Title: Hierarchical Adaptive Cruise Control System for Semi-Autonomous Electric Vehicles

Abstract:

The report discusses the development and implications of a hierarchical adaptive cruise control (ACC) system tailored for semi-autonomous electric vehicles. This system is designed to enhance driving safety and comfort by automatically adjusting the vehicle's speed to maintain a safe distance from other vehicles. The system utilizes a dual-controller mechanism with an upper controller based on model predictive control theory for longitudinal acceleration and a lower torque vectoring controller that manages wheel slip. Simulation tests under various conditions affirm the system's efficacy in improving collision avoidance, ride comfort, and adhesion utilization, suggesting significant implications for future automotive safety technologies.

1. Topic Research

1.1 Background

Adaptive cruise control (ACC) systems are advanced driver-assistance systems that control a vehicle's speed, enabling it to maintain a safe distance from vehicles ahead. ACC enhances driver comfort, particularly in long drives and congested traffic situations, by automatically adjusting the throttle and braking based on the traffic flow.

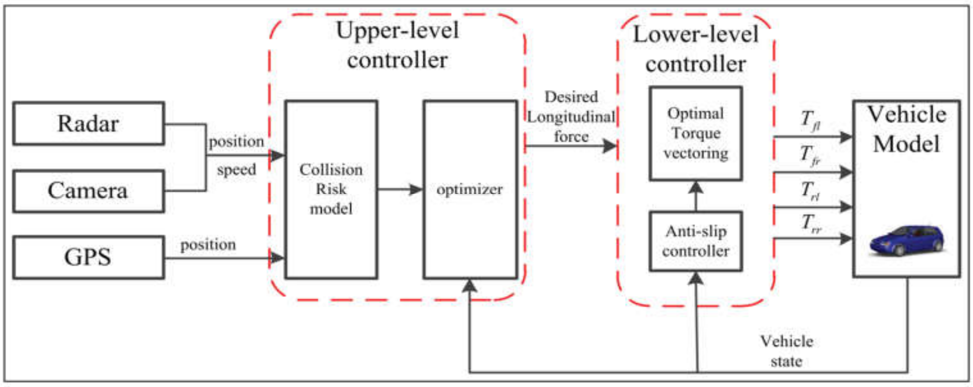
1.2 Importance of ACC in Semi-Autonomous Vehicles

In semi-autonomous electric vehicles, the integration of ACC systems plays a crucial role in facilitating autonomous driving features, leading to safer and more efficient driving experiences. The paper focuses on a hierarchical adaptive cruise control system that balances multiple factors such as driver expectations, collision risks, and ride comfort.

1.3 Research and Development Trends

Recent advancements in sensor technology and control theory have significantly improved the functionality and reliability of ACC systems. This research integrates model predictive control with a potential field method to effectively manage the dynamics of collision risk and vehicle trajectory.

2. System Design



2.1 System Architecture

The proposed ACC system is structured into two main components:

Upper Controller:

- **Collision Risk Model:**

This module receives data from the radar, camera, and GPS.

It uses this data to assess the risk of collision based on the relative position and speed of objects detected around the vehicle.

The collision risk is modeled using probabilistic methods, likely incorporating Gaussian distributions to predict potential future positions of these objects.

- **Optimizer:**

Processes the output from the collision risk model to determine the optimal vehicle response.

Calculates the desired longitudinal force (acceleration or deceleration) needed to maintain safety and comfort, balancing collision avoidance with ride smoothness and efficiency.

Lower Controller:

- **Optimal Torque Vectoring:**

This module receives the desired longitudinal force from the upper-level controller.

It computes the ideal distribution of torque across the vehicle's wheels to achieve the desired force while managing wheel slip, thus maintaining optimal traction and stability.

- **Anti-slip Controller:**

Works in conjunction with the torque vectoring system to prevent the wheels from slipping, especially under adverse conditions or sudden maneuvers.

Ensures that the torque applied to each wheel does not exceed the grip available from the road surface, which is crucial for maintaining control and stability.

2.2 Implementation Considerations

Sensor Integration:

As shown in Picture above, the whole control architecture is proposed based on the four-wheel independent driving (4WID) electric vehicle equipped with environmental awareness sensors including 3 main components:

1. **IMU (Inertial Measurement Unit) Radar:**

    - Combines radar capabilities with an Inertial Measurement Unit.
    - Radar part detects objects by emitting radio waves and measuring the reflection to determine distance and relative speed of surrounding objects.
    - The IMU component provides additional data on the vehicle's own movement and orientation by measuring acceleration and rotational rates. This helps in accurately determining the vehicle's behavior in terms of yaw, pitch, and roll, which is crucial for advanced control systems in dynamic environments.

2. **Camera:**

    - Captures visual data from the vehicle's environment.

- Utilizes advanced image processing to detect and classify objects such as vehicles, pedestrians, traffic signs, and lane markings.
- Provides detailed information on the position, speed, and trajectory of objects, particularly those in front of the vehicle, enhancing object recognition and situational awareness.

3. **GPS (Global Positioning System):**

- Receives signals from satellites to determine the vehicle's precise location on the globe.
- Assists in navigation by providing real-time positional data, which is essential for route planning and tracking.
- Integrates with digital maps to enable automated driving features like route optimization and turn-by-turn directions.

3. Solution Algorithm Proposals

As explained in the hardware section, this ACC model is divided into 2 main processing stages:

- Stage 1: uses MPC to calculate desired longitudinal acceleration based on collision risk modeled by Gaussian distribution.
- Stage 2: It uses the result of stage 1 and it focuses on torque vectoring based on vehicle dynamics to generate required acceleration while managing anti-wheel slip constraints, ensuring vehicle stability under various road conditions.

3.1 The upper-level controller

3.1.1 The vehicle longitudinal kinematics

- x(k): the longitudinal position at the time k.
- y(k): speed at the time k
- a(k): acceleration input at this time
- Δt: discretization time step
- The vehicle longitudinal kinematics could be discretized as:
    - $x(k+1) = x(k) + v(k).\Delta t$

      It mean the longitudinal postion at time k+1 equal the longitudinal postion at time k plus distance traveled in Δt with speed at k.

    - $v(k+1) = v(k) + a(k).\Delta t$ (similar to x(k+1))
- Then both of them can be written as the state space:

$$X(k) = \begin{pmatrix} x(k) \\ y(k) \end{pmatrix}$$

- So that:

$$X(k+1) = \begin{pmatrix} x(k+1) \\ v(k+1) \end{pmatrix} = \begin{pmatrix} x(k) + v(k).\Delta t \\ v(k) + a(k).\Delta t \end{pmatrix}$$

$$= \begin{pmatrix} x(k) + v(k).\Delta t \\ v(k) \end{pmatrix} + \begin{pmatrix} 0 \\ a(k).\Delta t \end{pmatrix}$$

$$= \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x(k) \\ y(k) \end{pmatrix} + \begin{pmatrix} 0 \\ \Delta t \end{pmatrix} .a(k)$$

=> X(k+1) = A.X(k) + B.u(k) (With A, B, u corresponding to the above formula)

=> X(k+2) = A.X(k+1) + B.u(k+1) = A^2 .X(k) + A.B.u(k) + B.u(k+1)

=> X(K+3) = A^3 X(k) + A^2.B.u(k) + A.B.u(k+1) + B.u(k+2).

Through iteration, the future state during the prediction horizon Np within the future input during the control horizon Nc could be expressed as follow:

=>

$$X(k+N_c) = A^{N_c}.X(k) + B.\left[\sum_{n=0}^{N_c-1} A^n.u(k+N_c-1-n)\right]$$

=>

$$X(k+N_p) = A^{N_p}.X(k) + B.\left[\sum_{n=N_p-N_c}^{N_p-1} A^n.u(k+N_p-1-n)\right]$$

- It can be seen that the output of the model depends on the variable Np. The output is more accurate the larger Np is. When Np is larger, the calculation cost increases, but if Np is not enough, the system may predict incorrectly or correctly but will not be able to control it in time because of the vehicle's inertia and depends on the friction of the wheels compared to the road surface.
- When the vehicle is approaching a static obstacle, the minimum safe distance to avoid collision if vehicle fully brakes could be expressed as follows:

$$X_r \geqslant v_0 t + \frac{1}{2} a_{\max} t^2$$

- Vo : the initial speed of the host vehicle
- t = v0/a_{max} : the minimum time that the vehicle decelerates from the initial speed to zero with the maximun deceleration

- To ensure safety, the predictive distance should not be less than the minimum safe distance. So the predictive time is set as follows:

$$T_p = w_s \frac{X_r}{v_0} = w_s \frac{v_0}{2 \cdot \mu g}$$

- W_s : the safe scaling parameter
- μ : the road adhesion coeficient

- The function shows that the larger v0 the smaller the coefficient, the larger the minimum distance must be and vice versa.

3.1.2 Optimization

The ACC system has the function of maintaining the driver's desired speed. However, as the risk of collision increases, the vehicle needs to slow down to avoid a collision, resulting in the vehicle speed deviating from the driver's desired speed. To resolve this contradiction, an optimal cost function based on the potential field is established over the prediction period. The total potential energy can be expressed by the following equation:

$$U = \sum_{i=k}^{k+N_p} (U_{at} + U_{re} + U_c) \quad (1)$$

- U_{at} : the attractive potential that guides vehicle moving with the desired speed
- U_{re} : the repulsive potential that keeps the vehicle away from the obstacle
- U_c : the potential function of the control input that could minimize the jerk

U_{at} could be presented as follows: $$U_{at}(i) = w_{at}(v_x(i) - v_d(i))^2 \quad (2)$$

- wat : the positive scalar positive parameter
- v_x (i) : the actual vehicle speed at time i
- v_y (i): desired vehicle speed at time i (is considered const)

Constraints : vmin <= v <= vmax, for the most cases, there should be no vehicle speed limit. So convention that $v_{min} = 0$, $v_{max} = v_y.10\%$. (3)

U_c could be presented as follows: $$U_c(i) = w_c(a(i) - a(i-1))^2 \quad (4)$$

- w_c : the positive scaling factor. (For this optimization problem, w_c could be very small but cannot be ignored)
- a(i) : the control input at time

U_{re} is established based on the Gaussian distribution as follows:

$$U_{re}(i) = w_{re} e^{-b\left(x_p(i) - x_o(i)\right)^2} \quad (5)$$

- W_{re} : the scaling factor of the repulsive potential function
- $x_p(i)$: vehicle position at time i
- xo(i) : obstacle position at time i
- b : the shape parameter of the potential function. It could be presented as

$$b = \frac{1}{v_p(i) - v_o(i) + s}$$

- $v_p(i)$: vehicle speed at time i
- $v_o(i)$: abstacle speed at time i

=> $v_p > v_o$ -> b is smaller -> U_{re} is larger

- There are two more constraints: (6)

- $x_p + \mathcal{E}_s <= x_0$: It's mean the vehicle's coordinates plus the minimum safe distance must be less than or equal to the obstacle's coordinates.
- $|a| <= \min(\mu_i g)$ i = fl, fr, rl, rr : the adhesion coeficient of each wheel.

From (1),(2),(3),(4),(5),(6) :

$$\min = \sum_{i=1}^{N_p} \left(U_{re} + w_{at} \underbrace{\left\| v_{k+i,k} - v_{des_{k+i,k}} \right\|^2}_{U_{at}} + w_c \underbrace{\left\| a_{k+i-1,k} - a_{k+i-2,k} \right\|^2}_{U_c} \right.$$

$$\begin{aligned} &s.t. \\ &x_{k+i,k} \leqslant x_{o_{k+1,k}} - \varepsilon_s \\ &v_{\min} \leqslant v_{k+i,k} \leqslant v_{\max} \\ &\left| a_{k+i,k} \right| \leqslant \mu g \end{aligned} \quad (7)$$

- It can be seen that U_{at} and U_c can be written as standard quadratic form. However, U_{re} is nonlinear so that the optimization problem is still nonlinear. Compared with the quadratic convex problem, the nonlinear problem has much more expensive computation cost. Thus, if the repulsive potential function could be written as quadratic convex form, the optimization problem is converted to a quadratic convex problem.

- To convert people use the Taylor method:

$$f(x) = f(x_0) + f'(x_0).(x - x_0) + \frac{f''(x_0).(x-x_0)^2}{2}$$

Choose x_0 = xp(i) =>

$$U_{re}(x) = U_{re}(x_p(i)) + U'_{re}(x_p(i)).(x - x_p(i)) + \frac{U''_{re}(x_p(i)).(x-x_p(i))^2}{2} \quad (8)$$

- Adopting (7) (8), the optimization problem is converted to the standard convex quadratic problem which can be solved by the Sequential Quadratic Programming. At each time step, we can get the desired acceleration for the next time. Then the optimization is resolved at each time step recursively.

- Sequential Quadratic Programming in basic:

Step 1.

Initialization: Choose an initial point close to the optimal solution.

Step 2.

Compute Gradient and Hessian: Calculate the gradient and Hessian of the objective function and constraints at the current point.

Step 3.

Solve Linear Optimization Problem: Use the approximation Hessian and gradient to solve a linear optimization problem to find a new direction.

Step 4.

Check Stopping Condition: Check whether the algorithm has met the stopping condition. If yes, stop and output the current solution. If not, proceed to the next step.

Step 5.

Update: Update the current point by moving a small distance in the new direction found.

Step 6.

Iterate: Repeat the process from step 2 until the stopping condition is met.

3.1.3 Advantages and disadvantages

Advantages:

+ MPC can directly integrate input and output constraints into the optimization process. This is crucial in industrial applications where safety and operational constraints are essential.

+ MPC uses a predictive model of the system to foresee future behavior and makes optimal decisions based on these predictions. This allows it to react to events before they occur.

+ MPC can adjust its inputs to achieve the desired output goals, even under changing environmental conditions. This makes MPC very effective in controlling processes with large delays and complex dynamics.

Disadvantages:

+ MPC requires solving an optimization problem at every time step, which can be computationally expensive, especially with complex systems or when fast response times are required.

3.2 The lower-level controller

3.2.1 Optimal torque vectoring algorithm

- We implement the vehicle longitudinal motion based on the desired acceleration calculated by the upper-level controller.
- With Newton's second law and vehicle longitudinal dynamics, we can calculate the force for each wheel.
- Make sure the slip ratio of each wheel is in a stable region.
- The required longitudinal traction force can be produced by equally distributing the torque to each wheel.

3.2.2 Anti-wheel-slip controller

- Based on the sliding mode control to generate the torque vectoring boundary and also the nonlinear tire model (Magic tire formula) to make the relationship between tire force and slip ratio becomes nonlinear.
- The desired anti-wheel-slip limit can be achieved using the torque vectoring algorithm

4. Work Package Breakdown

4.1 Planning and Preparation

Pre-Ideathon Setup:

- Prior to the start of the Ideathon, convene a preparatory meeting to finalize the roles, understand the tools and resources available, and set up the necessary software and work environment.
- Ensure all members are clear on their roles and the expectations.

4.2 Milestone Planning

Timeline:

- Start-Up Meeting (3 PM, May 6th): Brief team alignment on goals and immediate tasks.
- Mid-Event Check-In (3 AM, May 7th): Review progress, adjust task allocations if necessary, and plan for the final push.
- Final Review (2 PM, May 7th): Finalize all outputs, prepare presentation or submission materials.

Key Deliverables:

- Prototype or proof of concept ready for demonstration.
- Presentation slides or documentation as required by the event rules.

4.3 Task Assignment and Scheduling

Roles and Responsibilities:

- Le Hoang Viet: (Team Leader) Oversees the overall project, ensuring that the team remains on schedule. Responsible for assembling the final presentation and handling any logistical issues.
- Nguyen Tat Dat (Technical Lead): Leads the development and testing of the algorithm. Manages technical problems and ensures the prototype functions as intended.
- An Phuc Hoa (Documentation and Support): Responsible for documenting the development process and prepares presentation materials and ensures all submission requirements are met.

Task Breakdown:

- Hour 1-6: Brainstorming and finalizing the concept.
- Hour 7-18: Development phase, focusing on building the core functionalities.
- Hour 19-23: Preparing presentation/documentation.
- Final Hour: Review and prepare for submission/presentation.

4.4 Monitoring and Adjustments

Continuous Monitoring:

The Team Leader will check in hourly through quick stand-up meetings to ensure tasks are on track and identify any immediate issues.

Adjustment Strategy:

Should a significant delay or problem occur, the team will prioritize core functionalities and scale down features to meet the deadline effectively.

4.5 Communication Plan

Internal Communication:

Use a dedicated Slack channel or similar tool for continuous communication.

Set up a shared document for real-time updates on task progress.

External Queries:

Designate the Team Leader to handle any communications with event organizers or external stakeholders to maintain consistency and manage time effectively.

5. Reference

(1) (PDF) Potential field–based hierarchical adaptive cruise control for semi-autonomous electric vehicle (researchgate.net)

sqp.dvi (ohio-state.edu)