2021

Mitigating Insider Threats in a Cooperative Adaptive Cruise Control System Using Local Intra-Vehicle Data

Alexander Francis Colon
West Virginia University, afcolon@mix.wvu.edu

Recommended Citation

Colon, Alexander Francis, "Mitigating Insider Threats in a Cooperative Adaptive Cruise Control System Using Local Intra-Vehicle Data" (2021). *Graduate Theses, Dissertations, and Problem Reports*. 10241.
https://researchrepository.wvu.edu/etd/10241

Mitigating Insider Threats in a Cooperative Adaptive Cruise Control System Using Local Intra-Vehicle Data

Alexander Francis Colon

Thesis submitted to the Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science with Area of Emphasis in Cybersecurity

Brian Woerner, Ph.D., Chair
Roy Nutter, Ph.D.
Patrick Browning, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2021

Keywords: CACC, V2X, Internal, Local, Security

ABSTRACT

Mitigating Insider Threats in a Cooperative Adaptive Cruise Control System Using Local Intra-Vehicle Data

Alexander Francis Colon

With the rise of Connected-and-Automated-Vehicle (CAV) technologies on roadways, transportation networks have become increasingly connected through Vehicle-to-Everything (V2X) systems. With access to the additional data from V2X, modern cruise control systems like Adaptive Cruise Control (ACC) are further improved upon to develop systems like Cooperative ACC (CACC) which reduces traffic congestion and increases driver safety and energy efficiency. With that increased connectivity, previously closed vehicle systems are now vulnerable to new security threats which pose new technical challenges. Significant research has been done to strengthen the network against external threats such as denial-of-service attacks (DoS) or passive eavesdropping attacks using network management and cryptographic strategies. Internal threats like data falsification are more challenging to address because they originate from already authenticated sources on the network.

This work suggests a method to locally determine if network data can be trusted utilizing only the intra-vehicle sensors against the network data. It functions by leveraging the synchronization of CACC vehicle stream members to identify potentially malicious data. In the event the network data is determined to be untrustworthy, the vehicle will change its mode of operation to basic ACC where it will disconnect from the vehicle stream and increase the distance from the preceding vehicle. In order to test this approach, an ACC system was created and then modified into a simple CACC system that includes the V2X network data streams. Two common V2X applications were used to show the functionality of both the simple CACC system and the work: a Vehicle-to-Infrastructure (V2I) enabled traffic light and a Vehicle-to-Vehicle (V2V) vehicle stream.

Acknowledgements

I would like to begin by thanking Dr. Woerner for his continued support, patience, and for the opportunity to work with and learn from him as a graduate research assistant on the EcoCar project. I gained a lot of experience in working with him and the other team members that will prove to be invaluable in my next ventures. I also want to give thanks to my other committee members, Dr. Nutter and Dr. Browning for their input and guidance working on this research topic. Lastly, I give thanks to my friends and family for constantly pushing me to do my best in all that I do.

List of Figures

Acronyms

ACC Adaptive Cruise Control

BD Bidirectional Topology

CACC Cooperative Adaptive Cruise Control

CAV Connected and Automated Vehicle

CW/CA Collision Warning/ Collision Avoidance

DSRC Dedicated Short Range Communication

DoS Denial of Service

MPF Multiple-Preceding-Following Topology

PF Predecessor-Following Topology

PID Proportional Integral Derivative

PLF Predecessor-Leader-Following Topology

VAN Vehicle Area Network

V2B Vehicle-to-Broadband Cloud

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

Table of Contents

1 Introduction

As transportation infrastructure continues to be modernized and more Connected and Automated Vehicle (CAV) technologies are introduced on roadways, vehicle systems are becoming more connected than ever. This increased connectivity between vehicles and infrastructure aims to provide ways of improving traffic flow, energy efficiency, and driver safety. Transportation networks, or Vehicle Area Networks (VANs), are now closer to that of computer networks as cities develop infrastructure to become "smart" cities. While this technology does have its benefits, it comes at the cost of opening previously closed vehicle systems to external attacks.

As is the case with computer networks, VANs are also vulnerable to cyber-attacks that can result in instability or loss of function in the network. Instability in these networks can eliminate the benefits of the network connectivity and in some cases cause collisions. Vehicles that rely on the positional information from the network data will be more likely to violate their safe distance boundaries and activate their Collision Warning or Collision Avoidance (CW/CA) systems to prevent accidents. The threats that can impact these networks can originate from both outside and inside the VAN. Significant work has been done to identify and manage external threats. Internal threats, however, are more difficult to address as they come from already authenticated sources.

This paper suggests a method of using the many on-board vehicle sensors to validate the received network data locally and provide a way for the control system to determine whether the data should be trusted even if it originated from an already authenticated source. This idea of providing a way for the control system to continue to function prior to violating safe distance boundaries will lead to vehicles being more robust against abnormalities in the network data regardless of why they are present. To test this system, the approach was designed with a rudimentary Adaptive Cruise Control (ACC) algorithm that was modified to a simplified approach for Cooperative ACC (CACC) within Simulink. The model was tested using two common applications in VANs: interacting with a lead vehicle in a three-member vehicle stream and a traffic light that is transmitting data to the VAN.

1.1 Structure of the Paper

Section 2 addresses background information and relevant work covering the different aspects of CACC including the technology and challenges associated with it. Section 3 outlines the experimental procedures used to develop and simulate the work proposed in this paper. Section 4 presents the simulation results of the experiments and discusses the takeaways from those results. Sections 5 and 6 summarize the work of the paper and reflect on areas that could use further research and development.

2 Background

2.1 Vehicle Area Networks

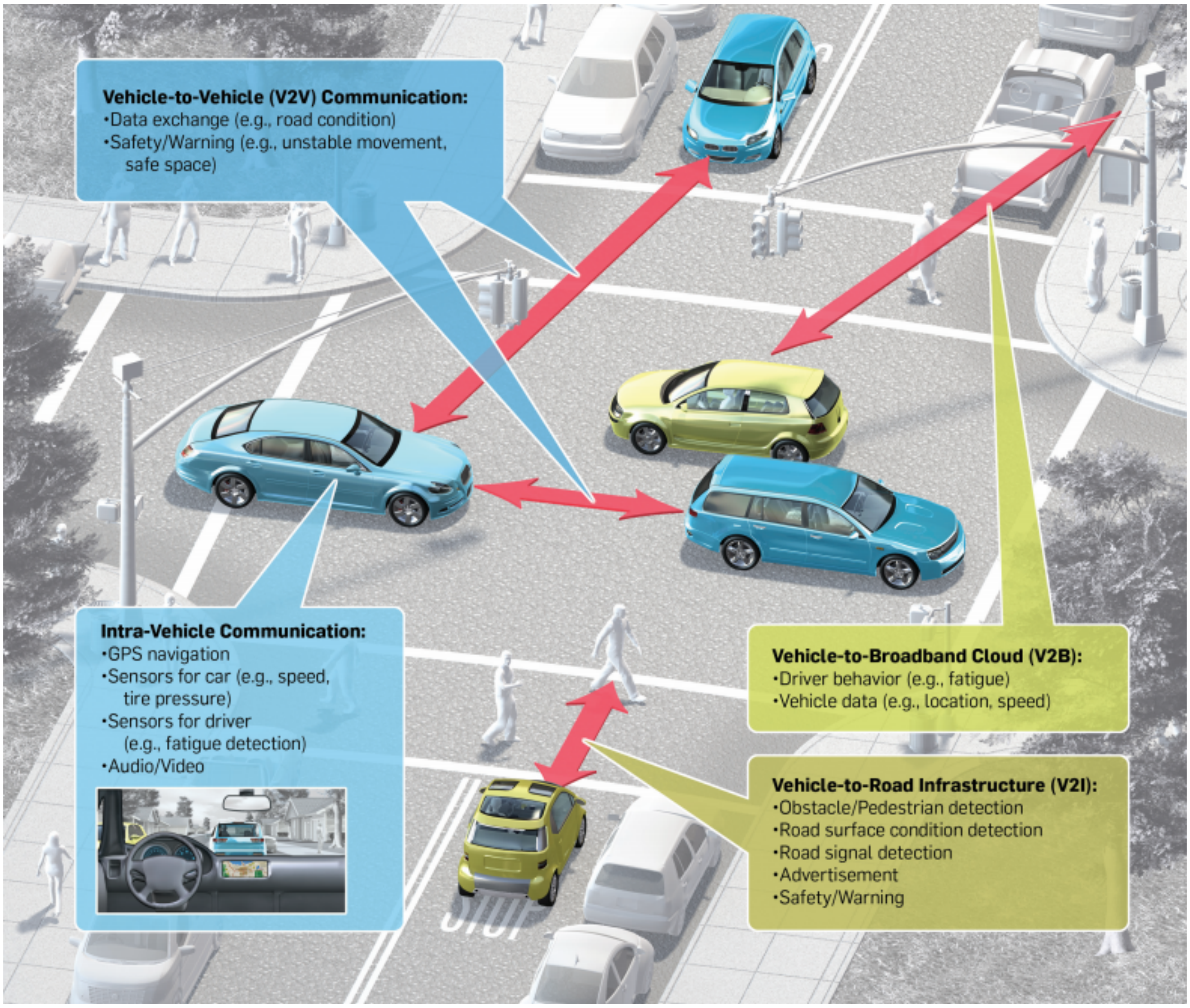

Vehicle-to-Vehicle (V2V) Communication:
•Data exchange (e.g., road condition)
•Safety/Warning (e.g., unstable movement, safe space)
Intra-Vehicle Communication:
•GPS navigation
•Sensors for car (e.g., speed, tire pressure)
•Sensors for driver (e.g., fatigue detection)
•Audio/Video
Vehicle-to-Broadband Cloud (V2B):
•Driver behavior (e.g., fatigue)
•Vehicle data (e.g., location, speed)
Vehicle-to-Road Infrastructure (V2I):
•Obstacle/Pedestrian detection
•Road surface condition detection
•Road signal detection
•Advertisement
•Safety/Warning

Figure 1: Control Vision of Vehicle Area Network [1]

Vehicle Area Networks (VANs) can be divided into four different areas as illustrated in Figure 1 above: Intra-Vehicle communications, Vehicle-to-Broadband Cloud (V2B) communications, Vehicle-to-Vehicle (V2V) communications, and Vehicle-to-Infrastructure (V2I) communications [1]. Intra-Vehicle communications include the data from the host, or ego, vehicle's onboard sensors and hardware transmissions. V2B communications are used to transmit ego vehicle data to cloud services. V2V communications are data exchanges between the ego vehicle and another vehicle which includes safety messages and road condition information. V2I communications are the data exchanges with roadside technologies like crosswalks and traffic lights. V2B, V2V, and V2I communication technologies are commonly grouped together under the term Vehicle-to-Everything (V2X) communications.

V2X technology is often referred to as the next step to intelligent transportation systems as it improves the functionality of current vehicle systems by expanding the environmental data that is available to the vehicle [2]. With the increased perception of the environment ahead of the vehicle, systems can be designed to be more anticipatory to changes in road conditions that would go unseen by vehicle sensors. Applications in this area generally function using Dedicated Short-Range Communication (DSRC) techniques that follow SAE J2735 standards [15]. These standards outline many messages but specify an especially important message type referred to as the basic safety message. This message type contains information about the ego vehicle's id, position, speed, heading, etc. in Appendix A, and is constantly broadcast to nearby receivers in other vehicles or roadside units. The data in the basic safety message acts as the core component of many V2X applications. One such application being CACC.

2.2 Cruise Control Technologies

Cruise control systems have been around for many years and are now considered to be a standard feature in consumer vehicles. Conventional cruise control systems regulate the acceleration of the vehicle to maintain a driver-set speed with the goal of reducing driver fatigue. As new technologies have been developed, cruise control systems have improved as well to meet the needs of modern consumers and work towards accident-free roadways. ACC systems, for example, are an advanced version of cruise control that emphasizes maintaining a safe distance from a preceding vehicle that is travelling slower than the ego vehicle's set speed.

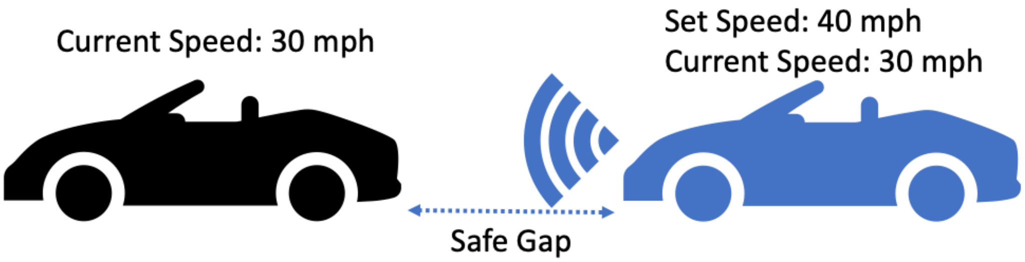


Figure 2: Adaptive Cruise Control Diagram

Figure 2 illustrates this by showing the ego vehicle, in blue, slowing down to reach a safe distance and match the slower speed of the preceding vehicle, in black. ACC systems further reduce driver fatigue by eliminating the need for the driver to adjust the speed manually to prevent collisions with other vehicles, and, in doing so, also improve driver safety, energy efficiency, and driver comfort when compared with conventional cruise control systems. It functions using intra-vehicle ranging sensors like radar to perceive a preceding vehicle's speed and location relative to the ego vehicle. That information is then used to command the ego vehicle as needed by a given situation. In the event there is no preceding vehicle present, the system will act as a basic cruise control system would by only maintaining the driver-set speed. Once a slower preceding vehicle appears, the system will adjust the ego vehicle's speed to reach a safe distance and match that vehicle's speed automatically. ACC systems are more effective in highway scenarios as opposed to complex urban environments due to their limited anticipatory capabilities to sudden maneuvers from the preceding vehicle [3]. The introduction of V2X technology improves those capabilities by extending the vehicle's ability to get information from vehicles and roadside transmitters ahead of the preceding vehicle. This V2X data can be used to extend ACC into CACC.

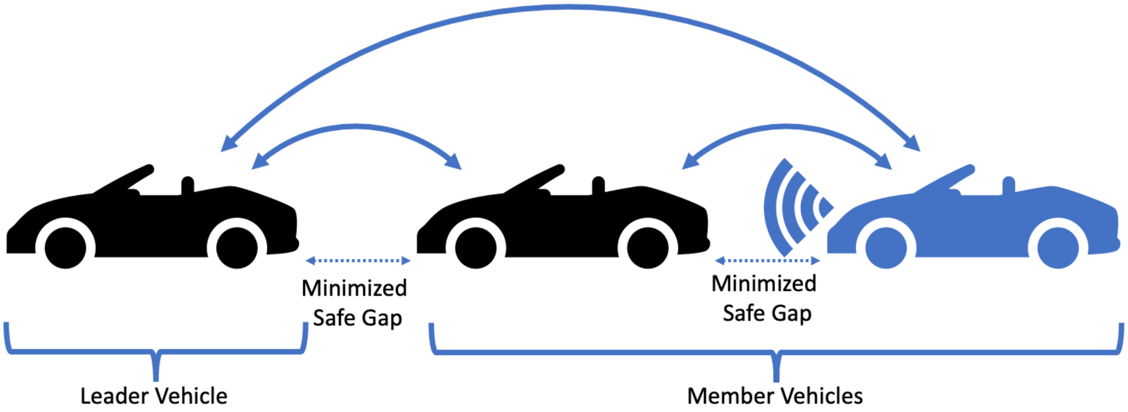
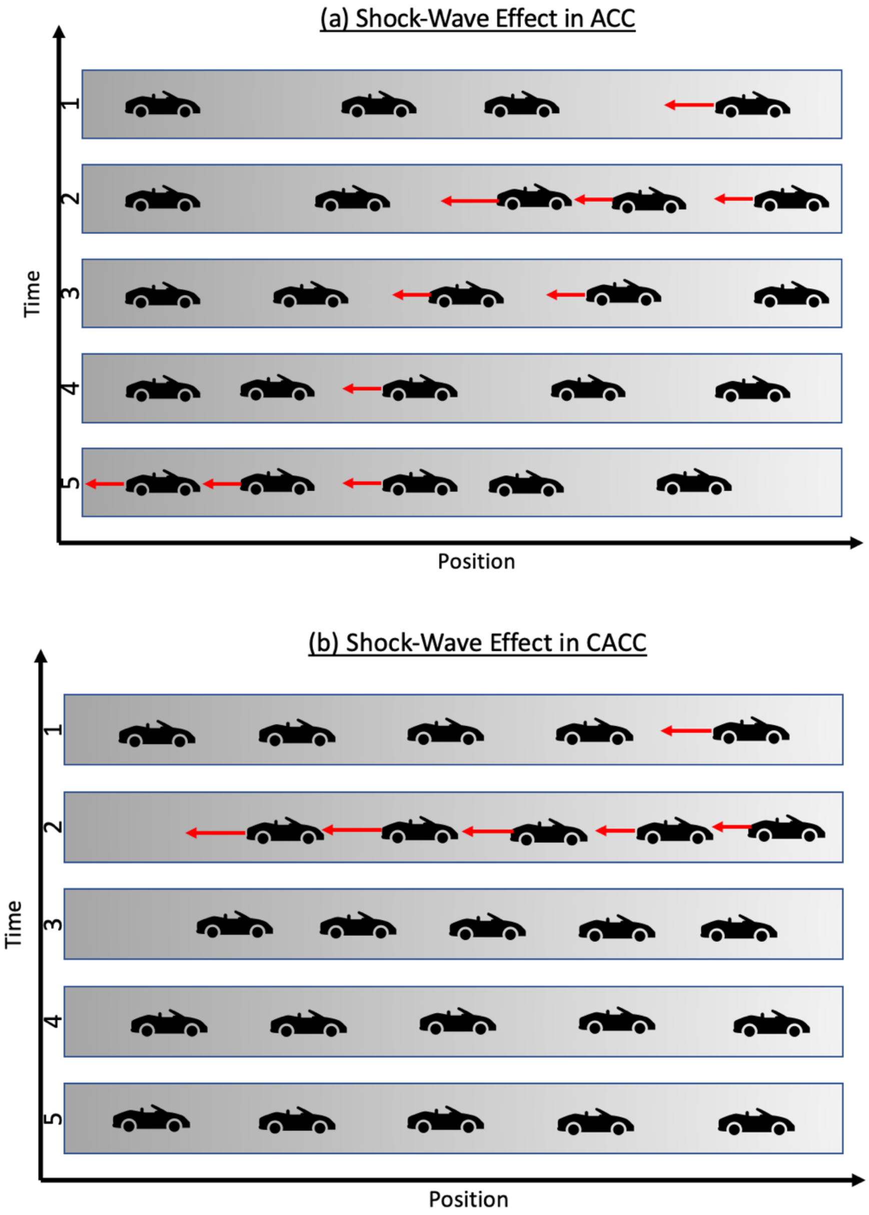


Figure 3: Cooperative Adaptive Cruise Control Vehicle Stream

CACC connects multiple vehicles together into a unit called a vehicle stream illustrated in Figure 3. A vehicle stream consists of a single lane of vehicles with a lead vehicle in front and any number of following vehicles as its members. This system utilizes V2V information that is transmitted between stream members using DSRC techniques with the addition of the intra-vehicle sensor data that ACC functions on. With the speed and positional data provided in the basic safety message, each member vehicle can match the lead vehicle's speed and behavior resulting in a tightly clustered group of vehicles with improved response times to sudden events [4]. The minimized time gap between member vehicles that this system enables maximizes road capacity, traffic flow, and studies have shown that CACC can meaningfully reduce energy consumption and air pollution [5]. Additionally, CACC systems address a shortcoming of ACC by mitigating the shock-wave effect found in congested vehicle scenarios [6].



(a) Shock-Wave Effect in ACC

(b) Shock-Wave Effect in CACC

Figure 4: Shock-wave Effect: (a) ACC System, (b) CACC System

The shock-wave effect in ACC, shown in Figure 4a, is the result of delays between the sensor data and the system response. When the lead vehicle begins to slow down, only the vehicle directly behind it will identify the change and begin to slow down. This braking event will then propagate to the remainder of the vehicles over a time period. Figure 4b shows how the functionality of CACC works to mitigate that using the data from the V2V network. In this case, the lead vehicle notifies subsequent vehicles when it slows down so they can do the same which reduces the propagation delay and results in a smoother and more efficient drive for passengers. The direction of this flow of information across the vehicle stream is how different CACC algorithms are classified as it dictates how the vehicle stream will respond to sudden events such as this.

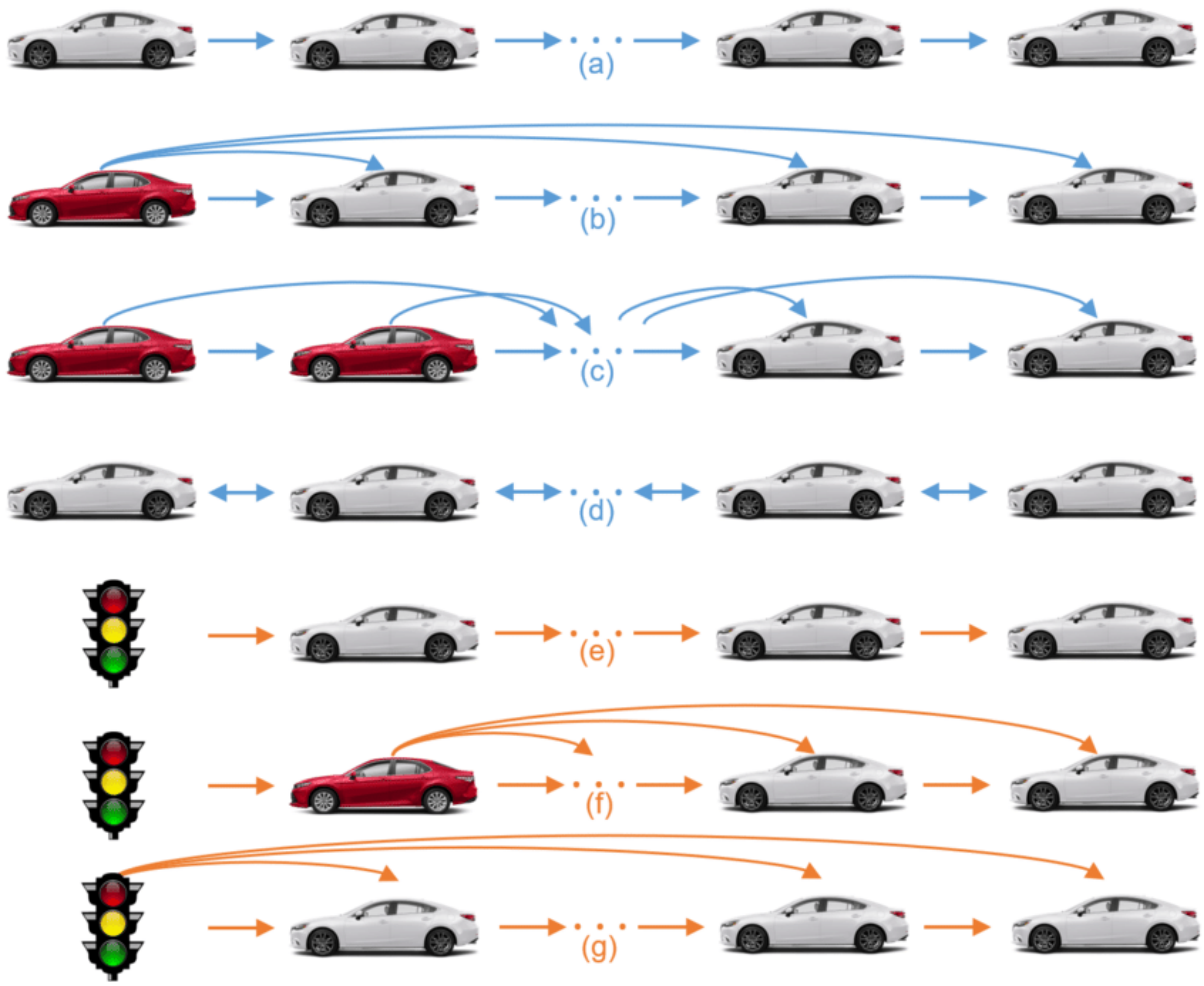


Figure 5: Information Flow Topologies: (a) predecessor-following (PF), (b) predecessor-leader-following (PLF), (c) multiple-predecessor-following (MPF), (d) Bidirectional (BD), (e) PF with infrastructure sending information to leader, (f) PLF with infrastructure sending information to leader, (g) PF with infrastructure broadcasting information [7]

The different communication flow topologies, shown in Figure 5 above, are as follows: Predecessor-Following (PF), Predecessor-Leader-Following (PLF), Bidirectional (BD), and n-Preceding-Following. Each communication topology has been proved to be string-stable in [8], which means any disturbances propagated from the lead vehicle to the rest of the vehicle stream can be attenuated [9]. This paper will only consider PLF as the CACC topology in the interest of simplicity. With this topology, information from the lead vehicle will be transmitted to all vehicles in the vehicle stream. But string stability in real-world environments can be impacted by both unpredictable road conditions as well as security threats.

2.3 Security Concerns in Cooperative Adaptive Cruise Control

The benefits of CACC are gained by achieving stability between the members of the vehicle stream. When a vehicle stream loses its stability, the safe distance parameters of the members are likely to be violated which can initiate the CW/CA systems to prevent a collision at the cost of driver comfort. The stability can be deliberately disrupted by both internal and external network threats shown below.

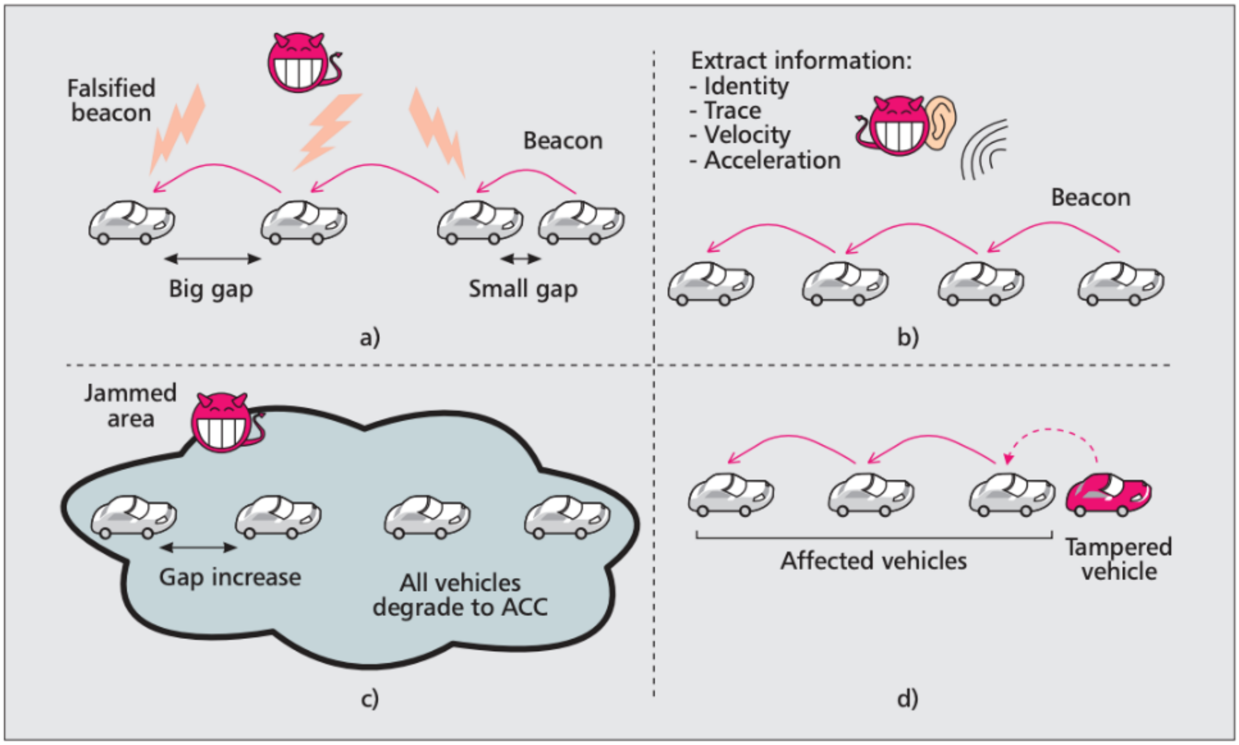


Figure 6: Security Attacks on a CACC vehicle stream: a) falsification attack, b) eavesdropping attack, c) radio jamming attack, d) tampering attack [10]

Figure 6 represents four possible attacks on a CACC vehicle stream. External threats like the ones represented in Figures 6b and 6c refer to attacks from outside of the vehicle network. Figure 6b shows an eavesdropping attack where the attacker seeks to extract information from the steam. In this case encryption techniques are used to prevent access or anonymity techniques using group signatures [11] or short-term certificates [12]. Figure 6c shows the effect of radio jamming or denial of service attacks on the vehicle network. In this case, vehicles would be unable to properly form or maintain a platoon because they would be unable to transmit or receive any data through the network, so they default to ACC functionality. Here, researchers consider a system to detect these attacks [13] in combination with other network countermeasures like frequency hopping to prevent the disruption. Internal threats like the ones in Figures 6a and 6d are attacks from within the network by an already authenticated entity. These are the type of attack this work focuses on addressing. Figure 6a represents a falsification or spoofing attack where an attacker modifies components of a broadcasted message like the basic safety message or acts as a nonexistent vehicle in the stream with the goal of disrupting the vehicle stream stability. Figure 6d represents a compromised lead vehicle where the data to be transmitted in the basic safety messages is altered before it is sent. There are many practical challenges to addressing internal threats due to the scale and interpretation of what constitutes "security and privacy" [10]. Typical approaches to the problem of trusted insiders involve some level of anomaly detection technique like the one in [14] which requires multiple streams of data from other sources. [14] considers a compromised lead vehicle disseminating false acceleration information which would lead to instability and potential collisions. The authors propose an information sharing model and real-time anomaly detection mechanism that uses the leader's information and the information from the vehicles around the leader.

3 Methodology

This paper proposes a design methodology that locally validates network data based on the data available from the intra-vehicle sensors to further address internal threats and reduce vehicle reliance on network data. Two common V2X applications will be used to evaluate the effectiveness of the approach: 1) interacting with a V2I enabled stoplight and 2) interacting with a lead vehicle in a V2V vehicle stream. The first application is a simple implementation using different scenarios interacting with a traffic light that is broadcasting different status information. This establishes the premise of the design methodology. The second application considers the different speed information a lead vehicle could disseminate across a vehicle stream and applies the design methodology in a more complex example. In order to test this design, an ACC system needed to be developed and then modified into a simple CACC system.

3.1 Adaptive Cruise Control Model

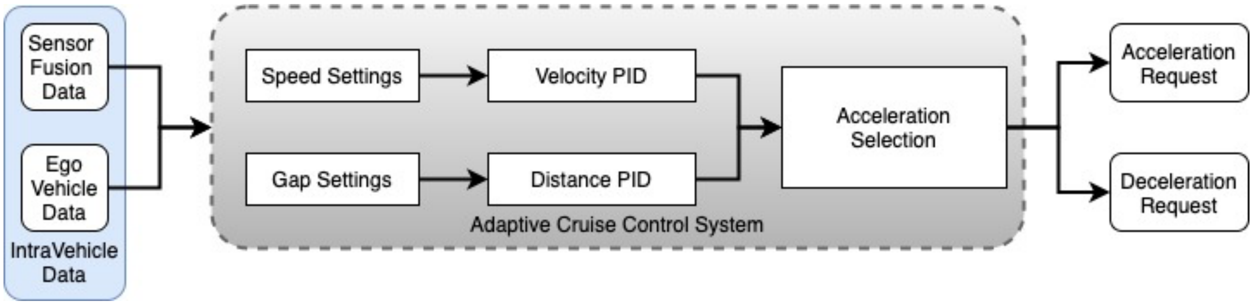


Figure 7: ACC System

The initial control algorithm was a rudimentary ACC system developed in Simulink which is represented in Figure 7. The system used a combination of sensor fusion data with ego vehicle information to output acceleration and deceleration requests. These requests were then looped through a longitudinal vehicle model to update dynamic vehicle information. The value of each request is determined by the distance error and velocity error. The distance error is the difference between the relative distance from the preceding vehicle and the calculated safe distance boundary between the two vehicles. The velocity error is the difference between the driver selected vehicle speed and the current vehicle speed. The error values for both distance and velocity were then input into separately tuned Proportional, Integral, Derivative (PID) controllers to determine the best acceleration to reach zero error [16]. The velocity error would primarily drive the acceleration requests when there is no preceding vehicle presently in view. Once a target vehicle is recognized, the request is the minimum acceleration between the velocity and distance error until the vehicle reaches the calculated safe distance boundary. Once the safe distance is reached, the distance error drives acceleration requests to maintain the safe distance to the preceding vehicle.

3.1.1 ACC System: Speed Settings

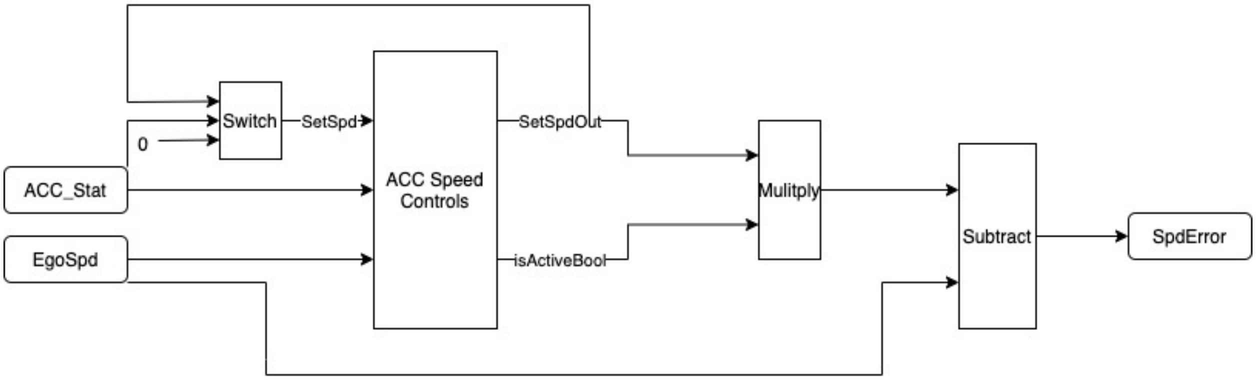


Figure 8: ACC System, Speed Settings Subsystem

The Speed Settings subsystem of the ACC system enables the driver interface controls such as setting and modifying the vehicle speed as well as toggling the ACC system. Seen in Figure 8 above, this subsystem takes in the current speed of the ego vehicle and a control variable called ACC_Stat. ACC_Stat bundles the driver interface functionality into a single control variable. That includes setting or adjusting value of the set speed and turning the ACC System on or off. Once the command to set the speed has been received, the current ego speed is captured and stored. When the system is active, the new set speed is compared to the current ego speed and the difference is used as the speed error that is passed to the velocity PID. Other things to note in this system are the status check to make sure ACC is currently active and the set speed feedback loop. The status check is a redundancy that will nullify any output if the system is currently off. The feedback loop is used to adjust the set speed to the desired level according to the current value of ACC_Stat.

3.1.2 ACC System: Gap Settings

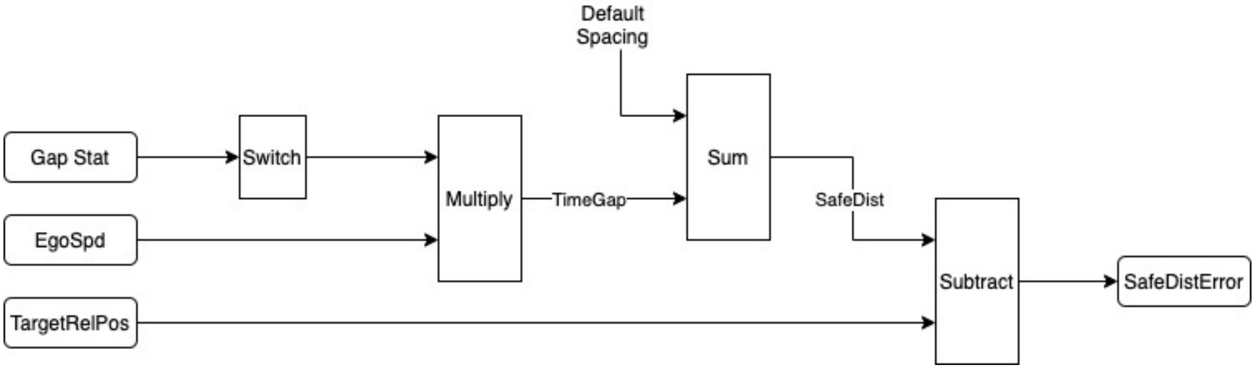


Figure 9: ACC System, Gap Settings Subsystem

The Gap Settings subsystem, illustrated above, is used to calculate the safe distance from the preceding vehicle and calculate the difference between that safe distance measure and the current distance. The gap stat input determines which distance setting is currently used between Near-1, Medium-2, Far-3. The gap stat input is multiplied by the speed of the ego vehicle to a time-to-impact gap value. This result is added to a default spacing buffer to calculate the safe distance from the preceding vehicle. The difference between the safe distance value and the relative position of the preceding vehicle is the safe distance error used in the Distance PID.

3.1.3 ACC System: Acceleration Selection Block

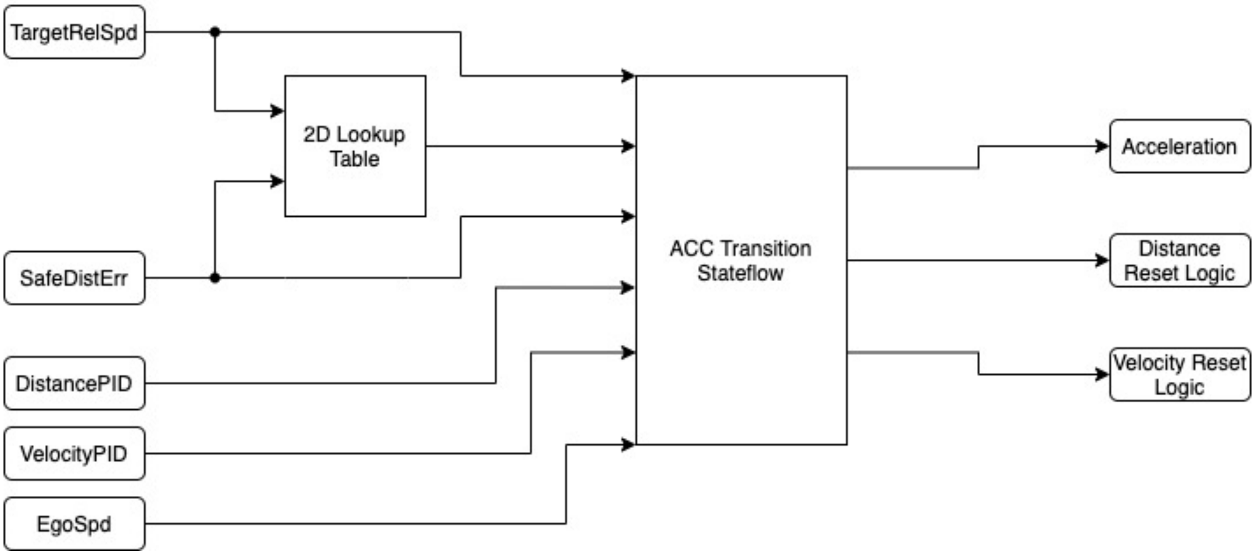


Figure 10: ACC System, Acceleration Selection Subsystem

The acceleration selection subsystem determines which controller's output will be honored by the vehicle model. Using the ego vehicle speed, relative target speed, calculated safe distance error, and outputs of the two PIDs, this subsystem uses a state flow diagram to switch between each controller. Additionally, this subsystem outputs reset logic for each PID controller.

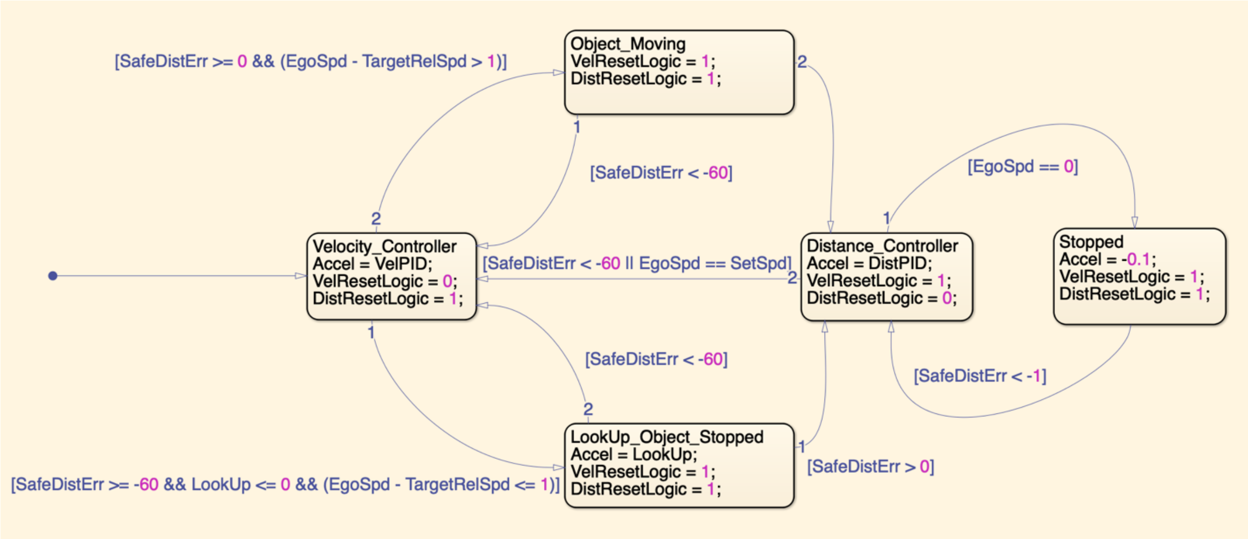


Figure 11: ACC System, Acceleration Selection Subsystem, State Flow Diagram

Figure 11 shows the two controller states and three intermediate states in between them. The velocity controller is the first controller because the initial goal of ACC is to maintain the driver selected speed. This controller passes the output of the Velocity PID and outputs reset logic to the Distance PID. The reset logic is important as it prevents a phenomenon in PID feedback controllers known as integrator windup. Integrator windup occurs when the integrator component of the controller accumulates a significant amount of error. The accumulation happens as a result of the absence of feedback to the Distance PID controller since only the Velocity PID controller output is being honored by the system. This is prevented here by resetting the accumulated error of the integrator to 0.

The state will change to one of the two transitionary states when a preceding vehicle is detected. If the vehicle appears to be moving, the state will briefly change to the object moving state and then immediately to the distance controller. If the vehicle appears to not be moving, the state will change to the object stopped state where acceleration control will be dictated by the 2-D lookup table seen below.

Distance (m) Speed (m/s)	-1	0	2	5	15	30	60
0	-0.2	-0.07	0	0	0	0	0
1	-0.2	-0.1	-0.05	0	0	0	0
3	-0.2	-0.1	-0.07	-0.001	0	0	0
6.66	-0.3	-0.1	-0.09	-0.005	0	0	0
10	-0.6	-0.4	-0.1	-0.1	-0.05	0	0
13.33	-0.8	-0.5	-0.2	-0.15	-0.15	0	0
16.66	-0.9	-0.7	-0.5	-0.5	-0.4	-0.1	0
20	-1	-1	-0.8	-0.7	-0.6	-0.3	0
23.33	-1	-1	-0.9	-0.8	-0.7	-0.4	-0.1
26.66	-1	-1	-1	-0.9	-0.8	-0.5	-0.1
30	-1	-1	-1	-1	-0.9	-0.6	-0.1

Figure 12: ACC System, Acceleration Selection Subsystem, 2D Lookup Table

The two dimensions in the table are distance from the preceding vehicle on the x-axis and the speed of the ego vehicle on the y-axis. Each of the values in the table was empirically tested based on the desired stopping behavior of the ACC system. When the ego vehicle is going slower it won’t have to start slowing down as quickly and can begin to stop closer to the preceding vehicle. The faster the ego vehicle is going as it approaches the preceding vehicle, the harder the vehicle will decelerate.

In either case, once the ego vehicle reaches the safe distance boundary, the state will shift to the distance controller and the distance PID’s requests will be honored by the system. At this point the reset logic will be applied to the velocity PID controller as the vehicle is now operating using the distance parameter. If the ego vehicle comes to a complete stop the state will shift to the stopped state which is a quality-of-life addition to keep the deceleration command constant. The state will shift back to the velocity controller once the preceding vehicle is no longer detected, or the ego vehicle reaches the set speed.

3.1.4 ACC System: System Performance

Three different scenarios were used to evaluate this ACC system’s performance: approaching a stopped vehicle, approaching a constantly moving vehicle, and following a dynamically moving vehicle. Each scenario generalizes a specific aspect of a full-range ACC which is a version of ACC that can bring the vehicle to a complete stop as opposed to ACC that can only be used above 25mph. Each is illustrated in the following figures.

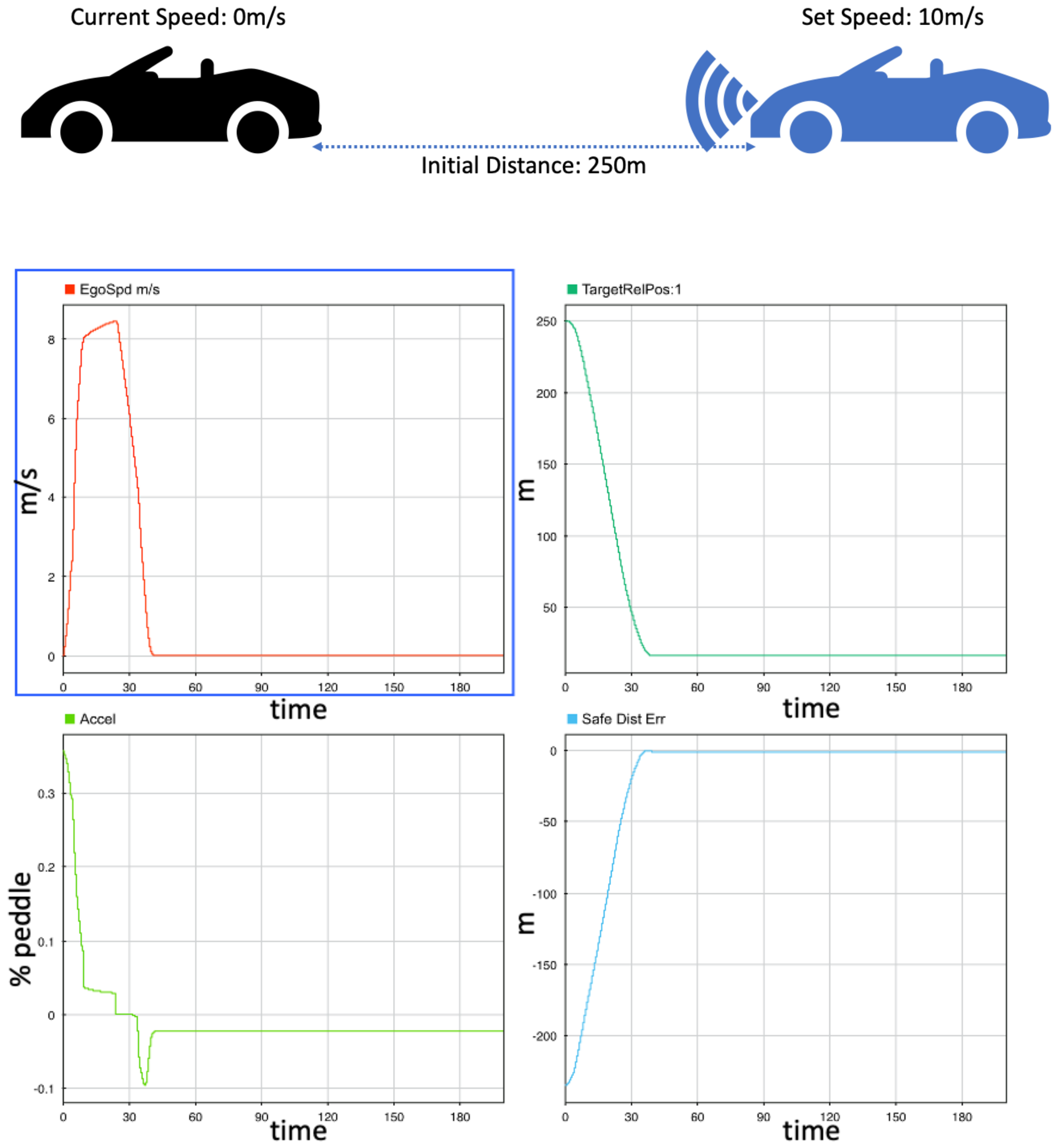


Figure 13: ACC System, Drive Cycle 1, Stopped Lead Vehicle

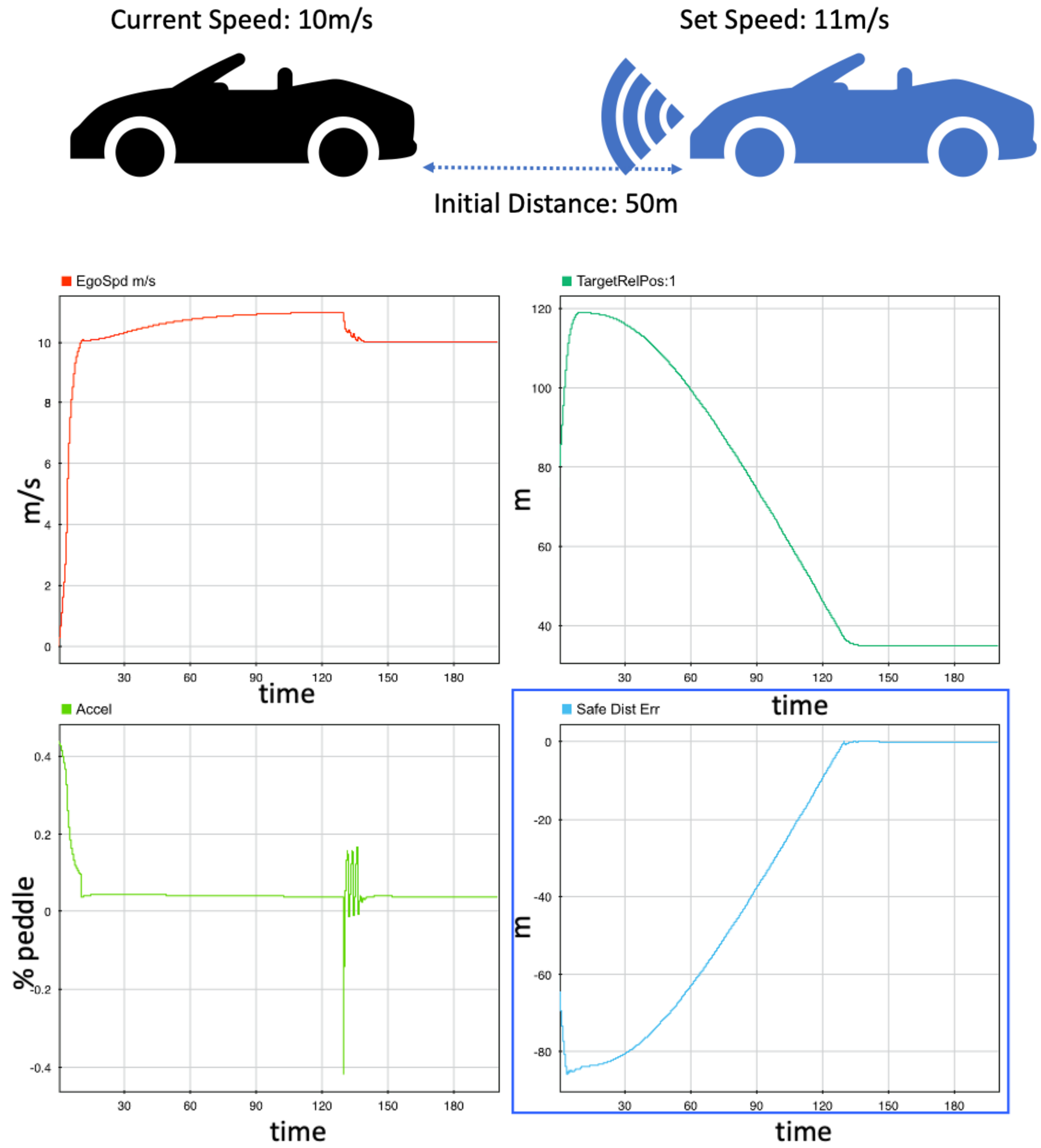


Figure 14: ACC System, Drive Cycle 2, Lead Vehicle with Constant 10m/s

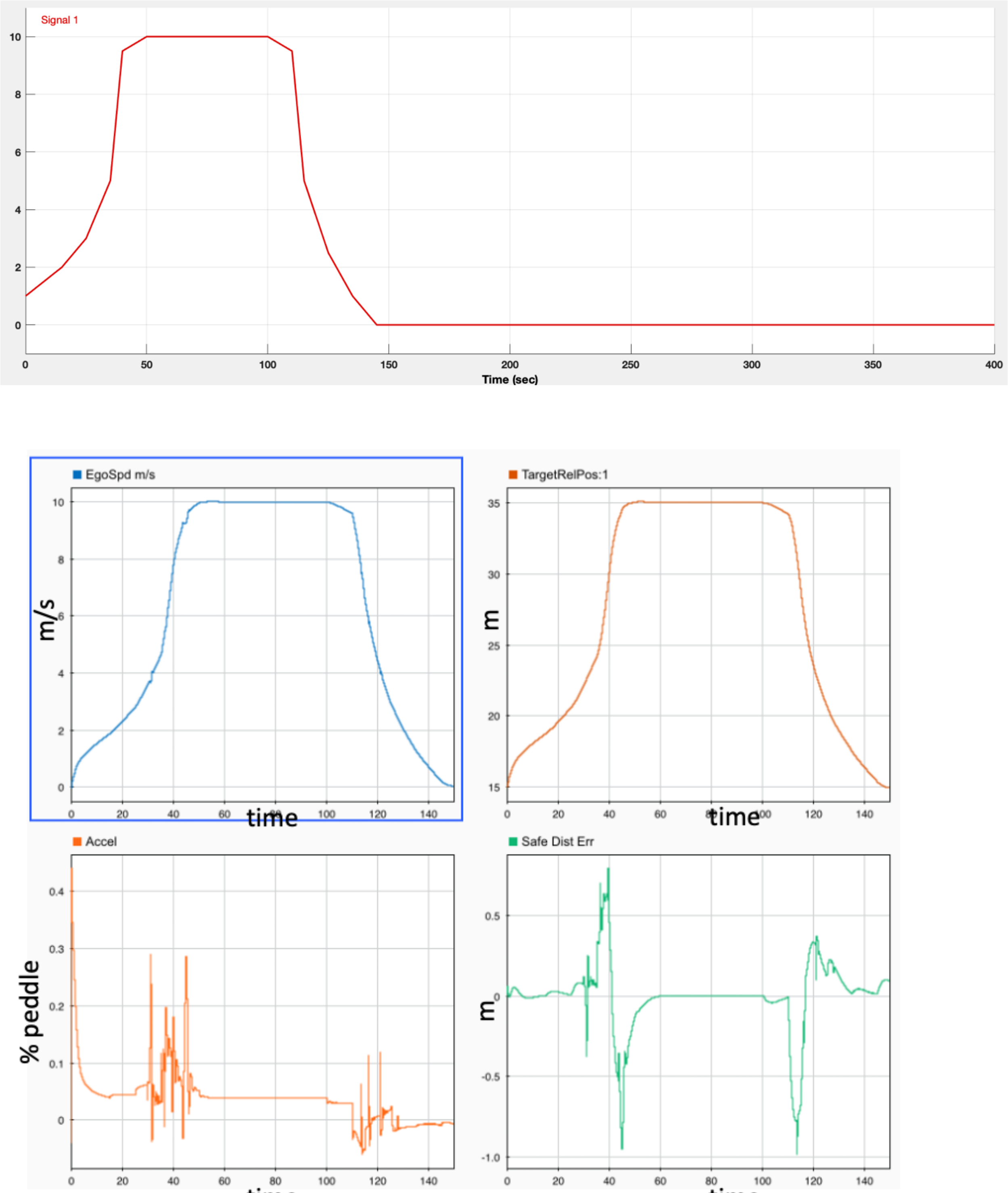


Figure 15: ACC System, Drive Cycle 3, Follow Lead Vehicle Speeding Up 10m/s then Stop

The performance of each scenario is determined using the ego speed, acceleration values, target position, and safe distance. The first scenario depicted in Figure 13 involves the ego vehicle speeding up to its set speed and approaching a stopped vehicle at an initial distance of 250m. In the acceleration graph, the output shows the ego vehicle initially accelerating to its set speed until it detects the stationary vehicle. At this point, the acceleration control switches to the object stopped state and the ego vehicle begins to slow down. In the second scenario, shown in Figure 14, the ego vehicle speeds up to its set speed and encounters a vehicle moving at a constant 10m/s. The acceleration oscillation is caused by control switching from the velocity controller to the distance controller due to the integrator component resetting until that point. Soon after taking control, acceleration values normalize and smooth out. Figure 15 shows the final scenario with the ego vehicle following a moving vehicle experiencing varying speeds. Control is immediately switched to the distance controller and the ego vehicle follows at the speed varying safe distance away. The two points of large oscillation in acceleration values are the result of instability due to the preceding vehicle's changing speed. This could be resolved with additional tuning of the distance controller, but the test is still a success as the acceleration values are not too extreme.

It is worth noting that this ACC system continued to be tested and evaluated on hardware with the WVU EcoCar team. Using an Intel Tank as the processor, this system received data from a Bosch front radar and Intel MobilEye camera and generated acceleration requests to a MABX controller. It was then retested once the hardware was integrated into a vehicle and functioned as expected.

3.2 Simple Cooperative Adaptive Cruise Control System Modifications

With the ACC model functioning, now V2X functionality needed to be included to enable communication with a traffic light and additional vehicle data from a lead vehicle. To accomplish the goals of the design, the ACC control strategy was modified into a simple CACC control strategy. It will be limited to longitudinal controls with a single lead vehicle as the vehicle stream leader and a single preceding vehicle in between the lead and ego vehicles. Traffic light cases do not consider edge cases where the traffic light will change as the ego vehicle approaches the intersection. The traffic light will broadcast positional information and light status information: Green-1, Yellow-2, Red-3. The lead vehicle will broadcast positional and speed information. With these changes, the algorithm will instead use the V2X broadcasted information as opposed to the intra-vehicle sensor data until specific criteria are violated. Like the response to jammed network data discussed earlier, the system will default to ACC functionality if no network data is present.

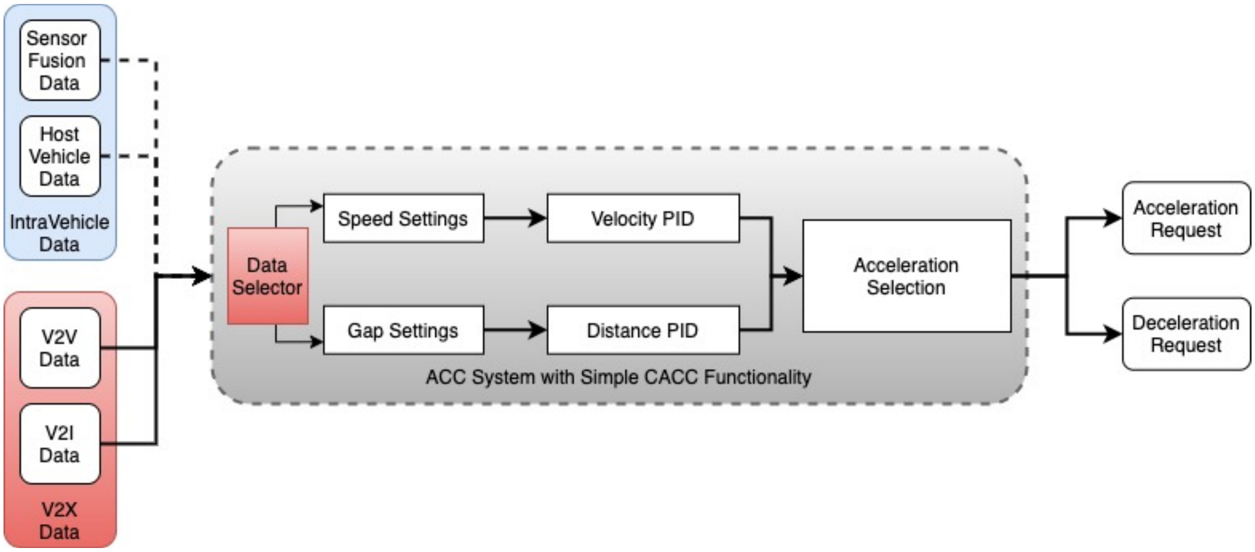


Figure 16: Simple CACC System

Depicted above in Figure 16 is the simple CACC system used for testing this security application. Designed on top of the existing ACC system, the key changes highlighted are the inclusion of additional data streams representing the broadcasted V2X information as well as a data selector block that can switch between ACC and CACC control strategies as needed. Along with these additions, existing blocks in the ACC system were modified to accommodate the extended functionality.

V2I traffic light data was simulated in a similar way to a static preceding vehicle. It broadcasts a distance from the ego vehicle along with a status indicator to specify the current traffic light status. In addition to the broadcasted V2I data, a camera sensor input was simulated using positional data to imitate the shorter detection range of the camera. V2V lead vehicle data was simulated using a combination of the preceding vehicle's dynamics and the adjusted ego vehicle's speed. The result is lead vehicle data for a vehicle in front of the preceding member vehicle that is traveling at the same speed as the ego vehicle. Both inputs are used to define the time gap between the three vehicles.

3.2.1 CACC System: Data Selector

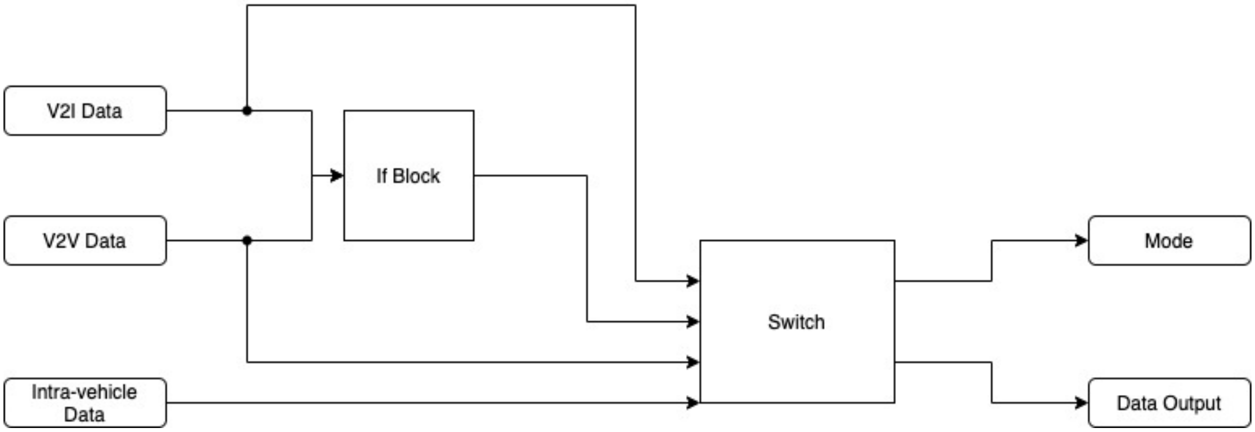


Figure 17: Simple CACC System, Data Selector Subsystem

The Data Selector subsystem was designed to check for the presence of data from the V2I and V2V data streams. It accomplishes this using an if-elseif-else block that checks each input for a nonzero value where it will determine the mode of operation: ACC Functionality-1, Traffic Light-2, CACC Functionality-3. Once that is determined, the switch will select the appropriate dataset to pass downstream to the remainder of the system. The mode of operation is also output to control additional variables in later subsystems. In addition to the data selector block, subsystems downstream needed to be modified to accommodate the new functionality.

3.2.2 CACC System: V2I Traffic Light Modifications

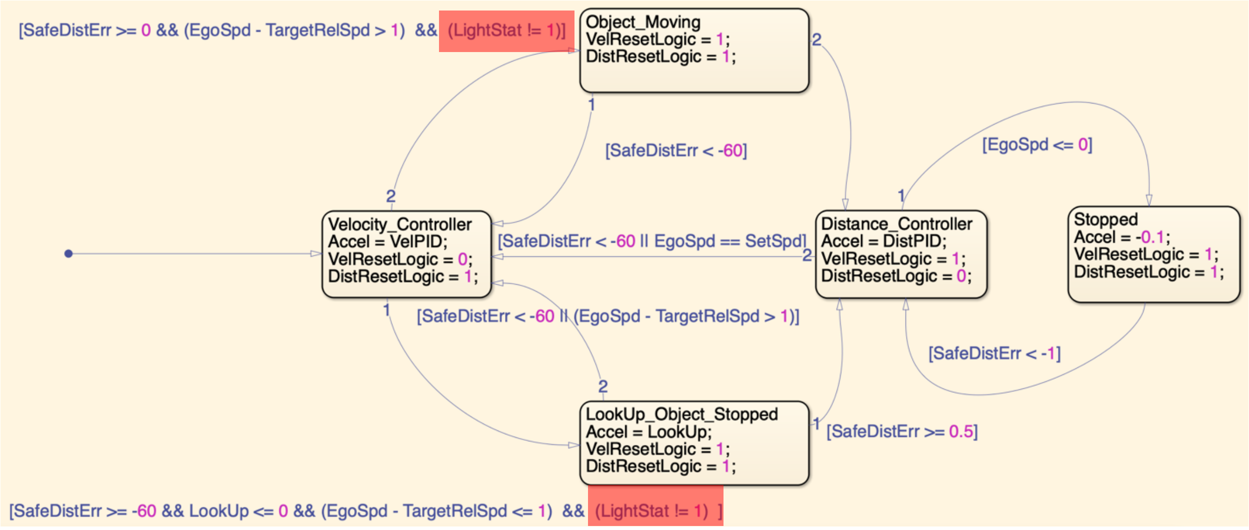


Figure 18: Simple CACC System, Acceleration Selection, State Flow Diagram, Modified

The traffic light functionality required modifications in both the acceleration selection subsystem state flow chart and the gap settings subsystem. The state flow diagram in the Acceleration Selection subsystem, shown in Figure 18, was adjusted to allow the system to stop the ego vehicle or continue based on the status of the light. An additional requirement was included to keep the velocity controller in control of the ego vehicle in the event the light status reads Green-1. If it is not green, then the ego vehicle needs to switch to the lookup table and distance controllers to stop the vehicle smoothly before the intersection is reached.

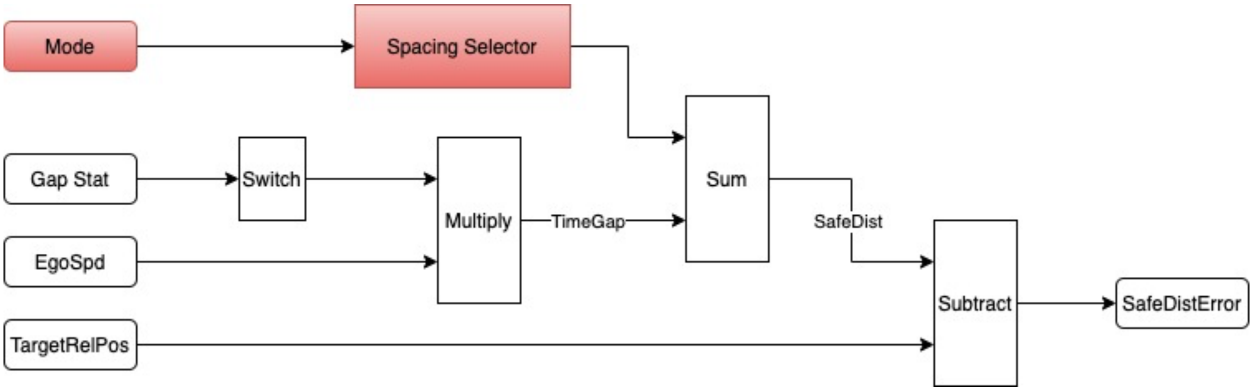


Figure 19: Simple CACC System, Gap Settings Subsystem, Modified for V2I

In the Gap Settings subsystem, shown above, a new block was created to switch between default spacing values to ensure the vehicle stops with an appropriate distance from the object based on the current mode of operation. In the case of the traffic light, the ego vehicle should stop at the intersection as opposed to the default spacing defined in the ACC algorithm, so traffic light functionality requires a default spacing of zero.

3.2.3 CACC System: V2V Vehicle Stream Modifications

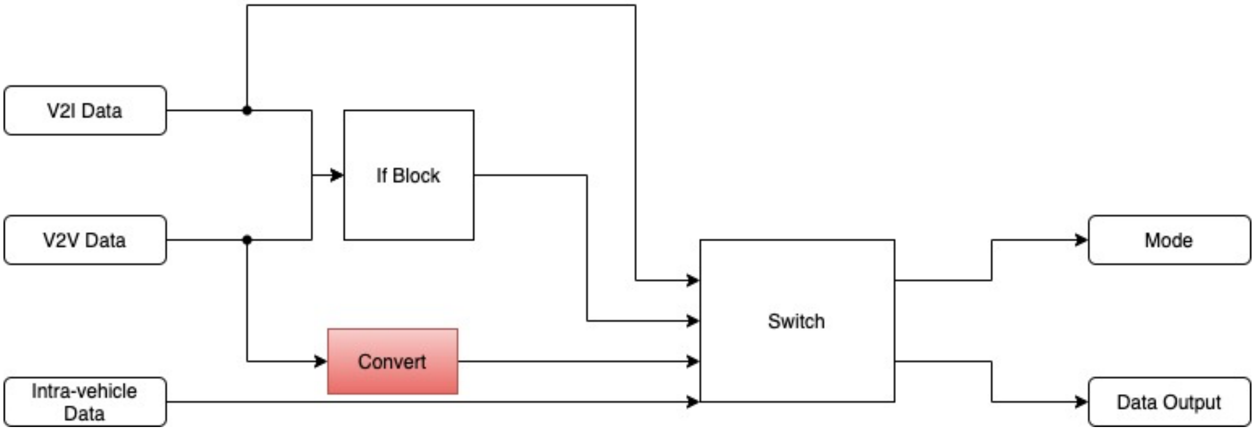


Figure 20: Simple CACC System, Data Selector Subsystem, Modified

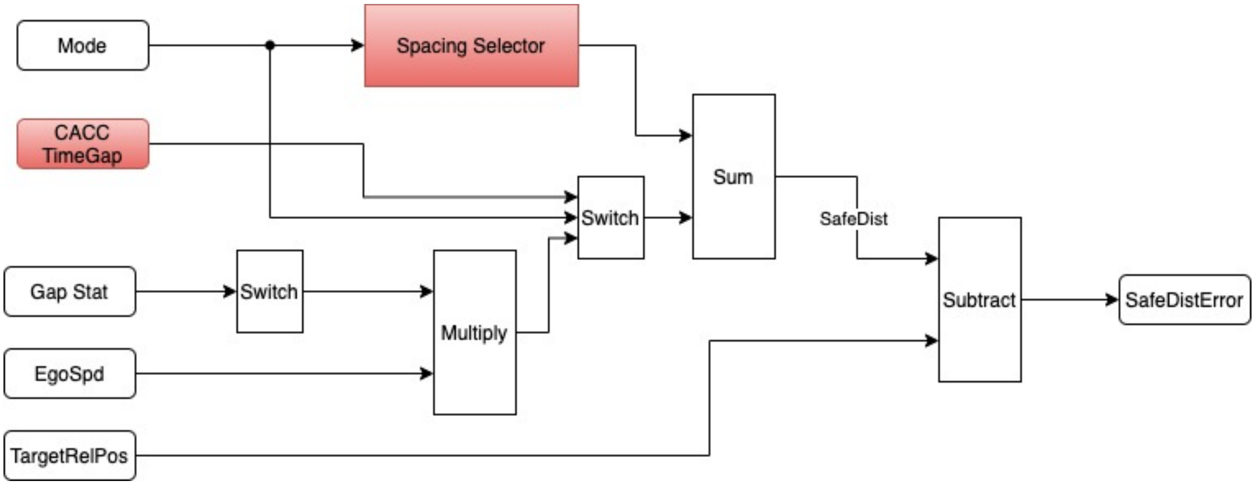


Figure 21: Simple CACC System, Gap Settings Subsystem, Modified for V2V

To enable V2V vehicle stream functionality, the lead vehicle's data was converted to a time gap to be passed to the Gap Settings subsystem. The time gap is input into the distance controller with the preceding vehicle's position data. An additional default spacing setting for CACC was also added to the spacing selector. In terms of the Acceleration Selection subsystem, only the distance controller was used to maintain the time gap between vehicles. A second state flow chart was created to simplify the design. Functionally, the vehicle is solely controlled by the distance controller to maintain the time gap.

3.2.4 CACC System: System Performance

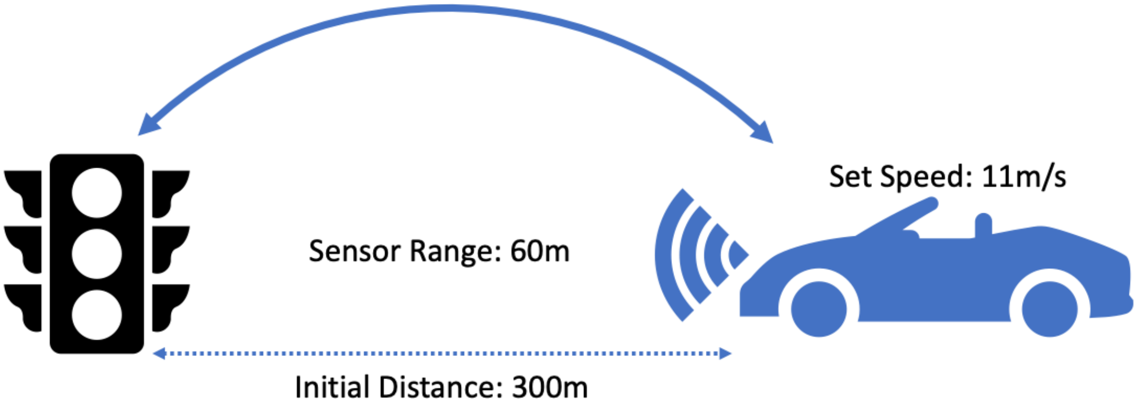


Figure 22: Simple CACC System, Traffic Light Scenario

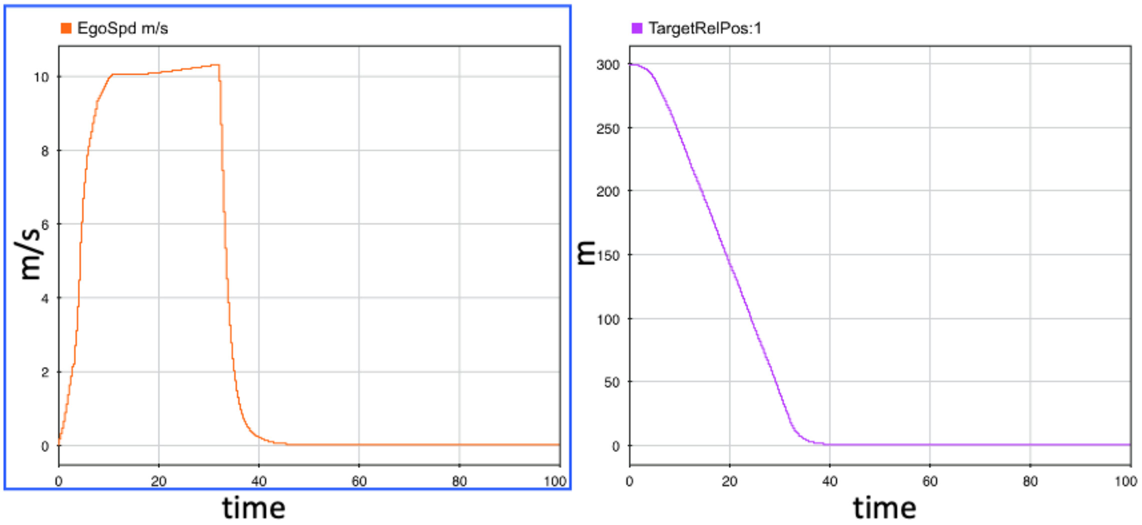


Figure 23: Simple CACC System, Traffic Light Performance – RED

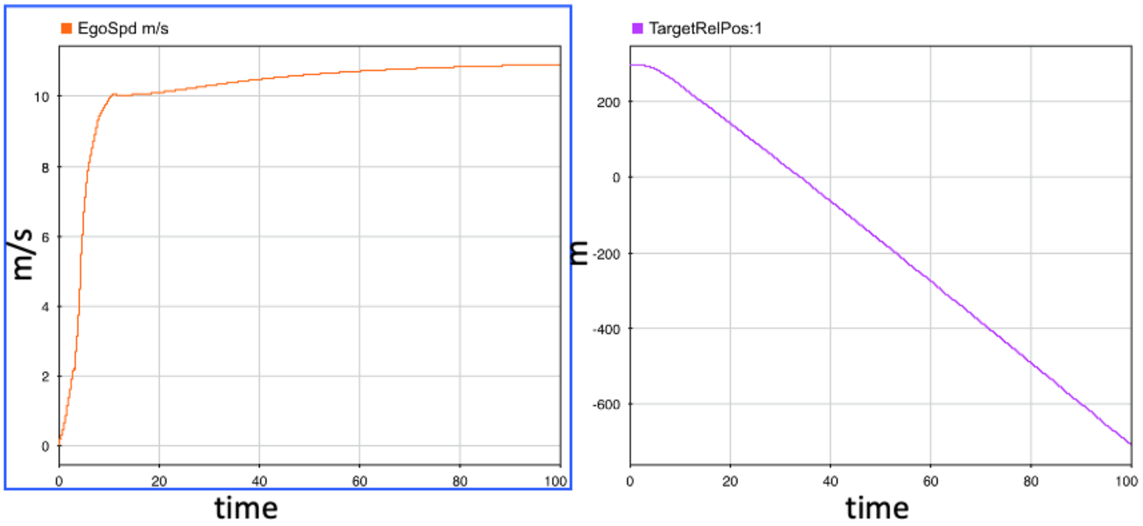


Figure 24: Simple CACC System, Traffic Light Performance – GREEN

Figure 23 shows the ego vehicle's performance when encountering a red traffic light. At approximately 30 seconds into the run the ego vehicle applies the brakes to slow the vehicle down as it reaches the intersection. Conversely in Figure 24, the green traffic light case, the ego vehicle maintains its set speed and drives through the light without issue. This functionality will be important in explaining the basic premise behind this strategy of managing internal threats. Interacting with traffic lights include a limited number of cases when excluding outside factors like other vehicles or edge cases.

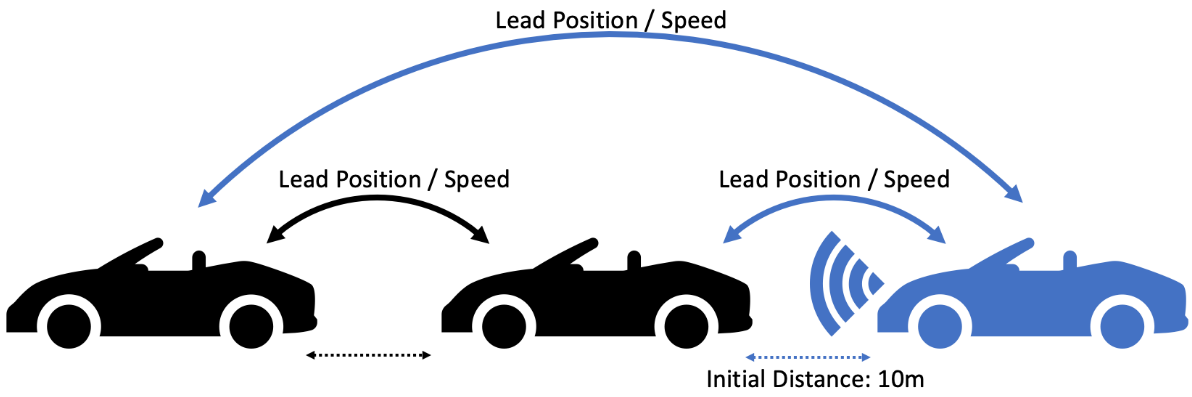


Figure 25: Simple CACC System, V2V Vehicle Stream Scenario

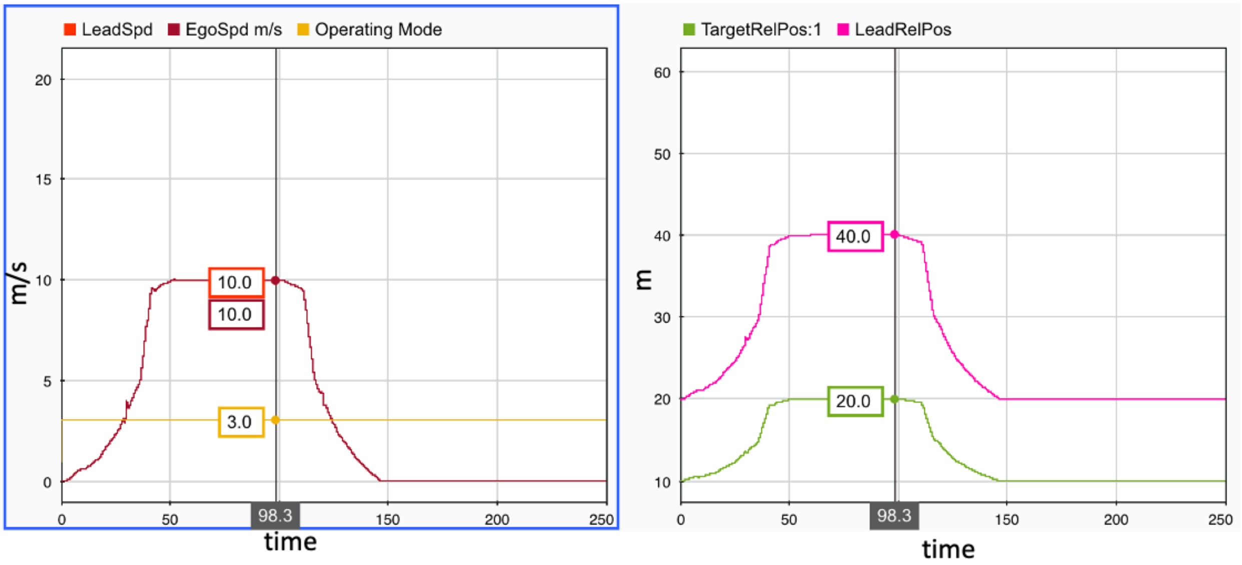


Figure 26: Simple CACC System, V2V Vehicle Stream Performance, Speeding Up 10m/s then Stop

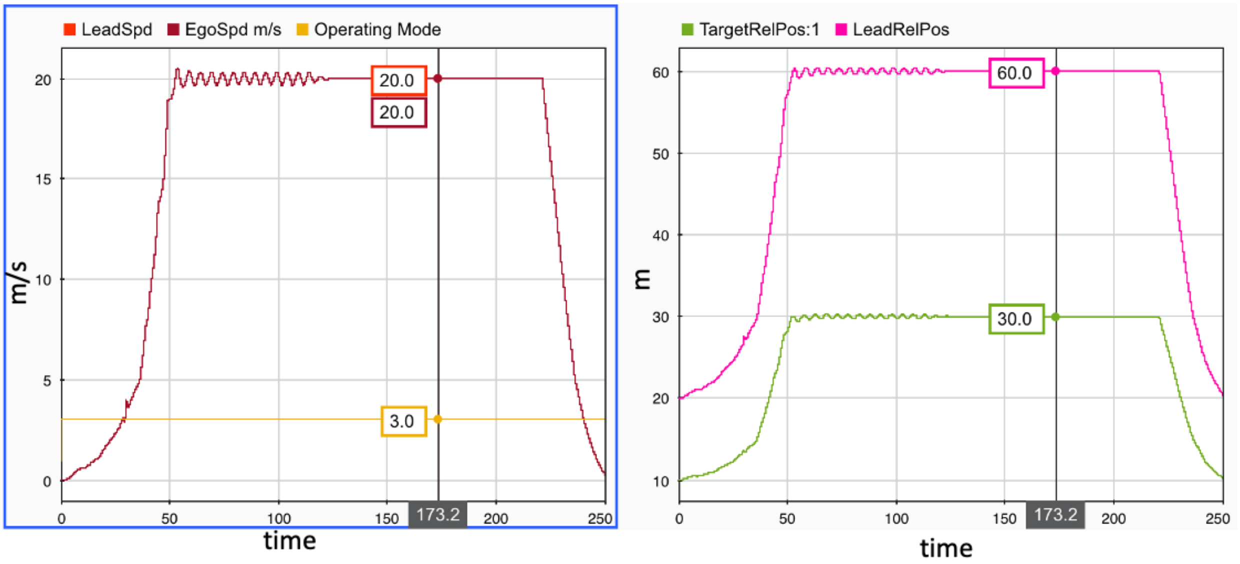


Figure 27: Simple CACC System, V2V Vehicle Stream Performance, Speeding Up 20m/s then Stop

Both Figures 26 and 27 show the effects the lead time gap has on the spacing between each vehicle. This is CACC in action. As the Lead vehicle slows down, the time gap will reduce, and the vehicles will get closer together. If the lead vehicle accelerates, the time gap will increase, and the vehicles will have more space between them. In the event the vehicles get too close and violate an arbitrary constant distance parameter of 2m, the system will enter CA/CW and disable the CACC system. It is a redundancy built into other CACC systems, so it was included. The oscillation seen in Figure 27 seems to be the combined result of the distance controller not being tuned for CACC functionality and sensitivity to high speed changes.

3.3 Identifying Erroneous Data

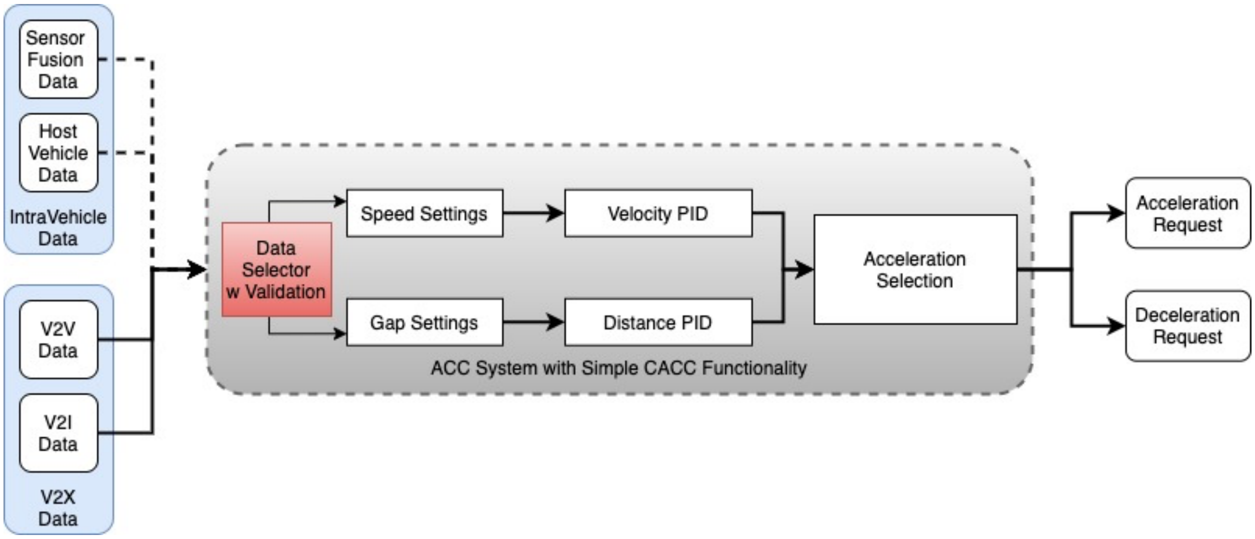


Figure 28: Simple CACC System with Data Validation

Now, with a semi functioning CACC system to use for testing, the proposed method for further addressing internal security vulnerabilities could be included. This was implemented into the data selector block, as shown in Figure 28, where the data from each stream can be used for validation. In addition to the range finding intra-vehicle sensors, this approach also assumes to have the use of camera technologies that can detect traffic lights and their statuses like that of the MobilEye camera system.

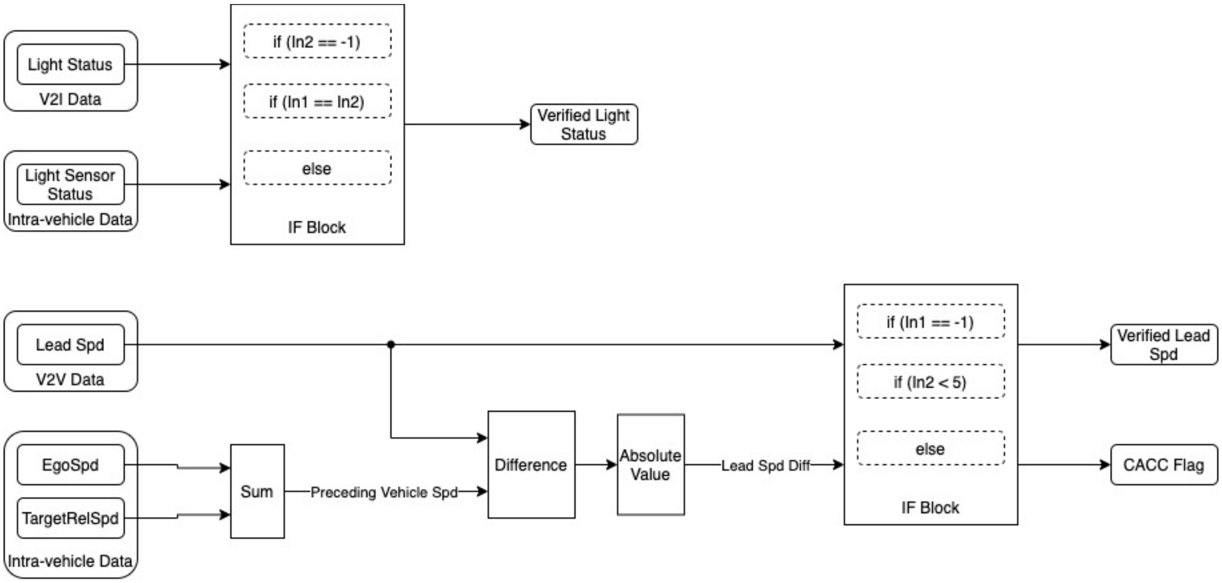


Figure 29: Data Validation

To implement this method, data inputs were compared with a 5m/s tolerance of error of between values. In the case of the V2V vehicle stream, the positional and speed data from the broadcasted safety message for both the lead and preceding vehicles were compared against the intra-vehicle sensor inputs based on sensor observations of the preceding vehicle. If the data was relatively close, the system would proceed normally as a CACC system. Alternatively, if there is a larger discrepancy in the values, the system would then operate as an ACC system. In the case of the V2I enabled traffic light, the broadcasted positional and status information is compared against the camera detection values to determine light status accuracy. If the camera is not in range of the traffic light, the system can only use the network data available.

This design method leverages the synchronization between vehicles in an active vehicle stream to detect irregularities in transmitted data using intra-vehicle sensor data. In the event the lead vehicle is broadcasting information that differs from the observed behavior of the preceding vehicle, the system can flag that as an anomaly regardless of why it is present. Whether it is falsified or erroneous data, once the anomaly is detected, the system can preemptively change the control strategy to ACC. This would result in more distance from the preceding vehicle, improve driver safety, and maintain driver comfort as it avoids instability in the stream. This system behavior also mimics human behavior in terms of creating more distance between vehicles in uncertain situations and environments.

4 Simulation Results

In order to test the functionality of the proposed method, different scenarios are considered within the two test applications. These scenarios will have varying degrees of potential consequences, however, if the system prevents them from occurring by either switching to ACC control or continuing normal CACC functionality, it will be deemed a success. The first application, the V2I traffic light, will be used to explain the simple operation of the design method: the ego vehicle either will or will not stop for the light based on the perceivable data. The second, the V2V vehicle stream, will be applied in a more complex application. The ego vehicle's behavior will be more subtle here than in the traffic light scenarios.

4.1 Application 1: V2I Traffic Light Simulation Results

Figure 30: Application 1, V2I Enabled Traffic Light

Several different scenarios can take place when interacting with a potentially malicious V2I enabled stoplight.

Test Num	Light Status	Network Status	If Ego Trusts Network Information	If Ego Does Not Trust Network Information
1	Green	Green	Vehicle continues driving normally	Would trust once light is in range
2	Green	Red	Vehicle brakes for light - Driver could disengage and proceed - Possible accident if rear vehicle isn't paying attention	Vehicle identifies green light and proceeds normally
3	Red	Red	Vehicle brakes for light	Would trust once light is in range
4	Red	Green	Vehicle drives through intersection - High likelihood of accident	Vehicle identifies red light and stops at intersection
5	Fake Roadside – No Light	Red	Acts as if light is present	Identifies the absence of the light and ignores

Figure 31: Application 1, V2I Enabled Traffic Light Scenario Table

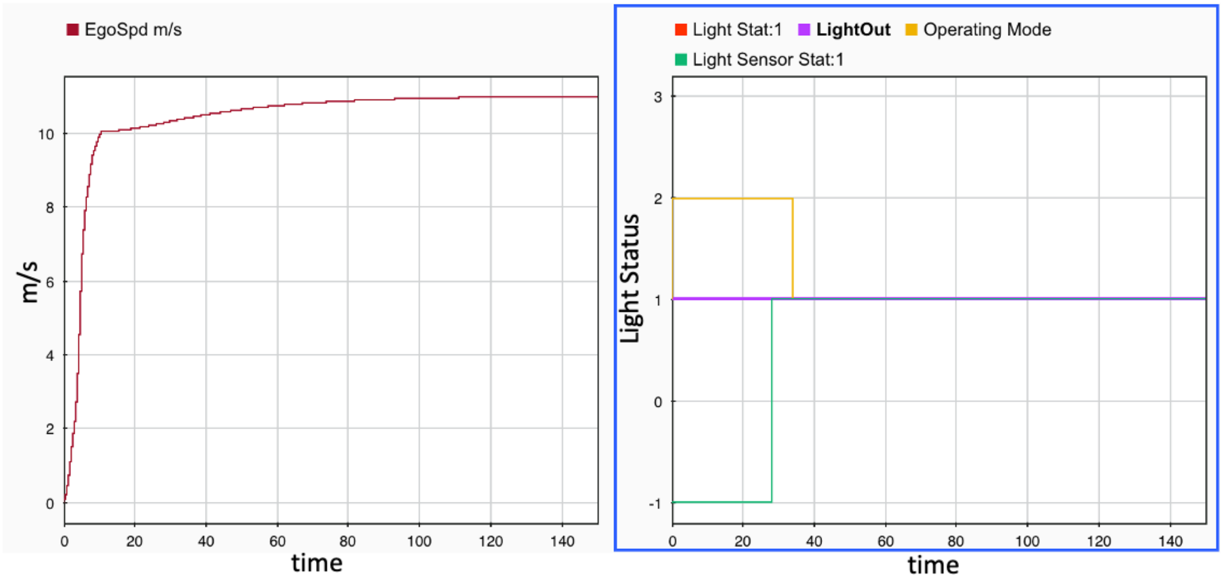


Figure 32: Traffic Light Application Performance, Test 1, Network-GREEN Sensor-GREEN

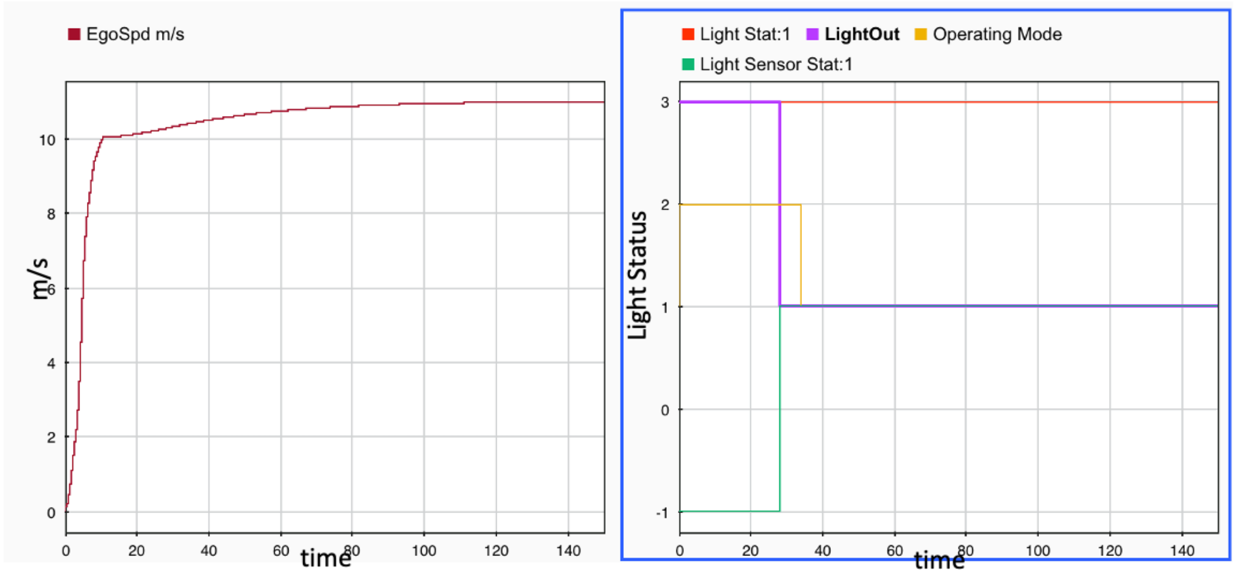


Figure 33: Traffic Light Application Performance, Test 2, Network-RED Sensor-GREEN

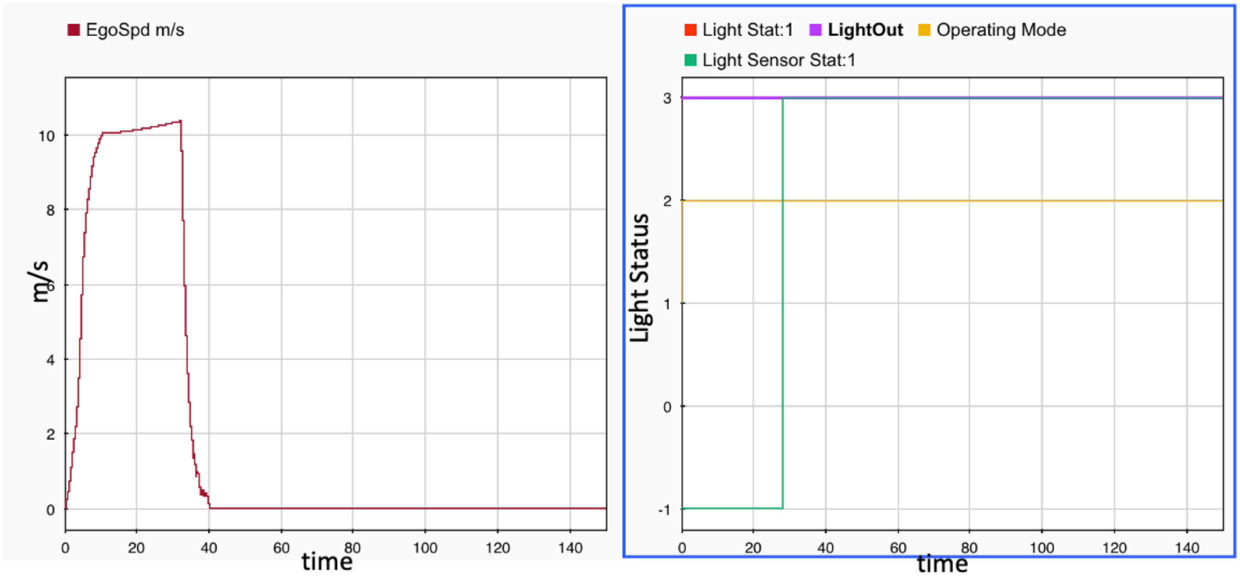


Figure 34: Traffic Light Application Performance, Test 3, Network-RED Sensor-RED

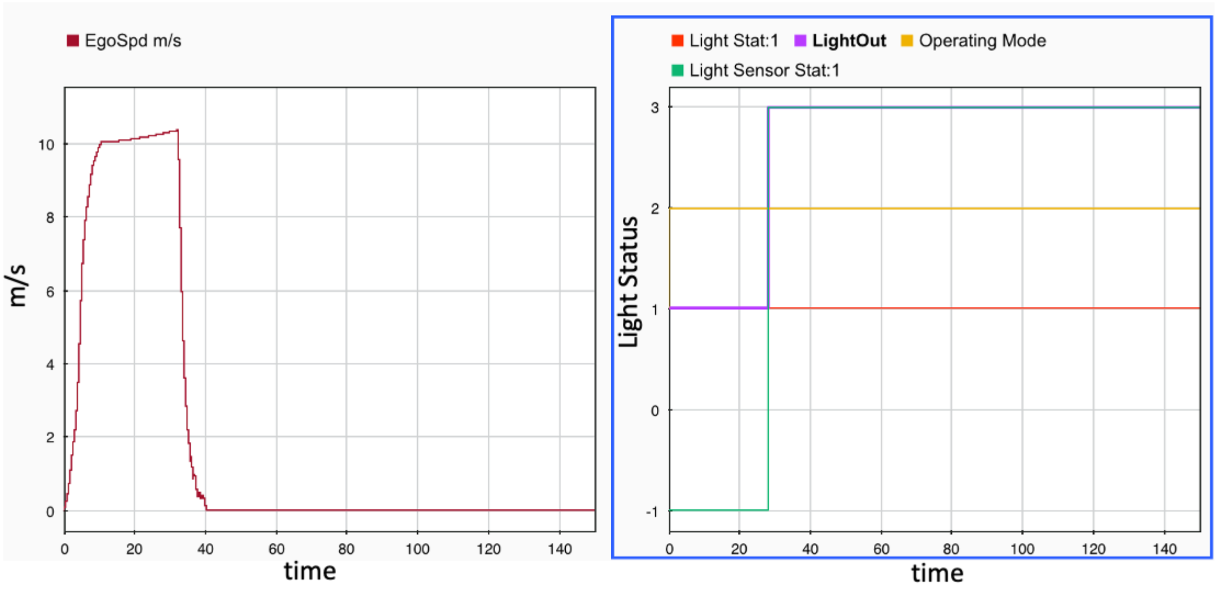


Figure 35: Traffic Light Application Performance, Test 4, Network-GREEN Sensor-RED

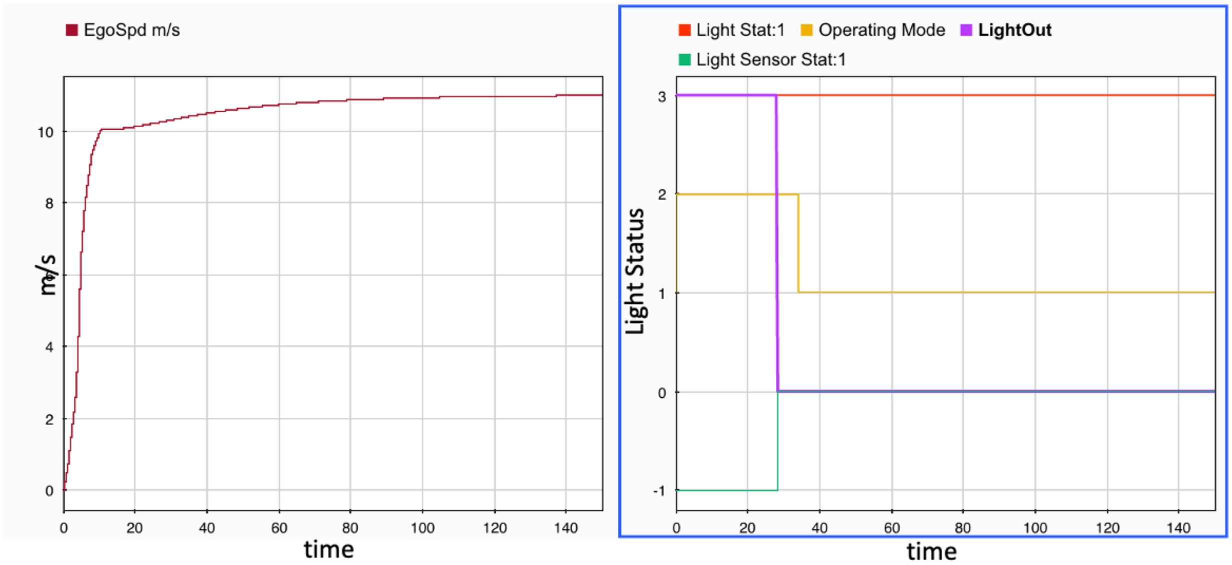


Figure 36: Traffic Light Application Performance, Test 5, Network-RED Sensor-NO LIGHT

4.2 Application 1: V2I Traffic Light Simulation Discussion

In the V2I traffic light application, shown in Figures 32 to 36, each test shows the system's performance in a simpler application. This application's possible outcomes are evident in whether the ego vehicle stops at the light or not. Without the local verification system, the ego vehicle would rely on the network data which could potentially be spoofed or altered. Once the ego vehicle gets in range of the traffic light for the camera sensor to read the light status, that data output is used over the network data if they are unequal. This is shown in the right graph of each figure where Light Out, the data that is used to make decisions, changes depending on whether Light Sensor agrees with Light Data. In tests 1 and 3 where the network data agrees with the sensor, no change is needed, and the network data continues to be used to control the ego vehicle. In other test cases, tests 2 and 4, the check prevents the ego vehicle from either running through the traffic light when its red or stopping at a green light. In either case, it could prevent a potentially dangerous situation and protect the driver. In the case where no light was present, the system initially operated normally until the vehicle was within range for the sensor to detect a light. Once it recognized that no traffic light was present, the vehicle continued normally.

4.3 Application 2: V2V CACC Vehicle Stream Simulation Results

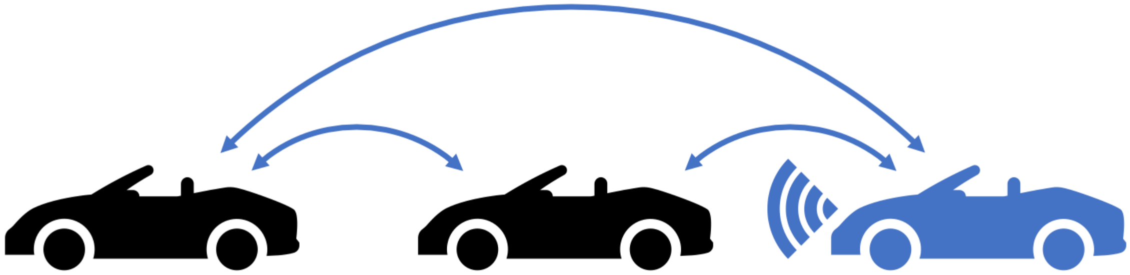


Figure 37: Application 2, V2V Vehicle Stream

Test Num	Lead Vehicle Broadcast Speed	If Ego Trusts Network Information	If Ego Does Not Trust Network Information
1	Matches Preceding	Vehicle maintains lead specified time gap from preceding vehicle	Would trust
2	Faster	Time gap would increase which would cause the ego vehicle to slow compared to the preceding member vehicle - Unstable vehicle stream - Wasted efficiency	Disengage from vehicle stream and switch to ACC functionality - Increase distance from preceding vehicle - Could join or lead another CACC stream
3	Slower	Time gap would decrease which would cause ego vehicle to get closer to the preceding vehicle - Unstable vehicle stream - Unsafe - Wasted efficiency - Lack of driver comfort	Disengage from vehicle stream and switch to ACC functionality - Increase distance from preceding vehicle - Could join or lead another CACC stream

Figure 38: Application 2, V2V Vehicle Stream Scenario Table

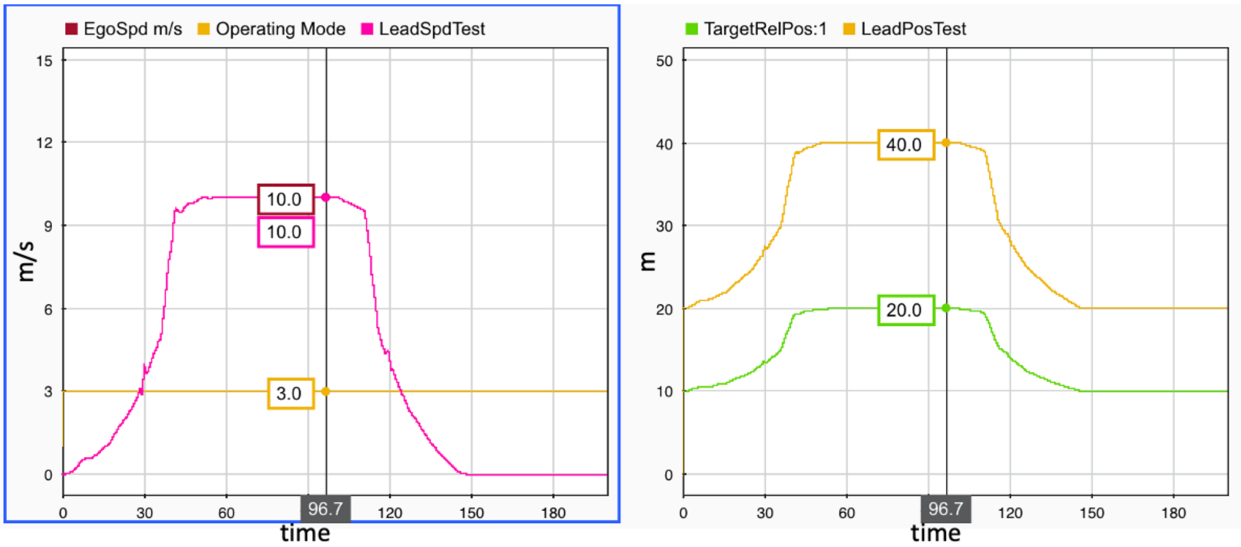


Figure 39: Vehicle Stream Application Performance, Test 1, With Validation

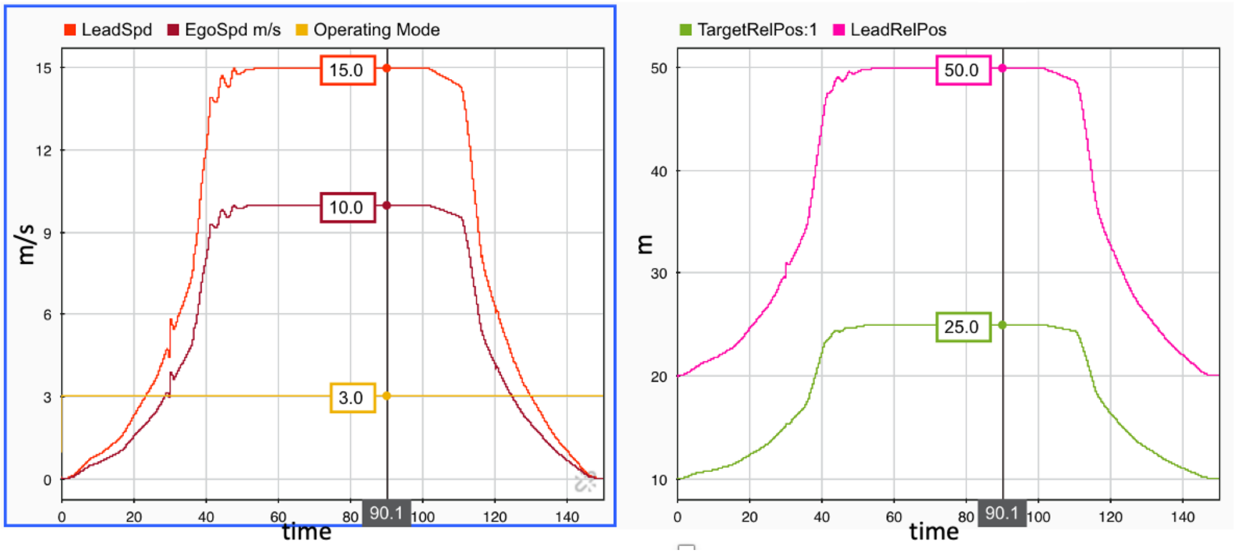


Figure 40: Vehicle Stream Application Performance, Test 2, No Validation

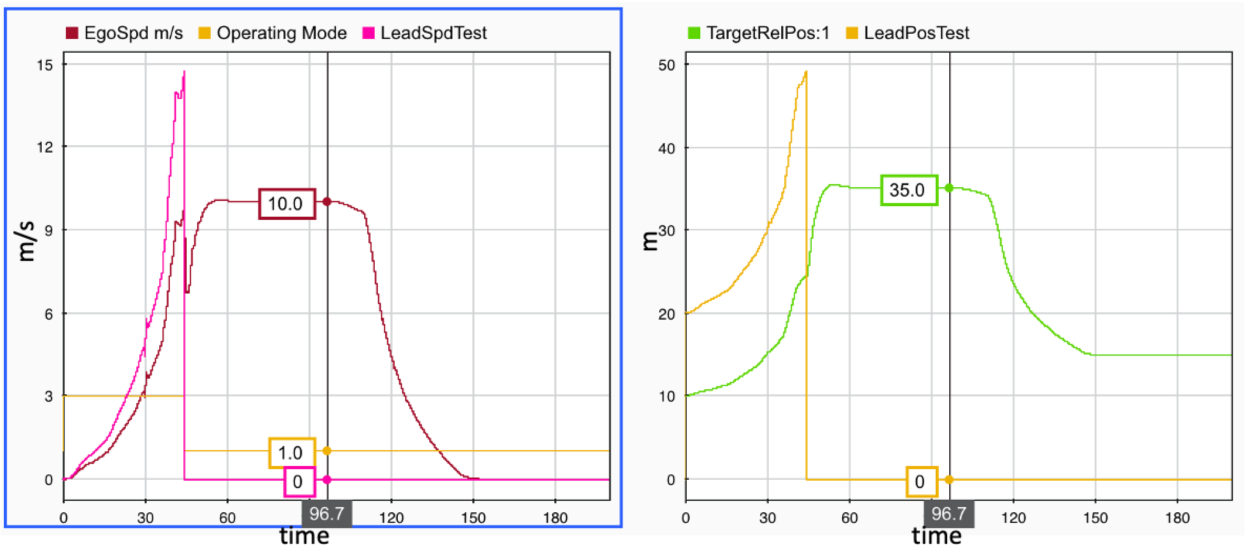


Figure 41: Vehicle Stream Application Performance, Test 2, With Validation

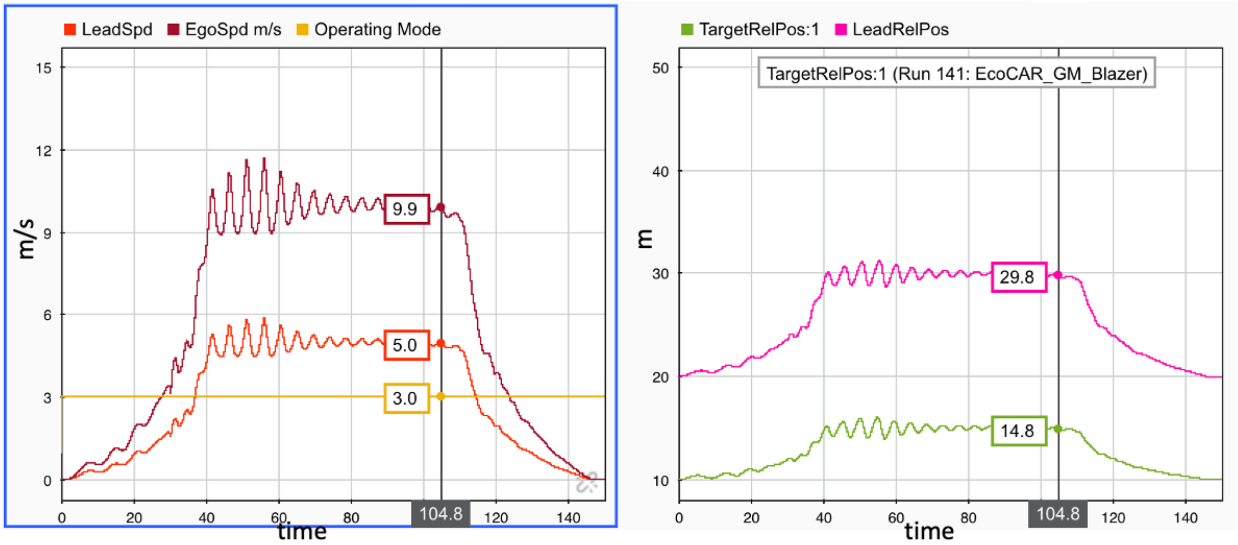


Figure 42: Vehicle Stream Application Performance, Test 3, No Validation

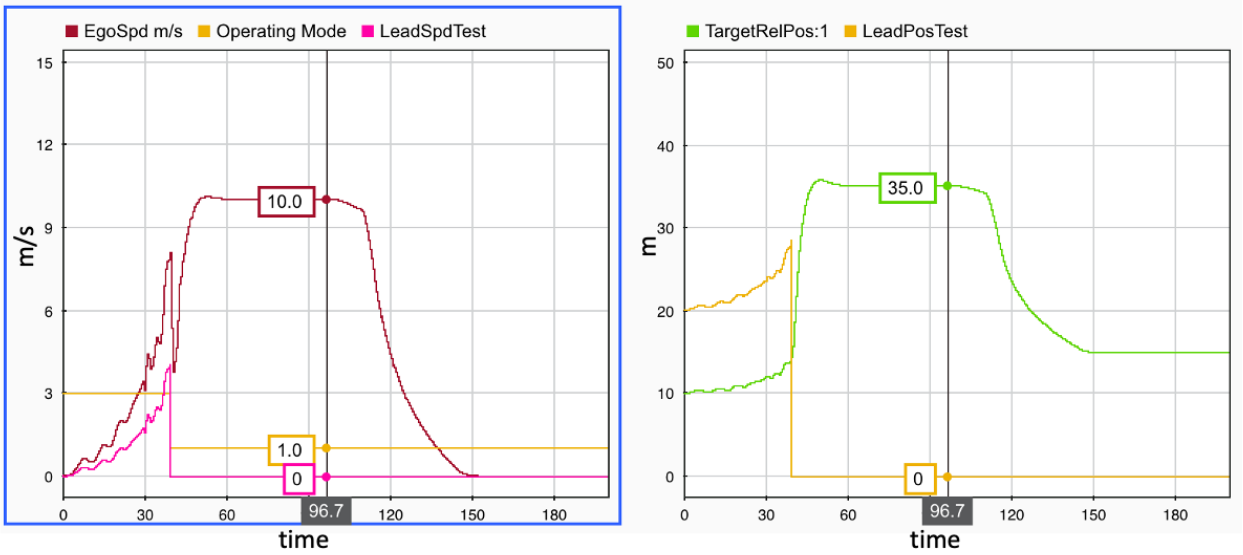


Figure 43: Vehicle Stream Application Performance, Test 3, With Validation

4.4 Application 2: V2V CACC Vehicle Stream Simulation Discussion

The vehicle stream application performance is slightly more nuanced in function compared to the traffic light, but the same principle is applied. Unlike the traffic light application where the light is either red or green, here, a threshold is used to determine whether the ego vehicle should switch function to ACC control. For testing purposes, a threshold of 5m was chosen. This can be seen in tests 2 and 3, shown in Figures 40 to 43, where the lead vehicle is transmitting a speed greater or less than that of the preceding vehicle, and control switches to ACC. Also observed is that in each case, the ego vehicle increases the safe distance from the preceding vehicle according to the differences in gap settings between the two systems. Once the threshold is met and the switch takes place, the ego vehicle brakes to increase the gap from the preceding vehicle before reaching the new safe distance and rematching the preceding vehicle's speed. The significant oscillation observed in Figure 42 is a consequence of the way lead data is represented with the PID controller. Halving the lead data for this case results in halving the feedback the Distance PID controller receives as input. As such, it takes twice as long for the Distance PID to smooth out the response.

4.5 System Review

Overall, the method of identifying error in the data streams and proactively switching function back to ACC proved to be successful. The idea leverages the synchronization between members of a vehicle stream to locally determine if the data from the lead vehicle is to be trusted. If the lead vehicle transmits data that does not agree with the observable vehicle dynamics past a threshold, then the ego vehicle should disconnect from the vehicle stream for the sake of safety and driver comfort. The decision to switch the vehicle back to ACC functionality is not negative. As with other fail cases within CACC, if something goes wrong with the network data, the system should default back to ACC functionality. This approach extends that to include a preemptive switch to protect against forms of internal threats like spoofing or data falsification.

Proactively disconnecting from a potentially malicious lead vehicle will enable the vehicle to resume CACC functionality through joining another available CACC vehicle stream behind a different leader or start its own vehicle stream. This method used in conjunction with other methods such as cloud-based trust certificates using vehicle IDs could effectively mitigate risks brought about through internal threats.

5 Conclusions

As new CAV technologies are researched and developed, transportation VANs are becoming more connected than ever. Using technologies like V2I and V2V, new data streams enable vehicle systems that produce less congestion on roadways, more energy efficiency, and improved driver safety. However, those new data streams also open previously closed vehicle systems to external attacks. Attacks towards the VAN can cause instability or loss of function in the network which eliminates the benefits of the connectivity. In applications like CACC where the distance between vehicles is minimized, instability could cause member vehicles to violate their safe distance boundaries and activate their CW/CA systems to prevent a collision. These attacks can originate from external or internal sources. Significant work has been done to address the different external threats posed to the network. Internal threats are more challenging to address because they originate from already authenticated sources.

This paper suggests a method to further address internal threats by using the many on-board vehicle sensors to validate the received network data locally. The goal is to provide an additional way for the control system to determine whether the data should be used even it originated from an authenticated source. This way, the control system can continue to function without violating safe distance boundaries and will lead to vehicles being more robust against abnormalities present in the network regardless of the reason. Combined with other work in this area such as ID flagging identified malicious vehicles, this system can mitigate internal threats on the VAN while maintaining driver safety and driver comfort.

6 Future Work

Due to the limitations put on the design by only considering general longitudinal cases, the validation methodology could be further refined when considering lateral cases as well. CACC systems include functionality that enables vehicles outside of the vehicle stream to merge into and join in the middle of the stream. This is an example of a set of cases where the threshold requirements would need to be modified to account for this case. Here, the synchronization across the member vehicles is sacrificed to allow the external vehicle to join the stream. Other edge cases could also be considered to further refine the methodology.

Another potential area of focus is considering the impact of noise from hardware on the data. The different control parameters could be reevaluated to account for noise or other accuracies within the intra-vehicle sensor data when determining if the data from the network is accurate.

7 Bibliography

1. S. Addepalli, “Progress and challenges in intelligent vehicle area networks,” Communications of the ACM, vol. 55, Issue 2, pp. 90 - 100, February 2012
2. B. Liu and A. El Kamel, "V2X-Based Decentralized Cooperative Adaptive Cruise Control in the Vicinity of Intersections," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 3, pp. 644-658, March 2016, doi: 10.1109/TITS.2015.2486140.
3. C. Desjardins and B. Chaib-draa, "Cooperative adaptive cruise control: A reinforcement learning approach", *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1248-1260, Dec. 2011.
4. A. Eskandarian, Handbook of Intelligent Vehicles, vol. 2. London, U.K.: Springer, 2012
5. B. Park, K. Malakorn, and J. Lee, “Quantifying benefits of cooperative adaptive cruise control toward sustainable transportation system,” Center Transp. Stud., Univ. Virginia, Charlottesville, VA, USA, Tech. Rep., May 2011.
6. S. Calvert, T. Van den Broek and M. Van Noort, "Modelling cooperative driving in congestion shockwaves on a freeway network", Proc. 14th IEEE ITSC, pp. 614-619, 2011.
7. Wang, Ziran & Bian, Yougang & Shladover, Steven & Wu, Guoyuan & Li, Shengbo & Barth, Matthew. (2020). A Survey on Cooperative Longitudinal Motion Control of Multiple Connected and Automated Vehicles. IEEE Intelligent Transportation Systems Magazine. 2020. 4-24. 10.1109/MITS.2019.2953562.
8. Influence of Information Flow Topology on Closed-loop Stability of Vehicle Platoon with Rigid Formation
9. J. Ploeg, B. Scheepers, E. Van Nunen, N. Van de Wouw and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control", Proc. 14th IEEE ITSC, pp. 260-265, 2011.
10. M. Amoozadeh et al., “Security vulnerabilities of connected vehicle streams and their impact on cooperative driving,” IEEE Commun. Mag., vol. 53, no. 6, pp. 126–132, Jun. 2015
11. X. Lin et al., “GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications,” IEEE Trans. Vehic. Tech., vol. 56, no. 6, 2007, pp. 3442–56.
12. P. Papadimitratos et al., “Architecture for Secure and Private Vehicular Communications,” Proc. IEEE 7th Int’l. Conf. ITS Telecommun., 2007, pp. 1–6.
13. A. Hamieh, J. Ben-Othman, and L. Mokdad, “Detection of Radio Interference Attacks in Vanet,” Proc. IEEE GLOBECOM ’09, 2009, pp. 1–5.

14. F. Alotibi, M. Abdelhakim, “Anomaly Detection for Cooperative Adaptive Cruise Control in Autonomous Vehicles Using Statistical Learning and Kinematic Model,” in IEEE Transactions on Intelligent Transportation Systems, 2020

15. Dimitrakopoulos G. (2017) Vehicular Communications Standards. In: Current Technologies in Vehicular Communication. Springer, Cham. https://doi-org.wvu.idm.oclc.org/10.1007/978-3-319-47244-7_2

16. Kiam Heong Ang, G. Chong and Yun Li, "PID control system analysis, design, and technology," in IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp. 559-576, July 2005, doi: 10.1109/TCST.2005.847331.

8 Appendix

Appendix A: Basic Safety Message Major Attributes

- Temporary ID
- Time
- Latitude
- Longitude
- Elevation
- Positional Accuracy
- Speed and Transmission
- Heading
- Acceleration
- Steering Wheel Angle
- Brake System Status
- Vehicle Size

Appendix B: Simulink Model

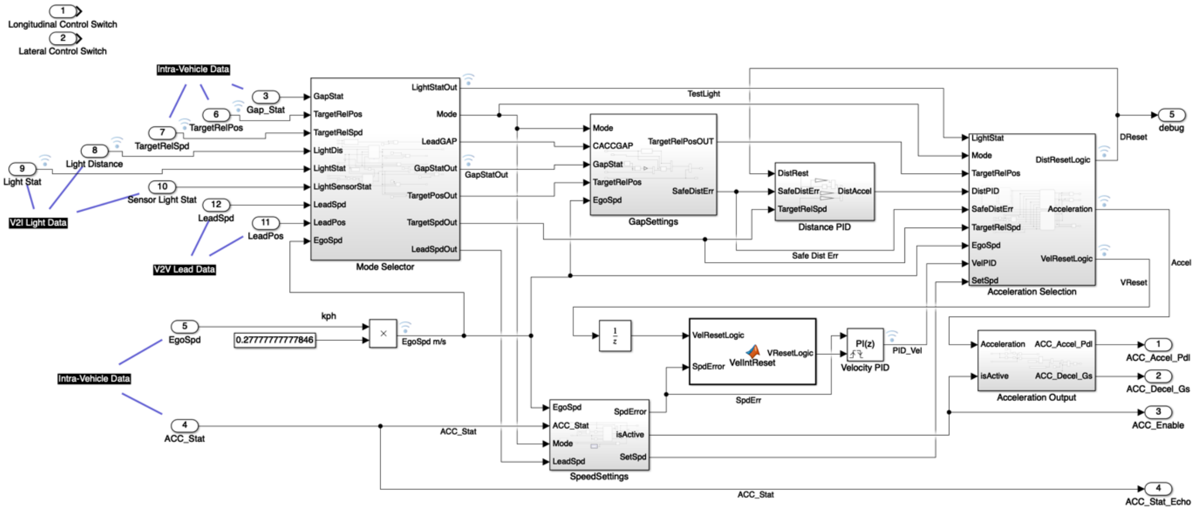


Figure B-1: Top Level System

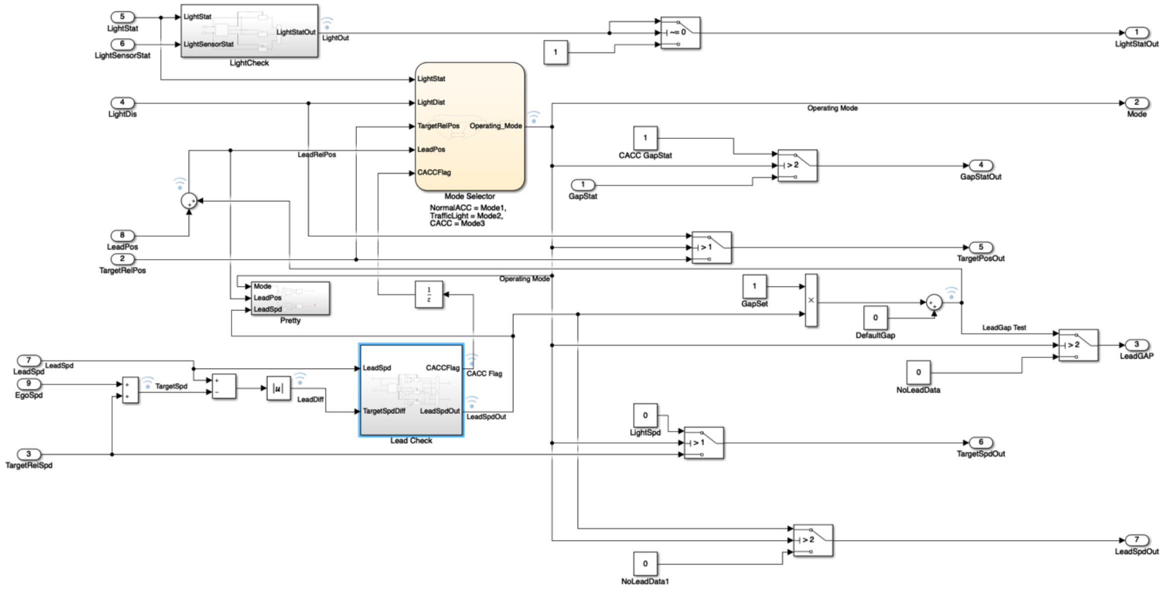


Figure B-2: Mode Selector Subsystem

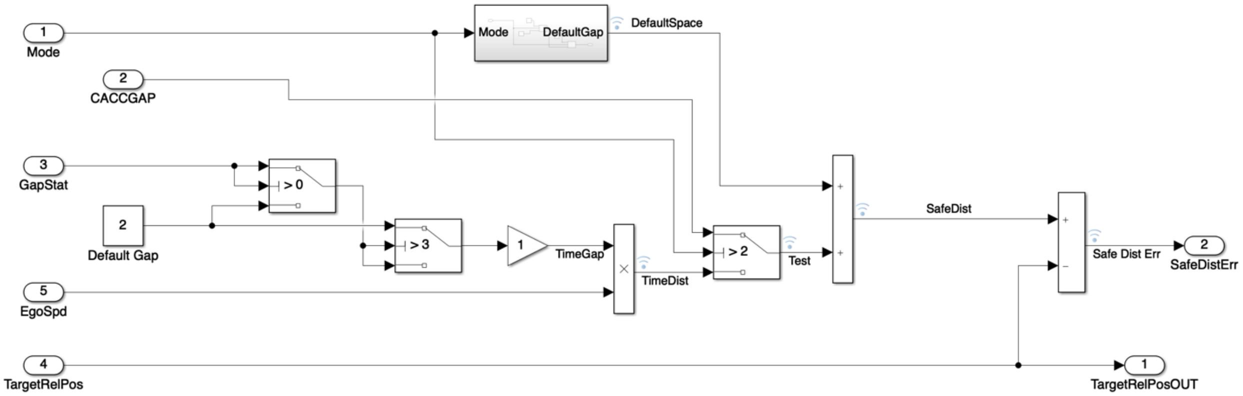


Figure B-3: Gap Settings Subsystem

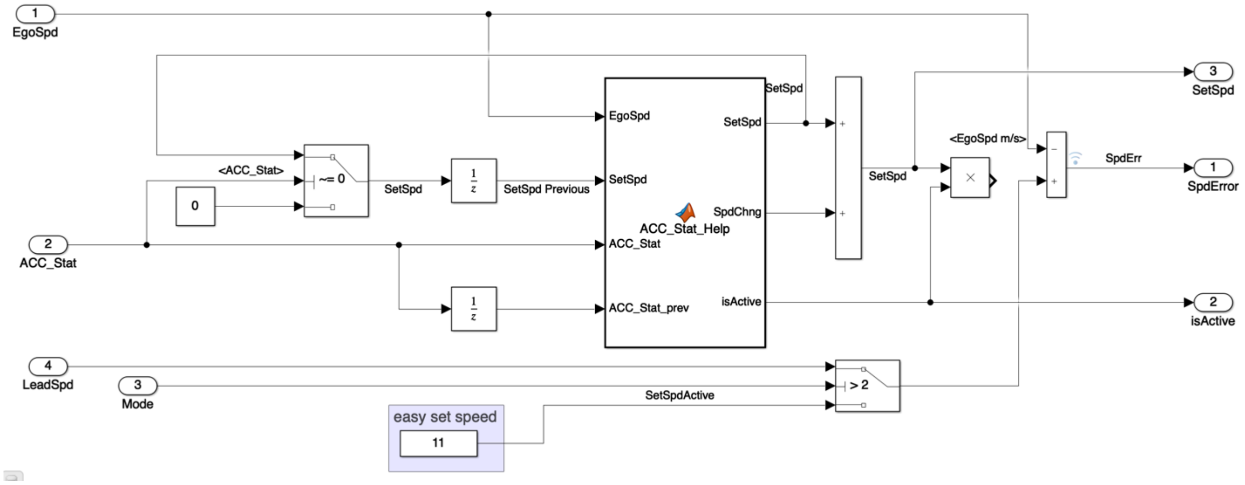


Figure B-4: Speed Settings Subsystem

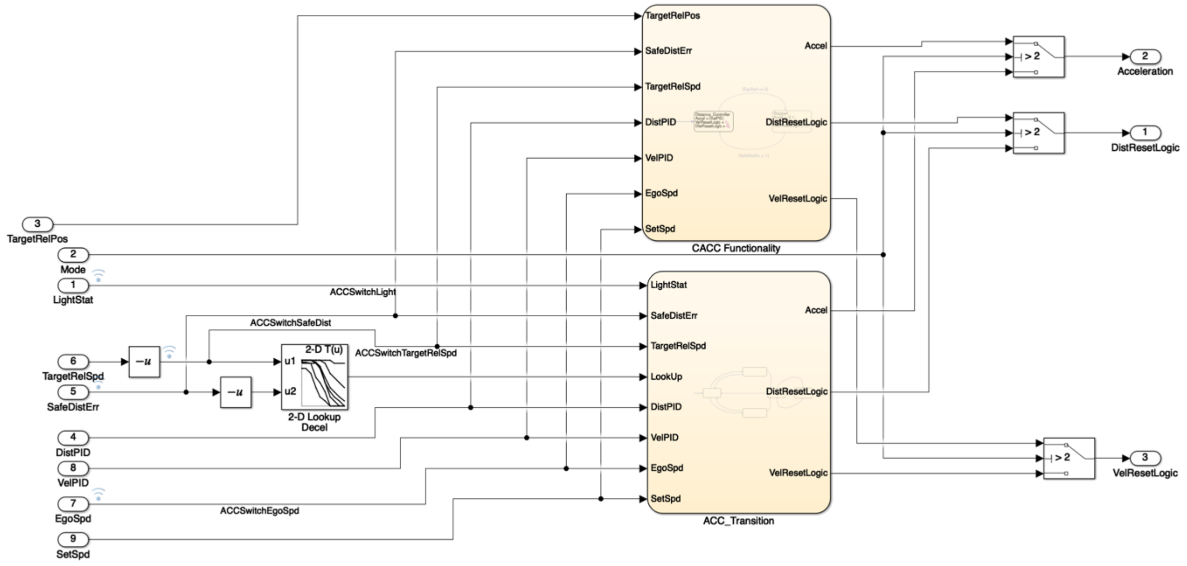


Figure B-5: Acceleration Selection Subsystem