Formula Bharat Challenge

System Intelligence

Problem Statement

An electric vehicle is today like a computer on wheels. Along with the basic functions expected from a vehicle, advanced driver assistance features are now possible thanks to the coordinated working of software, sensors, and actuators. When implemented correctly, they can improve the overall safety of the vehicle, add new capabilities and make vehicles more convenient and fun to operate. When implemented poorly, they can add unnecessary complexity, increase risks and end up annoying the user. Some examples of driver assistance features include Cruise Control, Slip Control, Hill Assist, and Collision Avoidance.

In this assignment, we would like to challenge the Formula Bharat teams to present any one advanced rider assistance system that could suit an electric two wheeler. The feature can be one of the above-mentioned, or some other feature that improves riding ease, excitement or safety of the scooter.

Teams will be judged on how deeply they analyse their chosen feature. They will NOT be judged on the selection of the feature itself. Any rider assistance feature is acceptable.

Here are some points to consider while analysing your chosen feature:

- Is the feature already well-established in cars or elsewhere? Are there multiple variations of it? What variation is most suitable for a scooter and why?
- Under what conditions will the feature be useful?
- What resources would be required from the scooter to make it work? (Eg.: sensors, actuators, energy, processing power)
- Are any hardware modifications required for this feature? For instance, additional buttons, additional sensors, more powerful processor, actuators and so on.
    - What will be the cost impact of this modification – including part, manufacturing and assembly costs?
    - Does it need to be obtained from vendors? What options exist?
    - Would it be possible to manage without it? What will be the impact of this, on the performance of the feature?
- Explain the implementation. **This is the most important part.**
    - Explain the algorithm. This includes fault-handling and edge cases.
    - Provide relevant illustrations and flowcharts.
    - Are there any control loops and system dynamics to consider? Provide relevant mathematical formulations.
- What are the risks of this feature?
    - How can the feature fail? Can it function with reduced performance in some cases?
    - What should be done in the case of a failure (partial or complete) to ensure safety?

- What kind of testing would be required before this feature is considered safe and ready for use?
    - What data would be required to monitor the feature and resolve issues if they come up?
- How might users react to this feature?
    - Should it be deployed in phases? Is feedback needed to improve the feature?
    - Will there be a learning curve for the user? Why?
    - Will it be polarising, with some users loving it and others hating it?

Submission

A presentation format with all details.

Judging Criteria

Parameter	Weightage
Analysis of requirements	20%
Explanation of implementation	40%
Analysis of risks and testing	20%
Analysis of user experience	10%
Overall depth of analysis and quality of presentation	10%