Flatness-Based Control: Kinematic Car

Tutorial 3: closed-loop dynamic controller

D. Kastelan
d.kastelan@lsr.uni-saarland.de
Chair of Systems Theory and Control Engineering
Saarland University

In this tutorial, a closed-loop dynamic controller will be implemented for the kinematic car that builds on the open-loop controller from from Tutorial 2. The controller equations with $K_0, K_1 > 0$ are repeated here for convenience:

$$\lambda(v, y_1, y_2, \theta, y_r, \dot{y}_r, \ddot{y}_r) = \begin{bmatrix} \ddot{y}_{1,r} \\ \ddot{y}_{2,r} \end{bmatrix} - K_1 \begin{bmatrix} v\cos\theta - \dot{y}_{1,r} \\ v\sin\theta - \dot{y}_{2,r} \end{bmatrix} - K_0 \begin{bmatrix} y_1 - y_{1,r} \\ y_2 - y_{2,r} \end{bmatrix} \tag{1.14a}$$

$$v(0) = \sqrt{\dot{y}_{1,r}^2(0) + \dot{y}_{2,r}^2(0)} \tag{1.14b}$$

$$\dot{v} = \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix} \lambda(v, y_1, y_2, \theta, y_r, \dot{y}_r, \ddot{y}_r) \tag{1.14c}$$

$$\varphi = \arctan\left(\frac{l}{v^2} \begin{bmatrix} -\sin\theta & \cos\theta \end{bmatrix} \lambda(v, y_1, y_2, \theta, y_r, \dot{y}_r, \ddot{y}_r) \right). \tag{1.14d}$$

1. Consider effect of initial condition errors with open-loop controller.
2. Implement the flatness-based dynamic controller:
   (i) Implement (1.14a) and (1.14d) in the `eval` method of the `KinematicCarDynamicControl` class. The Matlab function `dot(A,B,DIM)` is useful in (1.14d) for compactly performing the dot product of vectors arranged in two matrices.
   (ii) Update `carODE` in `KinematicCarSim.m` to accept feedback variables and an argument of type `KinematicCarDynamicControl`. The integration of the controller's dynamical subsystem described by the differential equation (1.14c) is to be performed by the existing ODE solver. Extend the state of the system of ODEs accordingly. You will also need to modify helper function `x2state`.

   Now update the execution of the simulation in `KinematicCarSim.m`:

   (iii) Initialize the extended ODE solver state appropriately.
   (iv) Initialize the instance of `KinematicCarDynamicControl` using the class constructor.
   (v) Modify the call to `ode45` to reflect the new `carODE`.
   (vi) After calling the solver, make sure that the `KinematicCarDynamicControl` object is evaluated at the desired sample times.
   (vii) Include the `KinematicCarDynamicControl` controller variables (v, φ) in the plot results.
   (viii) Pass the the `KinematicCarDynamicControl` object to `KinematicCarAnimate` for animation.
3. Investigate how the dynamic controller is able to reject initial condition errors as a function of controller gains.