

Ackermann kinematic model

Autoware now supports control inputs for vehicles based on an Ackermann kinematic model. This section introduces you a brief concept of the Ackermann kinematic model and explains how Autoware controls it. Please remember, Autoware control output (/control/command/control_cmd) publishes lateral and longitudinal commands according to the Ackermann kinematic model.

- If your vehicle does not suit the Ackermann kinematic model, you have to modify the control commands. [Another document gives you an example how to convert your Ackermann kinematic model control inputs into a differential drive model.](#)

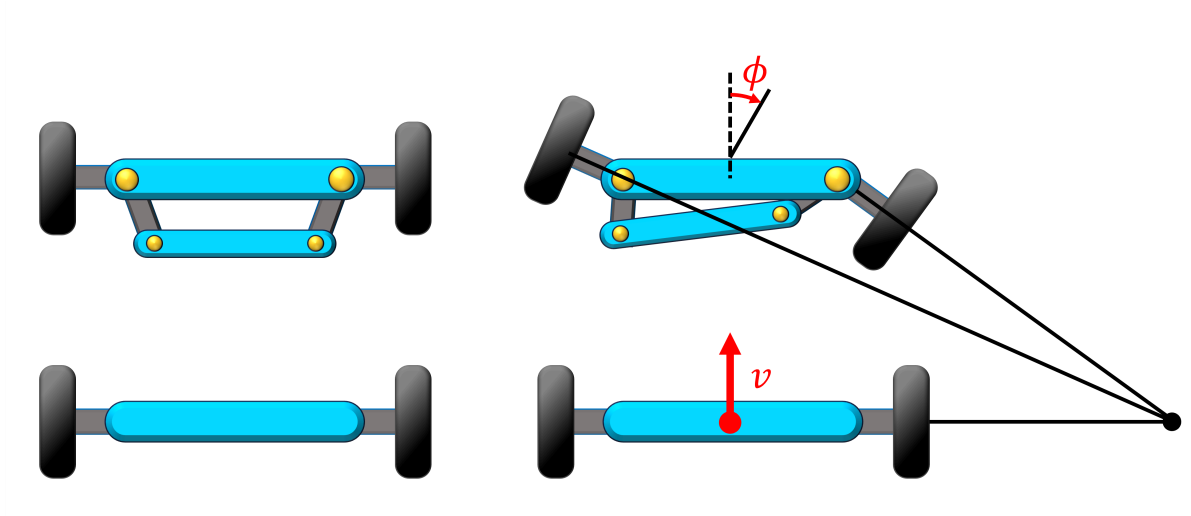
Geometry

The basic style of the Ackermann kinematic model has four wheels with an Ackermann link on the front, and it is powered by the rear wheels.

The key point of Ackermann kinematic model is that the axes of all wheels intersect at the same point, which means all wheels will trace a circular trajectory with a different radii but a common center point (See the figure below).

Therefore, this model has a great advantage that it minimizes the slippage of the wheels and prevents tires from getting worn soon.

In general, the Ackermann kinematic model accepts the longitudinal speed v and the steering angle ϕ as inputs. In autoware, ϕ is positive if it is steered counterclockwise, so the steering angle in the figure below is actually negative.



The basic style of an Ackermann kinematic model. The left figure shows a vehicle facing straight forward, while the right figure shows a vehicle steering to the right.

Control

Autoware publishes a ROS 2 topic named `control_cmd` from several types of publishers.

A `control_cmd` topic is a `AckermannControlCommand` type message that contains

AckermannControlCommand

```
builtin_interfaces/Time stamp
autoware_auto_control_msgs/AckermannLateralCommand lateral
autoware_auto_control_msgs/LongitudinalCommand longitudinal
```

where,

AckermannLateralCommand

```
builtin_interfaces/Time stamp
float32 steering_tire_angle
float32 steering_tire_rotation_rate
```

LongitudinalCommand

```
builtin_interfaces/Time stamp
float32 speed
float32 acceleration
float32 jerk
```

See the [AckermannLateralCommand.idl](#) and [LongitudinalCommand.idl](#) for details.

The vehicle interface should realize these control commands through your vehicle's control device.

Moreover, Autoware also provides brake commands, light commands, and more (see [vehicle interface design](#)), so the vehicle interface module should be applicable to these commands as long as there are devices available to handle them.