Date:	13.06.2019
Author:	Sophie Schäferle
Release No.:	CarMaker 8.0

How do I choose the correct sensor model for my specific application?

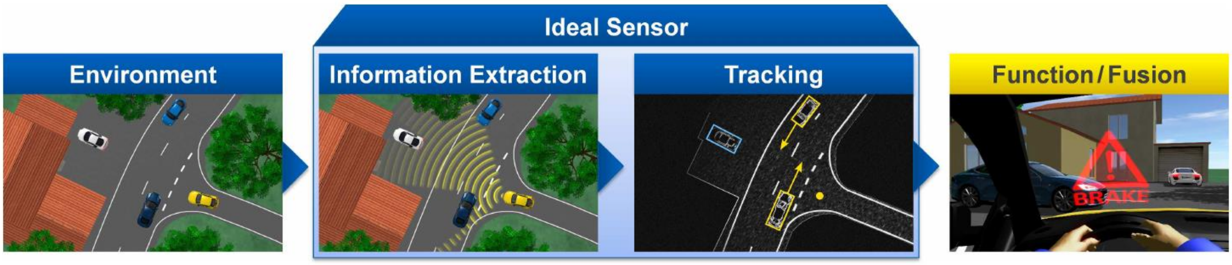
In our simulation software we offer a variety of different sensor models. These differ in complexity and output and can generally be divided into three levels: ideal sensors, HiFi (High Fidelity) sensors and RSI (Raw Signal Interface) sensors. In order to avoid unnecessary performance-loss whilst still achieving your simulation goals, it's important to select the correct model. To do so, you must understand what functions each class of sensor has. This is what I'd like to briefly explain in this article.

Ideal Sensor Models

The ideal sensor models make up the majority of all sensors in CarMaker. We have an *Inertial, Slip Angle, Object, Free Space, Traffic Sign, Line, Road* and *Collision Sensor*. We call them *ideal* because they deliver *ground truth information*. This means that the sensor models don't have any specific post-processing functions and the user doesn't need to implement any of his own either. All the ideal sensor model actually does is extract existing information from the simulation environment and deliver it to the user. You can imagine each ideal sensor as a sort of filter giving us only relevant information out of the large, overwhelming pool of all available quantities in CarMaker.

Since physical effects don't have an influence in regard to these models, we don't only receive information that we would from an actual sensor. Extracting ground truth information allows us to also track objects or know things beyond what we could really see. This is relevant in understanding which field of application the ideal sensors are designed for – *function development*.

Here's an example: An engineer works in the department for vehicle control systems and is designing a control algorithm for an ACC controller. A different department delivers a tool that tells him when and where an object is being detected. He doesn't care *how* this information is being determined and *where* it comes from, he just uses it to develop his own ACC relevant functions like braking, speeding up, etc. In the simulation this means that the object sensor is absolutely sufficient for this specific use case.

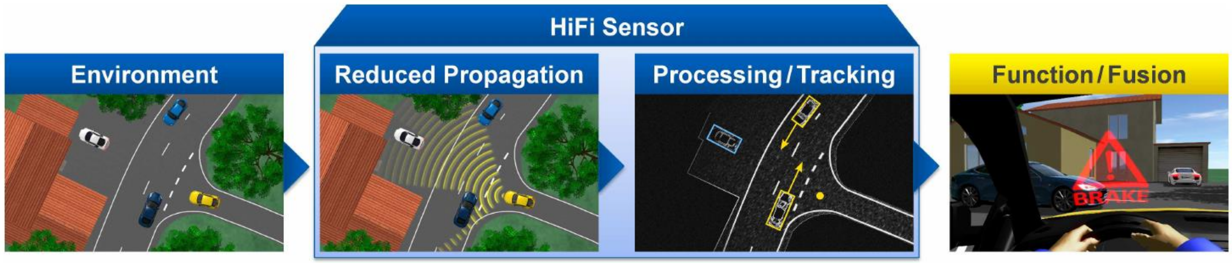


To visualize what I've just explained, have a look at the image above. The model extracts information from the environment, derives additional ground truth information and leaves the user with ideal data with which he can test his own functions.

HiFi (High Fidelity) Sensor Models

CarMaker offers three HiFi sensor models: *Free Space Sensor Plus (extension of the ideal Free Space Senor), Radar* and *Global Navigation sensor*. What mainly distinguishes the HiFi sensors from the ideal sensors is that not *all* ground truth information is conveyed to the user but some is filtered out and some is deliberately added in order to consider few basic but prominent physical effects. These include:

- Object merging due to the effects of high/low resolution
- Object occlusion when one object is being covered by another and can therefore not be detected
- False positives and false negatives
- Measurement noise

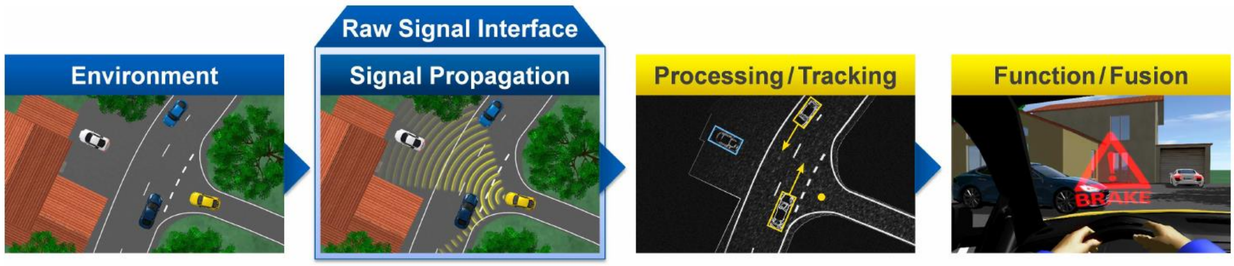


Due to the consideration of a variety of physical effects the HiFi sensor models aren't just extracting all available information in the environment, they simulate a *reduced propagation*. Different factors like the determination of signal strength are included in the subsequent processing whereas object tracking is also still available.

The output of these sensor models is best used in the fields of function development and testing. Coming back to our example from before: Our engineer has developed an algorithm for his vehicle controller. He's tested it several times using the Object Sensor but now he wants to know whether it also works when he isn't always fed with perfectly ideal data. Using the Radar Sensor he can put his algorithm to the test and see how it reacts to possibly false information and noise.

RSI (Raw Signal Interface) Sensor Models

In respect of current developments in the fields of ADAS and AD, the RSI Sensor models are becoming increasingly popular among all CarMaker users. The available models are *Camera, Lidar, Radar* and *Ultrasonic RSI*. These offer no tracking components whatsoever and can deliver only the *raw* information that an actual sensor would, too. This way the user himself is responsible for sorting, handling and interpreting the information that's given to him. This is a lot of work and it's important to decide whether it's relevant or not. Depending on the use-case, some models offer a few post processing functionalities that can support the user.



The RSI sensor models are calculated on the *GPU*. This way they can not only detect Traffic objects but all 3D objects and surfaces in the environment. These are scanned by a ray-tracing algorithm that simulates the propagation of waves or rays in consideration of a large number of environmental effects. This is where the high demand on hardware of these models originates from and why it may be critical for the performance of the system. One exception is the Camera RSI sensor model that works with the raw image data it receives from IPGMovie.

Visualization of the Lidar RSI sensor detecting all objects and surfaces in the environment

Again, I'd like to give a small example of a field of application where an RSI sensor model is relevant: Having a look back at the engineer using the object sensor to develop functions for an ACC controller, we remember that the ideal data he was receiving was sufficient for him because he didn't need to bother about where it came from. He just needed to work with it. The colleague who's actually responsible for generating this information has to work with the raw data, interpret it, write sensor algorithms and prepare it for the next step. The RSI sensor models help us in simulating exactly this output from the sensor hardware.

Overall you could say that the RSI sensors are the most detailed, but also those with the highest hardware demands and risks of performance reduction. "More detailed" doesn't always equal "better" because not every use case is the same. I hope I was able to shed a little light on this topic and that I might be able to help some of your struggling with the decision of which model would be best suited.