Integration of PRODRIVER® with Autoware

This document presents the integration of PRODRIVER® (motion-planning software from Embotech) with Autoware.

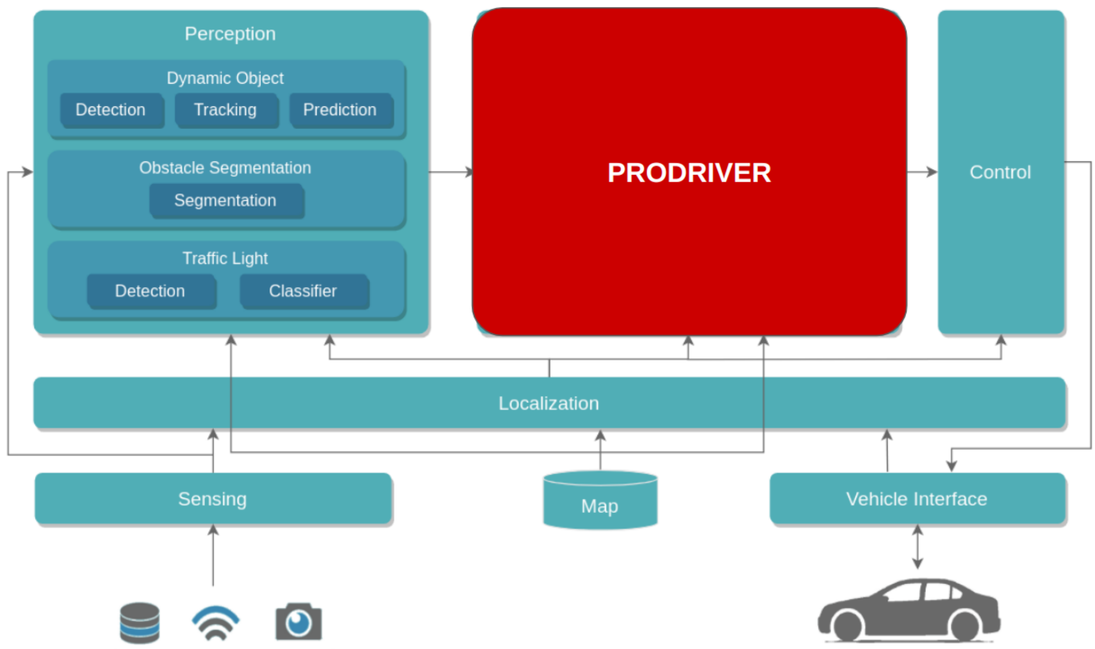
Autoware and PRODRIVER®

Autoware is an autonomous driving, open-source platform which can be customized for each desired use case/ODD. PRODRIVER® is a motion-planning stack provided by Embotech which computes a safe, collision-free, and efficient trajectory to be followed by the vehicle. This document reports the possibility of enhancing Autoware with PRODRIVER®.

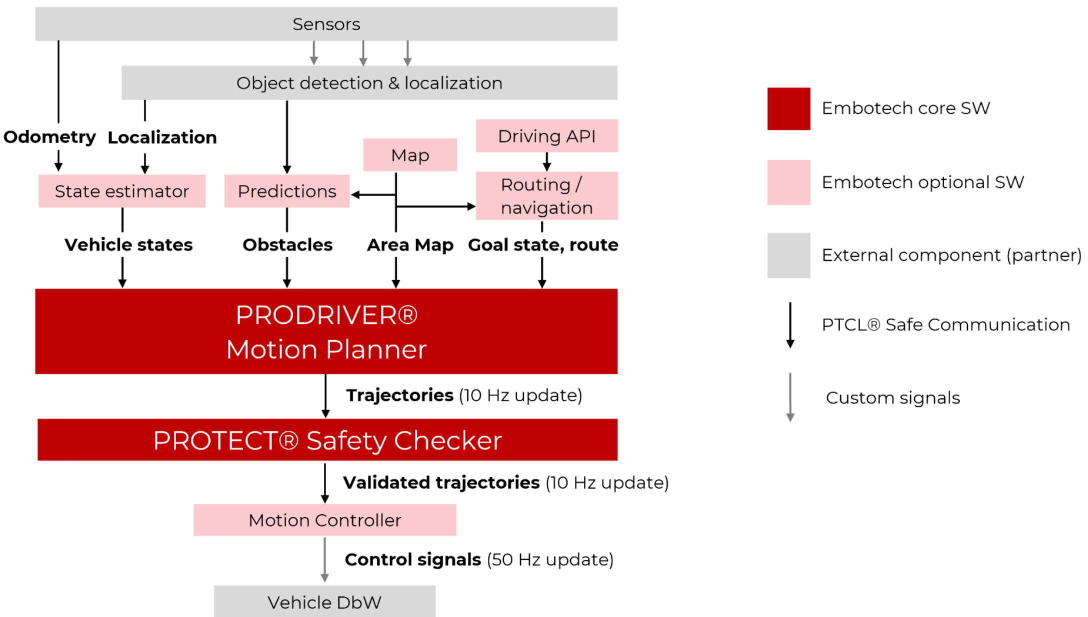
Architecture of Autoware integrated with PRODRIVER®

The diagram below shows the high-level architecture of Autoware using PRODRIVER® as the planning component. In this integration, PRODRIVER® completely replaces the planning module of Autoware, receiving messages from Autoware's perception and localization modules and sending back a trajectory to the controller.
Note that PRODRIVER® is also able to directly generate control commands for the vehicle interface, replacing both the planner and the controller modules of Autoware. In this document however, we focus on using PRODRIVER® as a planning module only.
For more details about Autoware, please see this document.

High-level architecture design



The high-level architecture of PRODRIVER® is shown below. Note that PTCL® is the safe communication protocol developed and used by Embotech. The core component of PRODRIVER® is the motion planner. It takes inputs from the vehicle and the environment and outputs a trajectory which is then checked by PROTECT®, and forwarded to the vehicle only if it is considered safe.



Features of PRODRIVER®

PRODRIVER® has numerous notable features, some of which are presented below.

Overall:

- No ML, no training data – **fast adaptation** to different vehicle types
- Builds on **state-of-the-art real-time optimization technology**
- **Same motion planning core** for all ADAS/AD functions – from ACC to full autonomy
- Real-time execution on **next-generation ECUs**
- Comes with debugging, visualization and diagnostic tools

L**ow speed (<30km/h):**

- **Complex maneuvers** with highest accuracy and minimum time
- **Fully generic:** covers all use cases with the same software, no case-specific programming
- **Multi-vehicle** coordination
- **Vehicle agnostic**
- Support for **trailer parking**
- Lateral and longitudinal combined planning

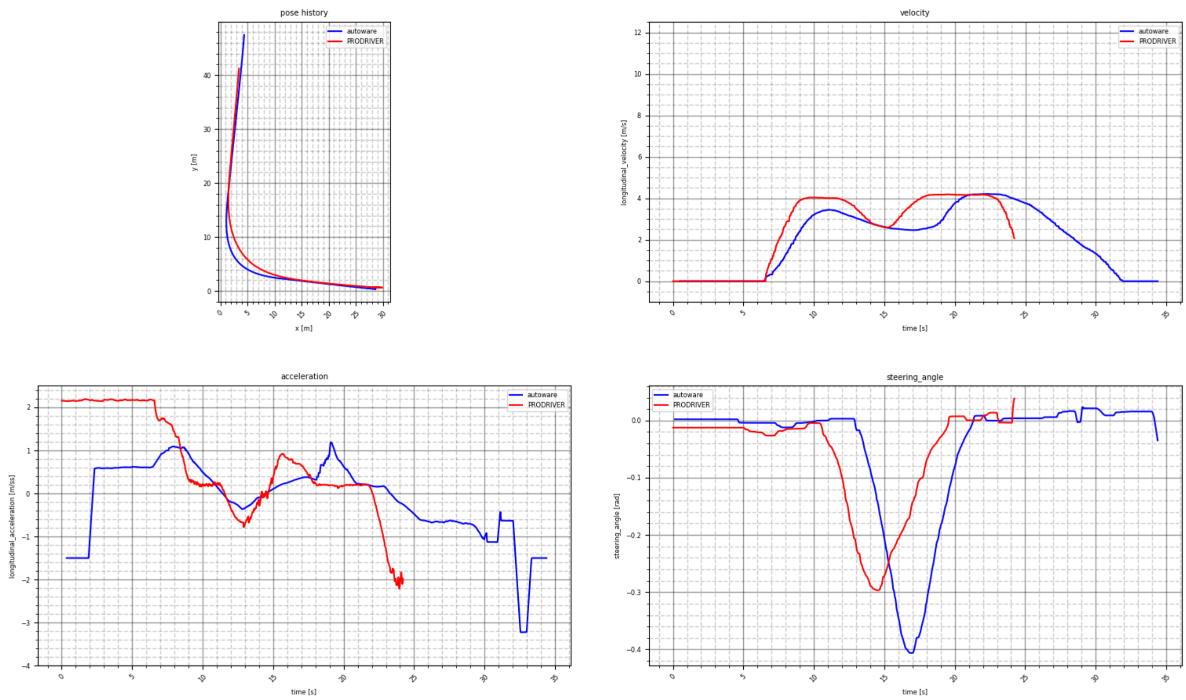
H**igh speed (>30km/h):**

- **Complex maneuvers** at the physical limits
- Highest quality of **safe, comfortable motion** – 12D, physically valid, smooth trajectories
- Customization of **driving style** by customer needs & vehicle type
- Full use of vehicle capabilities

Experiments and results

The successful integration was validated on multiple simulation scenarios as well as on TIER IV's test vehicle. Experiments on the real vehicle were conducted to compare the performance of Autoware's default planner and PRODRIVER® in simple scenarios such as driving straight, turning right/left at an intersection, and avoiding static obstacles. Next, we show detailed results for the intersection scenario on the real vehicle.

This video shows the integrated PRODRIVER® driving in the intersection scenario, while this video shows Autoware driving without PRODRIVER®.
A comparison of the results obtained with both motion planners in terms of pose history, measured velocity, acceleration command and steering angle command is shown below:



Initially, the vehicle is not in autonomous driving mode, meaning that the acceleration and steering commands are not actuated.
After around 7 seconds, the vehicle is engaged and its velocity starts to increase. Looking at the velocity graph, we see that PRODRIVER® (in red) is able to reach the maximum velocity of 4 m/s before slowing down for the left turn. Meanwhile, Autoware (in blue) accelerates more slowly and also starts decelerating earlier. Looking at the graph of the steering angle, we observe that during the turn, PRODRIVER® uses a steering angle of at most -0.3 rad compared to -0.4 rad for Autoware.
After completing the turn, both PRODRIVER® and Autoware accelerate until reaching maximum velocity and decelerate once close to the destination.

In summary, PRODRIVER® achieves a larger average velocity through the corner, as well as a smaller steering angle amplitude, compared to Autoware's default planner.
As a passenger, PRODRIVER®, which runs along the inside of the lane, feels more like a human driving than Autoware, which runs along the center of the lane.
Note that in this setup, PRODRIVER® and the map were initially designed for gated areas, where temporarily crossing an oncoming lane is considered a desirable behavior. Indeed, in such driving scenarios, it can improve driving efficiency and in some cases it can even be necessary to go around tight corners. This behavior is similar to what human drivers do in the same circumstances. It can be modified easily in PRODRIVER®, and it is only allowed if the oncoming lane is explicitly marked as a lane that PRODRIVER® can use if needed.

Steps to integrate PRODRIVER® with Autoware

This section describes how to integrate PRODRIVER® with Autoware. For more detailed information, please visit https://github.com/tier4/embotech_prodriver_connector.

Integrating an external component with Autoware can be done through the following steps:

- Prepare the inputs required by the external component
  - Identify the corresponding Autoware topics and message types
  - Prepare the conversion from the Autoware format to the external component's format
- Prepare the outputs of the external component to be used by Autoware
  - Identify the corresponding Autoware topics and message types
  - Prepare the conversion from the external component's format to the Autoware format
- Identify the communication method between Autoware and the external component
  - E.g., ROS2, network, library calls, …
- Implement the Autoware interface node
  - Subscribes to required inputs, publishes the generated outputs
  - Conversion functions: Autoware↔external component
  - Communication functions: send to and receive data from the external component

Next, we will go through each of these steps for the integration of PRODRIVER® with Autoware.

Inputs

To function, PRODRIVER® needs the following information:

- Map: proprietary format (PTCL®)
  - Note that Lanelet2 → PTCL® conversion can be provided by Embotech
- State of the ego vehicle:
  - Pose (x ,y , heading)
  - Velocities (longitudinal and lateral)
  - Accelerations (longitudinal and lateral)
  - Steering angle
  - Yaw rate
  - Lateral velocity is a plus
- Detected objects:
  - Type (e.g., car, pedestrian, …)
  - Predicted path with the shape and predicted pose at each time step
  - Confidence level

- Goal pose (x, y, heading)

The map is directly given to PRODRIVER® when it is started. The other information must be communicated from Autoware during execution.

- State of the ego vehicle: from localization and vehicle interface
    - Pose and velocities: from Autoware localization
        - Topic: `/localization/kinematic_state`
        - Message: [`nav_msgs/Odometry`]
    - Steering angle: from Autoware vehicle status
        - Topic: `/vehicle/status/steering_status`
        - Message: [`autoware_auto_vehicle_msgs/SteeringReport`]
- Detected objects: from Autoware perception
    - Topic: `/perception/object_recognition/objects`
    - Message: [`autoware_auto_perception_msgs/PredictedObject`]
- Goal state: from Autoware API or HMI
    - Topic: `/planning/mission_planning/goal`
    - Message: [`geometry_msgs/PoseStamped`]

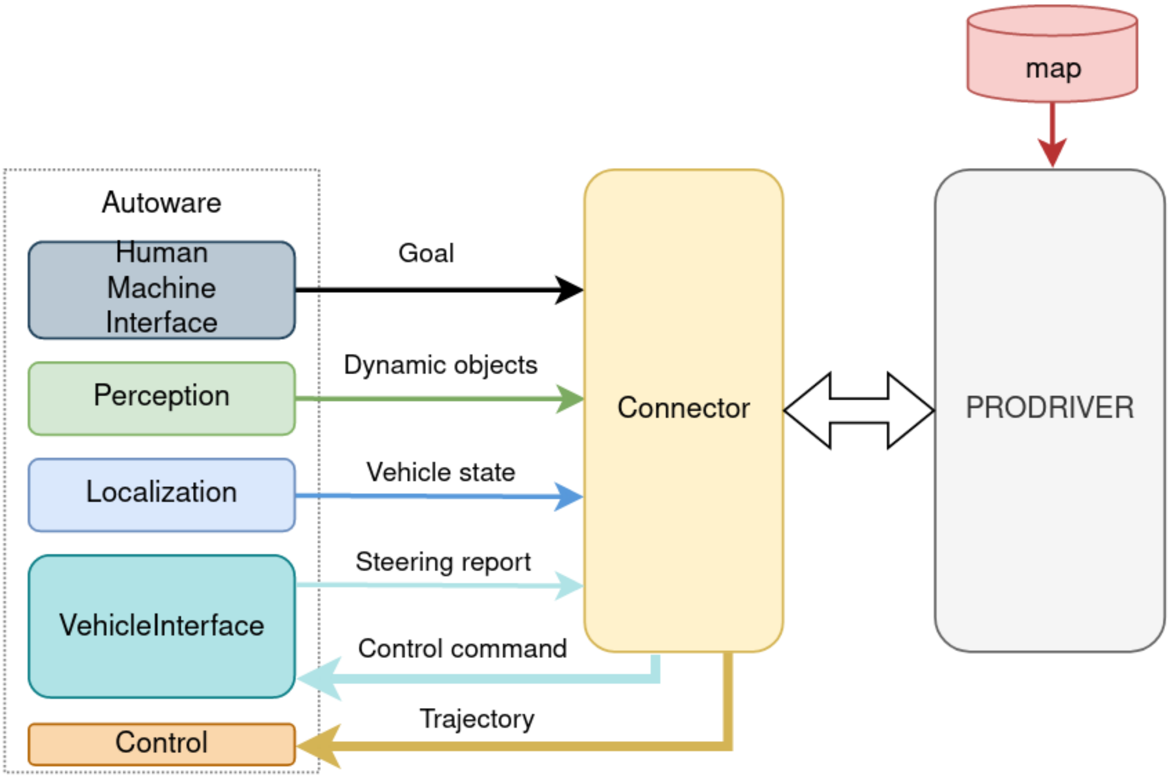
Outputs

PRODRIVER® generates a trajectory that the ego vehicle should follow. This trajectory is made of a sequence of points. Each point consists of:

- A timestamp
- The state of the ego vehicle (pose, velocities, accelerations, yaw rate)
- The feedforward control commands that should be sent to the vehicle (longitudinal acceleration and curvature)

We have two options for using this information to control the ego vehicle in Autoware:

- The trajectory can be sent to the Autoware controller
- The feedforward control commands can be extracted from the trajectory to be directly sent to the Autoware vehicle interface



Conversion

We implemented some conversion functions between the Autoware messages and the data format used by PRODRIVER®. These conversion functions mostly require changing the structure of the information, from Autoware ROS2 messages to PRODRIVER® C structs, and vice-versa.

Outside of the structure, the main change is in the cartesian (x-y) coordinate system. PRODRIVER® uses its own map representation, which is different from Autoware's. Autoware uses the MGRS coordinate system whereas PRODRIVER® uses the UTM system. This means that each position must be converted when communicated by the connector (e.g. when sending a goal pose to PRODRIVER®, or when sending a trajectory to Autoware). To convert between the two coordinate systems, we rely on the `lanelet2_projection` library which allows to easily convert MGRS and UTM coordinates to and from intermediate latitudinal and longitudinal coordinates. The MGRS projector must be configured with a code to identify the 100km grid square containing the map. The UTM projector must be configured with the latitudinal and longitudinal coordinates of the origin of the PRODRIVER® map.

Communication with PRODRIVER®

PRODRIVER® uses UDP communication to send and receive the previously discussed information. A C-language library called PTCL® is provided together with PRODRIVER® to set up the UDP connections and to send or receive messages.

ROS2 connector node

The connector node combines all the previously prepared functions to provide the full integration of PRODRIVER® with Autoware. It is written as a ROS2 C++ node.

Parameters

The node uses parameters defined at launch time to configure the network and the map coordinates. The network parameters must match the configuration of PRODRIVER® and whether it is running on the same machine as the connector, or on a remote machine. The coordinate parameters are map-specific, with the MGRS code and the latitudinal and longitudinal coordinates of the origin of the PRODRIVER® map.

Setup

When started, the connector sets up multiple callback functions defining what to do when receiving ROS2 messages from Autoware, or UDP messages from PRODRIVER®.
Another callback function is created on a timer to periodically send trajectories and control commands to Autoware.
The connector also sets up the network used to communicate with PRODRIVER®. Functions provided by the PRODRIVER® library (PTCL®) are called with the network parameters. Finally, the projectors used for coordinate conversions are set up with the coordinate parameters.

Callbacks

The callback functions for steering and acceleration messages simply cache the received messages. For the dynamic obstacles, goal, and vehicle state messages, the information is not cached but is directly converted and forwarded to PRODRIVER®. In the case of vehicle state messages, the conversion also makes use of the cached steering and acceleration which are then combined into a single PRODRIVER® structure. The callback function for receiving trajectory messages from PRODRIVER® simply caches the received message. The timer callback function then periodically uses the latest received trajectory message to generate the trajectory and control command messages to send to Autoware.