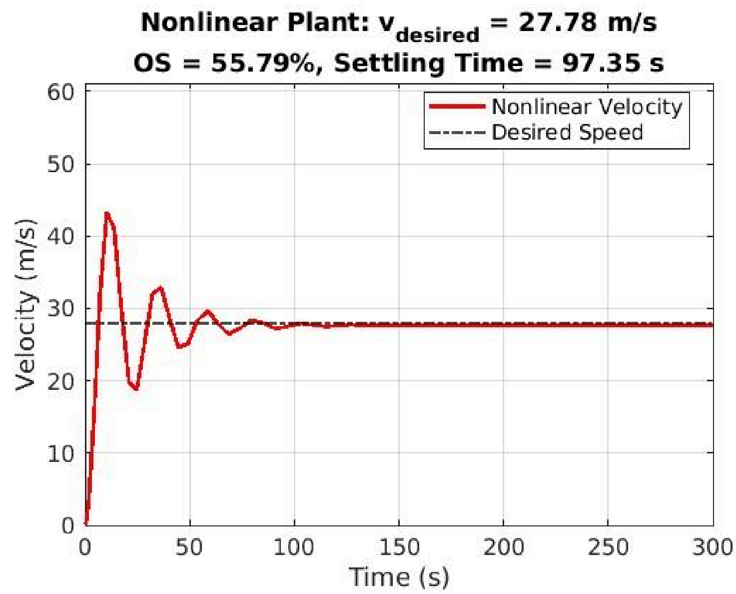

Figure 9: Non-Linear system converging at a value of 27.63 m/s using a Lead-Lag controller when the desired velocity value is 27.78 m/s.

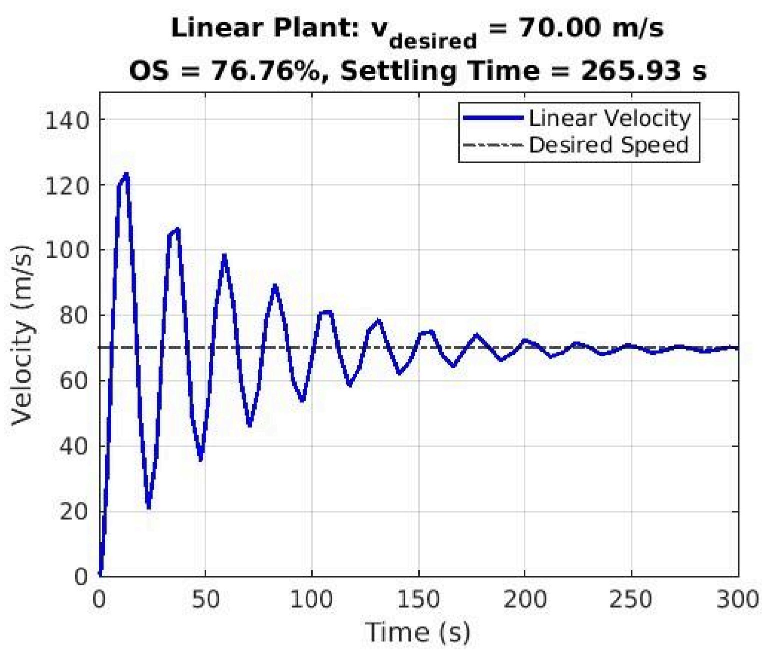

Figure 10: Linear system converging at a value of 69.97 m/s using a Lead-Lag controller when the desired velocity value is 70 m/s.

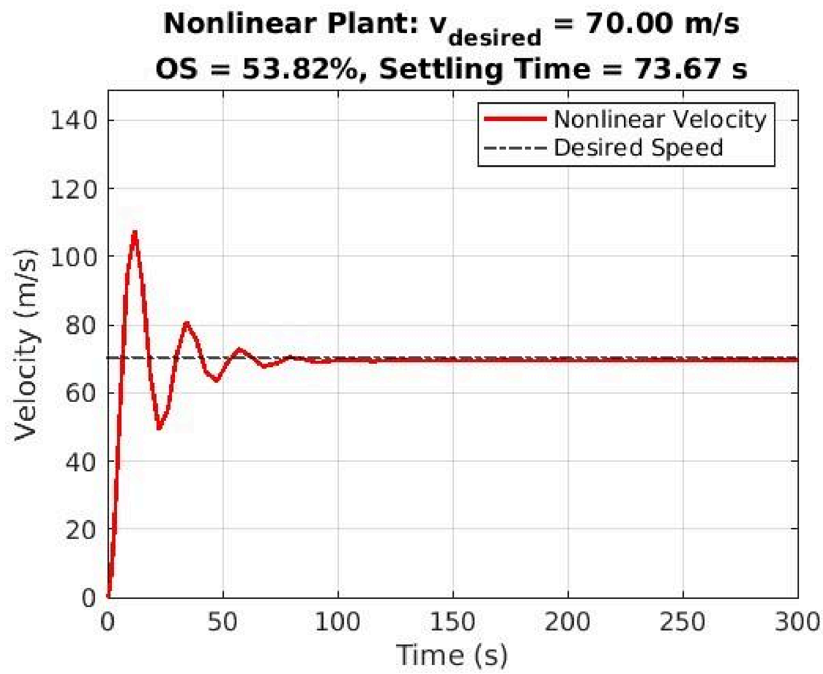

Figure 11: Non-Linear system converging at a value of 69.42 m/s using a Lead-Lag controller when the desired velocity value is 70 m/s.

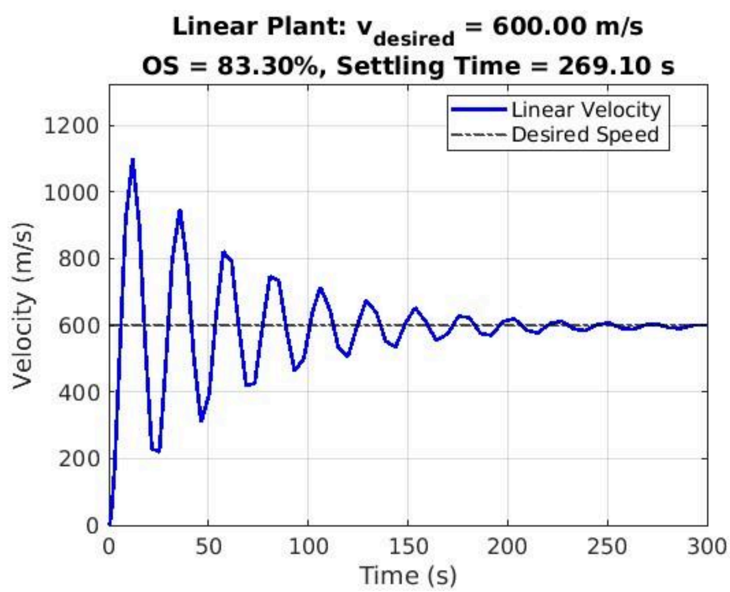

Figure 12: Linear system converging at a value of 599.80 m/s using a Lead-Lag controller when the desired velocity value is 600 m/s.

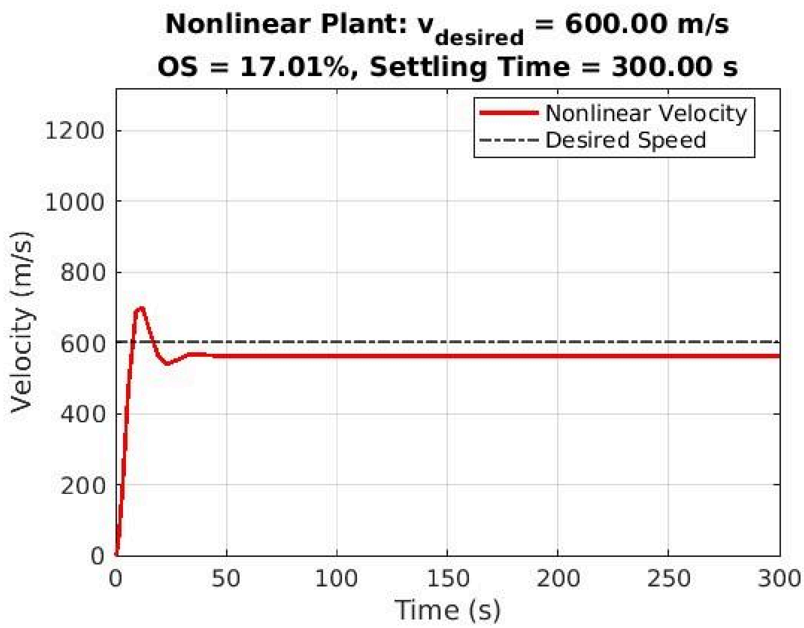

Figure 13: Non-Linear system converging at a value of 563.80 m/s using a Lead-Lag controller when the desired velocity value is 600 m/s.

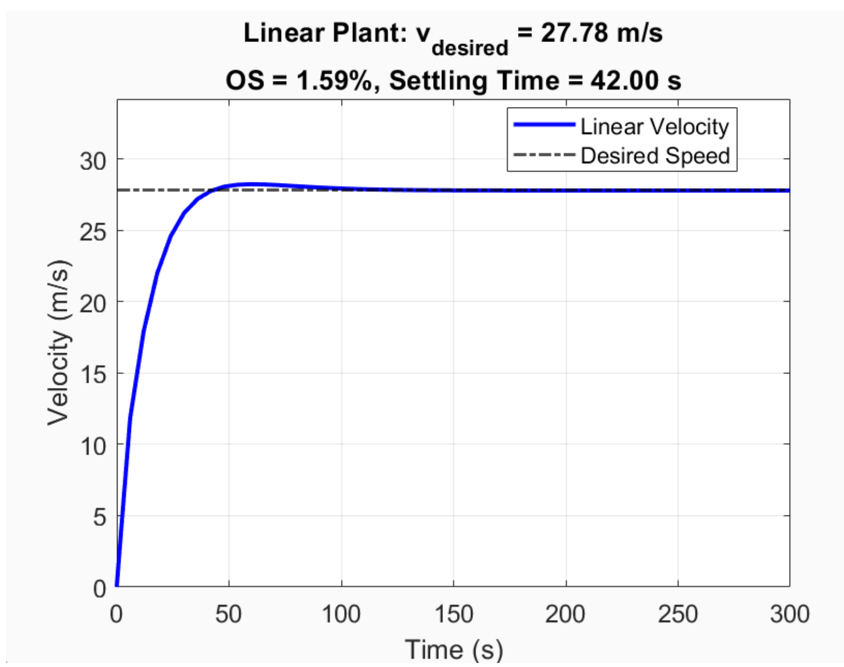

Figure 14: Linear system converging at a value of 27.780 m/s using a PID controller when the desired velocity value is 27.78 m/s.

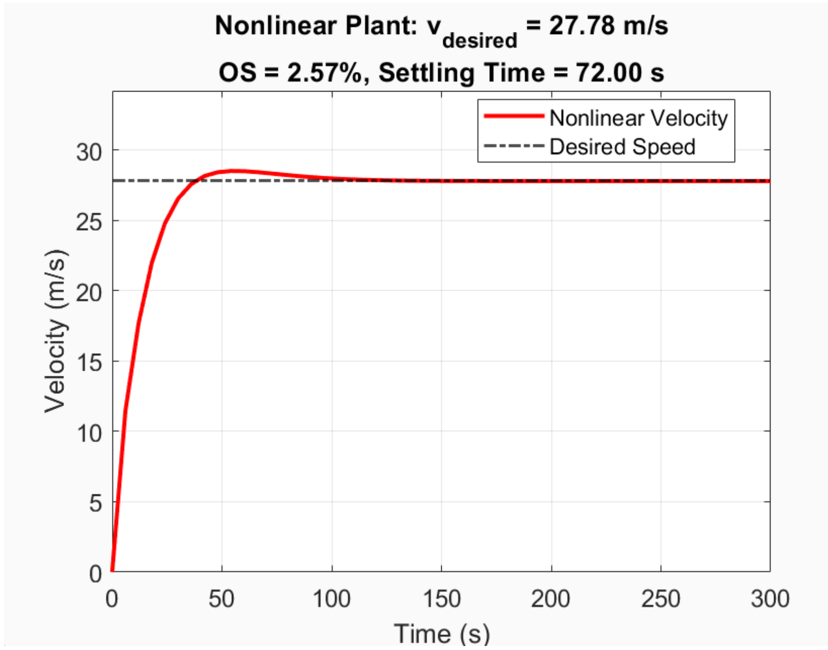

Figure 15: Non-Linear system converging at a value of 27.780 m/s using a PID controller when the desired velocity value is 27.78 m/s.

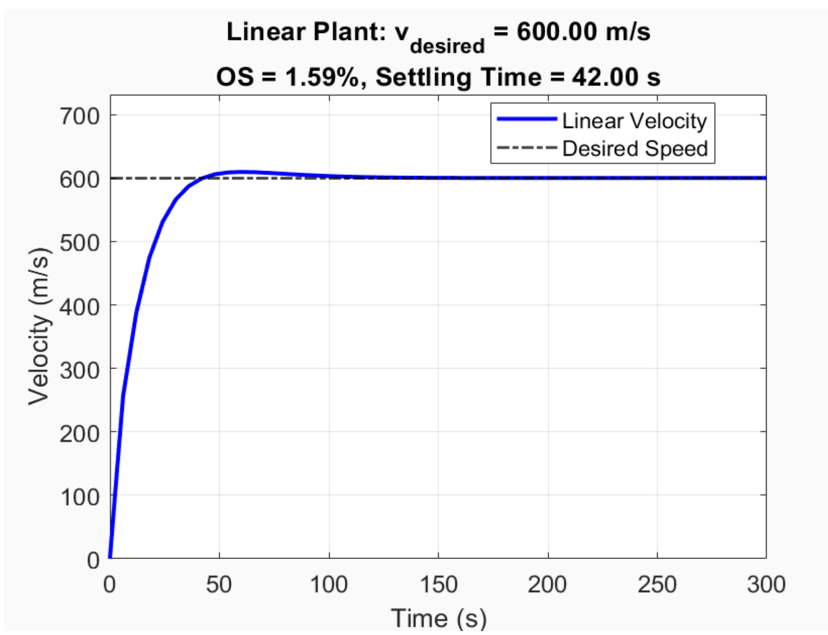

Figure 16: Linear system converging at a value of 599.99 m/s using a PID controller when the desired velocity value is 600 m/s.

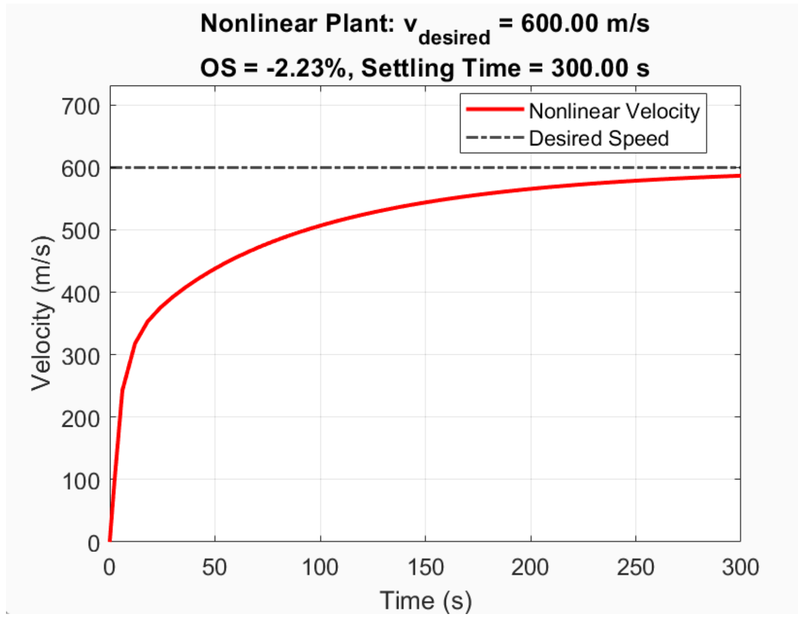

Figure 17: Non-Linear system converging at a value of 586.59 m/s using a PID controller when the desired velocity value is 600 m/s.

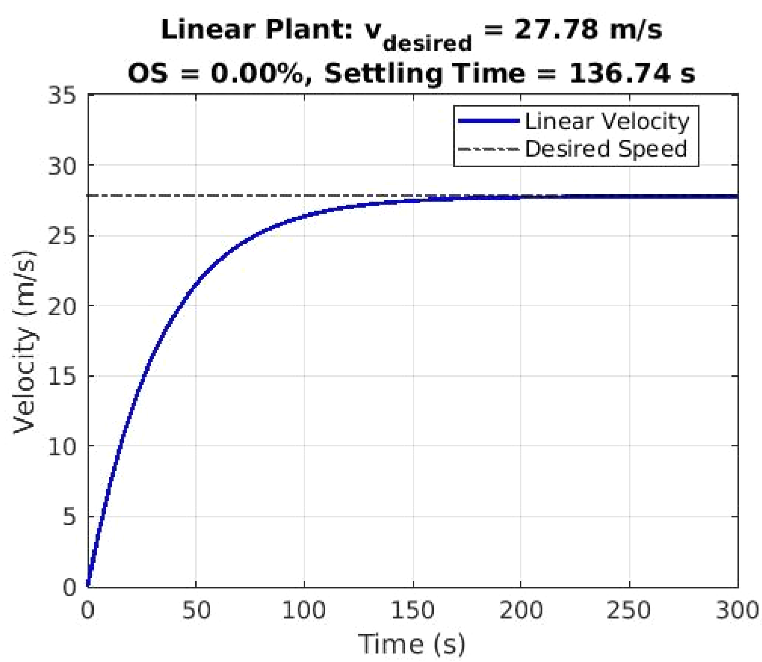

Figure 18: Linear system converging at a value of 27.78 m/s using a LQR controller when the desired velocity value is 27.78 m/s.

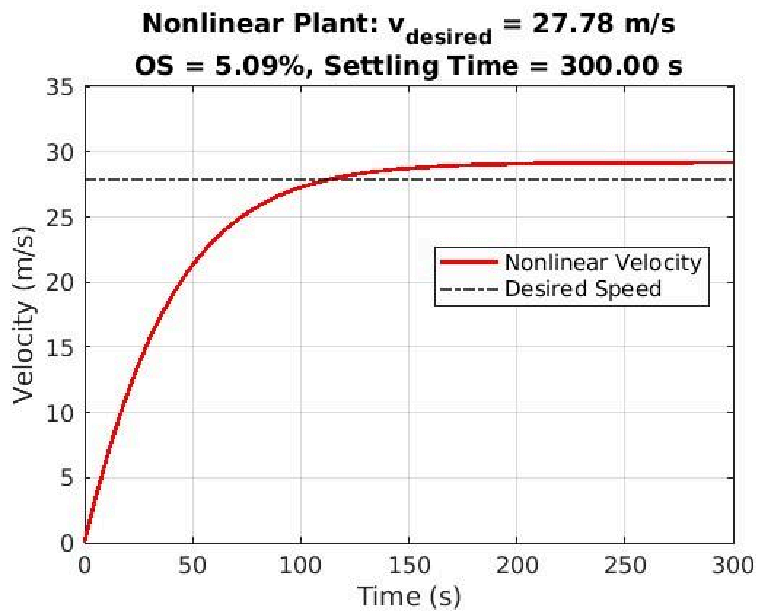

Figure 19: Non-Linear system converging at a value of 29.20 m/s using a LQR controller when the desired velocity value is 27.78 m/s.

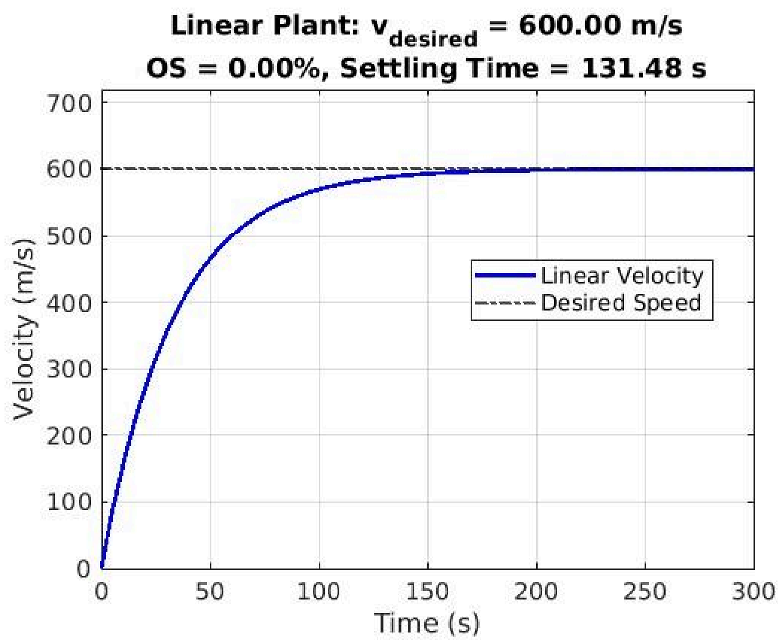

Figure 20: Linear system converging at a value of 599.90 m/s using a LQR controller when the desired velocity value is 600 m/s.

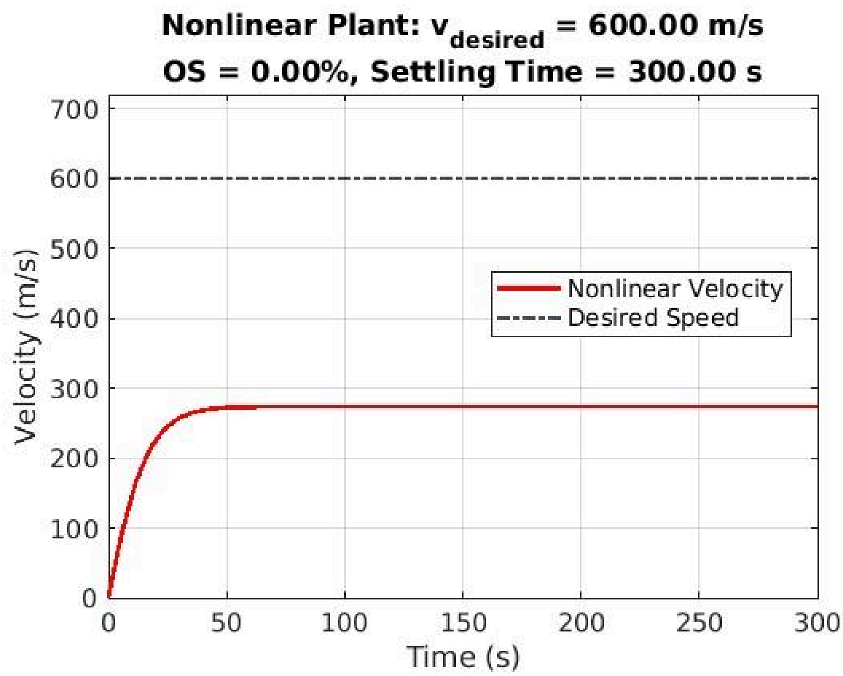

Figure 21: Non-Linear system converging at a value of 273.70 m/s using a LQR controller when the desired velocity value is 600 m/s.