

On stability for state-lattice trajectory tracking control

Oskar Ljungqvist, Daniel Axehill and Johan Löfberg

The self-archived postprint version of this conference article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-152455>

N.B.: When citing this work, cite the original publication.

Ljungqvist, O., Axehill, D., Löfberg, J., (2018), On stability for state-lattice trajectory tracking control, *2018 Annual American Control Conference (ACC)*, , 5868-5875.

<https://doi.org/10.23919/ACC.2018.8430822>

Original publication available at:

<https://doi.org/10.23919/ACC.2018.8430822>

Copyright: IEEE

<http://www.ieee.org/>

©2018 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.



On stability for state-lattice trajectory tracking control*

Oskar Ljungqvist¹, Daniel Axehill¹, Johan Löfberg¹

Abstract—In order to guarantee that a self-driving vehicle is behaving as expected, stability of the closed-loop system needs to be rigorously analyzed. The key components for the lowest levels of control in self-driving vehicles are the controlled vehicle, the low-level controller and the local planner. The local planner that is considered in this work constructs a feasible trajectory by combining a finite number of precomputed motions. When this local planner is considered, we show that the closed-loop system can be modeled as a nonlinear hybrid system. Based on this, we propose a novel method for analyzing the behavior of the tracking error, how to design the low-level controller and how to potentially impose constraints on the local planner, in order to guarantee that the tracking error is bounded and decays towards zero. The proposed method is applied on a truck and trailer system and the results are illustrated in two simulation examples.

I. INTRODUCTION

In self-driving vehicles, a typical controller structure for the two lowest levels of control are the *local planner* and the *low-level controller*, see Fig. 1. There also exists a higher-level planner, called a *task planner*, which usually handles logical decisions but will not be considered in this work. The vehicle is assumed to be modeled as a time invariant nonlinear system

$$\dot{x}(t) = f(x, u) \quad (1)$$

where $x \in \mathbb{R}^n$ denotes the states and $u \in \mathbb{R}^m$ denotes the control signals. We assume that the task planner is feeding the local planner with a desired goal position of the system. The role of the two lowest levels of control is as follows: the local planner is constructing a feasible nominal trajectory from a starting position to a desired goal position and the low-level controller's objective is to control the system such that the nominal trajectory is tracked with a small tracking error. More specifically, the local planner that is considered in this work is the so called *lattice planner* that was originally introduced in [1]. The lattice planner is an efficient planning algorithm that has been commonly used as motion planner in self-driving vehicle applications [2], [3], [4]. The lattice planner uses a finite number of feasible motions that are constructed offline. These motions will be further referred to as *motion primitives*. The lattice planner finds a solution, called a *plan*, to the motion planning problem by combining a finite number of motion primitives. This sequence of motion primitives constructs a collision free trajectory that moves the system from its starting position to a desired goal position while minimizing a given performance measure.

In particular for vehicular systems, their dynamics are abruptly changing depending on the direction of motion. Thus, a natural approach is to use different low-level controllers in forward and backward motion. This results in

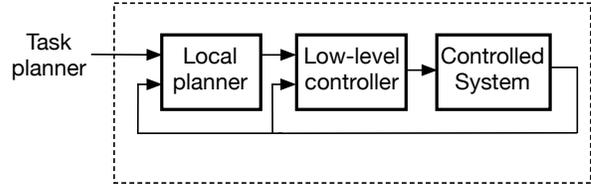


Fig. 1: The closed-loop system architecture considered in this work consists of the controlled system, the low-level controller and the local planner.

a switched closed-loop system depending on the direction of motion [5]. Furthermore, since the nominal trajectory is changing over time, the dynamics of the closed-loop system will become time-varying and will depend on the sequence of motion primitives the local planner is feeding the low-level controller.

Based upon these facts, this paper is dedicated to analysis of the behavior of the closed-loop system consisting of the controlled system and the low-level controller executing a trajectory computed by a lattice planner. This is performed by modeling it as a hybrid system [6]. The motivation for this work is the fact that it is well-known from the theory of hybrid systems that switching between stable systems in an inappropriate way can lead to instability of the switched system [6].

We exploit the structure of the lattice planner and propose a parallelizable framework that, individually for each motion primitive, synthesizes a low-level controller such that the tracking error is bounded and exponentially decays towards zero along each motion primitive. In the controller synthesis, the nonlinear system is described as a Linear Time-Varying (LTV) system and the low-level controller is synthesized by solving a set of Linear Matrix Inequalities (LMIs). Furthermore, a local discrete representation of the continuous-time hybrid system as in [7] is derived which describes the state evolution of the tracking error between each switching point. We show that if this discrete-time switched system has a common quadratic Lyapunov function, the tracking error at the switching points is bounded and will exponentially decay towards zero for any arbitrary trajectory the lattice planner is constructing. We show that this implies that the tracking error in continuous-time has an upper bound, where the upper bound decays towards zero as a function of the number of executed motion primitives.

A. Related work

One way of performing stability analysis of nonlinear systems is to describe them as Linear Differential Inclusions (LDIs) for which a common quadratic Lyapunov function can be searched for using LMI techniques [8]. In general, this method is rather conservative and it is only a sufficient stability condition. However, application results can be found in our previous work in [9] where path-following control

*The research leading to these results has been funded by FFI/VINNOVA.

¹Division of Automatic Control, Linköping University, Sweden, (e-mail: {oskar.ljungqvist, daniel.axehill, johan.lofberg}@liu.se)

for a reversing general 2-trailer system is considered and a method for analyzing local quadratic stability of the closed-loop system is presented. However, the method only considers motions in one direction and in the current work it will be shown that the method in [9] will fail to be able to guarantee quadratic stability when the nominal path is constructed by combining forward and backward motion segments.

Concerning stability analysis of hybrid systems, Lyapunov stability can be guaranteed if a common Lyapunov function exists for all the subsystems simultaneously [6]. The hybrid system is then Lyapunov stable under arbitrary switching strategies. In practice, this is rarely the case and the stability analysis of hybrid systems is dominated by extended Lyapunov techniques [6]. A common approach is to use several Lyapunov functions for the different subsystems and then impose constraints on the switching points, in order to guarantee stability [6], [10]. In [11] a method for constructing a piecewise quadratic Lyapunov function for piecewise affine systems is presented. It is shown that the stability condition can be posed in terms of LMIs when the switching occurs w.r.t. switching planes. However, in this work the switching between each subsystem is not w.r.t. switching planes. Actually, we will show that the closed-loop system considered in this work can be modeled as a time-switched continuous-time hybrid system.

A method for analyzing stability of a time-switched continuous-time hybrid system is presented in [7]. First, a discretization of the continuous-time hybrid systems is performed, where the discrete-time linear hybrid system is describing the state evolution of the continuous-time linear hybrid system between each switching point. The stability analysis is then performed using discrete-time Lyapunov techniques. However, that work only considered continuous-time linear hybrid systems. In this paper, a similar approach as in [7] will be used, but will be applied to nonlinear hybrid systems.

The remainder of the paper is organized as follows. The lattice planning framework is briefly described in Section II. In Section III, the connection between the closed-loop system and a continuous-time hybrid system is explained. In Section IV, the framework for designing the hybrid low-level controller is presented and the behavior of the closed-loop system along each precomputed motion primitive is analyzed. In Section V, the behavior of the continuous-time hybrid system at the switching points is analyzed by performing a discrete-time approximation of the continuous-time hybrid system and in Section VI the method is applied to a general 2-trailer system.

II. THE LATTICE PLANNER FRAMEWORK

The lattice planner is solving a motion planning problem for a system by constraining the motion of the system to a lattice graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The lattice graph is constructed by sampling the state space of the system in a regular fashion [1]. Each vertex $v \in \mathcal{V}$ represents a discrete state x_d of the system while each edge $e \in \mathcal{E}$ encodes a motion which respects the nonholonomic and physical constraints of the system. The discretization of the lattice defines which discrete states $x_d \in X_d$ the vehicle can reach and the system constraints are encoded in the motion primitive set \mathcal{P} . The size of the motion primitive set is M , i.e., $|\mathcal{P}| = M$, where $M \in \mathbb{Z}_{++} \setminus \infty$. A typical assumption when using lattice

planners is that the system is position-invariant and thus \mathcal{P} can be designed to be so. Every motion primitive $p_i \in \mathcal{P}$ is constructed offline by solving a Two-Point Boundary Value Problem (TPBVP), as in [4], to connect a discrete set of initial states to a discrete set of neighboring states in a bounded neighborhood in free space. The TPBVP solver guarantees that the motions respect the physical constraints imposed by the system, while the position-invariance property ensures that the motion primitives are possible to translate to other discrete states. A motion primitive $p_i \in \mathcal{P}$ is a trajectory $(x_0^i(t), u_0^i(t))$, $t \in [0, t_f^i]$ that satisfies the following properties:

$$\dot{x}_0^i = f(x_0^i, u_0^i) \quad (2a)$$

$$x_0^i(0) = x_s^i \in X_d \quad (2b)$$

$$x_0^i(t_f^i) = x_f^i \in X_d \quad (2c)$$

where x_s^i and x_f^i denotes the initial and final position of the system (1), respectively. An example of a motion primitive set for a general 2-trailer system from an initial state x_s^i where the system is standing straight can be seen in Fig. 3a. In this figure, the motion of the midpoint of the rear axle of the trailer is displayed. Each motion primitive $p_i \in \mathcal{P}$ is finally assigned with a cost J_i .

A planning problem is defined with an initial state x_S , a goal state x_G and a world representation \mathcal{W} in which all known obstacles are included. A *feasible solution* to the planning problem is an ordered sequence of m collision-free motion primitives which together construct a feasible trajectory connecting x_S to x_G . Finally, given all feasible solutions, the optimal solution is the one with minimum total cost.

III. CONNECTION TO HYBRID SYSTEMS

In this section it is shown that the closed-loop system can be modeled as a hybrid system consisting of the controlled system and the low-level controller when executing a trajectory constructed by the lattice-planner. The nominal trajectory is constructed online by the lattice planner and is thus a priori unknown. However, we know that it is constructed by combining a finite number of precomputed motion primitives, where each motion primitive is chosen from the set of M possible motion primitives. For motion primitive $p_i \in \mathcal{P}$, denote the nominal trajectory $\dot{x}_0^i(t) = f(x_0^i, u_0^i)$, the control signal deviation $\tilde{u} = u - u_0^i$ and the tracking error state $\tilde{x} = x - x_0^i$. Along motion primitive $p_i \in \mathcal{P}$, the tracking error system can be written as

$$\dot{\tilde{x}} = f(x, u) - f(x_0^i, u_0^i) \triangleq \tilde{f}_i(t, \tilde{x}, \tilde{u}), \quad t \in [0, t_f^i] \quad (3)$$

where $\tilde{f}_i(t, 0, 0) = 0$, $\forall t \in [0, t_f^i]$, i.e., the origin $(\tilde{x}, \tilde{u}) = (0, 0)$ is an equilibrium point. The hybrid system description follows from the fact that the sequence of motion primitives is selected by the lattice planner. Define $q: \mathbb{R}_+ \rightarrow \{1, \dots, M\}$ as a piecewise constant control signal that is controlled by the lattice planner. The tracking error system can be written as a time-switched continuous-time hybrid system:

$$\dot{\tilde{x}} = \tilde{f}_{q(t)}(t, \tilde{x}, \tilde{u}) \quad (4)$$

Thus, (4) is a continuous-time hybrid system composed by M different subsystems where only one subsystem is active at a time. We make the assumptions that $q(t)$ is piece-wise continuous from the right and is chosen such that there are

finitely many switches in finite time [6]. We also assume that each \tilde{f}_i is locally Lipschitz continuous around the origin $\tilde{x} = 0$ uniformly in $t \in [0, t_f^i]$. We are interested in analyzing the behavior of the closed-loop system by designing the hybrid low-level controller $\tilde{u} = g_{q(t)}(\tilde{x})$ and potentially by imposing constraints on the switching sequence $q(t)$. In the next section we will focus on the design of the hybrid low-level controller such that the tracking error is bounded and decaying during the execution of each motion primitive $p_i \in \mathcal{P}$ individually.

IV. LOW-LEVEL CONTROLLER SYNTHESIS

In this section the controller synthesis of the hybrid low-level controller is considered. The synthesis is performed for each motion primitive $p_i \in \mathcal{P}$, separately. The class of hybrid low-level controllers considered is limited to piecewise linear state feedback controllers with feed-forward action. Denote the low-level controller dedicated for motion primitive $p_i \in \mathcal{P}$ as $u(t) = u_0^i(t) + K_i \tilde{x}(t)$. When applying this feedback controller to the nonlinear system in (4), the nonlinear closed-loop system can, in a compact form, be written as

$$\dot{\tilde{x}} = \tilde{f}_i(t, \tilde{x}, K_i \tilde{x}) \triangleq \tilde{f}_{cl,i}(t, \tilde{x}), \quad t \in [0, t_f^i] \quad (5)$$

where $\tilde{x} = \bar{0}$ is an equilibrium point, since $\tilde{f}_{cl,i}(t, \bar{0}) = \bar{0}$, $\forall t \in [0, t_f^i]$. The feedback controller $\tilde{u} = K_i \tilde{x}$ is intended to be designed such that the tracking error is bounded and decays toward zero during the execution of primitive $p_i \in \mathcal{P}$. This is guaranteed by Theorem 1 from [12] where \mathbb{S}_{++}^n denotes the set of symmetric positive definite matrices.

Assumption 1: Assume $\tilde{f}_{cl,i} : [0, t_f^i] \times \tilde{\mathbb{X}} \rightarrow \mathbb{R}^n$ is continuously differentiable w.r.t. $\tilde{x} \in \tilde{\mathbb{X}} = \{\tilde{x} \in \mathbb{R}^n \mid \|\tilde{x}\|_2 < r\}$ and the Jacobian matrix $[\partial \tilde{f}_{cl,i} / \partial \tilde{x}]$ is bounded and Lipschitz on $\tilde{\mathbb{X}}$, uniformly in $t \in [0, t_f^i]$.

Theorem 1: Consider the closed-loop system in (5). Under Assumption 1, let

$$A_{cl,i}(t) = \frac{\partial \tilde{f}_{cl,i}}{\partial \tilde{x}}(t, 0) \quad (6)$$

If there exist a common matrix $P_i \in \mathbb{S}_{++}^n$ and a positive constant ε that satisfy

$$A_{cl,i}(t)^T P_i + P_i A_{cl,i}(t) \preceq -2\varepsilon P_i \quad \forall t \in [0, t_f^i] \quad (7)$$

then the following inequality holds

$$\|\tilde{x}(t)\| \leq \|\tilde{x}(0)\| \kappa(P_i)^{1/2} e^{-\varepsilon t}, \quad \forall t \in [0, t_f^i] \quad (8)$$

where $\kappa(P_i)$ is the condition number of P_i .

Proof: See e.g. [12]. ■

Remark 1: If the final time t_f^i is infinite, the origin of (5) is an uniformly exponentially stable equilibrium point [12].

Theorem 1 tells us that if there exists a common quadratic Lyapunov function $V_i(\tilde{x}) = \tilde{x}^T P_i \tilde{x}$ for (5) around the origin satisfying $\dot{V}_i \leq -2\varepsilon V_i$, $\varepsilon > 0$, then a small disturbance in the initial error state $\tilde{x}(0)$ will result in an error state trajectory $\tilde{x}(t)$ that is bounded and exponentially decays towards the origin. A practical use of Theorem 1 is stability analysis when the feedback gain K_i already is known, e.g., using Linear Quadratic (LQ) control [9].

In case the feedback gain K_i is unknown and to be determined, Theorem 1 can still be used. By using the chain rule, the matrix $A_{cl,i}(t)$ in (6) can be written as

$$\begin{aligned} A_{cl,i}(t) &= \frac{\partial \tilde{f}_i}{\partial \tilde{x}}(t, 0, 0) + \frac{\partial \tilde{f}_i}{\partial \tilde{u}}(t, 0, 0) K_i \\ &\triangleq A_i(t) + B_i(t) K_i \end{aligned} \quad (9)$$

Furthermore, assume the pairs $[A_i(t), B_i(t)]$ lie in the convex polytope \mathbb{P}_i , $\forall t \in [0, t_f^i]$, where \mathbb{P}_i is represented by its L_i vertices

$$[A_i(t), B_i(t)] \in \mathbb{P}_i = \mathbf{Co}\{[A_{i,1}, B_{i,1}], \dots, [A_{i,L_i}, B_{i,L_i}]\} \quad (10)$$

where \mathbf{Co} denotes the convex hull. Condition (7) in Theorem 1 can now be reformulated as [8]:

$$(A_{i,j} + B_{i,j} K_i)^T P_i + P_i (A_{i,j} + B_{i,j} K_i) \preceq -2\varepsilon P_i, \quad j = 1, \dots, L_i \quad (11)$$

This matrix inequality is not jointly convex in P_i and K_i . However, if $\varepsilon > 0$ is fixed, with the bijective transformation $Q_i = P_i^{-1} \in \mathbb{S}_{++}^n$ and $Y_i = K_i P_i^{-1} \in \mathbb{R}^{m \times n}$, the matrix inequality in (11) can be rewritten as an LMI in Q_i and Y_i [13]:

$$Q_i A_{i,j}^T + Y_i^T B_{i,j}^T + A_{i,j} Q_i + B_{i,j} Y_i + 2\varepsilon Q_i \preceq 0, \quad j = 1, \dots, L_i \quad (12)$$

In other words it is an LMI feasibility problem to find a linear state-feedback controller that satisfies condition (7) in Theorem 1. If Q_i and Y_i are feasible solutions to (12), the quadratic Lyapunov function is $V_i(\tilde{x}) = \tilde{x}^T Q_i^{-1} \tilde{x}$ and the linear state-feedback controller is $\tilde{u} = Y_i Q_i^{-1} \tilde{x}$. This controller synthesis can be performed in parallel for all motion primitives $p_i \in \mathcal{P}$ where the feedback controller $\tilde{u} = K_i \tilde{x}$ and the corresponding Lyapunov function $V_i(\tilde{x}) = \tilde{x}^T P_i \tilde{x}$ are dedicated for motion primitive $p_i \in \mathcal{P}$.

If a common Lyapunov function exists that satisfies (12) (i.e. $Q_i = Q$, but Y_i can vary), $\forall p_i \in \mathcal{P}$, then we can directly conclude that the tracking error will exponentially decay towards zero under arbitrary switches [8]. However, this is generally not the case. In proposition 1, we show this for a special case that is closely related to vehicular systems [9].

Proposition 1: Consider the switched linear system

$$\dot{x} = vAx + vBu, \quad v \in \{-1, 1\} \quad (13)$$

where $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$. When $\text{rank}(B) < n$, there exists no hybrid linear feedback control law in the form

$$u = \begin{cases} K_1 x, & v = 1 \\ K_2 x, & v = -1 \end{cases} \quad (14)$$

where $K_1, K_2 \in \mathbb{R}^{m \times n}$, such that the closed-loop system is quadratically stable with a quadratic Lyapunov function $V(x) = x^T P x$, $\dot{V}(x) < 0$ and $P \in \mathbb{S}_{++}^n$.

Proof: In order to find a common quadratic Lyapunov function $V(\tilde{x}) = \tilde{x}^T P \tilde{x}$, $\dot{V}(x) < 0$ and a hybrid feedback controller (14), the following matrix inequalities need to have a feasible solution [13]:

$$(A + BK_1)^T P + P(A + BK_1) \prec 0 \quad (15a)$$

$$-(A + BK_2)^T P - P(A + BK_2) \prec 0 \quad (15b)$$

in the variables K_1, K_2 and $P \succ 0$. Assume a feasible solution exists. Then, by adding together (15a) and (15b) we obtain

$$\tilde{K}^T B^T P + P B \tilde{K} \prec 0 \quad (16)$$

where $\bar{K} = K_1 - K_2 \in \mathbb{R}^{m \times n}$. Since $P \succ 0$, we can pre- and post multiply (16) with P^{-1} without affecting the negative definiteness:

$$P^{-1}\bar{K}^T B^T + B\bar{K}P^{-1} \prec 0 \quad (17)$$

Take a vector $z \in \mathbb{R}^n$ that lies in the nullspace of the B^T , i.e. $z^T B = 0$. This vector z can always be found since $\text{rank}(B) < n$. Now, pre- and postmultiply (17) with z^T and z , respectively:

$$z^T P^{-1} \bar{K}^T \underbrace{(z^T B)^T}_{=0} + \underbrace{z^T B}_{=0} \bar{K} P^{-1} z = 0 \not\prec 0 \quad (18)$$

Hence, a contradiction and we can conclude that there exists no feasible solution to (15) and the proposition follows. ■

From Proposition 1, we can conclude that for hybrid nonlinear systems where the Jacobian linearization can be written on the form (13), it is not possible to design a hybrid low-level controller in the form (14) such that local quadratic stability can be guaranteed. An example of such a hybrid system is the truck and trailer system in [9].

In the next section we will propose a method for analyzing the behavior of the time-switched continuous-time hybrid system in (4) when the hybrid low-level controller $\tilde{u} = K_{q(t)}\tilde{x}$ already has been designed. The framework is tailored for motion planners that are generating a trajectory by combining a finite number of precomputed motions.

V. CONVERGENCE ALONG A COMBINATION OF MOTION PRIMITIVES

Consider the continuous-time hybrid system in (4) with the hybrid low-level controller $\tilde{u} = K_{q(t)}\tilde{x}$ that has been synthesized by following the method presented in Section IV. Assume motion primitive $p_i \in \mathcal{P}$ is switched in at time t_k , i.e., $q(t) = i$, for $t \in [t_k, t_k + t_f^i]$. The evolution of the tracking error state $\tilde{x}(t)$ along motion primitive p_i then becomes

$$\tilde{x}(t_k + t_f^i) = \tilde{x}(t_k) + \int_{t_k}^{t_k + t_f^i} f_{cl,i}(t, \tilde{x}) dt \triangleq T_i(t_k, \tilde{x}(t_k), t_f^i) \quad (19)$$

where $\tilde{x}(t_k)$ denotes the tracking error state at time t_k and $f_{cl,i}$ is defined in (5). The solution to this integral has generally not an analytical expression. However, by the use of numerical integration, a local approximation of the state evolution of $\tilde{x}(t)$ between the two switching times t_k and $t_k + t_f^i$ can be obtained. A first-order Taylor series expansion of (19) around the origin $\tilde{x}(t_k) = 0$ is

$$\tilde{x}(t_k + t_f^i) \approx T_i(t_k, 0, t_f^i) + \underbrace{\frac{\partial T_i(t_k, 0, t_f^i)}{\partial \tilde{x}(t_k)}}_{=F_i} \tilde{x}(t_k) \quad (20)$$

The term $T_i(t_k, 0, t_f^i) = 0$, since $\tilde{f}_{cl,i}(t, 0) = 0$ for all motion primitives $p_i \in \mathcal{P}$. Denote $\tilde{x}[k] = \tilde{x}(t_k)$, $\tilde{x}[k+1] = \tilde{x}(t_k + t_f^i)$ and $q[k] = q(t_k) = i$. By, e.g., the use of finite differences, the evolution of the tracking error state (19) along motion primitive $p_i \in \mathcal{P}$ can be approximated as a linear discrete-time system

$$\tilde{x}[k+1] = F_i \tilde{x}[k] \quad (21)$$

Repeating this procedure for all motion primitives M , a set of M transition matrices, $\mathbb{F} = \{F_1, \dots, F_M\}$, can be computed.

Then, the discrete-time system that locally around the origin describes the state evolution (19) between the switching instances can be described as a linear discrete-time switched system:

$$\tilde{x}[k+1] = F_{q[k]} \tilde{x}[k], \quad q[k] \in \{1, \dots, M\} \quad (22)$$

where the sequence of motion primitives $q[k]$ is unknown at the time for the analysis. Exponential decay of the solution $\tilde{x}[k]$ to (22) is guaranteed by Theorem 2.

Theorem 2: Consider the linear discrete-time switched system in (22). Suppose there exist a matrix $S \in \mathbb{S}_{++}^n$ and a $\rho \geq 1$ that satisfy

$$I \preceq S \preceq \rho I \quad (23a)$$

$$F_j^T S F_j - S \preceq -\mu S, \quad \forall j \in \{1, \dots, M\} \quad (23b)$$

where $0 < \mu < 1$ is a constant. Then, under arbitrary switching for $k \geq 0$ the following inequality holds

$$\|\tilde{x}[k]\| \leq \|\tilde{x}[0]\| \kappa(S)^{1/2} \lambda^k \quad (24)$$

where $\lambda = \sqrt{1 - \mu}$ and $\kappa(S) = \rho$ denotes the condition number of S .

Proof: The proof is based on results in [14]. Define the quadratic Lyapunov function $V(\tilde{x}[k]) = \tilde{x}^T[k] S \tilde{x}[k]$. For any $\tilde{x}[0]$ and corresponding solution $\tilde{x}[k]$ of (22), the matrix inequality in (23b) implies

$$V(\tilde{x}[k+1]) - V(\tilde{x}[k]) \leq -\mu V(\tilde{x}[k])$$

Define $\lambda^2 = 1 - \mu > 0$. Then

$$V(\tilde{x}[k+1]) \leq \lambda^2 V(\tilde{x}[k]) \quad (25)$$

This inequality holds for all $k \geq 0$. By iterating (25) from $k = 0$ for $m > 0$ times, it follows that

$$V(\tilde{x}[m]) \leq \lambda^{2m} V(\tilde{x}[0]) \quad (26)$$

Using (23a) we can establish the inequalities

$$\|\tilde{x}[m]\|^2 \leq V(\tilde{x}[m]) \quad (27a)$$

$$V(\tilde{x}[0]) \leq \rho \|\tilde{x}[0]\|^2 \quad (27b)$$

By combining (26) and (27) we get

$$\|\tilde{x}[m]\|^2 \leq V(\tilde{x}[m]) \leq \lambda^{2m} V(\tilde{x}[0]) \leq \lambda^{2m} \rho \|\tilde{x}[0]\|^2 \quad (28)$$

Furthermore, since $\lambda > 0$ and $\rho > 0$, by taking the square root of (28) we obtain

$$\|\tilde{x}[m]\| \leq \|\tilde{x}[0]\| \kappa(S)^{1/2} \lambda^m \quad (29)$$

■

Remark 2: Theorem 2 establishes uniform exponential stability of the origin for the linear discrete-time switched system in (22) if we let $k \rightarrow \infty$. This is, however, not practically relevant in the lattice planner application since k is finite.

For a fixed μ , (23) is a set of LMIs in the variables S and ρ . With μ , ρ and S as variables, the matrix inequities in (23) are a generalized eigenvalue problem and bisection can be used to solve the optimization problem while, e.g., maximizing the decay rate μ and/or minimizing the condition number ρ of the matrix S .

The result in Theorem 2 concludes that the upper bound on the tracking error at the switching instances, exponentially decays towards zero. Thus, the norm of the initial tracking error $\|\tilde{x}(t_k)\|$, when starting the execution of a new motion primitive, will decrease as k grows. Moreover, using Theorem 1, this implies that the upper bound on the continuous-time tracking error $\tilde{x}(t)$ will exponentially decay towards zero. This result is formalized in Corollary 1.

Corollary 1: Consider the hybrid system in (4) with the controller $\tilde{u} = K_{q(t)}\tilde{x}$. Assume the conditions in Theorem 1 are satisfied for each mode $i \in \{1, \dots, M\}$ of (4) and assume the conditions in Theorem 2 are satisfied for the resulting discrete-time switched system (22). Then, $\forall k \in \mathbb{Z}_+$, $t \in [t_k, t_k + t_f^i]$ with $q(t_k) = i$, the continuous-time error state trajectory $\tilde{x}(t)$ satisfies

$$\|\tilde{x}(t)\| \leq \|\tilde{x}(t_0)\| \kappa(S)^{1/2} \kappa(P_i)^{1/2} \lambda^k \quad (30)$$

where $P_i, S \in \mathbb{S}_{++}^n$ and $0 < \lambda < 1$.

Proof: From inequality (8) in Theorem 1 it is obtained, for $t \in [t_k, t_k + t_f^i]$, that the tracking error $\tilde{x}(t)$ is bounded as

$$\|\tilde{x}(t)\| \leq \|\tilde{x}(t_k)\| \kappa(P_i)^{1/2} e^{-\varepsilon(t-t_k)} \leq \|\tilde{x}(t_k)\| \kappa(P_i)^{1/2} \quad (31)$$

since $\varepsilon > 0$ according to the assumptions. Furthermore, from inequality (24) in Theorem 2, it holds that

$$\|\tilde{x}(t_k)\| \leq \|\tilde{x}(t_0)\| \kappa(S)^{1/2} \lambda^k \quad (32)$$

By inserting (32) into (31) it follows that

$$\|\tilde{x}(t)\| \leq \|\tilde{x}(t_0)\| \kappa(S)^{1/2} \kappa(P_i)^{1/2} \lambda^k \quad (33)$$

■

The practical interpretation of Corollary 1 is that the upper bound on the continuous-time tracking error is exponentially decreasing in the number of executed motion primitives.

The section is concluded by summarizing the work flow that has been presented in Section IV-V as follows:

- 1) For each motion primitive $p_i \in \mathcal{P}$, design a low-level controller $\tilde{u} = K_i \tilde{x}$ such that Theorem 1 holds, by e.g., finding a solution to the LMIs in (12).
- 2) For each motion primitive $p_i \in \mathcal{P}$ derive a discrete-time linear system (21) that locally around the origin describes the state evolution of the error state during the execute the motion primitive (19).
- 3) In order to show that the origin to the continuous-time hybrid system in (4) with the hybrid low-level controller $\tilde{u} = K_{q(t)}\tilde{x}$ is stable, show that the derived discrete-time switched system in (22) satisfies Theorem 2.

VI. APPLICATION RESULTS

In this section we apply the proposed method to control of a general 2-trailer system, i.e., a truck with a dolly-steered trailer with off-axle hitching. The controlled system is schematically illustrated in Fig. 2. Throughout this section, the geometric lengths of the system correspond to our full-sized test vehicle with $L_1 = 4.66\text{m}$, $L_2 = 3.75\text{m}$, $L_3 = 7.59\text{m}$ and $M_1 = 0.8\text{m}$. The optimization problems that are posed in this section are solved in MATLAB using YALMIP [15].

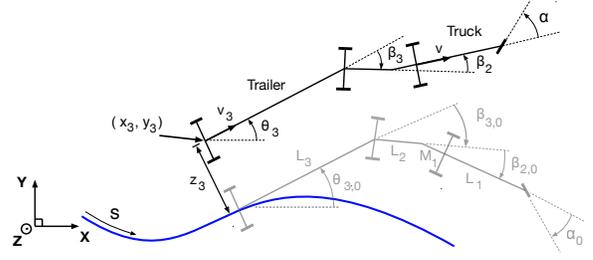


Fig. 2: An illustrative description of the general 2-trailer system represented in the Frenet frame with its moving coordinate system located at the orthogonal projection of the rear axle of the trailer to the blue path $(x_{3,0}(s), y_{3,0}(s))$. The black truck and trailer is the actual system and the gray represents the desired vehicle configuration at this specific value of $s(t)$.

A. The lattice planner

The full details of the lattice planner can be found in [4] but for sake of completeness the main steps are also presented here. Denote α as the truck's steering angle and v as the longitudinal velocity of the rear axle of the truck. With the generalized coordinates $\mathbf{p} = (x_3, y_3, \theta_3, \beta_3, \beta_2)$, the general 2-trailer system can be modeled as [9]:

$$\dot{x}_3 = v_3 \cos \theta_3 \quad (34a)$$

$$\dot{y}_3 = v_3 \sin \theta_3 \quad (34b)$$

$$\dot{\theta}_3 = v \frac{\sin \beta_3 \cos \beta_2}{L_3} \left(1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \quad (34c)$$

$$\dot{\beta}_3 = v \cos \beta_2 \left(\frac{1}{L_2} \left(\tan \beta_2 - \frac{M_1}{L_1} \tan \alpha \right) - \frac{\sin \beta_3}{L_3} \left(1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \right) \quad (34d)$$

$$\dot{\beta}_2 = v \left(\frac{\tan \alpha}{L_1} - \frac{\sin \beta_2}{L_2} + \frac{M_1}{L_1 L_2} \cos \beta_2 \tan \alpha \right) \quad (34e)$$

where v_3 denotes the longitudinal velocity of the rear axle of the trailer and satisfies the following static relationship

$$v_3 = v \cos \beta_3 \cos \beta_2 \left(1 + \frac{M_1}{L_1} \tan \beta_2 \tan \alpha \right) \quad (35)$$

Represent (34) as $\dot{\mathbf{p}} = v_3 f(\mathbf{p}, \tan \alpha)$. The model in (34) is a kinematic model that is derived based on no-slip assumptions, which can be motivated during low-speed maneuvers. Since v enters linearly in (34) a method known as time scaling can be applied to eliminate the speed dependence. Therefore, without loss of generality we hereafter assume that the longitudinal velocity of the truck only takes on the values $v = \{-1, 1\}$. We refer to [9] and the references therein for a more detailed description concerning the modeling. In order to generate a state lattice, the state space of (34) is discretized with an accuracy of $r = 1.0$ m for both $x_{3,d}$ and $y_{3,d}$, and the orientation $\theta_{3,d}$ is discretized into $|\Theta| = 16$ different angles. The system in (34) can be posed in circular equilibrium configurations $(\alpha_e, \beta_{3,e}, \beta_{2,e})$ [4] where $\dot{\beta}_2$ and $\dot{\beta}_3$ are equal to zero. In order to have a tractable dimension in the state lattice, this relationship is used in the discretization. The equilibrium steering angle $\alpha_{e,d}$ is discretized into $|\Phi| = 3$ angles, where $\alpha_{e,d} \in \Phi = \{-0.2117, 0, 0.2117\}$. Furthermore, the motion primitives are generated using numerical optimal

control [16], where the model in (34) is used together with several additional states and constraints, in order to generate steering angles $\alpha(t)$ that are sufficiently smooth to use in a real application [4]. In Fig. 3, the complete set of possible motions from the discrete state $\theta_{3,d} = 0$ are shown. Since, numerical optimal control [16] is used to generate the motion primitives, the resulting trajectories are sampled.

The size of the total motion primitive set is denoted $M = |\mathcal{P}| = 4096$. Hereafter we partition the total motion primitive set \mathcal{P} in two parts. The partitions are \mathcal{P}_{fwd} and \mathcal{P}_{rev} which are the sets of motion primitives that move the system forward ($v = 1$) and backward ($v = -1$), respectively. The size of the partitioned motion primitive sets are $|\mathcal{P}_{\text{fwd}}| = |\mathcal{P}_{\text{rev}}| = 2048$. A motion primitive $p_i \in \mathcal{P}$ here satisfies:

$$\dot{\mathbf{p}}_0^i = v_{3,0}^i f(\mathbf{p}_0^i, \tan \alpha_0^i), \quad t \in [0, t_f^i] \quad (36)$$

which moves the system from an initial position $(\mathbf{p}_0^i(0), \alpha_0^i(0))$ to a final position $(\mathbf{p}_0^i(t_f^i), \alpha_0^i(t_f^i))$ on the lattice grid.

B. Low-level controller synthesis

In this subsection we will synthesize the hybrid low-level controller used to control the lateral and angular tracking error around precomputed trajectories. The framework is partly based on our previous work in [9]. Around motion primitive $p_i \in \mathcal{P}$, denote its nominal trajectory as $(\mathbf{p}_0^i(t), \alpha_0^i(t))$, $t \in [0, t_f^i]$. Similarly as in [9], we can describe the model (34) in terms of deviation from this nominal trajectory, see Figure 2. Perform the input substitution $u = \tan \alpha$ and introduce the state $s(t)$ as the distance traveled by the rear axle of the trailer along its projection onto the path up to time t . The traveled distance along the trajectory $s(t)$ can be modeled as [9]:

$$\dot{s} = v_3 \frac{\cos \tilde{\theta}_3}{1 - \kappa_0(s) z_3} \quad (37)$$

where $\kappa_0(s) = \frac{\tan \beta_{3,0}(s)}{L_3}$ denotes the curvature of the path followed by the rear axle of the trailer. With this new parametrization s , the nominal trajectory in (36) can be written as

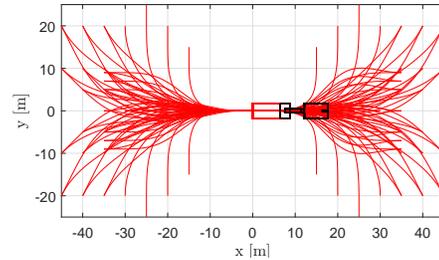
$$\frac{d\mathbf{p}_0^i}{ds} = f(\mathbf{p}_0^i(s), u_0^i(s)), \quad s \in [0, s_f^i] \quad (38)$$

since $ds = v_{3,0}^i dt$. Denote $z_3(s(t))$ as the signed lateral distance between the midpoint of the rear axle of the trailer and its projection onto the path at time t . Define the lateral and angular tracking error states as $\tilde{\theta}_3(t) = \theta_3(t) - \theta_{3,0}^i(s(t))$, $\tilde{\beta}_3(t) = \beta_3(t) - \beta_{3,0}^i(s(t))$, $\tilde{\beta}_2(t) = \beta_2(t) - \beta_{2,0}^i(s(t))$ and the control deviation $\tilde{u}(t) = u(t) - u_0^i(s(t))$. Now, using the chain rule, the lateral and angular tracking error $\tilde{\mathbf{p}} = (z_3, \tilde{\theta}_3, \tilde{\beta}_3, \tilde{\beta}_2)$ can be modeled as [9]:

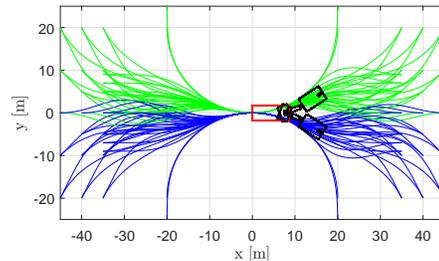
$$\dot{\tilde{\mathbf{p}}} = v \tilde{f}_i(s(t), \tilde{\mathbf{p}}, \tilde{u}), \quad s \in [0, s_f^i] \quad (39)$$

where $\tilde{f}_i(s(t), 0, 0) = 0$, $\forall s \in [0, s_f^i]$. The model in (39) corresponds to one mode of the system in (4). Also, we refer to [9] for details concerning the modeling and the transformation to the Frenet frame coordinate system. The LTV-system that locally around the origin describes the nonlinear system in (39) is

$$\dot{\tilde{\mathbf{p}}} = v A_i(s(t)) \tilde{\mathbf{p}} + v B_i(s(t)) \tilde{u}, \quad s \in [0, s_f^i] \quad (40)$$



(a) The set of motion primitives for $(\theta_{3,i}^d, \alpha_{e,i}^d) = (0, 0)$ to different final states on the lattice grid.



(b) The set of motion primitives for $(\theta_{3,i}^d, \alpha_{e,i}^d) = (0, 0.2117)$ (green) and $(\theta_{3,i}^d, \alpha_{e,i}^d) = (0, -0.2117)$ (blue) to different final states on the lattice grid.

Fig. 3: The set of motion primitives for zero initial heading of the trailer and different initial equilibrium configuration states to different final states on the lattice grid. The colored paths are the paths taken by the rear axle of the trailer (x_3, y_3) during the different motions.

where

$$A_i(s(t)) = \frac{\partial \tilde{f}_i}{\partial \tilde{\mathbf{p}}}(s(t), \bar{0}, 0), \quad B_i(s(t)) = \frac{\partial \tilde{f}_i}{\partial \tilde{u}}(s(t), \bar{0}, 0) \quad (41)$$

For the special case when the nominal trajectory is moving the system either straight forward or backward, the matrices in (41) become:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{L_3} & 0 \\ 0 & 0 & -\frac{1}{L_3} & \frac{1}{L_2} \\ 0 & 0 & 0 & -\frac{1}{L_2} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ -\frac{M_1}{L_1 L_2} \\ \frac{L_2 + M_1}{L_1 L_2} \end{bmatrix} \quad (42)$$

For the system in (40), Proposition 1 states that it is not possible to design the hybrid low-level controller $\tilde{u} = K_{q(t)} \tilde{\mathbf{p}}$ such that we can guarantee local quadratic stability in continuous-time for the closed-loop system when motion primitives in forward and backward motion are combined.

Instead we follow the work flow presented in Section IV and design a feedback gain K_i and a corresponding Lyapunov function $V_i(\tilde{\mathbf{p}}) = \tilde{\mathbf{p}}^T P_i \tilde{\mathbf{p}}$ for each motion primitive $p_i \in \mathcal{P}$, separately. As in Section IV, we do the bijective transformation $Q_i = P_i^{-1}$ and $Y_i = K_i P_i^{-1}$. Moreover, the convex polytope \mathbb{P}_i in (10) is estimated by evaluating the linearization (40) of (39) at each sampled point of the nominal trajectory (38). Each evaluation of the linearization is then assumed to be a vertex for the convex polytope \mathbb{P}_i in (10). The matrix inequalities that need to have a feasible solution in order to guarantee that the tracking error of the closed-loop system is bounded and decays toward zero is (12). For each motion primitive $p_i \in \mathcal{P}$, the low-level

controller synthesis is performed by solving the following optimization problem

$$\begin{aligned} & \underset{Y_i, Q_i}{\text{minimize}} && \|Y_i - K_{LQ}Q_i\|_2 && (43) \\ & \text{subject to} && (12) \text{ and } Q_i \succeq I \end{aligned}$$

with the decay rate $\varepsilon = 0.01$ and K_{LQ} is a nominal feedback gain that in general depends on the motion primitive $p_i \in \mathcal{P}$. Here, however, we have chosen to have only two nominal feedback gains; one for all forward primitives $p_i \in \mathcal{P}_{\text{fwd}}$ and one for all backward primitives $p_i \in \mathcal{P}_{\text{rev}}$. The nominal feedback gains have been generated using LQ-control for the two separate cases when the tracking error system has been linearized around a straight reference trajectory in forward and backward motion, respectively. For this specific case the Jacobian linearization is the matrices A and B defined in (42). The weight matrices used in the LQ-design were chosen as $Q_{LQ} = \text{diag}([1, 10, 8, 2])$ and $R_{LQ} = 1$, and the nominal feedback gains became

$$K_{LQ, \text{fwd}} = [-1.00 \quad -12.12 \quad -6.22 \quad -3.64] \quad (44a)$$

$$K_{LQ, \text{rev}} = [-1.00 \quad 10.52 \quad -8.49 \quad 4.12] \quad (44b)$$

where positive feedback is assumed. For the motion primitives $p_i \in \mathcal{P}_{\text{fwd}}$ the nominal feedback gain is $K_{LQ, \text{fwd}}$ and for the motion primitives $p_i \in \mathcal{P}_{\text{rev}}$ the nominal feedback gain is $K_{LQ, \text{rev}}$. The reason to choose the objective function in (43) is because we would like the feedback controller $\tilde{u} = K_i \tilde{\mathbf{p}}$ to inherit the nominal LQ-controller's properties. The optimization problem in (43) is solved for each motion primitive $p_i \in \mathcal{P}$ and each optimization generates a feedback gain $K_i = Y_i Q_i^{-1}$ and a quadratic Lyapunov function $V_i(\tilde{\mathbf{p}}) = \tilde{\mathbf{p}}^T Q_i^{-1} \tilde{\mathbf{p}}$. For this specific application $\forall p_i \in \mathcal{P}$, the optimal objective function value in (43) is zero, which means that $K_i = K_{LQ}$. Thus, for this specific application the hybrid low-level controller can be written as

$$\tilde{u} = \begin{cases} K_{LQ, \text{fwd}} \tilde{\mathbf{p}}, & p_i \in \mathcal{P}_{\text{fwd}} \\ K_{LQ, \text{rev}} \tilde{\mathbf{p}}, & p_i \in \mathcal{P}_{\text{rev}} \end{cases} \quad (45)$$

However, the quadratic Lyapunov functions are certainly not equal for all motion primitives $p_i \in \mathcal{P}$.

Remark 3: For this specific general 2-trailer system and motion primitive set \mathcal{P} , it was possible to find a common quadratic Lyapunov function $V_{\text{fwd}}(\tilde{\mathbf{p}})$ with a guaranteed decay rate $\varepsilon = 0.01$ and a common feedback controller $\tilde{u} = K_{\text{fwd}} \tilde{\mathbf{p}}$, $\forall p_i \in \mathcal{P}_{\text{fwd}}$. Similarly, it was also possible to find a common quadratic Lyapunov function $V_{\text{rev}}(\tilde{\mathbf{p}})$ with a guaranteed decay rate $\varepsilon = 0.01$ and a common feedback controller $\tilde{u} = K_{\text{rev}} \tilde{\mathbf{p}}$, $\forall p_i \in \mathcal{P}_{\text{rev}}$. However, it was not possible to find a common quadratic Lyapunov function $V(\tilde{\mathbf{p}})$ with a guaranteed decay rate $\varepsilon > 0$ for all motion primitives in both forward and backward motion. This follows directly from Proposition 1.

Practically, Remark 3 means that if we constrain the lattice planner to only generate nominal trajectories by combining motion primitives from either \mathcal{P}_{fwd} or \mathcal{P}_{rev} , we can guarantee that the tracking error is bounded and exponentially decays towards zero. However, in order to guarantee similar properties of the tracking error when the nominal trajectory is constructed by combining forward and backward motion

segments, we need to apply the method presented in Section V.

C. Analyzing the closed-loop hybrid system

To guarantee that the tracking error $\|\tilde{\mathbf{p}}(t)\|$ is bounded and decays toward zero when the nominal trajectory is constructed by any combination of motion primitives, backward as well as forward ones, we apply the method presented in Section V. The closed-loop system is implemented in MATLAB using Simulink. Central differences is used in order to derive the linear discrete-time system in (21) that describes the evolution of the error states (19) during the execution of a motion primitive $p_i \in \mathcal{P}$. With a step size δ the transition matrix $F_i \in \mathbb{R}^{n \times n}$ can be derived by simulating the closed-loop system with an initial error $\pm \delta$ in each tracking error state at a time. In this application, we used the step size $\delta = 0.01$. Since the number of error states are $n = 4$, eight simulations of the closed-loop system are performed in order to generate each transition matrix F_i . This numerical differentiation is performed $\forall p_i \in \mathcal{P}$ and M transition matrices are produced, i.e., $\mathbb{F} = \{F_1, \dots, F_M\}$.

The matrix inequalities in (23) are solved in order to show that the tracking error for the switched system in (22) exponentially decays towards the origin at the switching instants $\tilde{x}(t_k)$. By selecting $0 < \mu < 1$ the semidefinite optimization problem in (46) can be solved. The condition number of S is minimized such that the guaranteed upper bound of the tracking error in (24) is as tight as possible.

$$\begin{aligned} & \underset{\rho, S}{\text{minimize}} && \rho && (46) \\ & \text{subject to} && F_j^T S F_j - S \preceq -\mu S, \quad j = 1, \dots, M \\ & && I \preceq S \preceq \rho I \end{aligned}$$

It turned out that it was not possible to select $0 < \mu < 1$ such that a feasible solution to (46) exists for the original motion primitive set \mathcal{P} . The reason for this is because in the original motion primitive set \mathcal{P} there exist short trajectories of about 1 meter that moves the system either straight forward or straight backward. If these trajectories are switched between we can not guarantee that the tracking error at the switching points will exponentially decay towards zero, which makes sense from a practical point of view. In order to resolve this, the short motion primitives were extended to about 15 meters (as the size of the truck and trailer) and their corresponding discrete-time transition matrices F_i were again derived. With this adjusted motion primitive set \mathcal{P}_{adj} and $\mu = 0.3$ the optimization problem in (46) can be solved and the optimal solution is $\rho = 23.28$ and the common $S \in \mathbb{S}_{++}^n$ is

$$S = \begin{bmatrix} 1.06 & 0.94 & -0.71 & 0.016 \\ 0.94 & 23.019 & -0.22 & 2.24 \\ -0.71 & -0.22 & 22.99 & 2.46 \\ 0.016 & 2.24 & 2.46 & 1.51 \end{bmatrix} \quad (47)$$

Extending the short motion primitives manually is equivalent to adding constraints on the switching sequence $q[k]$ in the lattice planner. For this case, when a short motion primitive $p_i \in \mathcal{P}$ is activated, $q[k]$ needs to remain constant for a certain amount of switching instances. This is a constraint that can easily be added within the lattice planner.

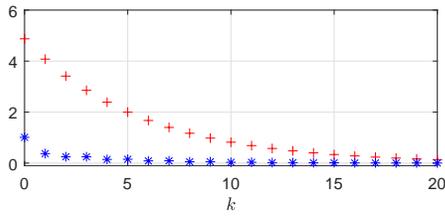


Fig. 4: The norm of the tracking error at the switching points $\|\hat{\mathbf{p}}[k]\|$ (blue stars) and its theoretical upper bound (red crosses) during the simulation where the lengths of the straight motion primitives are 15 meters.

D. Simulation results

To evaluate the tracking performance of the hybrid closed-loop system, the system is simulated when the lattice planner is constructing a nominal trajectory by combining motion primitives in forward and backward motion. As an illustrative example the nominal trajectory is constructed by switching between a motion primitive that moves the system 15 meters straight forward and a motion primitive that moves the system 15 meters straight backward. 20 motion primitive switches are performed. Simulation results are provided in Fig. 4–5 with the initial tracking error $\hat{\mathbf{p}}(0) = (1, 0, 0.1, -0.1)$. In Fig. 4 the norm of the tracking error $\|\hat{\mathbf{p}}[k]\|$ and its theoretical upper bound $\|\hat{\mathbf{p}}[0]\| \sqrt{\rho}(1 - \mu)^{k/2}$ is plotted. The closed-loop system behaves as expected and the tracking error at the switching points $\hat{\mathbf{p}}[k]$ is converging towards zero. In Fig. 5 the error state trajectories are presented. As expected, the tracking error in continuous-time is also bounded and is approaching zero. Another simulation example with the initial tracking error $\hat{\mathbf{p}}(0) = (5, 0, 0, 0)$ is visualized in Fig. 6. In this scenario the vehicle is tracking 90 degrees turns in forward and backward motion. As can be seen, also in this case the control error converges to zero.

VII. CONCLUSIONS AND FUTURE WORK

It is shown that the closed-loop system consisting of the controlled system, the low-level controller and the lattice-planner can be modeled as a nonlinear hybrid system. By exploiting the structure of the lattice-planner, we propose a framework that individually for each motion primitive synthesizes a low-level controller such that the tracking error is bounded and decays towards zero during the execution of this motion primitive. Moreover, in order to guarantee that the closed-loop system is stable for an arbitrary sequence of motion primitives, a discrete-time switched system is derived which locally around the origin describes the evolution of the tracking error between each switching point. Using discrete-time Lyapunov techniques it is shown that the tracking error at each switching point is bounded and exponentially decays towards zero. It is also shown that this implies that the upper bound on the continuous-time tracking error will exponentially decay towards zero as a function of the number of executed motion primitives. Finally, the proposed framework is applied on a general 2-trailer system and the theoretical results are confirmed in practice by simulations.

As future work we would like to rigorously analyze the approximation error in the derivation of the discrete-time switched system and include this in the stability analysis.

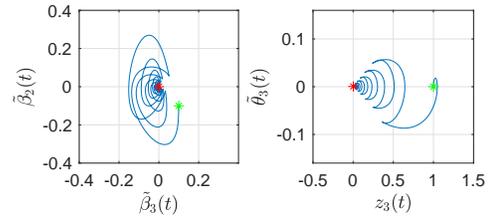


Fig. 5: The tracking error during the simulation where the lengths of the straight motion primitives are 15 meters. The green crosses mark the initial error state and the red crosses mark the final error state.

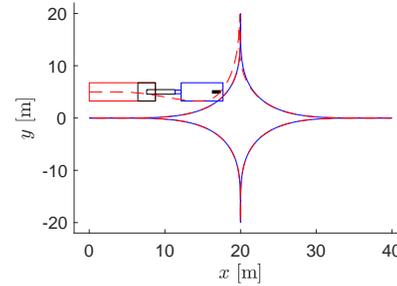


Fig. 6: Simulation when tracking 90 degrees turns in forward and backward motion. The blue path is the nominal trajectory of the rear axle of the trailer and the red dotted path is the path taken by the rear axle of the trailer.

REFERENCES

- [1] M. Pivtoraiko *et al.*, “Differentially constrained mobile robot motion planning in state lattices,” *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.
- [2] D. González *et al.*, “A review of motion planning techniques for automated vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [3] C. Urmson *et al.*, “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [4] O. Ljungqvist *et al.*, “Lattice-based motion planning for a general 2-trailer system,” in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium, Los Angeles, June 2017*, pp. 2455–2461.
- [5] C. Altafini *et al.*, *Hybrid Control of a Truck and Trailer Vehicle*. Springer Berlin Heidelberg, 2002.
- [6] R. A. DeCarlo *et al.*, “Perspectives and results on the stability and stabilizability of hybrid systems,” *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1069–1082, 2000.
- [7] M. Rubensson and B. Lennartson, “Stability and robustness of hybrid systems using discrete-time lyapunov techniques,” in *Proceedings of the 2000 American Control Conference*, vol. 1, no. 6, Sep 2000, pp. 210–214.
- [8] S. Boyd *et al.*, *Linear matrix inequalities in system and control theory*. SIAM, 1994.
- [9] O. Ljungqvist *et al.*, “Path following control for a reversing general 2-trailer system,” in *Proceedings of the 55th IEEE Conference on Decision and Control*, Dec 2016, pp. 2455–2461.
- [10] S. Pettersson and B. Lennartson, “Stability and robustness for hybrid systems,” in *Proceedings of 35th IEEE Conference on Decision and Control*, Dec 1996.
- [11] M. Johansson and A. Rantzer, “Computation of piecewise quadratic lyapunov functions for hybrid systems,” *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 555–559, Apr 1998.
- [12] H. K. Khalil, *Nonlinear Systems*. Prentice-Hall, New Jersey, 1996.
- [13] H. Wolkowicz, *et al.*, *Handbook of semidefinite programming: theory, algorithms, and applications*. Springer Science & Business Media, 2012, vol. 27.
- [14] W. J. Rugh and W. J. Rugh, *Linear system theory*. prentice hall Upper Saddle River, NJ, 1996, vol. 2.
- [15] J. Löfberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” in *2004 IEEE International Symposium on Computer Aided Control Systems Design*. IEEE, 2004, pp. 284–289.
- [16] B. Houska *et al.*, “ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization,” *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.