

白皮书

增强自动驾驶汽车开发的关键能力

对虚拟世界进行仿真，培养多领域技能，并交付用于复杂自主系统的软件

自动驾驶中的“自动”分很多级别，从高级驾驶辅助系统 (ADAS) 一直到完全自动驾驶 (AD)。随着自动化级别的提高，用例受限逐渐减少而测试需求随之增加，这使得在虚拟世界中进行场景仿真的需求变得愈发重要。开发这些自动驾驶应用需要掌握多领域技能，从规划与控制到如检测、定位、跟踪和融合等感知领域的技能，特别是在一个可以为日益复杂的软件提供设计、验证和部署支持的环境中。

本白皮书为您介绍了汽车工程师如何：

- 通过构建虚拟世界和利用仿真来管理验证复杂度
- 培养多领域技能以适应汽车工程的根本变革
- 开发符合 ISO 26262 标准的软件应用程序

ADAS/AD 开发

ADAS/AD 工程师经常面临几个问题：该如何分析和合成场景？该如何设计和部署算法？在系统层面，又该如何集成和测试整个 AD 系统？

很多时候关于 ADAS/AD 开发的讨论很快就变成了有关感知的讨论，而有关感知的讨论进而会变为 AI 和 AI 建模的讨论。然而，ADAS/AD 开发不仅限于感知。它跨越多种虚拟世界，需要使用多种工具开发算法和将这些算法部署为软件应用程序的多领域技能，如图 1 所示。

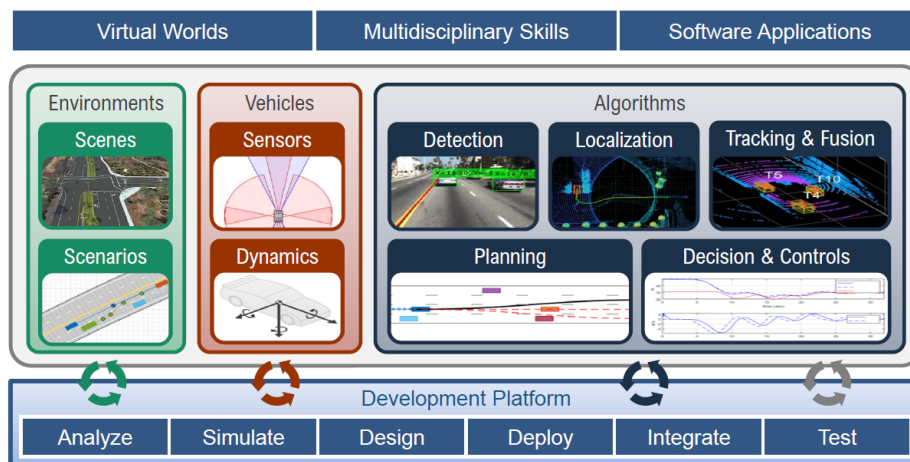


图 1. 工程师在开发 AD 应用时应考虑的三管齐下的方法。

此外，这些工程师通常会预留出大量时间开发和微调环境、车辆和算法的模型。建模的确是工作流中的一个重要步骤，但模型并不是工作流的尽头。在 ADAS/AD 应用的实际开发中，成功的关键因素是尽早发现问题，并知道将时间和资源投入工作流的哪些方面能获得最佳结果。

在深入研究典型工作流之前，应考虑两个重要方面：

- ADAS 和 AD 是多领域技术，有许多开发工具和供应商。这又要求支持集成仿真平台的优秀连接器。集成可将（在许多平台上开发的）所有算法放在一起执行系统仿真以深入了解情况。
- 除了集成，开发者还必须具备能够轻松实现可视化的工具或平台，以评估整个工作流中算法的性能。

典型的 ADAS/AD workflow

首先，您需要创建一个景象 (scene)。然后再创建一个场景，其中包括该景象 (scene)、交通参与者（车辆、行人）、天气和光源。接下来，需要对自主车辆进行建模，以包含作为 AV 传感器套件一部分的传感器以及车辆动力学（用于横向控制、纵向控制或同时用于两者）。在完成上述准备后，您就可以开始对该场景进行仿真，而这会进而支持对感知、规划和控制算法进行迭代改进。在对这些算法有足够信心后，您就可以开始创建软件了。软件代码或者是由工具自动生成的，或者是手写的。然后，集成代码以执行系统级仿真，从而确保代码在系统级能够正常工作。最后，作为测试的一部分，以交互方式或自动方式运行仿真（在您的桌面端、集群或云上）。

仿真虚拟世界

您可能听说过通过仿真运行百万场景的概念。仿真需要反映现实世界。在测试场景之前，您的虚拟世界中需要有可以仿真的景象 (scene)。景象 (scene) 需要反映现实世界，这可能是相当复杂的，例如交叉路口就是一个具有挑战性的道路景象 (scene)。以一个交通环岛为例，它既可以是一个相对简单的 3 个入口/出口的布局，也可以是一个复杂的 12 个入口/出口的布局，例如[法国巴黎的凯旋门](#)。

即使是现实中非常复杂的道路景象 (scene)，[RoadRunner](#) 也能快速、完整的重现（请参阅图 2）。对于 AD 来说，道路是景象 (scene) 的一个关键部分。

Quickly create accurate road models



Photo

Re-created Scene

图 2. 用 RoadRunner 重新创建现实世界的景象 (scene)。

重新创建的景象 (scene) 采用的格式需要能够导出并用于市面上流行的仿真器，如 CARLA、CarMaker® 和 NVIDIA® DRIVE Sim®。如果您需要创建很长的道路景象 (scene)，这种手动方法可能会很麻烦。此时，自动方法就会帮您省时省力。这时，您可以从 [HERE](#) 高清实时地图导入更长的三维路段。

您可以基于这些景象 (scene) 创建驾驶场景。场景的数据源之一是记录的数据。[福特汽车公司](#) 通过所记录的数据进行事件识别和场景生成，开发了主动驻车辅助功能。同样，[通用汽车公司](#) 根据所记录的车辆数据生成了场景并将其用于验证车道对中系统。

您可以根据记录的数据识别新场景。这种方法允许您从 CAN 日志中，或直接从摄像头或激光雷达中提取信息。您可以将数据可视化，然后对其进行标注。标注可以使用公共或自定义算法来实现自动化。之后，从记录并标注的数据中识别感兴趣的场景，用其重新创建仿真测试用例。这一过程通常是一个开环工作流。

您还可以从场景变体中识别新场景。在这种方法中，您首先需要创建一个场景。然后，创建场景变体并使用仿真来帮助识别感兴趣的新场景，再将其添加到您的回归测试中。此流程实现的是一个闭环工作流。

通过以上两种方法，您可以在设计和仿真工作流中识别和添加新测试用例。

景象 (scene) 和场景可以通过交互方式或编程方式创建。此外，您还可以：

- 将景象 (scene) 导入/导出为包括 OpenDRIVE 在内的各种格式
- 将 OpenStreetMap® 数据导入场景中
- 将场景导出到 OpenDRIVE
- 将场景导出到 OpenSCENARIO

虚拟世界的保真度可以根据仿真特定用例的需要来选择。例如，来自雷达的跟踪检测可用于开发规划和控制算法，而相机检测可用于开发感知算法。MathWorks 为虚拟世界提供两种环境：

- **立方体：**您可以使用立方体世界表示方式对驾驶场景进行仿真，使用传感器模型，并生成合成数据以在测试仿真环境中测试自动驾驶算法，包括控制、传感器融合和路径规划。例如，您可以使用这种方法来确定传感器的最佳位置和数量。
- **Unreal Engine®：**您可以在使用 Epic Games 的 Unreal Engine 渲染的三维仿真环境中开发、测试和可视化驾驶算法的性能。除了在立方体世界中提到的算法，您还可以开发和测试由来自不同相机模型的相机数据驱动的感知算法。

图 3 显示属于典型 AV 传感器套件的传感器。

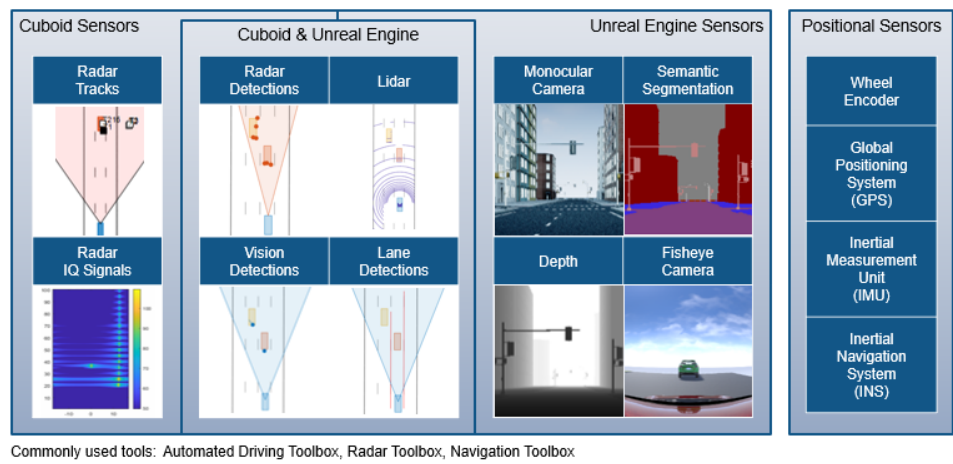


图3. 仿真用于 AD 应用的传感器。

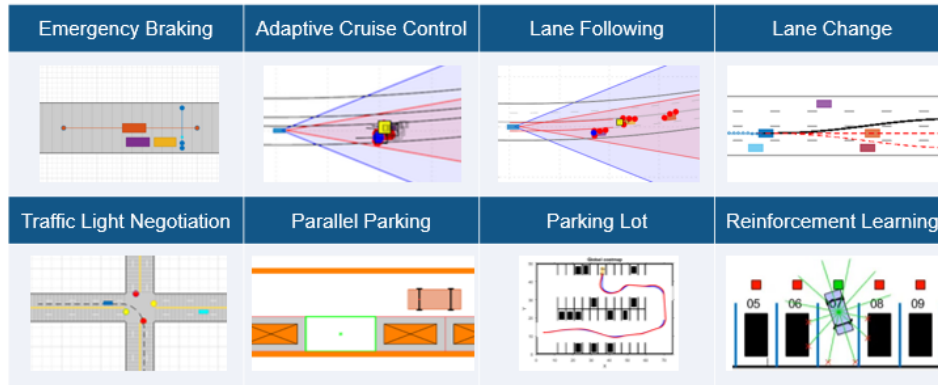
雷达、激光雷达和摄像头传感器用于在不同仿真环境中使用传感器和检测方法检测目标。位置传感器在这两种仿真环境中都可以使用。要对车辆动力学进行仿真，您需要多轴车辆、卡车和挂车、动力总成系统、转向、悬架系统、车轮和轮胎的模型。

重申一下，开发虚拟世界涉及创建景象 (scene)、创建场景、传感器建模和车辆动力学建模。此过程具备可扩展性，并让用户得以灵活应用自己的专业知识，而不必成为其他领域的专家。

培养多领域技能

AV 开发的多领域属性要求 ADAS/AD 算法存在于更大的系统中，并能与车辆系统的其他组成部分互操作。在 ADAS/AD 应用中，您不仅要有检测目标（行人、车辆、停车标志）的感知系统，该系统还必须与其他系统相集成，以进行定位、路径规划、控制等。

开发这种复杂的系统需要多领域技能来开发 ADAS/AD 功能的算法，如自适应巡航控制、自动紧急制动以及更高级别的功能，如高速公路变道和自动泊车/代客泊车。图 4 显示了一些 AD 功能示例。



Commonly used tools: Automated Driving Toolbox, Model Predictive Control Toolbox, Stateflow, Navigation Toolbox, Reinforcement Learning, Robotics System Toolbox

图 4. 典型 AD 规划和控制算法。

这些算法涵盖规划、控制和感知领域：

- **规划和控制**包括运动规划、决策逻辑以及纵向和横向控制。
- **感知**包括检测、目标跟踪和传感器融合、以及定位。

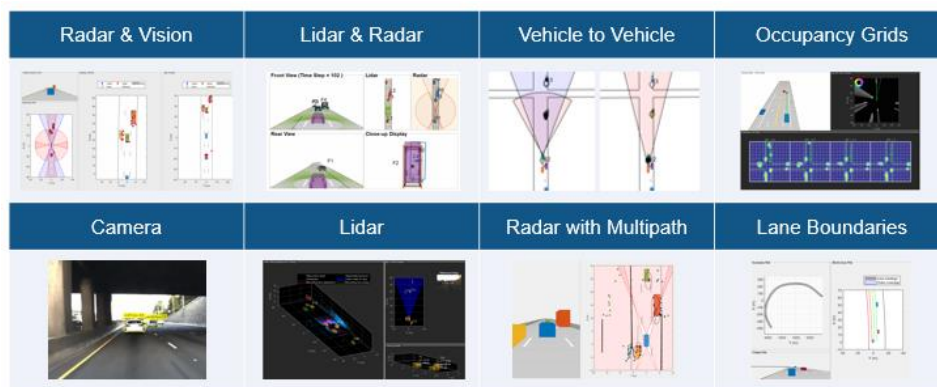
Automated Driving Toolbox™ 包括[示例](#)，帮助您以此作为起点开始设计自己的 ADAS/AD 功能。刚进入汽车行业的工程师需要了解典型的汽车结构及其组成子系统，包括控制系统。例如，他们可以使用 [Control System Toolbox™](#) 快速开始控制系统设计，使用 [Vehicle Dynamics Blockset™](#) 快速开始车辆动力学建模。鉴于 ADAS/AD 系统的复杂性和软件开发周期的快节奏，从其他领域进入这一领域的工程师可以使用 [Automated Driving Toolbox](#) 和 [Sensor Fusion and Tracking Toolbox™](#) 等工具快速入门。此外，他们可以开始使用 [Model Predictive Control Toolbox™](#) 开发高级控制系统算法，如模型预测控制 (MPC)。

以[高速公路变道](#)为例。开发此功能的工作流包括：在立方体世界中合成场景，设计规划器，使用 MPC 设计控制，车辆动力学建模，以及最后可视化结果以通过仿真深入了解情况。

另一个示例是[自动泊车服务](#)。开发此功能的工作流将引导您从路径规划开始，直到轨迹生成，再到车辆控制。这方面的其他示例包括 [使用非线性 MPC 的轨迹生成和跟踪](#)，以及用于[使用强化学习的自动搜索和泊车任务](#)的控制器。

MATLAB® 和 **Simulink®** 等工具为工程师提供在迭代环境中所需的支持。尽管算法和预置模型可作为良好的开端，但这还不是全部。工程师学习如何使用这些算法，并借助示例找到解决他们面临的特定问题的最佳方法。

规划和控制算法由跟踪和融合算法驱动。图 5 显示了典型的检测。



Commonly used tools: Automated Driving Toolbox, Sensor Fusion and Tracking Toolbox, Radar Toolbox

图 5.AD 跟踪和融合算法的典型检测。

您可以使用图 5 中所所示的示例和工具来设计跟踪和融合算法，以将雷达、激光雷达和摄像头等传感器的检测结果转换为目标、车道和网格等跟踪信息。

您可以根据相机和激光雷达数据设计检测和定位算法，还可以使用地图和惯性融合来增强定位。图 6 显示了 AD 的检测和定位算法的设计。



Commonly used tools: Automated Driving Toolbox, Computer Vision, Lidar Toolbox, Radar Toolbox, Deep Learning Toolbox, Navigation Toolbox

图 6.AD 的典型检测和定位算法。

请注意，激光雷达既可用于开发更高级别的自动化功能，也可作为额外的传感器用来验证来自更低级别的自动化功能的检测。来自传感器检测的输出用作了定位的输入。这些输出还用于与地图数据相关联以改进定位算法。您可以将相机和激光雷达的检测以及 **HERE** 高清实时地图数据和 **GPS** 结合使用，以提高车辆定位的准确度。在某些地图信息不可用的情况下，您可以使用同步定位与地图构建 (**SLAM**)。此技术使用来自激光雷达和摄像头传感器的数据。

交付 ADAS/AD 软件

在部署到产品之前，通过仿真和准确性测试来验证模型是否正常工作，以及多个系统构成的系统内所有成员能否良好地协同工作是至关重要的。

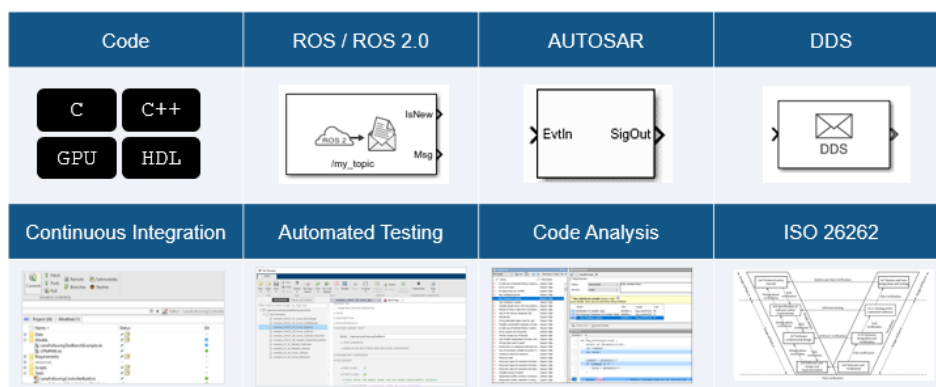
要在部署前实现上述程度的准确性和稳健性，工程师必须确保系统在任何情况下都能以预期方式作出响应。在此阶段应提出的问题包括：

- 每个算法/功能的整体性能如何？
- 系统的整体性能如何？
- 它在每种场景下的表现是否均符合预期？
- 模型是否已涵盖各种边缘情形？

一旦算法在功能上是正确的，就需要将其实现为嵌入式软件。在生成代码之前，请将设定添加到模型中，以确保仿真模型和实现的代码在整个开发过程中保持功能上的一致。

以最终实现这些算法所用的语言来准备它们。指定的硬件环境可以是桌面端、云、边缘或深度嵌入式设备。实现的灵活性使得工程师有充分的回旋余地在各种环境中部署算法，而无需重写原始代码。

工程师可以将算法部署为独立可执行文件（包括 **Web App**）或用于面向服务的架构（**ROS**、**AUTOSAR**）和实时硬件（**CPU**、**GPU**、**FPGA**）的代码（用于 **GPU**、**HDL** 的 **C**、**C++**、**CUDA** 代码）。通过这些部署途径，您可以与 150 多个工具接口进行集成。您还可以与 **CAN**、**FMI/FMU**、**Python®** 和 **ONNX™** 进行集成。此外，工具还需要适合通用软件开发工作流，例如持续集成、自动化测试、代码分析和 **ISO 26262**，如图 7 所示。



Commonly used tools: MATLAB Coder, Embedded Coder, GPU Coder, HDL Coder, ROS Toolbox, AUTOSAR Blockset, DDS Blockset, Simulink Test, Simulink Coverage, Polyspace, IEC Certification Kit

图 7. 开发 AD 软件应用程序。

整合

一旦您成功仿真和测试了算法/功能和系统预期遇到的所有情况，并且可以验证其性能，就能建立对其的信任。测试工作流程应包括需求的链接、单元级评估、单元集成，然后是系统级评估。评估应包括功能评估和代码评估。工程师可以在纯仿真模式、软件在环、处理器在环、硬件在环或真实系统中根据需求进行系统测试。需要测试的场景即使没有几千个，也至少有数百个，因此 AD 工程师最好采用自动化测试，而不是手动运行它们。此[自动化测试示例](#)说明了如何根据需求定义场景，并对组件和为这些组件生成的代码进行自动化测试，从而评估 ADAS/AD 功能。这种测试自动化也适用于[持续集成](#)工具，例如 Jenkins。

开发 ADAS/AD 应用是个令人激动的行业，它涉及多个工程领域。这些应用也给汽车行业带来前所未有的复杂性。为了让汽车工程师在构建 ADAS/AD 应用时能够成功管理这种复杂度，汽车工程，包括仿真的使用、工程师的技能以及软件的开发和部署，均需要进行根本性的变革。工程师需要工具来验证功能或系统对于所有预期的用例都能正常工作，从而避免浪费宝贵的金钱和时间来重新设计。[MATLAB](#)、[Simulink](#) 和 [RoadRunner](#) 可以帮助工程师驾驭这些不同领域，成功开发出 ADAS/AD 应用并将其投向市场。

了解更多

- [探索自动驾驶解决方案](#)
- [联系我们](#)