Enhanced Motion Control of a Self-Driving Vehicle Using Modelica, FMI and ROS

Nikolas Schröder[1] Oliver Lenord[2] Ralph Lange[2]

[1]Institute of Flight Mechanics and Control, University of Stuttgart, Germany,
lrt86824@stud.uni-stuttgart.de
[2]Robert Bosch GmbH, Germany, {oliver.lenord, ralph.lange}@de.bosch.com

Extended Abstract

Industrial logistics is an important application area of autonomous robotics. In the last years, a number of elaborate algorithms for task scheduling, coordination and path planning for fleets of self-driving vehicles in such applications have been proposed. Prerequisite to apply these strategies is a reliable vehicle motion control. Trajectories commanded by the planner need to be properly executed by the drive platform to ensure that the goals of the mission are met in time and space.

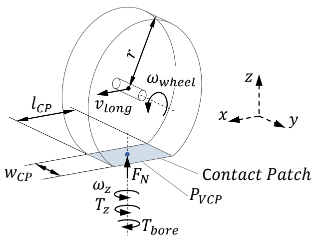

Figure 1. Tire road contact.

Model-based control design is a well established approach to design and apply motion control strategies. Model-in-the-loop (MiL) simulations allow to validate and test the control design early on. Optimized controllers can be designed that take the physical properties and system dynamics into account.

In this work, a common problem of motion platforms with differential drive and caster wheels is elaborated. By applying a model-based control design approach the reliability of the motion controller is significantly improved by the so called *Path Filter*. It is based on a new planar wheel model with bore friction (cf. Figure 1) and a control

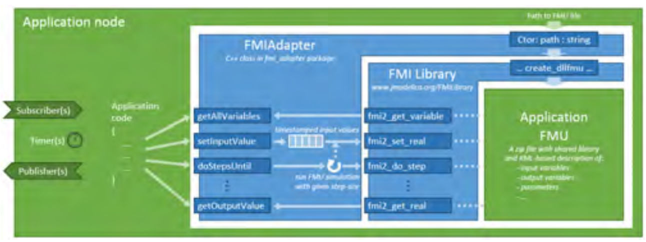

Figure 2. Architecture diagram from fmi_adapter package.

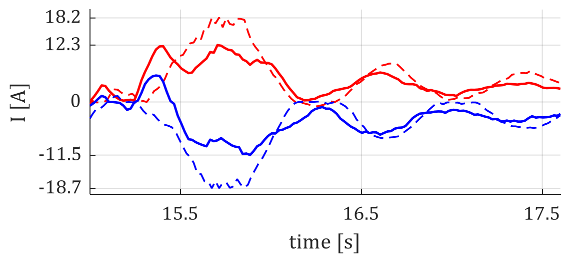

Figure 3. Measured motor currents for initiating a CW turn on the spot. The ActiveShuttle was driving forward prior to that. Dashed lines represent the results w/o path filter, solid lines the results with $\omega_{max} = 100$ rad/s and $T_\varphi = T_\omega = 0.01$ s. Red and blue lines represent the left and right drive, respectively.

strategy to avoid locking conditions caused by the caster wheels. The path filter has been designed as self-contained function that can be retrofitted into an existing control architecture between motion planer and motion controller.

The path filter has been applied to the *ActiveShuttle DevKit*, a self-driving vehicle (SDV) for industrial logistics. The software of the ActiveShuttle is based on the *Robot Operating System* (ROS). In order to enable a generic and efficient control design process, the integration of the Path Filter control function into ROS is facilitated through the Functional Mock-up Interface (FMI). For this purpose, we present the new FMI-Adapter software package (github.com/boschresearch/fmi_adapter/), which allows to integrate Functional Mock-Up Units into ROS nodes as illustrated in Figure 2. An introduction to ROS is given and the mapping between ROS and FMI concepts explained.

The functionality and performance of the Path Filter is tested on the ActiveShuttle DevKit prototype. Our experiments confirm, (cf. Figure 3), that the effort to perform movements that require the caster wheels to be turned on the spot is drastically reduced. Moreover, oscillations in the motor currents caused by the abrupt release of the counteracting bore torque decay much faster. Jerks are reduced, which increases the durability of hardware components and improves the stability of shaky loads. Most important, the risk of getting stuck in a lock condition is drastically reduced.