SMART THROTTLE CONTROL SYSTEM

Shiveish Chetty

240986

MIDTERM EVALUATION

Key Concepts Learned

Plant Modelling
Physical systems were represented using mathematical models. Transfer functions provided a structured way to describe system dynamics and predict behavior.

Open-Loop and Closed-Loop Control
Open-loop systems operate without feedback, while closed-loop systems use feedback to reduce error. Meaningful performance evaluation was carried out only in closed-loop systems.

Laplace Transform
The Laplace transform simplified system analysis by converting differential equations into algebraic expressions in the s-domain.

Transfer Functions
Transfer functions defined the input–output relationship of systems and formed the basis for stability analysis and controller design.

System Order and Type
System order was determined by the number of poles, while system type depended on the number of integrators. System type directly influenced steady-state tracking performance.

Time-Domain Performance Metrics
Rise time, settling time, overshoot, and steady-state error were used to quantify and compare system responses.

Stability and Pole Location
System stability was analyzed using pole locations in the s-plane. Left-half-plane poles indicated stable behavior, while right-half-plane poles caused instability.

Steady-State Error and Tracking
Integrators were shown to eliminate steady-state error for step inputs, while higher system type was required for accurate ramp tracking.

Controller Design Principles
Controller tuning involved balancing speed, accuracy, and stability rather than optimizing a single performance parameter.

Tools Used

MATLAB
Used for simulating system responses and validating analytical results.

MATLAB Live Scripts
Enabled organized analysis by combining theory, code, and plots.

Analytical Control Tools
Final Value Theorem and standard response formulas were used for steady-state and transient analysis.

Block Diagrams
Provided clear visualization of system structure, feedback paths, and error signals.

Techniques Applied

- System modelling using transfer functions
- Time-domain response analysis
- Steady-state error calculation in the s-domain
- Gain tuning to meet performance specifications
- Closed-loop pole adjustment for stability improvement
- Zero placement to enhance transient response
- Comparison of open-loop and closed-loop behavior

Challenges Faced

- Managing trade-offs between speed, overshoot, and robustness
- Distinguishing open-loop output from closed-loop error behavior
- Limited ramp tracking performance without integrators
- Interpreting s-domain results in physical terms
- Transitioning from formula-based to system-level reasoning

Category	Topic	Examples from Sessions & Assignments
Key Concepts	Plant Modeling	From **Session 1**, learned that a plant is a mathematical model of a physical system. Applied this by modeling systems such as (G(s)=4/s+2) and (G(s)=10/s(s+5)) in assignments.
	Open-loop vs Closed-loop Control	**Session 1 & 2** introduced feedback control. In Assignment 2, confusion between open-loop divergence and closed-loop zero steady-state error highlighted the importance of feedback definition.
	Laplace Transform	**Session 1** introduced Laplace transforms. Used extensively in assignments to convert time-domain equations into transfer functions and apply the Final Value Theorem.
	Transfer Function Representation	Across **Sessions 1–4**, transfer functions were the primary system representation. Used in all assignments to analyze dynamics, stability, and performance.
	System Order	**Session 2** explained first- and second-order systems. In Assignment 1, (G(s)={4}/{s+2}) was identified as first-order and analyzed using standard formulas.
	System Type	**Session 3** introduced Type-0, Type-1, Type-2 systems. In Assignment 2 , integrator count was used to predict step and ramp tracking behavior.
	Time-Domain Metrics	**Session 2** defined rise time, settling time, overshoot, and steady-state error. These metrics were computed analytically and verified using MATLAB plots in assignments.
	Stability	**Session 2** linked stability to pole location. Assignments demonstrated that poles moving left improve speed, while integrators affect stability and tracking.
	Steady-State Error	**Session 3** formalized steady-state error analysis. Assignments applied Final Value Theorem for step and ramp inputs in closed-loop systems.
	PID Limitations	**Session 4** discussed PID coupling and tuning limits. Assignments indirectly demonstrated why gain-only or zero-only control cannot solve all tracking problems.
	Compensators (Lead/Lag)	**Session 4** introduced lead compensation. Assignment 4 applied zero placement (lead-like behavior) to speed up response.

Category	Topic	Examples from Sessions & Assignments
	Feedforward Concept	**Session 4** explained proactive feedforward control. Although not directly implemented, assignments revealed why feedback alone has limits.
	MIMO Awareness	**Session 4** highlighted real systems as MIMO. Assignments remained SISO but built the foundation needed before handling coupled systems.
Tools	MATLAB Live Scripts (.mlx)	Used in all assignments to generate step and ramp responses, visualize transient behavior, and validate analytical results.
	MATLAB Plot Interpretation	Step-response and ramp-response plots were analyzed to confirm settling time, steady-state value, and tracking performance.
	Analytical Control Tools	Laplace transform, Final Value Theorem, first-order formulas, and system type rules were used consistently across assignments.
	Block Diagrams	Introduced in **Sessions 1–2** and used conceptually to distinguish open-loop vs closed-loop behavior in assignments.
Techniques	System Modeling	Modeled plants using transfer functions derived from sessions and applied them in assignments.
	Time-Domain Estimation	Used analytical expressions for rise time and settling time before verifying results via MATLAB (Assignment 1 & 3).
	Steady-State Error Calculation	Applied limit-based analysis for step and ramp inputs using closed-loop error expressions (Assignments 2).
	Gain Tuning	In Assignment 3, gain was selected to meet settling time and steady-state error specifications taught in **Session 3**.
	Pole Shifting	Demonstrated that increasing gain moves poles left, improving speed (Sessions 2–3, Assignment 3).

Category	Topic	Examples from Sessions & Assignments
	Zero Placement	In Assignment 4, a controller zero was added to improve transient response, reflecting compensator concepts from **Session 4**.
	Closed-Loop Evaluation	Compared open-loop and closed-loop behavior using MATLAB plots to assess improvement (Assignments 1–3).
	Ramp Tracking Evaluation	Used system type logic from **Session 3** to explain infinite ramp error despite improved step response (Assignment 2).
Challenges Faced	Open vs Closed-Loop Interpretation	Early confusion arose when open-loop output diverged while closed-loop error was zero, resolved through session concepts.
	Speed vs Accuracy Trade-off	Increasing gain improved settling time but risked overshoot and reduced robustness, reflecting tuning conflicts discussed in **Session 4**.
	Misleading Improvements	Adding zeros improved speed but did not improve ramp tracking, reinforcing system type theory from **Session 3**.
	Ramp Input Complexity	Ramp tracking required deeper understanding than step response, highlighting integrator importance.
	Mathematical Interpretation	Translating s-domain limits and pole locations into physical behavior required practice and iteration.
	Conceptual Progression	Transitioned from formula-based solving to system-level reasoning using poles, zeros, and feedback structure.
	Real-World Limitations	Sessions emphasized nonlinearities and coupling, helping contextualize why assignment models are simplifications.

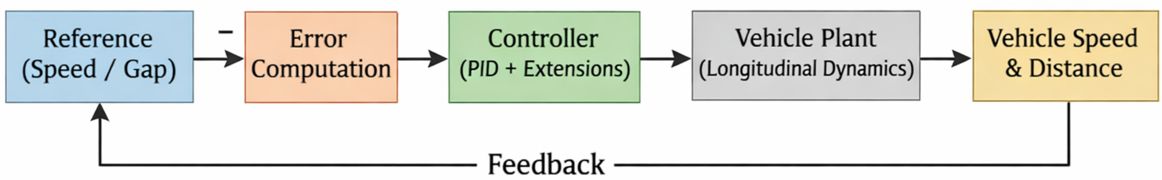
Adaptive Cruise Control

Introduction

Adaptive Cruise Control (ACC) with traffic response is a **closed-loop longitudinal control system** designed to regulate vehicle speed and inter-vehicle spacing in varying traffic conditions. Making an ACC can follow the same fundamental structure in classical control systems, like **plant, controller, reference input, sensors, and feedback loop**, included with supervisory safety logic.

Overall System Architecture

The ACC system can be structured as a closed loop system like.



Plant and System Modelling

In ACC, the **plant** corresponds to the vehicle's longitudinal dynamics. This is typically approximated using a **linearized transfer function model**, similar to the mass–damper systems:

$$G(s) = 1/(Ms^2 + bs + k)$$

where the input represents a throttle/brake command (or desired acceleration), and the output is vehicle speed or position. Although the real vehicle is nonlinear, linear models are sufficient within normal operating regions when combined with feedback.

Reference Generation

ACC operates using **two different references**, depending on traffic conditions:

1. Speed reference (free-flow traffic):

$$r_v(t) = v_s$$

This is analogous to a step input, similar to classic cruise control.

2. Distance reference (traffic present):

$$d_r(t) = d_0 + hv_e(t)$$

Where d_0 is the stand still distance and

h is the desired time headway.
This formulation ensures speed-dependent safety and introduces a **ramp-type tracking problem**, directly linking ACC to system-type and steady-state error concepts from control theory.

Error Signals and Tracking Objectives

Based on the active mode, the ACC controller computes one of the following tracking errors:

- Speed Error:

$$e_v(t) = v_s(t) - v_e(t)$$

- Distance Error:

$$e_d(t) = d(t) - d_r(t)$$

The system switches between these two references , making ACC a **hybrid control system** rather than a single fixed-reference controller. Because both speed and distance needs to be maintained to avoid collision and keep a safe distance.

Controller Design

Base Feedback Controller (PID)

The core ACC controller is the **PID-Based Feedback Controller.**

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

- K_p **(Proportional Action):** It mainly decreases rise time and increase the response Time to Speed or Distance error. It also decreases steady state error. However it results in overshooting. Which is handled by Derivative controller.

- K_i **(Integral Action):** It also have a similar function to proportional action, but completely eliminated the steady-state error which becomes especially important in maintaining consistent spacing. While it increases accuracy drastically it also increase Overshoot and settling time.

- K_d **(Derivative Action):** It introduces damping to the system which reduces Overshoot and oscillations. Which is essential when used with K_p and K_i as they both increase overshoot and hence destabilise the system.

Limitation of PID controller

In practice, PID control alone is insufficient for ACC due to:

- Parameter coupling (improving speed may reduce stability),
- Integral windup during stop-and-go traffic,
- Noise sensitivity of derivative action (e.g., radar measurements).

So we introduce some Enhancements-

- **Lead-Lag compensator:**

- **Lead Compensator:** It is used to improve transient response and reaction speed, especially during sudden braking by a lead vehicle.
- **Lag Compensator:** It eliminated Steady-State error with a much higher precision by increasing low frequency gain, which reduces the long-term Distance error.

These compensators decouple conflicting control objectives such as speed, stability, and accuracy.

- **Feedforward Controller(Traffic Anticipation):**

ACC incorporate **Feedforward Control** by using relative velocity to predict the upcoming traffic and reduce or increase speed accordingly.

$$u_{ff} \propto (v_l - v_e)$$

Feedforward action allows the system to anticipate predictable disturbances, such as lead-vehicle deceleration, while feedback control corrects unpredicted disturbances. This proactive-reactive combination significantly improves traffic response.

Feedback loop and Vehicle response

The controller output drives the complete system, as the resulting speed and distances are continuously measured and fed into the system, this ensures:

- Disturbances caused by wind, hills, potholes etc are rejected timely and ensures consistency.
- Stable and smooth vehicle behaviour, and robust tracking.

Safety and Supervision Layer

This supervision layer operates in parallel to all the controllers and enforces some absolute rules which should be followed like {max, min} speed, and minimum distance a vehicle should maintain.

$$a_{min} < u(t) < a_{max} \text{ and } d(t) > d_{min}$$

If the constraints are violated , then the system stops completely and return the control back to the driver.

Multiple Inputs and Outputs considerations

ACC is inherently MIMO system as:

Multiple inputs (throttle, breaks)

Multiple outputs (speed, distance)

Strong coupling between variables

FINAL SAY-

Adaptive Cruise Control with traffic response is a closed loop control system, which regulates vehicles speed and distance from other vehicles via PID based feedback system augmented with compensators and feedforward action, it operates on linearised vehicles by continuously keeping track of reference signals and enforcing strict rules on speed and spacing.