Virtual Differential

Development and verification

Development: Alessandro Luppi

Verification: Davide Capuzzo

Last revision on 03/06/2020

# Introduction

The virtual differential is a subsystem of the complete all-wheel-drive electric vehicle under development. Its aim is to produce a reference wheel speed, for each of the four wheels, which will be tracked by the electric motor driving the wheel, with the eventual addition of correction factors for enhanced vehicle stability. The introduced differential model is called “virtual” as there not exist a mechanical component transferring power from a mechanical engine to the wheels; rather, the virtual differential will be implemented by an electronic control unit actively controlling actuators which will drive the wheels’ rotation.

The objective of this dissertation is to introduce the virtual differential model, and how it has been developed and verified.

In the following there will be introduced:

* the requirements of the virtual differential subsystem which drove the design;
* the development process;
* use and interface of the developed subsystem;
* the verification and validation processes

## System requirements

In order to interface with the model, the following software modules are required:

* MATLAB R2019b and Simulink

No additional libraries are required.

# Design requirements

The virtual differential model must provide the reference speeds for the four wheels of the all-wheel-drive electric vehicle.

A list of the requirements which drove the design is reported:

* The virtual differential must provide the reference speed of all four wheels of the electric vehicle.
* The speeds must be compatible with those provided by a mechanical differential on a standard vehicle; in particular, they should comply with the speeds provided by the Ackerman model, assuming a front-steering vehicle: when turning, outer wheels must rotate faster than inner wheels.
* The reference wheels’ speeds must conform with the desired vehicle speed, set by the driver through the throttle pedal; in case of advanced driving (ADAS) systems, the speed may be set by a properly functioning ADAS feature.
* The reference wheels’ speeds must be positive (and grater than 0) when vehicle is driving forward, negative (and smaller than 0) when vehicle is reverse moving, and null (equal to 0) when vehicle is (required to be) stopped.
* The virtual differential is allowed to take advantage of some of the vehicle’s signals generated by in-vehicle sensors or by other electronic control units, such as throttle pedal position and steering angle.

# Model-based design

The design of the virtual differential subsystem has been based on the Ackerman steering geometry assuming ideal vehicle and road conditions.

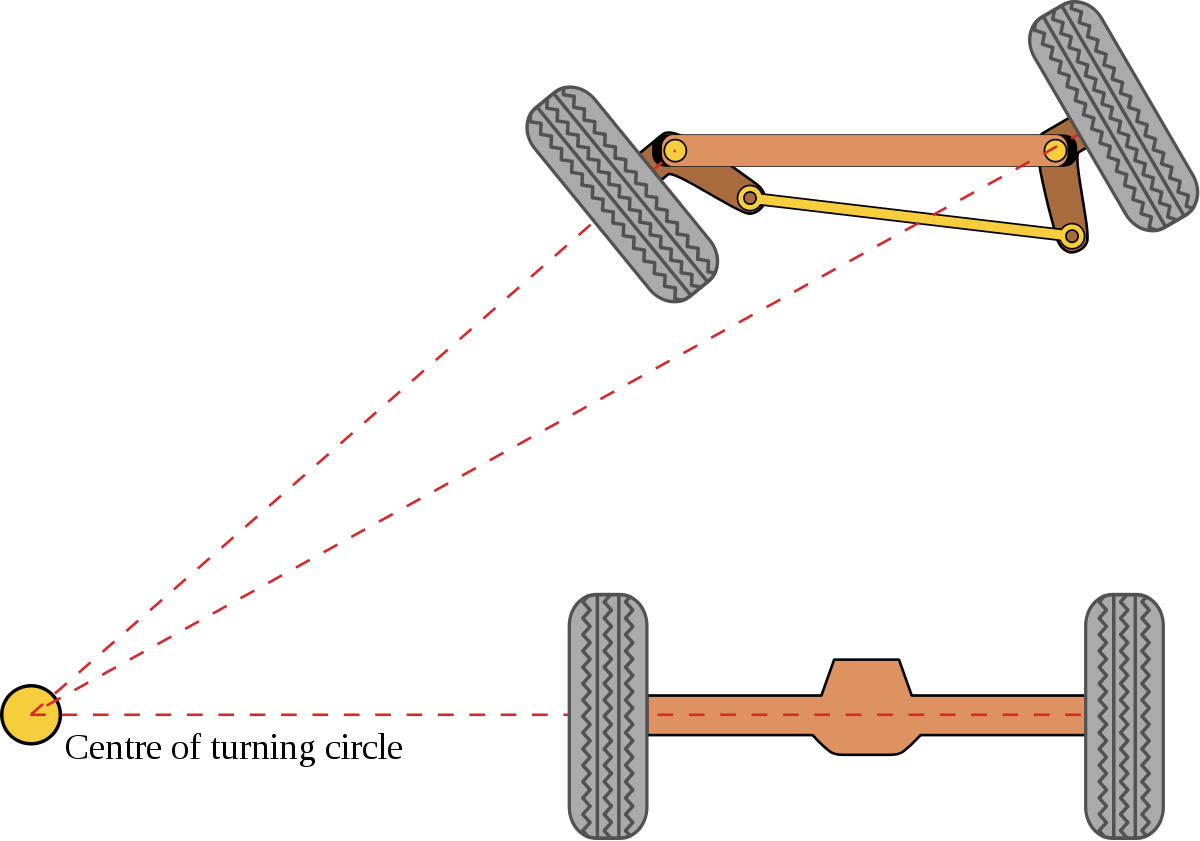
The assumptions for the design are:

* Ideal vehicle and road conditions, leading to the non-slip condition; road is dry asphalt, and vehicle moves on the road without any lateral skidding; eventual skidding will be addressed by other systems.
* Front and rear axles have equal length, and wheels have the same dimension (radius); these parameters are known, as well as the distance between front and rear axles.
* Throttle pedal position reflects the reference vehicle speed, and no other signal influence the requested vehicle speed; action of brake pedal will be addressed by other systems.
* There are sensors providing the instantaneous angular position of the two front wheels.

These assumptions conform with the other developed systems for the all-wheel-drive electric vehicle. The integration of the virtual differential subsystem with other vehicle’s systems introduce further complexity and eliminates some of the idealized assumptions.

The Ackerman steering model describes the reference steering geometry on which most of the vehicles rely on, as it introduces a situation in which none of the wheels of the vehicles is subject to longitudinal slip under ideal driving conditions. Variations from the model allow to reach different levels of performance. It is assumed that the all-wheel-drive electric vehicle under development implements a steering mechanism which conforms with the Ackerman steering model.

According to Ackerman steering geometry, at any steering angle the four vehicle’s wheels rotates around the same center of turning. It means that, during any turn, the lines perpendicular to the wheels’ axis meet at a single point, corresponding to the center of turning around which the vehicle is rotating. Assuming a front-steering vehicle, the center of turning lies on the line connecting the two rear wheels, corresponding, inside the vehicle, with the rear axle.



Once the geometry of the vehicle is known, the speeds of the four wheels can be computed from the knowledge of the steering angle and the (reference) speed of the vehicle.

Calling and the angles from the longitudinal axis of the vehicle to the longitudinal axes of the front wheels respectively, the distance between each front wheel to the center of turning is:

WB being the wheelbase, that is the distance between front and rear axles. The two angles of turning of the two front wheels must be known and they must comply with the Ackerman steering model, meaning that front wheels must be linked by a mechanism imposing Ackerman steering geometry.

The distance between each rear wheel to the center of turning is:

where wheel 3 is the wheel located behind front wheel 1 and wheel 4 is the wheel located behind front wheel 2. The relations do not change switching front wheels 1 and 2.

The angular speed at which the vehicle rotates around the center of turning can be computed as where is the (reference) vehicle speed and is the distance from the center of the vehicle to the center of turning, inferred from geometrical relations.

Finally, the angular speed of the four wheels is computed with the knowledge of the radius of the wheels as:

The angular speeds of the wheels comply with the Ackerman steering geometry and relates to steering angle and vehicle (reference) speed.

# Usage

{TODO, introduction to usage and use cases}

## System interfaces

{TODO, inputs and outputs}

## System parameters

{TODO, parameters required and how to set them}

## How to use

{TODO, suggestions on how to connect the model with other blocks}

# Verification

{TODO: describe verification process (i.e. does it satisfy requirements and interfaces?)}