Hybrid Control Design for a Wheeled Mobile Robot

Bak, Thomas; Bendtsen, Jan Dimon; Ravn, Anders Peter

Publication date:
2003

Document Version
Publisher's PDF, also known as Version of record

Hybrid Control Design for a Wheeled Mobile Robot

Thomas Bak[1], Jan Bendtsen[1], and Anders P. Ravn[2]

[1] Department of Control Engineering, Aalborg University, Fredrik Bajers Vej 7C, DK-9220 Aalborg, Denmark, {tb,dimon}@control.auc.dk
[2] Department of Computer Science, Aalborg University, Fredrik Bajers Vej 7E, DK-9220 Aalborg, Denmark, apr@cs.auc.dk

Abstract. We present a hybrid systems solution to the problem of trajectory tracking for a four-wheel steered four-wheel driven mobile robot. The robot is modelled as a non-holonomic dynamic system subject to pure rolling, no-slip constraints. Under normal driving conditions, a nonlinear trajectory tracking feedback control law based on dynamic feedback linearization is sufficient to stabilize the system and ensure asymptotically stable tracking. Transitions to other modes are derived systematically from this model, whenever the configuration space of the controlled system has some fundamental singular points. The stability of the hybrid control scheme is finally analyzed using Lyapunov-like arguments.

1 Introduction

Wheeled mobile robots is an active research area with promising new application domains. Mobile robots are mechanical systems characterized by challenging (nonintegrable) constraints on the velocities which have led to numerous interesting path tracking control solutions, see [16], [13], [4], and the recent survey of non-holonomic control problems in [11]. Recently, [3] and [1] have addressed the robot path tracking problem from a hybrid systems perspective. In this paper, we consider a problem of similar complexity and develop a systematic approach to derivation of a hybrid automaton and to stability analysis.

Our work is motivated by a project currently in progress, where an autonomous four-wheel driven, four-wheel steered robot (Figure 1) is being developed. The project needs a robot that is able to survey an agricultural field autonomously. The vehicle has to navigate to certain waypoints where measurements of the crop and weed density are obtained. This information is processed and combined into a digital map of the field, which will eventually allow the farm manager to deal with weed infestations in a spatially precise manner. The robot is equipped with GPS, gyros, magnetometer and odometers, which will not only help in the exact determination of the location where each image is taken, but also provide measurements for an estimate of the robot's position and orientation for a tracking algorithm. Actuation is achieved using independent steering and drive on four wheel assemblies (8 brushless DC motors in total). The robot navigates from waypoint to waypoint following spline-type trajectories between

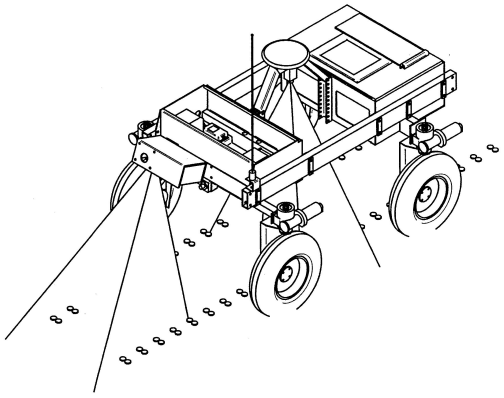


Fig. 1. Schematic model of the experimental platform. The robot is equipped with 8 independent steering and drive motors. Localization is based on fusion of GPS, gyro, magnetometer, and odometer data.

the waypoints to minimize damage to the crop. From a control point of view, this is a tracking problem. To solve this problem a dynamical model of the vehicle subject to pure rolling, no-slip constraints has been developed, following the approach taken in [5] and [6]. Based on this nonlinear model, we design a path tracking control law based on feedback linearization.

Feedback linearization designs have the potential of reaching a low degree of conservativeness, since they rely on explicit cancelling of nonlinearities. However, such designs can also be quite sensitive to noise, modelling errors, actuator saturation, etc. As pointed out in [8], uncertainties can cause instability under normal driving conditions. This instability is caused by loss of invertibility of the mapping representing the nonlinearities in the model. Furthermore, there are certain wheel and vehicle velocity configurations that lead to similar losses of invertibility. Since these phenomena are, in fact, linked to the chosen control strategy rather than the mechanics of the robot itself, we propose in this paper to switch between control strategies such that the aforementioned stability issues can be avoided. This idea is also treated in [15], where singularities in the feedback linearization control law of a ball-and-beam system is treated by switching to an approximate control scheme in the vicinity of the singular points in state space.

In this paper, we intend to motivate the rules for when and how to change between the individual control strategies directly from the mathematical-physical model. We will consider the conditions under which the description may break down during each step in the derivation of the model and control laws. These conditions will then define transitions in a hybrid automaton that will be used as a control supervisor.

However, introducing a hybrid control scheme in order to improve the operating range where the robot can operate in a stable manner comes at a cost:

The arguments for stability become more complex. Not only must each individual control scheme be stable; they must also be stable under transitions (refer to e.g., [12] and the references therein for further information on stability theory for switched systems). A straightforward analysis will show that the system can always be rendered unstable: Just vary the reference input such that transitions are always taken before the transition safe state. We therefore intend to apply the generalized Lyapunov stability theory as introduced by Branicky in [2] to add a second automaton that can constrain the change of the reference input (the trajectory) such that the resulting system remains stable.

We abstract the Lyapunov functions to constant rate functions, where the rates are equivalent to the convergence rate. Each mode or state of the original automaton is then replaced by three consecutive states. The first of these states models the initial transition cost and settling period where the function may increase, albeit for a bounded time, while the second and third state models the working mode with the local Lyapunov function. The third state is the transition safe state, where the Lyapunov function has decreased below its entry value. All three states are guarded by the original conditions for a mode change; but it is potentially unsafe to leave before the third state is entered.

This automaton thus defines safe operating conditions, or put another way: Constraints to be satisfied by the trajectory planner. The composed automaton is in a form where model checking tools can be employed for the analysis. The robot thereby has a tool for determining online whether or not a given candidate trajectory is safe from a stability point of view.

2 Dynamic Model and Linearization

In the following we derive the model and the normal mode control scheme. During the derivation we note conditions for mode changes.

We consider a four-wheel driven, four-wheel steered robot moving on a horizontal plane, constructed from a rigid frame with four identical wheels. Each wheel can turn freely around its horizontal and vertical axis. The contact points between each of the wheels and the ground must satisfy pure rolling and non-slip conditions.[3]

Consider a reference ('field') coordinate system (X_F, Y_F) in the plane of motion as illustrated in Figure 2. The robot position is then completely described by the coordinates (X, Y) of a reference point within the robot frame, which without loss of generality can be chosen as the center of mass, and the orientation θ relative to the field coordinate system of a ('vehicle') coordinate system (X_v, Y_v) fixed to the robot frame. These coordinates are collected in the *posture* vector $\xi = [X\ Y\ \theta]^{\mathrm{T}} \in \mathbb{R}^2 \times \mathbb{S}^1$.

[3] The pure rolling and non-slip conditions can obviously not be satisfied in the real-life application, where the robot drives in a muddy field. They are primarily employed here in order to enable us to derive control laws that minimize the amount of slip and the degree by which the wheels 'work against each other.'

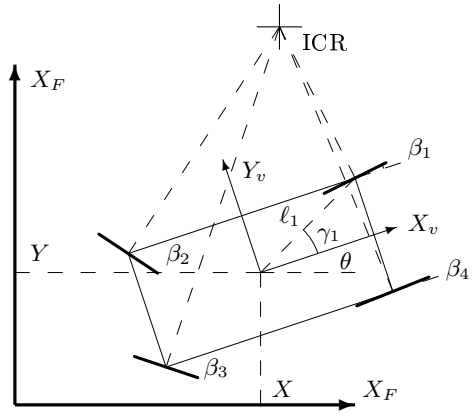


Fig. 2. Definition of the field coordinate system (X_F, Y_F), vehicle coordinate system (X_v, Y_v), vehicle orientation θ, distance ℓ_1, and direction γ_1 from the center of mass (X, Y) to wheel 1. Each wheel plane is perpendicular to the Instantaneous Center of Rotation (ICR).

The position of the i'th wheel $(1 \leq i \leq 4)$ in the vehicle coordinate system is characterized by the angle γ_i and the distance ℓ_i. As the wheels are not allowed to slip, the planes of each of the wheels must at all times be tangential to concentric circles with the center in the Instantaneous Center of Rotation (ICR). The orientation of the plane of the i'th wheel relative to X_v is denoted β_i. The vector $\beta = [\beta_1\ \beta_2\ \beta_3\ \beta_4]^{\mathrm{T}} \in \mathbb{S}^4$ define the wheel orientations.

From an operational point of view a relevant specification of the ICR is to give the orientation of two of the four wheels. We therefore partition β into $\beta_c \in \mathbb{S}^2$ containing the coordinates used to control the ICR location and $\beta_o \in \mathbb{S}^2$ containing the two remaining coordinates that may be derived from the first.

Cross Driving (Singular Wheel Configuration) An important ambiguity (or singular wheel configuration), is present in the approach taken above. For $\beta_1 = \pm\pi/2$ and $\beta_2 = \pm\pi/2$ the configuration of wheels 3 and 4 is not defined. The situation corresponds to the ICR being located on the line through wheel 1 and 2. The wheel configuration $\beta_c = [\beta_3\ \beta_4]^{\mathrm{T}}$ result in similar problems and both configurations fail during cross driving as all wheels are at $\pm\pi/2$. To ensure safe solutions to the trajectory tracking problem we must ensure that the singular configurations are avoided at all times. Based on this discussion we identify three discrete control modes, q_1, q_2 and q_3:

q_1: Trajectory tracking with $\beta_c = [\beta_1\ \beta_2]^{\mathrm{T}}$. This mode is conditioned on $|\beta_1| < (\frac{\pi}{2} - e_\beta) \vee |\beta_2| < (\frac{\pi}{2} - e_\beta)$.

q_2: Trajectory tracking with $\beta_c = [\beta_3\ \beta_4]^{\mathrm{T}}$. This mode is conditioned on $|\beta_3| < (\frac{\pi}{2} - e_\beta) \vee |\beta_4| < (\frac{\pi}{2} - e_\beta)$.

q_3: Cross Driving with $\beta_1 = \beta_2 = \beta_3 = \beta_4$. This mode is conditioned on $(|\beta_1| \geq (\frac{\pi}{2} - e_\beta) \vee |\beta_2| \geq (\frac{\pi}{2} - e_\beta)) \wedge (|\beta_3| \geq (\frac{\pi}{2} - e_\beta) \vee |\beta_4| \geq (\frac{\pi}{2} - e_\beta))$.

where e_β is a small positive number. The two first modes cover the situations where the ICR is governed by wheels 1 and 2 and by wheel 3 and 4, respectively. The last covers the remainder of the configuration space where ICR is approximately at infinity. For brevity of the exposition, we will consider $\beta_c = [\beta_1\ \beta_2]^{\mathrm{T}}$ in the following; the case with $\beta_c = [\beta_3\ \beta_4]^{\mathrm{T}}$ is analogous.

In general, no set of two variables is able to describe all wheel configurations without singularities [14]. The problem of singular configurations is hence not due to the representation used here, but is a general problem for this type of robotic systems.

2.1 Vehicle Model

Following the argumentation in Appendix A, the robot posture can be manipulated via one velocity input $\eta(t) \in \mathbb{R}$ in the instantaneous direction of the wheel orientation state $\Sigma(\beta_c) \in \mathbb{R}^3$, which is constructed to meet the pure-roll constraint. Similarly, it is possible to manipulate the orientation of the wheels via an orientation velocity input $\zeta(t) = [\dot{\beta}_1\ \dot{\beta}_2]^{\mathrm{T}} \in \mathbb{R}^2$. The no-slip condition on the wheels that constrain $\eta(t)$ is handled (see Appendix A) by applying Lagrange formalism and computed torque techniques. The result is the following extended dynamical model:

$$\dot{\chi} = \begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\beta}_c \end{bmatrix} = \begin{bmatrix} 0 & R^{\mathrm{T}}(\theta)\Sigma(\beta_c) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \chi + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \nu \\ \zeta \end{bmatrix} \tag{1}$$

where ν is a new exogenous input that is related to the torque applied to the drive motors, and $R^{\mathrm{T}}(\theta)$ is a coordinate rotation matrix. In equation (1) it is assumed that the β dynamics can be controlled via local servo loops, such that we can manipulate $\dot{\beta}$ as an exogenous input to the model.

2.2 Normal Trajectory Tracking Control

Provided we avoid the singular wheel configurations the standard approach from here on is to transform the states into normal form via an appropriate diffeomorphism followed by feedback linearization of the nonlinearities and a standard linear control design. We choose the new states

$$x_1 = T(\chi) = \begin{bmatrix} \xi_{ref} - \xi \\ \dot{\xi}_{ref} - \dot{\xi} \end{bmatrix}, \tag{2}$$

which yields the following dynamics:

$$\dot{x}_1 = A_1 x_1 + B_1 \left(\delta(\chi) \begin{bmatrix} \nu \\ \zeta \end{bmatrix} - \alpha(\chi) \right), \quad A_1 = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ I \end{bmatrix}. \tag{3}$$

Using the results from Appendix A, $\delta(\chi)$ and $\alpha(\chi)$ may be found to

$$\delta(\chi) = R^{\mathrm{T}}(\theta) \left[\Sigma(\beta_c)\ N(\beta_c)\eta \right] \tag{4}$$

and

$$\alpha(\chi) = \sin(\beta_1 - \beta_2)\eta^2 \begin{bmatrix} -\ell_1 \sin\beta_2 \cos(\beta_1 - \gamma_1) + \ell_2 \sin\beta_1 \cos(\beta_2 - \gamma_2) \\ \ell_1 \cos\beta_2 \cos(\beta_1 - \gamma_1) - \ell_2 \cos\beta_1 \cos(\beta_2 - \gamma_2) \\ 0 \end{bmatrix} \quad (5)$$

where $N(\beta_c) = [N_1\ N_2]$ is specified in equations (20) and (21). When we apply the control law

$$\begin{bmatrix} \nu \\ \zeta \end{bmatrix} = \delta(\chi)^{-1}(\alpha(\chi) - K_1 x_1) \quad (6)$$

we obtain the closed-loop dynamics $\dot{x}_1 = (A_1 - B_1 K_1)x_1$, which tends to 0 as $t \to \infty$ if K_1 is chosen such that $A_1 - B_1 K_1$ has eigenvalues with negative real parts. Similar dynamics can be obtained for the mode with $\beta_c = [\beta_3\ \beta_4]^{\mathrm{T}}$, resulting in closed-loop dynamics $\dot{x}_2 = (A_2 - B_2 K_2)x_2$.

2.3 Cross Driving Control

The normal trajectory tracking cannot be applied in the singular wheel configurations and a specific control must hence be derived that is able to control the vehicle when all wheels are parallel. Fortunately, the dynamics of the robot becomes particularly simple in this case. With $\dot{\theta} = 0$ the dynamics are immediately linear; hence, choosing the states

$$x_3 = T\chi = \begin{bmatrix} \xi_{ref} - \xi \\ \dot{\xi}_{ref} - \dot{\xi} \end{bmatrix}, \quad (7)$$

where T is an appropriate invertible matrix, yields the dynamics

$$\dot{x}_3 = A_3 x_3 + B_3 \begin{bmatrix} \nu \\ 0 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & I \\ A_{31} & A_{32} \end{bmatrix}, \quad B_3 = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (8)$$

which can be controlled by applying the feedback $\nu = -K_3 x_3$. Note that this controller does perform any control on the wheel orientation. In order not to remain in the mode q_3 we impose a new condition, based on the error in orientation, $|\theta_{ref} - \theta| < a$, where a is a small positive number.

2.4 Rest Configurations

During the feedback linearization design we detect another interesting condition due to the inversion of $\delta(\chi)$. If $\delta(\chi)$ looses rank, the control strategy breaks down and the control input grows to infinity. If we avoid the rest configuration, $\eta = 0$, then $\Sigma(\beta_c)$ specifies the current direction of movement and the column vectors N_1 and N_2 are perpendicular to this direction and to each other. To avoid an ill-conditioned $\delta(\chi)$ we must impose a new condition, $|\eta| \geq n$, where n is a small positive number, on our trajectory tracking modes.

To complete the construction, we add additional modes to handle the rest configuration. First assume that the robot is started with $\beta_1 = \beta_2 = \beta_3 = \beta_4$. We

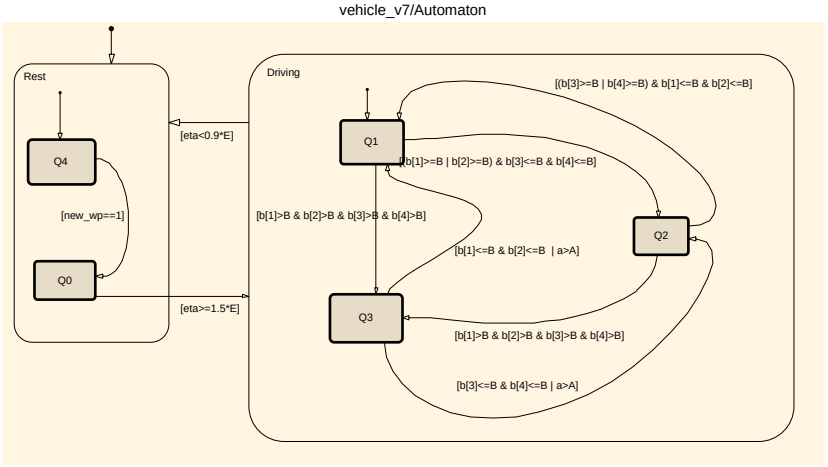
may then utilize the controller defined for the cross driving (q_3) mode, choosing ξ_{ref} as an appropriate point on the straight line originating from the center of mass in the direction defined by β along with

$$\dot{\xi}_{ref} = \begin{bmatrix} \eta_{ref} \\ \zeta_{ref} \end{bmatrix} = \begin{bmatrix} 2n \\ 0 \end{bmatrix} \tag{9}$$

This mode (q_0) allows the robot to start from rest. Finally we add a mode q_4 to handle a stop. Again we assume that the wheels have been oriented by the control laws in mode q_1 or q_2 such that the waypoint lies on the straight line from the center of mass in the direction defined by β. We may then apply the same state transformation as in equation (7) along with the same state feedback, and choosing ξ_{ref} as the target waypoint along with $\dot{\xi}_{ref} = 0$.

3 Hybrid Automaton Supervisor

The trajectory tracking problem for this particular robot may be solved by applying the different control laws, as outlined above for different modes. The conditions for exiting the modes have been defined as well. For each of these modes, we defined special control schemes, and conditions. Given that there are two modes where the robot is at rest, and three modes where the robot is driving, it is straightforward to introduce two super-modes, *Rest* and *Driving*. This gives rise to the hierarchical hybrid automaton implemented using Stateflow as shown in Figure 3.



Here `b[i]` is β_i, `B` is $\frac{\pi}{2} - e_\beta$ `a` is $|\theta_{ref} - \theta|$, and `A`, `E`, are small positive numbers.

Fig. 3. Stateflow representation of Automaton.

The hybrid automaton [10] consists of five discrete states, $\mathcal{Q} = \{q_0, q_1, q_2, q_3, q_4\}$ as defined during the model and controller derivation. The continuous state x

defined by equation (2) or (7) belongs to the state space $\mathcal{X} \subseteq \mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{R}^3$. The corresponding hybrid state space is $\mathcal{H} = \mathcal{Q} \times \mathcal{X}$. The vector fields are defined by

$$f(q,x) = \begin{cases} (A_3 - B_3K_3)x_0 & \text{if } q = q_0, \\ (A_1 - B_1K_1)x_1 & \text{if } q = q_1, \\ (A_2 - B_2K_2)x_2 & \text{if } q = q_2, \\ (A_3 - B_3K_3)x_3 & \text{if } q = q_3, \\ (A_3 - B_3K_3)x_4 & \text{if } q = q_4. \end{cases} \tag{10}$$

Conditions and guards are given in Figure 3 based on the derivations in Section 2. The system including the supervisor was simulated in Simulink and the tracking of an example trajectory is shown in Figure 4. The system is clearly

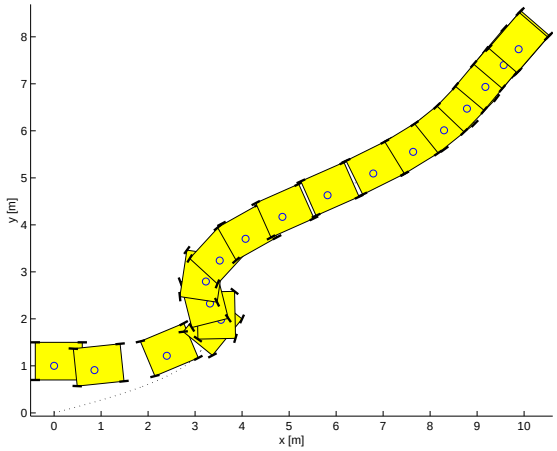


Fig. 4. Tracking a reference trajectory. The vehicle is initially at rest offset from the trajectory by 1 meter in the y-direction.

able to start from a rest configuration, track the trajectory and stop at a rest configuration. In this example the controller starts in the mode q_4, switches to a new waypoint and trajectory information becomes available. As η grows the mode is changed to q_3 and eventually q_1. As the conditions on steering wheels (β_c) are violated the control switches to mode q_2. Finally as the vehicle approaches the end waypoint the mode returns q_4 and stops. As the endpoint is defined by the direction of $\Sigma(\beta_c)$, and the orientation of the wheels are near parallel (and without control) the vehicle reaches the final waypoint with a small error.

Mode changes, tracking errors and wheel positions are given in Figure 5.

4 Stability Analysis

Stability analysis of the developed controller uses the notion of stability of switched systems introduced in [2] as summarized in Appendix B. In case of

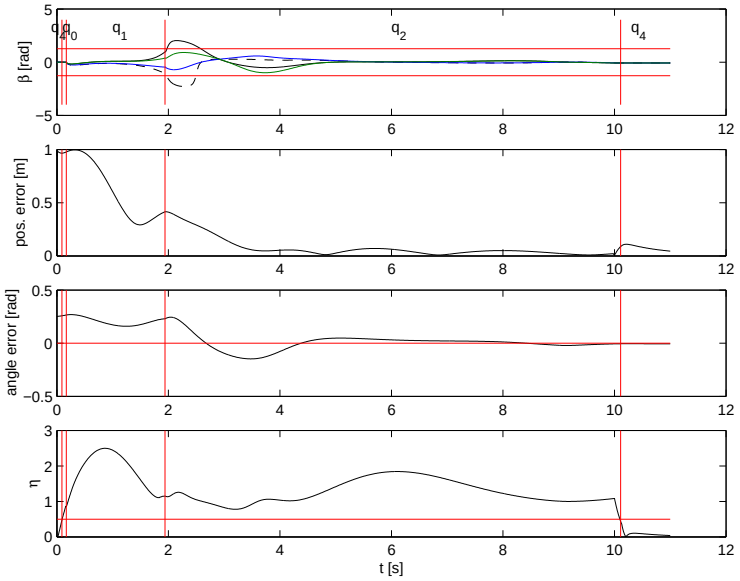


Fig. 5. Wheel orientation (β_i), tracking errors (position, angle) and the input η for the case given in Figure 4. The modes are indicated in the top subplot. The transition to rest is achieved by a step change in the planned trajectory, which result in a disturbance in the position tracking at approximately 10 sec.

the autonomous robot, we have the following Lyapunov functions for the individual control modes.

q_0: Starting from rest with $\beta_1 = \beta_2 = \beta_3 = \beta_4 :\ V_0(x_0) = x_0^{\mathrm{T}} P_3 x_0$.
q_1: Trajectory tracking with $\beta_c = [\beta_1\ \beta_2]^{\mathrm{T}} :\ V_1(x_1) = x_1^{\mathrm{T}} P_1 x_1$
q_2: Trajectory tracking with $\beta_c = [\beta_3\ \beta_4]^{\mathrm{T}} :\ V_2(x_2) = x_2^{\mathrm{T}} P_2 x_2$
q_3: Cross driving with $\beta_1 = \beta_2 = \beta_3 = \beta_4 :\ V_3(x_3) = x_3^{\mathrm{T}} P_3 x_3$
q_4: Stopping with $\beta_1 = \beta_2 = \beta_3 = \beta_4 :\ V_4(x_4) = x_4^{\mathrm{T}} P_3 x_4$.

In each of the cases listed above, $P_j = P_j^{\mathrm{T}} > 0$ is the positive definite solution to the Lyapunov equation $P_j(A_j - B_j K_j) + (A_j - B_j K_j)^{\mathrm{T}} P_j = -I$. Note that for modes q_0 and q_4, the same state feedback K_3 and solution matrix P_3 as in mode q_3 are used. Elementary calculations now yield

$$\dot{V}_j = \dot{x}_j^{\mathrm{T}} P_j x_j + x_j^{\mathrm{T}} P_j \dot{x}_j = -x_j^{\mathrm{T}} x_j.$$

With this in place, we can now attempt to analyze the combination of the Lyapunov functions using a hybrid automaton. We note that since we focus on stability only, we can in each mode abstract from the concrete evolution of the state and replace it by the evolution of the Lyapunov function. For each discrete state $q_j, j = 0, \ldots, 4$ in the automaton in Figure 3 we introduce three consecutive states $q'_{j,k}, k = 0, 1, 2$, which evaluate a constant rate variable Λ_j

that dominates the j'th Lyapunov function. These states are: An *entry* state $q'_{j,0}$, which represents the gain in the Lyapunov function $V_j(x_j)$ at the instant the hybrid control law switches to mode j; an *active* state $q'_{j,1}$, which represents the period where the feedback control $[\nu\ \zeta^{\mathrm{T}}]^{\mathrm{T}} = -K_j x_j$ is active, and where $V_j(x_j)$ is decreasing toward 0; and a state $q'_{j,2}$, where $V_j(x_j)$ has decreased below the entry level. The basic idea is depicted in Figure 4. When the control enters mode

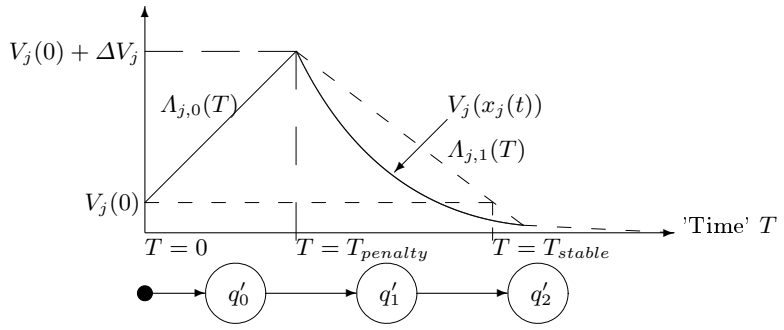


Fig. 6. *Three-state automaton abstracting the Lyapunov function of mode j. The entry, active and stabilized states are indicated below the figure.*

j at time t_j, the Lyapunov function will have gained an amount ΔV_j since the last time it was active. This is modelled abstractly as a constant rate function $\Lambda_{j,0}(T) = T + V_j(t_j), 0 \leq T \leq T_{penalty}$, where the 'time' $T_{penalty}$ is determined as $\Delta V_j = T_{penalty}$. Here, T is an abstract time used for the evaluation of the constant rate function that dominates the j'th Lyapunov function, and which is reset to 0 every time mode q_j is entered. At $T = T_{penalty}$, the system enters the *active* state, in which $\dot{V}_j$ is negative definite. Consequently, $V_j(x_j(t)), t_j \leq t \leq t + T_{stable} - T_{penalty}$ is bounded from above by the function $\Lambda_{j,1}(T) = -\alpha_o T + \Delta V_j + V_j(0), T_{penalty} \leq T \leq T_{stable}, \alpha_o \geq 0$, i.e., another constant rate automaton. T_{stable} is the time where $\Lambda_{j,1}(T) = V_j(0)$; at this point the state changes to $q'_{j,2}$. In order to complete the construction, we must, for each mode change, find the maximal transition penalty ΔV_j which determines $T_{penalty}$, and a suitable α_0.

In general, the transition penalty is the difference between two Lyapunov functions, at the transition point x_j from mode q_i to q_j. In our case, we note that the domains of the Lyapunov functions for the driving modes q_1, q_2, q_3 are identical, cf. equations (2) and (7). Thus the transition penalty is of the form $x_j^{\mathrm{T}}(P_j - P_i)x_j$. Here, we can choose to use the minimum of the P-matrices for all three Lyapunov functions, thus overapproximating the larger ones. This results in a transition penalty of zero, and we can conclude that the system is stable irrespective of the transition pattern while driving. For the transitions to stop mode q_4 and from start mode q_0, the domains are different, cf. equation (9). In the stop transition mode, we can safely ignore the term from the driving Lyapunov

function, thus we get a penalty less than $x_4^{\mathrm{T}} P_3 x_4$, where the magnitude of x_4 is determined by the difference $\xi_{ref} - \xi$ and the velocity $\dot{\xi}$. Assuming that the vehicle is stopped only after it has found the trajectory, the first term is close to zero, and the second term is of the order of n. A similar analysis applies to the start to drive mode transition.

The slope α_0 can be evaluated from the entry value $V_j(0)$ and the growth ΔV_j of the j'th Lyapunov function as follows. The solution of the linearized system during the time the j'th controller is active (i.e., the time where the automaton is in the state $q'_{j,k}$) can be written as $x_j(t) = e^{(A_j - B_j K_j)t} x_j(t_j)$. Hence, in the time interval $t \in [t_j; t + (T_{stable} - T_{penalty})]$,

$$\begin{aligned} V_j(t) &= \left(e^{(A_j - B_j K_j)t} x_j(t_j)\right)^{\mathrm{T}} P_j e^{(A_j - B_j K_j)t} x_j(t_j) \\ &= \| P^{\frac{1}{2}} e^{(A_j - B_j K_j)t} x_j(t_j) \| \end{aligned}$$

where $\| \cdot \|$ denotes the 2-norm of $(\cdot)$. Assuming the pair (A_j, B_j) is controllable, K_j can be chosen such that it is possible to diagonalize the closed-loop matrix, i.e., it is possible to find an invertible matrix S_j such that $(A_j - B_j K_j) = S_j D_j S_j^{-1}$, where D is a diagonal matrix of appropriate dimension containing the eigenvalues of $(A_j - B_j K_j)$ in the main diagonal. Thus we have

$$\begin{aligned} V_j(t)^{\frac{1}{2}} &= \| P_j^{\frac{1}{2}} e^{(A_j - B_j K_j)t} x_j(t_j) \| \\ &= \| P_j^{\frac{1}{2}} S_j e^{D_j t} S_j^{-1} x_j(t_j) \| \\ &\leq \| P_j^{\frac{1}{2}} S_j \| \, \| S_j^{-1} x_j(t_j) \| \, \| e^{D_j t} \| \\ &\leq \| P_j^{\frac{1}{2}} S_j \| \, \| S_j^{-1} x_j(t_j) \| \dim(x_j) e^{\lambda_{max}(D_j)} \end{aligned}$$

where $P^{\frac{1}{2}}$ is the uniquely determined square matrix satisfying $P = P^{\frac{1}{2}} P^{\frac{1}{2}}$. All the terms in front of the exponential are constants that can be evaluated at time t_j, implying that $(T_{stable} - T_{penalty})$ and α_0 can easily be found once $x_j(t_j)$ is known, i.e., at the transition to mode q_j. As indicated on Figure 4, using the upper bound constant rate functions allows for a certain margin to the actual Lyapunov function, which can be considered a form of 'robustness' of the scheme.

When evaluating the stability of the system for a given trajectory, it is clear that only control transitions from the stabilized state are guaranteed safe. To check for unsafe transitions due to the input we propose to add a second automaton constraining the change of the reference input (the trajectory). This automaton has three states, *startup*, *constant_speed* and *stop* which allows us to specify the basic operation of the path planning. The trajectory planner transitions conditions are guarded by the transitions in the automation describing the Lyapunov function. Mode transitions from the two unsafe states (*entry*, *stabilized*) are redirected to an error mode. When a composition with the path planner has error as an unreachable state – the system is safe. This analysis may be done offline, when trajectories are preplanned, or online, in the case of dynamic trajectory planning.

In the concrete case, the driving modes are safe throughout, while stop and start introduce a jump in the error and thus must be separated by some driving period.

5 Conclusion

We have developed a hybrid control scheme for a path-tracking four-wheel steered mobile robot, and shown how it can be analyzed for stability.

The basis for controller development is standard non-slipping and pure rolling conditions, which are used to establish a kinematic-dynamical model. A normal mode path tracking controller is designed according to feedback linearization methods. Other modes are introduced systematically, where the model has singularities. For each such case a transition condition and a new control mode is introduced. Specialized controllers are developed for such modes.

With the control automaton completed, we found for each mode, Lyapunov-like functions, which combine to prove stability. In order to simplify the analysis, we bound the Lyapunov functions by constant rate functions. This allows us to show stability by analyzing a version of the control automaton, where each mode contains a simple three state automaton that evaluates the constant rate functions.

Discussion and Further Work In the systematic approach to deriving modes, we list conditions when the normal mode model fails. Some of these, e.g. Cross Driving, are rather obvious; but others, e.g. the Rest Configuration, are less clear, because they are conditions that make the controlled system ill conditioned. Such problems are usually detected during simulation. Thus a practical rendering of the systematic approach is to use a tool like Stateflow and build the normal mode model. When the simulation has problems, one investigates the conditions and defines corresponding transitions. This is an approach that we believe is widely applicable to design of supervisory or mode switched control systems.

Such an approach is evidently only safe to the extent that it is followed by a rigorous stability analysis. The approach we develop is highly systematic. It ends up with a constant rate hybrid automaton which should allow model checking of its properties. In particular, whether it avoids unsafe transitions when composed with an automaton modelling the reference input. A systematic analysis of this combination is, however, future work.

Another point that must be investigated is, how the wheel reference output is made bumpless during mode transitions. Finally, the idealized non-slip and pure rolling conditions are of course impossible to meet in real-life applications (especially the non-slip condition), and the effect of such perturbations must be studied.

Acknowledgement The authors wish to express their sincere gratitude to the reviewers for their insightful comments. Part of this work was performed at UC Berkeley, and the first author wishes to thank Prof. Shankar Sastry for supporting his visit.

References

1. C. Altafini, A. Speranzon, K.H. Johansson. Hybrid Control of a Truck and Trailer Vehicle, In C. Tomlin, and M. R. Greenstreet, editors, *Hybrid Systems: Computation and Control* LNCS 2289, p. 21ff, Springer-Verlag, 2002.
2. M. S. Branicky. Analyzing and Synthesizing Hybrid Control Systems In G. Rozenberg, and F. Vaandrager, editors, *Lectures on Embedded Systems*, LNCS 1494, pp. 74-113, Springer-Verlag, 1998.
3. A. Balluchi, P. Souères, and A. Bicchi. Hybrid Feedback Control for Path Tracking by a Bounded-Curvature Vehicle In M.D. Di Benedetto, and A.L. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control*, LNCS 2034, pp. 133–146, Springer-Verlag, 2001.
4. G. Bastin, G. Campion. Feedback Control of Nonholonomic Mechanical Systems, *Advances in Robot Control*, 1991
5. B. D'Andrea-Novel, G. Campion, G. Bastin. Modeling and Control of Non Holonomic Wheeled Mobile Robots, in *Proc. of the 1991 IEEE International Conference on Robotics and Automation*, 1130–1135, 1991
6. G. Campion, G. Bastin, B. D'Andrea-Novel. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots, *IEEE Transactions on Robotics and Automation* Vol. 12, 1:47-62, 1996
7. L. Caracciolo, A. de Luca, S. Iannitti. Trajectory Tracking of a Four-Wheel Differentially Driven Mobile Robot, in *Proc. of the 1999 IEEE International Conference on Robotics and Automation*, 2632–2838, 1999
8. J.D. Bendtsen, P. Andersen, T.S. Pedersen. Robust Feedback Linearization-based Control Design for a Wheeled Mobile Robot, in *Proc. of the 6th International Symposium on Advanced Vehicle Control*, 2002
9. H. Goldstein. Classical Mechanics, Addison-Wesley, 2nd edition, 1980
10. T. A. Henzinger. The Theory of Hybrid Automata, In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science* (LICS 1996), pp. 278-292, 1996.
11. I. Kolmanovsky, N. H. McClamroch. Developments in Nonholonomic Control Problems, *IEEE Control Systems Magazine* Vol. 15, 6:20-36, 1995
12. D. Liberzon, A. S. Morse. Basic Problems in Stability and Design of Switched Systems, *IEEE Control Systems Magazine* Vol. 19, 5:59-70, 1999.
13. C. Samson. Feedback Stabilization of a Nonholonomic Car-like Mobile Robot, In *Proceedings of IEEE Conference on Decision and Control*, 1991.
14. B. Thuilot, B. D'Andrea-Novel, A. Micaelli. Modeling and Feedback Control of Mobile Robots Equipped with Several Steering Wheels, *IEEE Transactions on Robotics and Automation* Vol. 12, 2:375-391, 1996.
15. C. Tomlin, S. Sastry. Switching through Singularities, *Systems and Control Letters* Vol. 35, 3:145-154, 1998.
16. G. Walsh, D. Tilbury, S. Sastry, R. Murray, J.P. Laumond. Stabilization of Trajectories for Systems with Nonholonomic Constraints *IEEE Trans. Automatic Control* 39: (1) 216-222, 1994.

A Vehicle Dynamics

Denote the rotation coordinates describing the rotation of the wheels around their horizontal axes by $\phi = [\phi_1\ \phi_2\ \phi_3\ \phi_4]^{\mathrm{T}} \in \mathbb{S}^4$ and the radii of the wheels by

$r = [r_1\ r_2\ r_3\ r_4] \in \mathbb{R}^4$. The motion of the four-wheel driven, four-wheel steered robot is then completely described by the following 11 generalized coordinates:

$$\kappa = \left[X\ Y\ \theta\ \beta^{\mathrm{T}}\ \phi^{\mathrm{T}}\right]^{\mathrm{T}} = \left[\xi^{\mathrm{T}}\ \beta^{\mathrm{T}}\ \phi^{\mathrm{T}}\right]^{\mathrm{T}} \tag{11}$$

and we can write the pure rolling, no slip constraints on the compact matrix form

$$\mathcal{A}(\kappa)\dot{\kappa} = \begin{bmatrix} J_1(\beta)R(\theta)\ 0\ J_2 \\ C_1(\beta)R(\theta)\ 0\ 0 \end{bmatrix} \dot{\kappa} = 0 \tag{12}$$

in which

$$J_1(\beta) = \begin{bmatrix} \cos\beta_1\ \sin\beta_1\ \ell_1\sin(\beta_1-\gamma_1) \\ \cos\beta_2\ \sin\beta_2\ \ell_2\sin(\beta_2-\gamma_2) \\ \cos\beta_3\ \sin\beta_3\ \ell_3\sin(\beta_3-\gamma_3) \\ \cos\beta_4\ \sin\beta_4\ \ell_4\sin(\beta_4-\gamma_4) \end{bmatrix}, \quad J_2 = rI_{4\times 4},$$

$$C_1(\beta) = \begin{bmatrix} -\sin\beta_1\ \cos\beta_1\ \ell_1\cos(\beta_1-\gamma_1) \\ -\sin\beta_2\ \cos\beta_2\ \ell_2\cos(\beta_2-\gamma_2) \\ -\sin\beta_3\ \cos\beta_3\ \ell_3\cos(\beta_3-\gamma_3) \\ -\sin\beta_4\ \cos\beta_4\ \ell_4\cos(\beta_4-\gamma_4) \end{bmatrix}, \text{ and } R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Following the argumentation in [6], the posture velocity $\dot{\xi}$ is constrained to belong to a one-dimensional distribution here parametrized by the orientation angles of two wheels, say, β_1 and β_2. Thus,

$$\dot{\xi} \in \mathrm{span}\{\mathrm{col}\{R(\theta)^{\mathrm{T}}\Sigma(\beta_c)\}\}$$

where $\Sigma(\beta_c) \in \mathbb{R}^3$ is perpendicular to the space spanned by the columns of C_1, i.e., $C_1(\beta)\Sigma(\beta_c) \equiv 0\ \forall\beta$. Σ can be found by combining the expression for $C_1(\beta)$ with equations for the orientation of wheels 3 and 4 to

$$\Sigma = \begin{bmatrix} \ell_1\cos\beta_2\cos(\beta_1-\gamma_1) - \ell_2\cos\beta_1\cos(\beta_2-\gamma_2) \\ \ell_1\sin\beta_2\cos(\beta_1-\gamma_1) - \ell_2\sin\beta_1\cos(\beta_2-\gamma_2) \\ \sin(\beta_1-\beta_2) \end{bmatrix}.$$

The discussion above implies that the robot posture can be manipulated via one velocity input $\eta(t) \in \mathbb{R}$ in the instantaneous direction of $\Sigma(\beta_c)$, that is, $R(\theta)\dot{\xi}(t) = \Sigma(\beta_c)\eta(t)\ \forall t$. Similarly, it is possible to manipulate the orientations of the wheels via an orientation velocity input $\zeta(t) = [\dot{\beta}_1\ \dot{\beta}_2]^{\mathrm{T}} \in \mathbb{R}^2$.

The constrained dynamics of η are handled by applying Lagrange formalism and computed torque techniques as suggested in [5] and [6]. The Lagrange equations are written on the form [9]

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\kappa}_k}\right) - \frac{\partial T}{\partial \kappa_k} = c_k(\kappa)^{\mathrm{T}}\lambda + Q_k$$

in which T is the total kinetic energy of the system and κ_k is the k'th generalized coordinate. On the left-hand side, $c_k(\kappa)$ is the k'th column in the kinematic

constraint matrix $\mathcal{A}(\kappa)$ defined in (12), λ is a vector of Lagrange undetermined coefficients, and Q_k is a generalized force (or torque) acting on the k'th generalized coordinate.

The kinetic energy of the robot is calculated as

$$T = \frac{1}{2}\dot{\kappa}^{\mathrm{T}} \begin{bmatrix} R(\theta)^{\mathrm{T}} M R(\theta) & R(\theta)^{\mathrm{T}} V & 0 \\ V^{\mathrm{T}} R(\theta) & J_\beta & 0 \\ 0 & 0 & J_\phi \end{bmatrix} \dot{\kappa} \tag{13}$$

with appropriate choices of M, J_β and J_ϕ. In the case of the wheeled mobile robot we can derive the following expressions:

$$M = \begin{bmatrix} m_f + 4m_w & 0 & -m_w \sum_{i=1}^{4} \ell_i \sin\gamma_i \\ 0 & m_f + 4m_w & m_w \sum_{i=1}^{4} \ell_i \cos\gamma_i \\ -m_w \sum_{i=1}^{4} \ell_i \sin\gamma_i & m_w \sum_{i=1}^{4} \ell_i \cos\gamma_i & \mathcal{I}_f + m_w \sum_{i=1}^{4} \gamma_i^2 \end{bmatrix}. \tag{14}$$

Here, $\mathcal{I}_f$ is the moment of inertia of the frame around the center of mass, and m_f and m_w are the masses of the robot frame and each wheel, respectively. We note that since the wheels are placed symmetrically around the x_v and y_v axes, the off-diagonal terms should vanish. However, this may not be possible to achieve completely in practice, due to uneven distribution of equipment within the robot.

We denote the moment of inertia of each wheel by $\mathcal{I}_w$ and find

$$J_\beta = \frac{1}{2}\mathcal{I}_w I_{4\times 4} \quad \text{and} \quad J_\phi = \mathcal{I}_w I_{4\times 4} \tag{15}$$

and

$$V = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mathcal{I}_w & \mathcal{I}_w & \mathcal{I}_w & \mathcal{I}_w \end{bmatrix}. \tag{16}$$

The Lagrange undetermined coefficients are then eliminated in order to arrive at the following dynamics:

$$h_1(\beta)\dot{\eta} + \Phi_1(\beta)\zeta\eta = \Sigma^{\mathrm{T}} E \tau_\phi \tag{17}$$

in which $E = J_1^{\mathrm{T}} J_2^{-1} \in \mathbb{R}^{3\times 4}$ and $\tau_\phi \in \mathbb{R}^4$ is a vector of torques applied to drive the wheels. The quadratic function $h_1(\beta)$ is given by

$$h_1(\beta) = \Sigma^{\mathrm{T}}(M + E J_\phi E^{\mathrm{T}})\Sigma > 0 \tag{18}$$

and $\Phi_1(\beta) \in \mathbb{R}$ is given by

$$\Phi_1(\beta) = \Sigma^{\mathrm{T}}(M + E J_\phi E^{\mathrm{T}}) N(\beta_c) \tag{19}$$

and $N(\beta_c) = [N_1\ N_2]$, where

$$N_1 = \begin{bmatrix} -\ell_1 \cos\beta_2 \sin(\beta_1 - \gamma_1) + \ell_2 \sin\beta_1 \cos(\beta_2 - \gamma_2) \\ -\ell_1 \sin\beta_2 \sin(\beta_1 - \gamma_1) - \ell_2 \cos\beta_1 \cos(\beta_2 - \gamma_2) \\ \cos(\beta_1 - \beta_2) \end{bmatrix} \tag{20}$$

$$N_2 = \begin{bmatrix} -\ell_1 \sin\beta_2 \cos(\beta_1 - \gamma_1) + \ell_2 \cos\beta_1 \sin(\beta_2 - \gamma_2) \\ \ell_1 \cos\beta_2 \cos(\beta_1 - \gamma_1) + \ell_2 \sin\beta_1 \sin(\beta_2 - \gamma_2) \\ -\cos(\beta_1 - \beta_2) \end{bmatrix} \tag{21}$$

Equation (17) can be linearized by using a computed torque approach and choosing τ_ϕ appropriately. The torques are simply distributed evenly to each wheel; we observe that

$$\Sigma^{\mathrm{T}} E \tau_\phi = [a_1\ a_2\ a_3\ a_4][\tau_1\ \tau_2\ \tau_3\ \tau_4]^{\mathrm{T}} = L$$

where L is the left-hand side of equation (17). Then we set $\tau_\phi = H\tau_0,\ H \in \mathbb{R}^4$ and choose $H_i = L\mathrm{sign}(a_i)/\sigma$, where σ is the sum of the four entries in the vector $\Sigma^{\mathrm{T}} E$. This distribution policy ensures that the largest torque applied to the individual wheels is as small as possible. By now applying the torque

$$\tau_0 = \frac{1}{\Sigma^{\mathrm{T}} E H}\left(h_1(\beta)\nu + \Phi_1(\beta)\zeta\eta\right), \tag{22}$$

we obtain $\dot{\eta} = \nu$, where ν is a new exogenous input. The result of the extension is the dynamical model given in equation (1).

B Stability of Switched Systems

Consider a dynamic system whose behavior at any given time $t \geq t_0$, where t_0 is an appropriate initial time, is described by one out of several possible individual sets of continuous-time differential equations $\Sigma_0, \Sigma_1, \ldots, \Sigma_\mu$, and let $x_0(t), x_1(t), \ldots, x_\mu(t)$ denote the corresponding state vectors for the individual systems:

$$\Sigma_j: \quad \dot{x}_j = f_j(x_j(t)), \quad j = 0, 1, \ldots, \mu$$

The governing set of differential equations is switched at discrete instances $t_i, i = 0, 1, 2, \ldots$ ordered such that $t_i < t_{i+1} \forall i$. That is, the system behavior is governed by Σ_j in the time interval $t_i < t \leq t_{i+1}$, then by Σ_k in the time interval $t_{i+1} < t \leq t_{i+2}$, and so forth. Assume furthermore that for each Σ_j there exists a Lyapunov function, i.e., a scalar function $V_j(x_j(t))$ satisfying $V_j(0) = 0, V_j(x_j) \geq 0$, and $\dot{V}(x_j) \leq 0$ for $x_j \neq 0$. It is noted that, by the last requirement, V_j is a non-increasing function of time in the interval where Σ_j is active. Hence, it can be deduced that the switched system governed by the sequence of sets of differential equations is stable if it can be shown that

$$V_j(x_j(t_q)) \geq V_j(x_j(t_r))$$

for all $0 \leq j \leq \mu$ and $t_q, t_r \in \{t_i\}$, where $t_q < t_r$ are the last and current switching time where Σ_j became active, respectively.