DRIVE SAFELY

Verifying and validating the ability of autonomous driving systems to operate in a complex environment presents an enormous challenge.

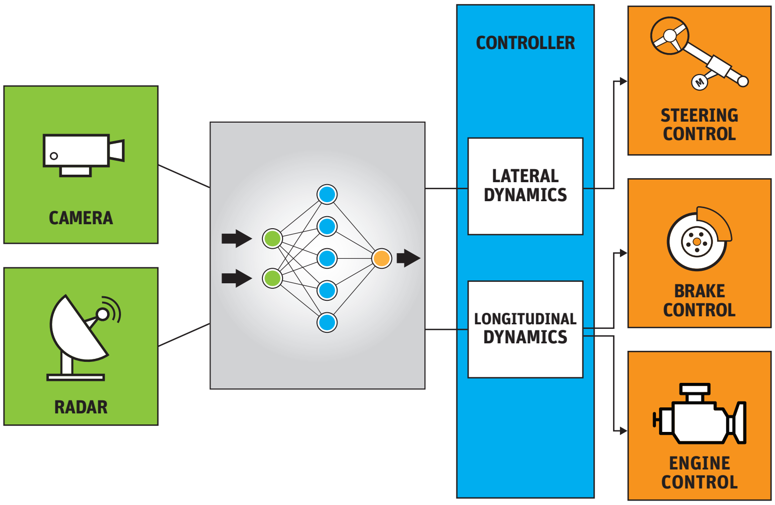
It would be impossible to program a computer to handle every possible driving scenario, so today's autonomous driving systems feature programs that learn and think like human beings to make the right decision for almost every situation. But how can these programs be verified for safety? The answer is by carefully designing an embedded software architecture that maximizes safety and a simulation platform that bombards autonomous driving software with billions of difficult driving cases to quickly identify its weaknesses.

By **Michael Wagner**, Chief Executive Officer,
Edge Case Research, Pittsburgh, USA

Bernard Dion, Chief Technical Officer – Systems,
ANSYS

Delivering an autonomous driving system, one that has the ability to understand every conceivable driving situation and make judgments to ensure the safety of vehicle occupants and pedestrians, is a complex and demanding task. For example, consider the challenge of developing rules for identifying any imaginable pedestrian, vehicle or other object that could appear on a city street. Conventional requirements-driven programming methods are not capable of mastering the huge number of potential situations that could occur on today's roads and highways.

Hands-off autonomous driving systems rely upon deep learning algorithms that can be trained to develop human-like capabilities to recognize patterns without having to be exposed to every possible situation that could arise on a trip to the grocery store. These systems lack the defined detailed requirements



Conventional software cannot do the job, so machine learning and deep learning are at the heart of the latest autonomous driving software.

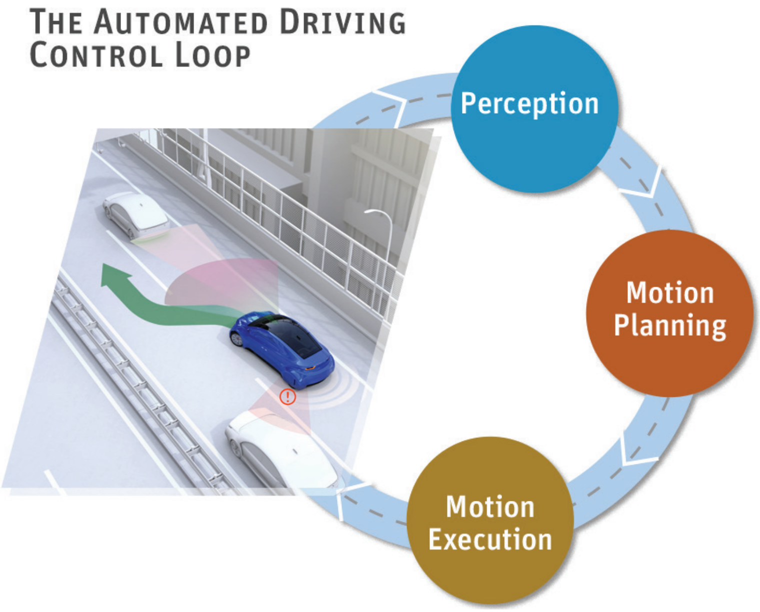
and architecture that are used to validate conventional safety-critical software. Road testing is not a practical verification method because billions of miles would be required to demonstrate safety and reliability. The ANSYS ADAS/autonomous vehicle open simulation platform integrates physics, electronics, embedded systems and software simulation to accurately simulate complete autonomous driving systems. By linking the ANSYS simulation platform and ANSYS SCADE model-based development tools with Switchboard™ automated robustness testing technology from Edge Case Research (ECR), together with ANSYS medini functional safety analysis, it is possible to achieve end-to-end safety in autonomous driving systems, including those that use deep learning.

From ADAS to Autonomous Driving

Advanced driver assistance systems (ADAS) are increasingly being used in today's automobiles to alert drivers to potential problems or even to take control of the vehicle to avoid a collision.

These safety systems are normally validated using the system and embedded software lifecycle V-model defined in ISO 26262. Using the V-model, developers carefully define the detailed requirements and architecture of the system and then methodically verify the ability of the system to meet each of the requirements. The ANSYS SCADE Suite complete end-to-end model-based system engineering (MBSE) solution is used in the development of safety-related systems for leading automobile manufacturers.

Developing a fully autonomous driving system is much more sophisticated, and must be based on a combination of machine learning/deep learning and control logic to implement the full autonomous vehicle control loop. The control loop is composed of perception (what the car observes), motion planning (what behavior the car is planning) and motion execution (how the car will complete the plan). This control loop is executed in a cyclic fashion so that the vehicle can respond to constant changes in the environment. But autonomous driving systems based on machine learning can only be released to the public after developers have demonstrated their ability to achieve extremely high levels of safety. Road testing is clearly an essential part of the vehicle development process, but it is not the answer to safety validation. The problem is that road testing primarily consists of routine occurrences that are not difficult for human or autonomous drivers. Billions of miles of road testing would be required to validate



“Hands-off ***autonomous driving systems*** rely upon deep learning ***algorithms*** that can be trained to ***recognize patterns*** without having to be exposed to every possible situation.”

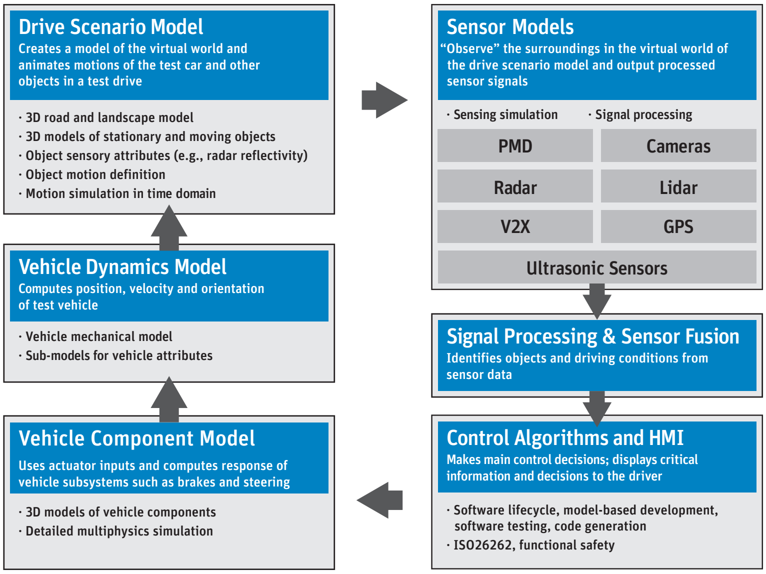
safety, and, even then, a failure or a change of code would potentially require starting over from zero.

Overcoming the Safety Verification Challenge

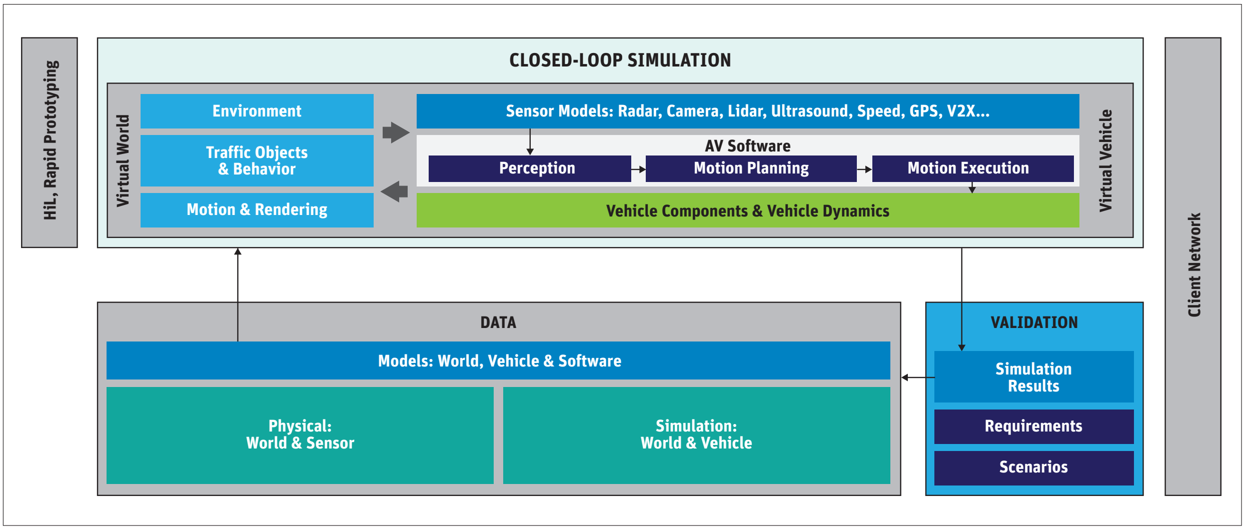
The ANSYS ADAS/autonomous vehicle open simulation platform can test many more scenarios in a fraction of the time and cost required for road testing by incorporating:

- Simulation of driving scenarios, including modeling of both the virtual world in which the autonomous car is operating and the virtual vehicle itself with accurate sensor simulation (radar, lidar, cameras, GPS, etc.) as well as vehicle dynamics.
- ISO 26262 qualified model-based development tools for control and human machine interface (HMI) software.
- Optimization of the signal integrity, thermal, structural and electromagnetic reliability of semiconductors and electronics systems.

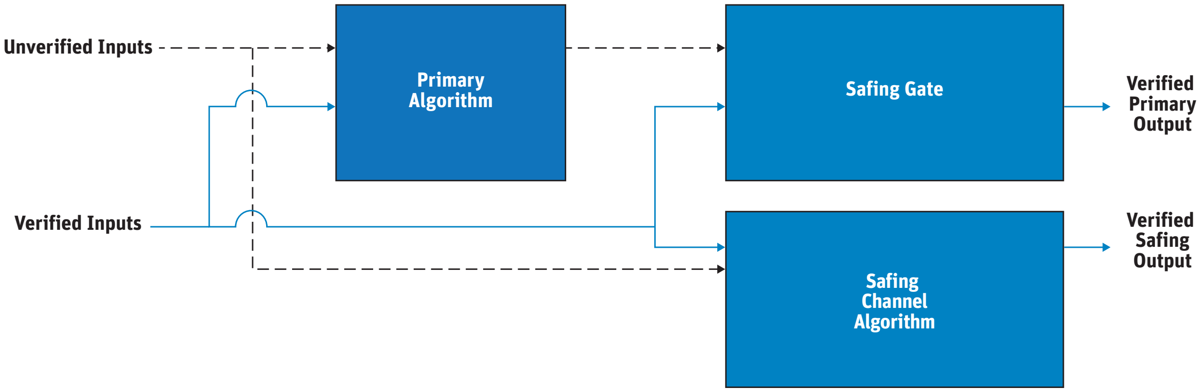
The integration of all physics, embedded systems, software simulation and code generation enables developers of autonomous systems to accurately simulate the complete automated driving control loop on a single platform. The drive scenario model animates the motion of the test car and other vehicles and objects in a test drive. Sensor models observe the surroundings in the


Drive Scenario Model
Creates a model of the virtual world and animates motions of the test car and other objects in a test drive
· 3D road and landscape model
· 3D models of stationary and moving objects
· Object sensory attributes (e.g., radar reflectivity)
· Object motion definition
· Motion simulation in time domain
Sensor Models
“Observe” the surroundings in the virtual world of the drive scenario model and output processed sensor signals
· Sensing simulation
· Signal processing
PMD
Cameras
Radar
Lidar
V2X
GPS
Ultrasonic Sensors
Vehicle Dynamics Model
Computes position, velocity and orientation of test vehicle
· Vehicle mechanical model
· Sub-models for vehicle attributes
Signal Processing & Sensor Fusion
Identifies objects and driving conditions from sensor data
Vehicle Component Model
Uses actuator inputs and computes response of vehicle subsystems such as brakes and steering
· 3D models of vehicle components
· Detailed multiphysics simulation
Control Algorithms and HMI
Makes main control decisions; displays critical information and decisions to the driver
· Software lifecycle, model-based development, software testing, code generation
· ISO26262, functional safety

Simulation of the automated driving control loop

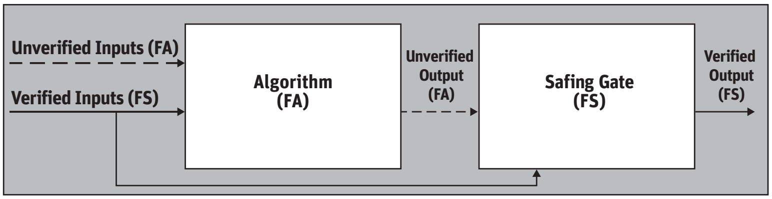


ANSYS autonomous vehicle simulation architecture



The primary channel produces a long-duration mission with no defined end state, while the safing channel produces a short-duration mission that ends in a safe state.

virtual world and output sensor signals. Signal processing models and deep learning identify objects and driving conditions from sensor data. Control algorithms make control decisions, generate actuator inputs, and display information and decisions to the passenger/operator. Vehicle component models use actuator inputs and compute the response of vehicle subsystems such as steering and braking. The vehicle dynamics model computes position, velocity and orientation of the test vehicle.



Source: Carnegie Mellon University

The algorithm (the "DOER") can fail arbitrarily (FA) meaning it can do wrong things in the worst possible way.

The safing gate (the "CHECKER") turns the algorithm into a fail silent (FS) component, only producing correct data or shutting down.

The safing algorithm for the planning phase

Safe Architecture for Safe Vehicles

While simulation is far faster and more efficient than road testing, it does not on its own answer the question of how to verify the safety of the complex autonomy algorithms used for perception, motion planning and execution functions.

To do this, first engineers must break down the overall autonomous vehicle software architecture into a meaningful set of components based on perception, planning and execution. Next, they must design an architecture that will guarantee safety for each of these components. This architecture is based on a DOER-CHECKER principle.

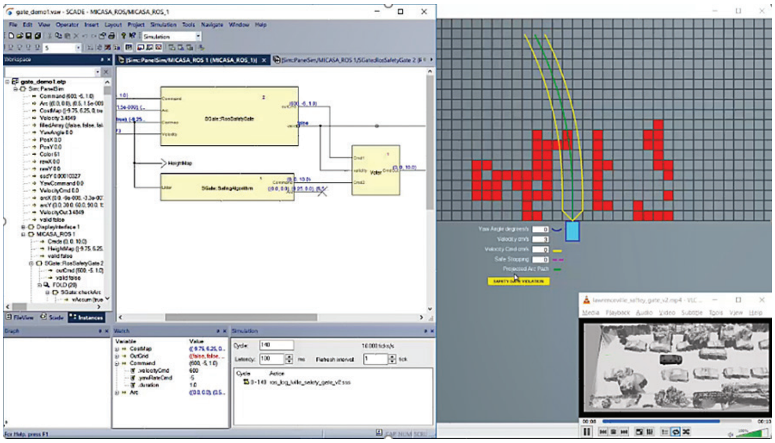
The detailed architecture is composed of a primary algorithm (DOER) that may be extremely complex, undergo frequent updates and be difficult to verify. This primary algorithm is paired with a corresponding safing gate (CHECKER) that verifies that the outputs of the primary algorithm are correct. If the safing gate detects a problem, a safing channel algorithm takes control. This can be the basis for the two-channel architecture developed by members of the ECR team while at Carnegie Mellon University (see diagram). This architecture comprises a primary channel that produces a long-duration mission and a safing channel that produces a short-duration mission, such as pulling the car to the side of the road.

Using this architecture, the plan can be checked for safety during the planning phase. The primary algorithm need not satisfy safety objectives at the highest level (ASIL D in ISO 26262); rather, this responsibility is allocated to the safing gate. What makes this possible is that the detailed safety requirements of the safing gates can be established so that their implementation meets the objectives of ISO 26262 at ASIL D. This is depicted by the example shown, in which the car is going to stop because a double-parked car has been detected.

Automated Robustness Testing Identifies and Diagnoses Failures for Perception

Assuring the safety of perception is more complex; it is not possible to create a safing gate to check that perception outputs are correct and safe. Therefore, safety of perception must be validated using different techniques. ECR Switchboard addresses this challenge (and some others) by providing automated robustness testing to find failures.

What is needed to prove perception safety is large-scale exposure to the difficult cases that can challenge autonomous driving systems (and often human drivers). ECR Switchboard uses a novel algorithm to cut through the potentially endless number of possible tests to quickly find test cases that cause software to fail and understand why the failure occurred. It sifts through the high-dimensional input space to identify exceptional queries that are informative for testing the model. It bombards the automated driving system with a mixed stream of nominal and exceptional inputs until a failure occurs. The failures are then diagnosed by generalizing a single fault-triggering input to produce a set of inputs that serve as hints in implicating field-value assignments in triggering the failure. This approach is highly effective at finding edge cases that cause system failures.



ECR Switchboard identifies perception failure: strong detection becomes extremely weak after barely perceptible environmental changes. The deep-learning algorithm can be augmented under test using any failures the Switchboard finds.

> “The ANSYS/ECR partnership can deliver a complete solution to verify and validate the safety of the most advanced autonomous driving systems.”

ECR switchboard finds a path planning failure

Perhaps the greatest challenge remaining in the large-scale deployment of autonomous driving systems is testing and debugging machine learning and deep learning algorithms that work without defined requirements and design to ensure their robustness and safety. ANSYS has leveraged its vast experience in multiple physics simulation and simulating safety-critical embedded software to deliver a complete automatic driving simulation platform that includes the world’s only ISO 26262–compliant code generator. This platform is now integrated with the ECR Switchboard robustness testing platform, which runs huge numbers of simulation scenarios while biasing toward difficult scenarios to mitigate residual safety validation risk. This partnership can deliver a complete solution to verify and validate the safety of the most advanced autonomous driving systems.

Analysis and Development of Safety-Critical Embedded Systems: The Need for an Integrated Toolkit
ansys.com/safety-critical