Hybrid Control Design for a Wheeled Mobile Robot

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

Published in:
Lecture Notes in Computer Science

Publication date:
2003

Document Version
Early version, also known as pre-print

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.

Keywords: hybrid systems, robot, trajectory tracking, stability analysis

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 lead to numerous interesting path tracking control solutions, [12], [11], [4]. Recently [3] and [1] have addressed the problem from a Hybrid systems perspective. 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 are taken of the crop and weed density. This information is processed and combined to a digital map of the field, which eventually will allow the farm manager to manage 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 ia achieved using independent steering and drive motors (8 brushless DC motors in total).

The robot navigates from waypoint to waypoint following spline-type trajectories between the waypoints to minimize damage to the crop. From a control point of view, this is a tracking problem. To solve this problem we first derived a dynamics model of

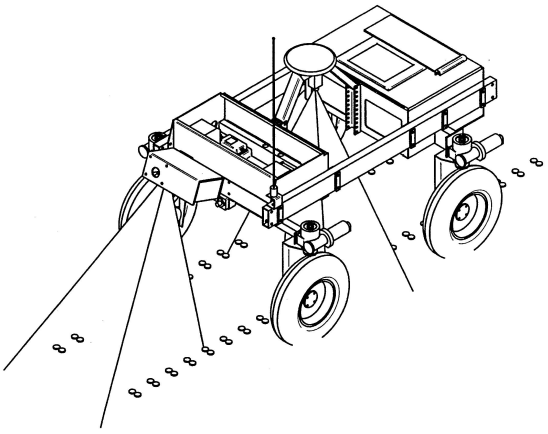

Fig. 1. Schematic model of the experimental platform. The robot is equipped with 8 independent steering and drive motors. Localization is based on fusion of GPS, gyro, magnetometer, and odometer date.

the vehicle subject to pure rolling, no-slip constraints, following the approach taken in [5] and [6].

Next 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. 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. As one of the main contributions of this work, We will therefore apply the generalized Lyapunov stability theory as introduced by Branicky in [2].

Even when given suitable Lyapunov functions for each mode, reasoning about overall stability is still a major effort. We thus abstract the Lyapunov functions to a constant rate function, where the rate is 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.

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 thus add a second automaton constraining the change of the reference input (the trajectory) such that the resulting system remains stable. This automaton defines safe operating conditions, or put another way: Constraints to be satisfied by the trajectory planner. The composed automaton is in a form, such that it can be analyzed using model checking tools.

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.

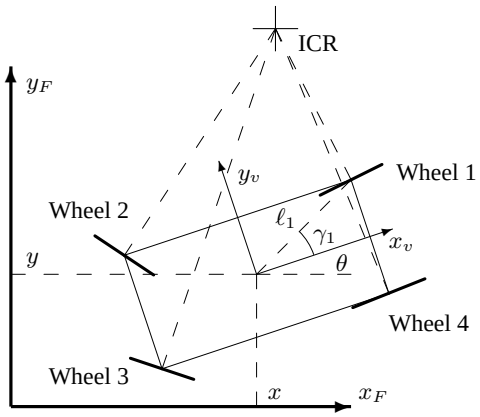

Fig. 2. Definition of the field coordinate system (x_F, y_F), vehicle coordinate system (x_v, y_v), vehicle orientation θ, distance ℓ_1, and direction γ_1 from the center of mass (x, y) to wheel 1. Each wheel plane is perpendicular to the Instantaneous Center of Rotation (ICR).

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 $\xi = [x\ y\ \theta]^{\mathrm{T}} \in \mathbb{R}^2 \times \mathbf{S}^1$.

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 $\beta = [\beta_1\ \beta_2\ \beta_3\ \beta_4]^{\mathrm{T}} \in \mathbf{S}^4$ 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 $\beta_c \in \mathbf{S}^2$ containing the coordinates used to control the ICR location and $\beta_o \in \mathbf{S}^2$ 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 $\pm\pi/2$ i.e. ICR at infinity (and $\dot{\theta} = 0$). 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 $|\beta_1| < (\pi/2 - b) \wedge |\beta_2| < (\pi/2 - b) \wedge |\dot{\theta}| \geq o$.
q_2: Trajectory tracking with $\beta_c = [\beta_3\ \beta_4]^{\mathrm{T}}$. This mode is conditioned on $|\beta_3| < (\pi/2 - b) \wedge |\beta_4| < (\pi/2 - b) \wedge |\dot{\theta}| \geq o$.
q_3: Cross Driving with $\beta_1 = \beta_2 = \beta_3 = \beta_4$. This mode is conditioned on $|\dot{\theta}| < o$.

where b and o are small positive numbers. The two first modes cover the situations where the ICR is governed by wheels 1 and 2 and by wheel 3 and 4, respectively, while the latter covers the situation where the ICR is 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.

As presented here, the problem is specific to the proposed representation of the ICR, but in general, no set of two variables is able to describe all wheel configurations without singularities [13]. That is, the problem of such singular configurations is not due to the representation used, but is a general problem for this type of robotic systems.

2.1 Vehicle Model

Following the argumentation in Appendix A, robot posture can be manipulated via one velocity input $\eta(t) \in \mathbb{R}$ in the instantaneous direction of $\Sigma(\beta_c) \in \mathbb{R}^3$ which is constructed to meet the pure-roll constraint. 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

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. 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 that 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

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

where $N(\beta_c) = [N_1 \; N_2]$ is specified in equations (19) and (20). If we then apply the control law

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

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 situation 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 Parallel Wheels Control

Due to the conditions imposed on the trajectory tracking controllers, an additional control scheme must be derived that is able to control the vehicle when all four wheels are parallel, which forces $\dot{\theta}$ to 0. 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

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

where T is an appropriate invertible matrix, then yields the dynamics

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

which can be controlled by applying the feedback $\nu = -K_3 x_3$.

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, q_1 and q_2.

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 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

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

This mode would allow the robot to start from rest and adds a mode (q_0). Finally we add a mode q_4 to handle the stop situation. 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 (7) along with the same state feedback, and choosing ξ_{ref} as the target waypoint along with $\dot{\xi}_{ref} = 0$. As a result $|\eta| \geq n$ is added to the mode conditions for q_1, q_2, and two new modes q_0 and q_4 are added to handle the start from rest and the stop situations respectively.

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.

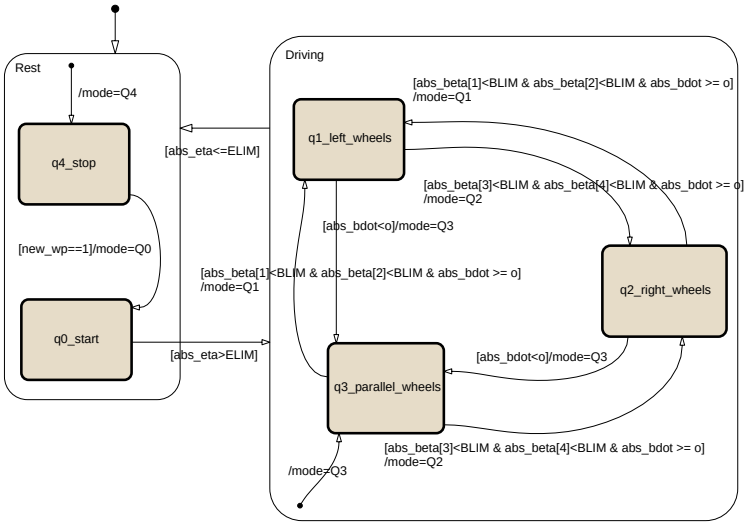

Fig. 3. Stateflow representation of Automaton.

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

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

$$f(q,x) = \begin{cases} (A_3 - B_3K_3)x_0 & \text{if } q = q_0, \\ (A_1 - B_1K_1)x_1 & \text{if } q = q_1, \\ (A_2 - B_2K_2)x_2 & \text{if } q = q_2, \\ (A_3 - B_3K_3)x_3 & \text{if } q = q_3, \\ (A_3 - B_3K_3)x_4 & \text{if } q = q_4. \end{cases} \tag{9}$$

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 able to start from an 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 where it remains until the vehicle returns to q_4 and stops. Mode changes, tracking errors and wheel positions are given in Figure 5.

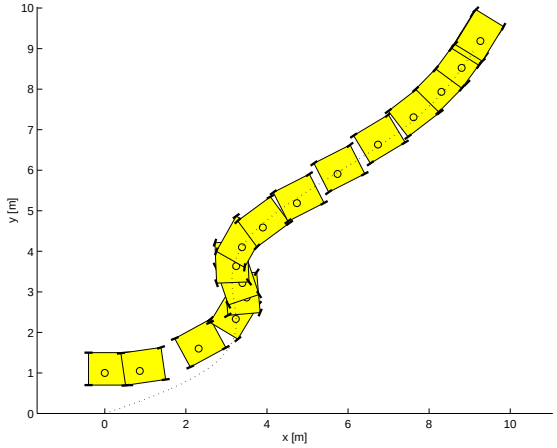

Fig. 4. Tracking a reference trajectory. The vehicle is initially at rest offset from the trajectory by 1 meter in the y-direction.

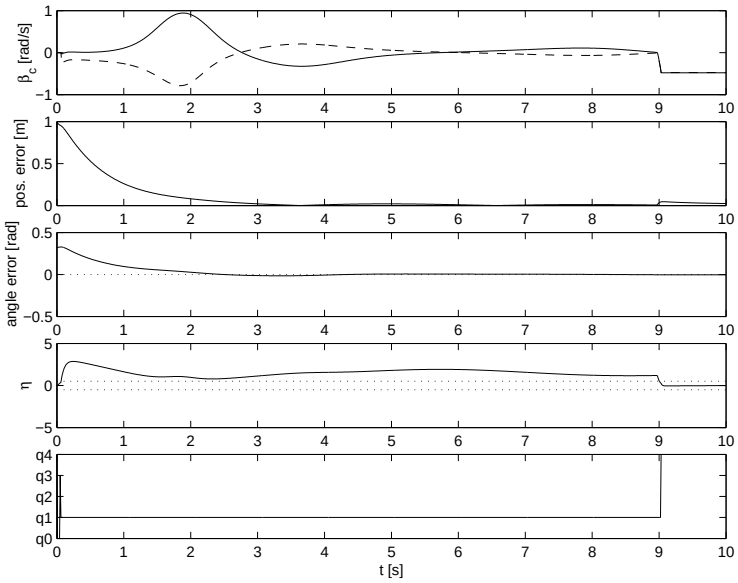

Fig. 5. Wheel orientation (β_1, β_2), tracking errors and modes for the case given in Figure 4.

4 Stability Analysis

In order to ensure that the hybrid control law specified in the previous Section does not cause the robot to become unstable, we will carry out a stability analysis of the closed loop system. Here, we will use the notion of stability of switched systems introduced in [2], which is summarized in Appendix B.

In case of 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 evaluate the Lyapunov functions for each mode using a hybrid automaton.

We first observe, 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. As mentioned in Section 1, we furthermore divide each discrete state $q_j, j = 0, \ldots, 4$ in the automaton in Figure 3 into three consecutive states that evaluate a constant rate variable which dominates the Lyapunov function. These states are: An *entry* state, which represents the gain in the Lyapunov function $V_j(x_j)$ at the instant the hybrid control law switches to mode j; an *operation* state, which represents the period where the feedback control $\nu = -K_j x_j$ is active, and where $V_j(x_j)$ is decreasing toward 0; and a *final* state, where $V_j(x_j)$ is small enough to indicate that the system has become stable (i.e., the tracking error is sufficiently small to ignore). The basic idea is depicted in Figure 4. When the control enters mode 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 $V_j(T) = \alpha_p T + V_j(0), 0 \leq T \leq T_{penalty}, \alpha_p \geq 0$, where the time $T_{penalty}$ is determined such that $\Delta V_j = \alpha_p T_{penalty}$. Subsequently, the *operation* state becomes active, in which $\dot{V}_j$ is negative definite. Consequently, $V_j(T)$ is bounded from above by the function $V'(T) = -\alpha_o T + \Delta V_j + V_j(0), T_{penalty} \leq T \leq T_{stable}, \alpha_o \geq 0$, i.e., another constant rate automaton. Finally, when T_{stable} is reached, the system is considered stable and stays in this state with the trivial constant rate function $V_{stable}(T) = V_{stable}(0)$ until another mode change is enforced, e.g., because the target waypoint is reached. In Stateflow the hierarchy is achieved using subcharted states, which allows us to divide the states in Figure 3 into three new substates. It is clear that only the transition from the final state is guaranteed safe. To avoid

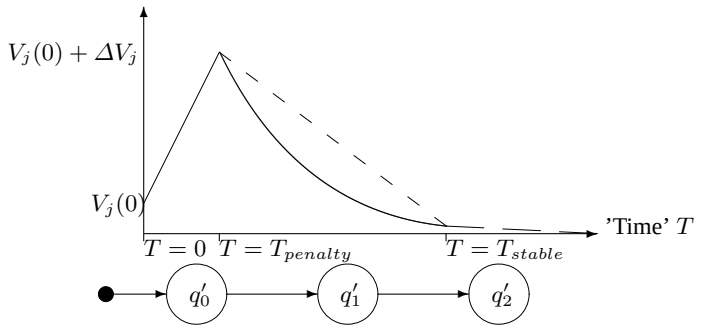

Fig. 6. *Abstract three-state automaton simulating the Lyapunov function of mode j. The entry, operation and final states are indicated below the figure.*

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 ut 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. If all mode transitions from the two unsafe states (*entry*, *operation*) are redirected to an error mode, and the parallel composition with the path planner has error as an unreachable state – the system is safe.

5 Conclusion

We have developed a hybrid control scheme for a path-tracking four-wheel driven, four-wheel steered autonomous robot, and shown how it is 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. This is used to find a partial linearization of the dynamics using computed torques and local servo loops around the steering motors. Then, a normal mode path tracking controller is designed according to the feedback linearization method. 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, Lyaponov-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 when developing the model; but others, e.g. the Rest Configuration, are less clear, because they are not outright singularities, but more conditions that make the model 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. An approach that we believe is widely applicable to design of supervisory or mode switched control systems.

Such a divide and conquer approach is evidently only safe to the extent that it is followed by a rigorous stability analysis.The approach which we develop is very 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.

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. C. Samson. Feedback Stabilization of a Nonholonomic Car-like Mobile Robot, In *Proceedings of IEEE Conference on Decision and Control*, 1991.
12. 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
13. 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

A Vehicle Dynamics

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

$$q = \left[x \ y \ \theta \ \beta^{\mathrm{T}} \ \phi^{\mathrm{T}}\right]^{\mathrm{T}} = \left[\xi^{\mathrm{T}} \ \beta^{\mathrm{T}} \ \phi^{\mathrm{T}}\right]^{\mathrm{T}} \tag{10}$$

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

$$\mathcal{A}(q)\dot{q} = \begin{bmatrix} J_1(\beta)R(\theta) & 0 & J_2 \\ C_1(\beta)R(\theta) & 0 & 0 \end{bmatrix} \dot{q} = 0 \tag{11}$$

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,

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

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 for non-holonomic systems are written on the form [9]

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

in which T is the total kinetic energy of the system and q_k is the k'th generalized coordinate. On the left-hand side, $c_k(q)$ is the k'th column in the kinematic constraint matrix $\mathcal{A}(q)$ defined in (11), λ is a vector of so-called *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

$$T = \frac{1}{2}\dot{q}^{\mathrm{T}} \begin{bmatrix} R(\theta)^{\mathrm{T}} M R(\theta) & R(\theta)^{\mathrm{T}} V & 0 \\ V^{\mathrm{T}} R(\theta) & J_\beta & 0 \\ 0 & 0 & J_\phi \end{bmatrix} \dot{q} \tag{12}$$

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

$$M = \begin{bmatrix} m_f + 4m_w & 0 & -m_w \sum_{i=1}^{4} \ell_i \sin\gamma_i \\ 0 & m_f + 4m_w & m_w \sum_{i=1}^{4} \ell_i \cos\gamma_i \\ -m_w \sum_{i=1}^{4} \ell_i \sin\gamma_i & m_w \sum_{i=1}^{4} \ell_i \cos\gamma_i & \mathcal{I}_f + m_w \sum_{i=1}^{4} \gamma_i^2 \end{bmatrix}. \tag{13}$$

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.

Turning to the wheels, we denote the moment of inertia of each wheel by $\mathcal{I}_w$ and find

$$J_\beta = \frac{1}{2}\mathcal{I}_w I_{4\times 4} \quad \text{and} \quad J_\phi = \mathcal{I}_w I_{4\times 4} \tag{14}$$

and

$$V = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \mathcal{I}_w & \mathcal{I}_w & \mathcal{I}_w & \mathcal{I}_w \end{bmatrix}. \tag{15}$$

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

$$h_1(\beta)\dot{\eta} + \Phi_1(\beta)\zeta\eta = \Sigma^{\mathrm{T}} E \tau_\phi \tag{16}$$

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

$$h_1(\beta) = \Sigma^{\mathrm{T}}(M + E J_\phi E^{\mathrm{T}})\Sigma > 0 \tag{17}$$

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

$$\Phi_1(\beta) = \Sigma^{\mathrm{T}}(M + E J_\phi E^{\mathrm{T}}) N(\beta_c) \tag{18}$$

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

$$N_1 = \begin{bmatrix} -\ell_1 \cos\beta_2 \sin(\beta_1 - \gamma_1) + \ell_2 \sin\beta_1 \cos(\beta_2 - \gamma_2) \\ -\ell_1 \sin\beta_2 \sin(\beta_1 - \gamma_1) - \ell_2 \cos\beta_1 \cos(\beta_2 - \gamma_2) \\ \cos(\beta_1 - \beta_2) \end{bmatrix} \tag{19}$$

$$N_2 = \begin{bmatrix} -\ell_1 \sin\beta_2 \cos(\beta_1 - \gamma_1) + \ell_2 \cos\beta_1 \sin(\beta_2 - \gamma_2) \\ \ell_1 \cos\beta_2 \cos(\beta_1 - \gamma_1) + \ell_2 \sin\beta_1 \sin(\beta_2 - \gamma_2) \\ -\cos(\beta_1 - \beta_2) \end{bmatrix} \tag{20}$$

Equation (16) 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

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

where L is the left-hand side of (16). Then we set $\tau_\phi = H\tau_0,\ H \in \mathbb{R}^4$ and choose $H_i = L\text{sign}(a_i)/\sigma$, 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.

Hence, by applying the torque

$$\tau_0 = \frac{1}{\Sigma^{\mathrm{T}} E H} \left(h_1(\beta)\nu + \Phi_1(\beta)\zeta\eta \right), \tag{21}$$

we obtain

$$\dot{\eta} = \nu$$

where ν is a new exogenous input. The result of the extension and partial linearization 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.