Ensuring Safe Self-Driving

AImotive's Development Puts Safety First

Tamás Csizmadia

AImotive has always understood the inherent risks of developing self-driving technology. However, without public road testing these innovative solutions would never mature to a level where they are safe for public use. Over the last few months, we have offered insights into our safety procedures, giving readers of the AImotive Blog a behind the scenes look at our safety driver training, a brief overview of our most important safety measures in testing, and the technical background of using simulation technology to test our solutions. The goal of this white paper is to provide a more in-depth understanding of how we ensure the safety of everyone on roads next to us at AImotive's testing locations around the world.

Keywords: safety, simulation, testing and verification, validation, self-driving, test driven development.

Introduction

A driving force behind the development of autonomous vehicles is the idea of future road safety and drastically decreasing the number of lives lost on roads around the world. However, as the technology matures self-driving vehicles themselves may be perceived as threats to safety if their public road testing is not properly controlled.

To ensure the risks are kept to a minimum AImotive heavily relies on simulation testing. In fact, simulation is the only viable solution to reach the 8 billion km (8,000,000,000 kilometers) industry experts believe necessary to achieve true safety. Self-driving vehicles must be tested in diverse weather and road conditions, in different driving cultures to be adaptable and scalable. This is a huge logistic operation, one that poses a challenge to even the largest corporations. Meanwhile one computer running one simulator node equals a test car running at 1800 km/h 24 hours a day, every day of the year. Scale these numbers to data center levels and the gains are enormous.

Simulation is not only more scalable and economical, it is also safer. Allowing our team to test and repeat dangerous scenarios any number of times, to ensure that our solutions can deal with them without problem. Extreme weather conditions, and serious accidents are rare on the road network as a whole, but in a simulator, they are always only the click of a button away.

However, relying on simulation in this way poses its own challenges. In the following this whitepaper details the challenges of testing of autonomous vehicles safely. Examining both simulated and real worlds and the solutions AImotive relies on to solve these difficulties.

A Pipeline for Increased Safety in Autonomous Development

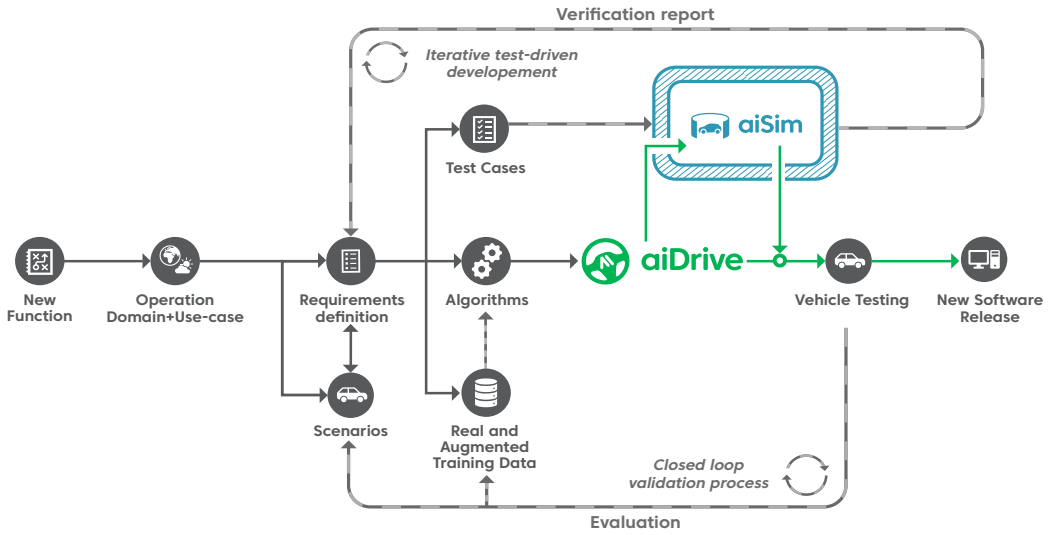


Figure 1: The AImotive technology pipeline heavily relies on simulation technology and feedback from road tests at different locations to ensure safe and quick development.

All safety considerations and measures in effect at AImotive can be summarized through our development pipeline. New features start with our Safety Team who define the functional requirements of the features to be developed based on the operational domain and use case. As a result, functional safety requirements steer the development process from the beginning. Our safety team also defines test scenarios which will be used to benchmark the solution throughout development and before road testing in aiSim, our in-house developed highly realistic simulator for autonomous systems.

Once development begins the pipeline provides a series of feedback loops to ensure our teams have the largest possible amount of information to make decisions. Running thousands of tests in our simulator day and night allows for extremely short iteration times pushing our solutions towards road test quickly while ensuring they are safe.

Only after a feature passes a minimum number of scenarios in the simulator can it be tested on public roads. As a result, the simulator acts as a safety barrier by only allowing sufficiently mature builds into the real world. All information from road tests is then collected, evaluated, and fed back into the development process, ensuring that issues that arise even only once on the road are examined and dealt with quickly.

Simulated Testing for Real-World Safety

As enumerated in a presentation by László Kishonti, AImotive CEO at the Embedded Vision Summit (May 2018) simulation testing creates its own challenges. There are several factors to consider. First, test scenarios must be defined with both functional and system-level requirements in mind and must ensure safe development. Once these scenarios are at the disposal of our development teams they must be able to access the simulator as a resource efficiently, while fully utilizing all hardware is also a challenge.

AImotive relies on both software-in-the-loop and hardware-in-the-loop testing. The former allows us to gain an understanding of how our algorithms perform, the stability of their detections and the functionality of the full self-driving stack. The latter ensures that the hardware platforms we rely on while testing are capable of processing the data our sensors collect in the extremely low latency environment of high-speed autonomous driving.

Modular testing is our go-to solution on the bug hunt and for identifying specific problems in certain areas of the self-driving software stack. Beyond testing certain small areas of aiDrive through mutual benchmarking, scenario creation, simulation development and self-driving development can support each other and maintain a stable development trajectory. Hitting the open roads of the real world with a new build is always a big decision, and that's why the immense amount of data collected from simulation testing is so extremely useful, in essence, forming a safety barrier

between development builds and the real world. The following subchapters provide more information on each of the aspects, or you can revisit our blog from after the Summit for a brief overview.

Scenarios Mimic Reality

Test scenarios can be divided into three main classes: verification scenarios, real-world scenarios and fault injection tests. Each of these serves different processes and come in to play at different parts of the development process. However, only a properly curated set of scenarios can ensure the safe development of self-driving systems.

Verification scenarios are those created when the requirements of a functionality are defined at the beginning of the development process. Their main purpose is to test the specific functionality in question and support its development. To this aim, they are carefully planned by AImotive's Functional Safety Team. Verification scenarios have the most limited goals, as they are created to measure the performance of a certain functionality.

Real-world scenarios are broader in scope, and rather test the whole self-driving stack than a single functionality. They are modelled based on information collected from international data bases of road accidents, or on disengagement reports from our own real-world tests. Simulation offers a huge advantage over real-world testing in this regard: repeatability. If we encounter a situation aiDrive cannot handle during road testing recreating it in the simulator allows our team to identify the root of the problem and implement possible fixes without having to encounter the same scenario again on public roads.

Finally, fault injection tests begin when the technology is mature enough for developers to expect stable operation even in the case of certain adverse conditions and failures. Through these scenarios, we gain an understanding of how our solutions react to

sensor failures, software errors and myriad other possible problems. This is also how fault tolerance time intervals are measured and safe state behavior examined. Through this testing, we ensure that our solutions are not a danger to others should any critical issues arise during road tests.

The three types come together to form a constantly growing library of several thousands of scenarios. This carefully engineered set ensures that our systems are constantly tested on interesting, risky or even dangerous situations. Many of these are difficult, or impossible to test in the real world properly. Either because they are too dangerous, or because of extremely limited repeatability. Very rarely will a self-driving vehicle have to react to a high-speed accident on a highway ahead of it but through these scenarios, we know how aiDrive would react and can plan changes or further development if needed. As a result, we are never blind when road tests commence, our safety drivers have a clear understanding of what to expect from the current version.

Simulation in Development

However, creating and curating this library of scenarios is not enough, if they are not properly accessible to, and utilized by development teams. At AImotive different forms of simulation, testing happen at different levels with different goals. We rely on both fixed-time step and real-time simulation to verify our technology. Automated tests are run every night on a large batch of scenarios with the most current version of the software stack. Developers can also request tests from their own workstation whenever needed to see the effects of the changes they make to the code.

Fixed time-step and real-time simulation both happen within the same simulation software, our uniquely realistic in-house developed aiSim. The two methods serve very different purposes and place different demands on hardware. Fixed time simulation can run on heterogeneous hardware setups, and through its deterministic

nature is best used to verify the logic of our algorithms. On the other hand, real-time simulation, despite its high performance requirements, gives us an idea of the runtime of algorithms and allows developers to visually monitor the movements of the vehicle in the simulated environment.

To ensure our servers are always running at peak capacity AImotive has a unique system in place that gives development teams direct access to the simulator as a resource. When working on new areas of code or improving on existing lines developers can request simulator tests on a specifically selected group of scenarios to measure the effects of their changes. This form of almost instantaneous feedback means that developers are not stuck waiting for information about their solutions for days, and can adapt, change or improve their ideas immediately.

There are also measures in place to guard against bugs reaching larger versions of aiDrive and code regression. The most important of these are pre-commit tests, which happen automatically on a predefined and largely unchanged set of scenarios. Previous builds of aiDrive have always completed these scenarios without error, and if the new code causes any glitches it is automatically rejected. This happens even before code review begins to ensure our resources are properly utilized and the team has time to concentrate on the most important tasks at any given time.

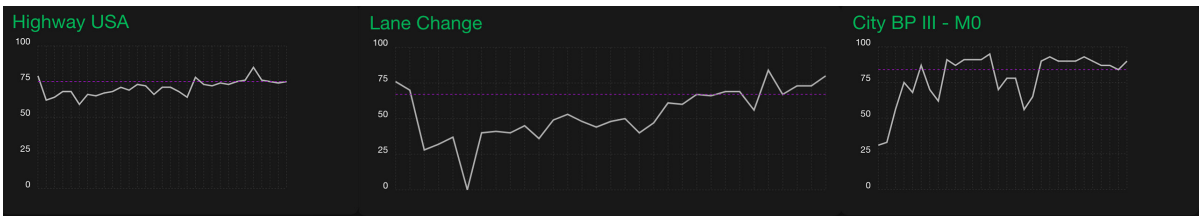


Figure 2: Vast amounts of data collected from simulation testing are aggregated on an internal development dashboard to display trends and progress.

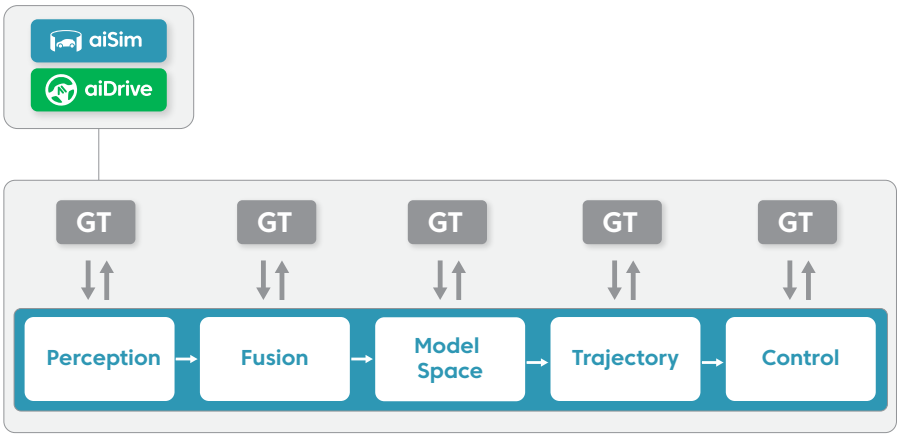
Finally, we run large-scale tests every night on the current version of aiDrive including thousands of different scenarios. This diverse selection of situations means to test the whole software stack. The vast amount of information from these nightly tests provides us with metrics regarding the overall performance of our self-driving technology as well as long-term trends in scenario pass rates. The immense amount of data collected from the simulator means that effectively reviewing it is itself a challenge. Automated systems constantly update a live dashboard with the most important graphs and metrics but reviewing the hours of real-time simulation footage is still completed manually.

Naturally, AImotive's testing methodology includes measures to mitigate some portions of this workload. Bug tracking, to ensure that buggy scenarios are not reviewed multiple times is one, and concentrated development of one functionality at a time is another. A semi-automated review system has also been considered. In such a solutions tests would be flagged for human review based on certain predetermined criteria. Naturally, the question arises, is it truly safe to have a semi-automated system review a fully automated one?

Going Modular for Mutual Benchmarking

Beyond a vast library of scenarios and a wide range of test possibilities more can still be done to extract the full potential of simulation testing. Running the full self-driving stack in the simulator provides the most important feedback, however, the root of a problem is not always apparent. This is where modular testing comes in. Modules are smaller sections of the stack responsible for well-defined tasks. In aiSim, any of these modules can be replaced with ground truth data provided by the simulator itself. If all but one module is replaced, the performance of the single running module can be examined.

Figure 3: Any section of code, or module, can be replaced with ground truth data to accelerate evaluation and bug fixing.



The ability to benchmark each module separately is immensely important for several reasons. One is bug and error fixing. By quickly cycling through a failed scenario running each module one by one developers can identify which area an error is located in. A module is vastly smaller and less complex than the full stack, which makes finding and fixing these bugs a lot quicker. Another is resource allocation. Through modular testing we are always aware of which module needs more work and can then examine whether the team responsible for the module has all the resources, including workforce, it needs to move forward effectively. This means that our performance in testing not only offers feedback on the quality of aiDrive but on the effectiveness of our methods and developers. Knowing that a module is less stable than others allows us to shuffle resources to expedite development. Furthermore, it ensures that our safety drivers will be extra aware of certain situations when on the road.

Different than, but deeply connected to modular testing is how the different elements of the AImotive toolchain can mutually benchmark our technology. Different versions of scenarios, aiDrive and aiSim can be combined in a way that reduces the number of possible errors drastically. When a scenario fails the cause is either the self-driving stack, the simulator, or the scenario, and

mutual benchmarking is designed to help us find the culprit. New versions of scenarios are first tested on stable builds of aiSim and aiDrive (naturally a version that is prepared to handle the task in the scenario). If an error occurs, then the scenario will be the first to be analyzed. The same is done for every build of aiSim and aiDrive as well. A new build of aiSim will be tested using a version of aiDrive running in scenarios it has previously always passed. If it fails in any of them, our team will look at the new version of the simulator, and so on.

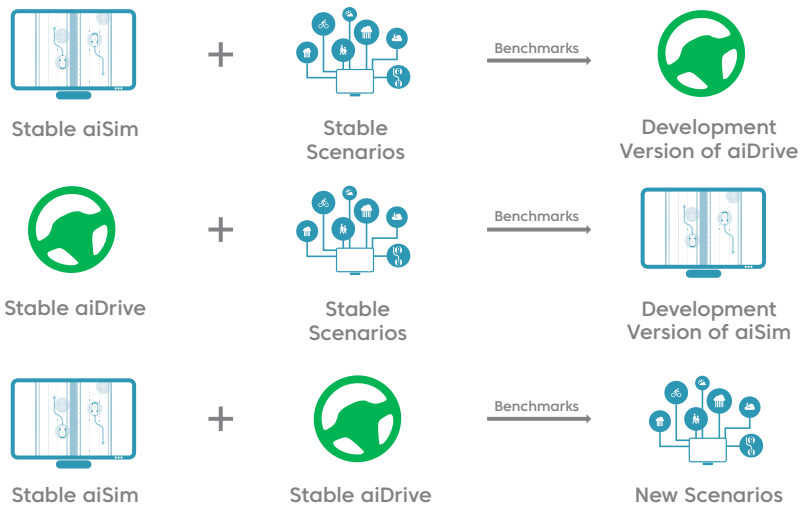


Figure 4: aiDrive, aiSim and created scenarios can be used to mutually benchmark one another. We rely on stable builds as benchmarks to ensure regression free development of each element of our testing.

Simulation as a Safety Barrier

Simulation testing is an extremely powerful tool for self-driving development but only if properly used. Beyond the technical considerations, properly curated scenarios and resource management all play a role. However, when the difficulties detailed above are overcome simulation technology becomes a veritable safety barrier between development and road testing, ensuring that only mature and stable software versions are ever tested in the real world on closed tracks and public roads.

Furthermore, as AImotive recreates each test location in aiSim before commencing road tests in any location. As a result, we always have location-specific data on the performance of our systems and our safety drivers will be aware of any difficult situations in which extra caution is required. Modelling a diverse range of different environments, in some cases, even ones where real-world tests are not being carried out provides our engineers with an understanding of aiDrive's performance in exotic, difficult or rare road conditions.

While this increased safety during development is one of the most important benefits of simulation testing, it is, in fact, the only viable solution to the enormous amount of mileage a self-driving software needs to cover to be safe for public use. Conservative estimates claim 8,000,000,000 km or 5,000,000,000 miles must be covered in variable conditions and interesting situations. This is an immense logistic and economic task for even the largest companies, and simulation remains the only true alternative.

Out on the Road

Despite its countless advantages and an emphasized need, simulation does have its limits. The main one being that no matter how realistically a simulator can render the world, or emulate physics, it isn't reality. Simulation can reduce the amount of real-world testing needed but cannot replace public road tests entirely. This means that development vehicles must share the road with human drivers from early on, even when they are unprepared for all situations. Simulation is one method for mitigating some of the risks involved, as is taking functional safety considerations into account from the beginning of development. However, testing on public roads requires careful consideration and specific preparations to ensure safety. The following chapters provide an overview of the measures taken at AImotive to ensure the safety of our testing process.

Safety Drivers and Safety Limits

AImotive's public road testing operations can be described as part of a five-step safety process. The first step is software and hardware-in-the-loop testing as described above. This is followed by tests in the simulated environment of the target location. In the third step, closed track tests are carried out to verify basic stable functionality if required. Road tests begin in Hungary, with the development team on-site. As a result, they receive feedback from our testing teams directly and can go out into the field to examine operation if needed. Once any basic problems or bugs have been rectified international public road testing will begin, this is step five. Beyond this high-level overview, there are several smaller steps in the process to serve testing safety.

All real-world tests, regardless of whether they happen in a closed parking garage, a closed test track or public roads are overseen by our team of trained safety drivers and operators. At AImotive, to ensure a higher degree of safety, there must always be at least two people in the vehicle during tests. The first is the safety driver, who is responsible only for how the car itself acts and monitors traffic.

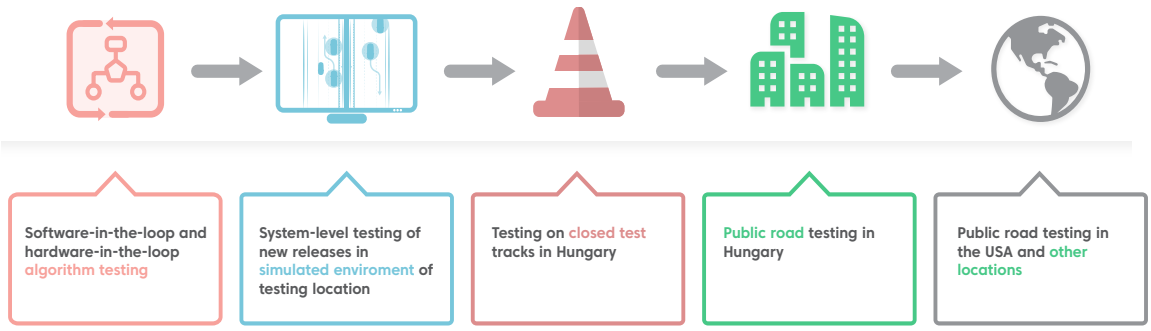


Figure 5: AImotive's real-world testing happens as part of a 5-step process. Meetings of key decision makers conclude on when a certain build can move to the next step of the process based on all collected and available data.

The second is a test operator, one of our developers or engineers who monitors the self-driving software stack through a debug screen. Should either of them decide it is unsafe to continue in autonomous mode for any reason control is immediately taken back into human hands.

Naturally, overseeing autonomous testing has a set of unique challenges. To prepare safety drivers for these AImotive enforces strict training practices and testing policies. Every new member of the safety driver team goes through a rigorous training program in their first weeks at AImotive. This includes theoretical information about the self driving system, control handover and disengagements. Other lessons are more practical and focus on high-speed vehicle control and defensive driving. A short video of this high-speed training is available on our website. All safety drivers must also participate in regular refresher training and sessions.

As of writing, all AImotive safety drivers are experienced automotive test engineers. Nevertheless, the training program has been set up in a way to ensure that should this practice become unattainable as our fleet expands in the future, testing safety is not compromised. The training program is evaluated based on a complex set of measurable performance tests and on the general readiness of the safety driver as seen by the Head of Testing and other members of our safety team.

Test operators are also faced with their own unique challenges. The debug screen of a complex autonomous system is incomprehensible to the untrained eye. These engineers and developers have a deep understanding of the code at work in our prototypes allowing them, at times, to predict when the system may fail. This allows our test crews to retake control of the vehicle preemptively, in a controlled manner.

Following training but before public road tests commence a committee of key individuals meets every morning to review which build of aiDrive may be tested safely. This decision is based on information from simulation and closed track tests. However, regardless of the decision made our safety drivers continuously analyze their surroundings to ensure self-driving mode is only activated when it is safe.

Before heading out onto the roads a series of mechanical and systems checks are carried out at the office to verify that the prototype is road ready. Tests are carried out on predefined routes selected by the Head of Testing based on information from previous tests and general traffic flow, using the software version(s) approved in the daily meetings.

Furthermore, the AImotive Safety Driver Policy sets out a set of time limits and rules to ensure safety drivers are less affected by fatigue and the, at times, monotonous nature of testing. A safety driver can oversee at maximum four hours of autonomous operation a day in sessions that can last no longer than 90 minutes. Safety drivers must also rest at least half an hour after each self-driving test session. The same policy outlines the fundamental considerations and expectations safety drivers must adhere to, beyond the broader rules of the highway code.

Figure 6: Stringently enforced time limits on testing guard our safety drivers against fatigue and the monotone nature of self-driving testing.

The most important of these is the emphasis laid on how the safety driver must dedicate his undivided attention to the vehicle and traffic conditions.

To further enhance road safety while testing control safety limits are incorporated in the drive by wire system through an independent hardware setup. These predefined limits are calibrated and tested on closed tracks to ensure that the self-driving system cannot give the vehicle erratic driving commands which may pose a threat at high speeds. Through preventing overly aggressive throttle and brake application or steering angles these limits support our safety drivers by giving them enough time to intervene in any situation.

Always going back to the drawing board

Developing autonomous technology is a highly iterative process, one that requires vast amounts of data and testing. However, any and all testing is completely useless if the information collected it not utilized. As discussed above, all simulation testing data is collected and reviewed to ensure quick development. However, the highly automated process of simulation testing facilitates data collection, the same cannot be said for real-world tests.

To ensure that exact real world-data is available to engineers, developers and testers AImotive records video feeds, debug information and telemetry from all road tests. On returning to the office the safety driver leading the test is charged with uploading the collected data to an informational hub through a semi-automated process. Engagements and intended or unintended disengagements are automatically flagged by the system, however, those participating in the test can also flag additional events, and add written comments to any of the above, to further explain the conditions.

These two data sets are continuously monitored and correlated to ensure that development is focused on the most critical areas, and

to ensure that the differences between simulated and real-world performance are not too great. This two-pronged approach to test-driven development allows AImotive to accelerate the creation of autonomous technology without compromising on safety, which must remain the ultimate goal of self-driving technology.

Conclusions

It would be naïve and irresponsible to claim that testing autonomous vehicles on public roads is without its risks. However, as there is no alternative to public road testing to fully verify the technology before deployment the best approach is to identify, monitor and mitigate risk factors. The above have detailed AImotive's approach to the safe development and testing of self-driving cars.

Heavily relying on simulation improves testing safety and brings down development times while allowing our team to manage resources effectively and focus on the most pressing areas of software development. Placing simulation as a safety barrier between development and real-world tests means that only reliable and mature versions of aiDrive are ever tested on public roads.

Our testing procedures rely on the knowledge and experience of our team supported by specific training for monitoring autonomous vehicles even at high speeds. Further guidelines, policies and rules ensure that our vehicles are road ready and our safety drivers are never fatigued or distracted. With two persons involved in testing one is only ever monitoring road conditions, while the other reviews information from the self-driving system. Additional hardware safety barriers ensure that the self-driving software cannot transmit erratic behavior to the drive by wire system of the vehicle.

It is AImotive's belief that our development pipeline and the safety considerations outlined herein drastically improve the safety of autonomous vehicle testing everywhere and contribute to the creation of a truly safe self-driving system, operable in any location, any climate, at any time.