

Robust Hybrid Control for Autonomous Vehicle Motion Planning

Emilio Frazzoli¹

Munther A. Dahleh²

Eric Feron³

Abstract

The operation of an autonomous vehicle in an unknown, dynamic environment is a very complex problem, especially when the vehicle is required to use its full maneuvering capabilities, and to react in real time to changes in their operational environment. A possible approach to reduce the computational complexity of the motion planning problem for a nonlinear, high dimensional system, is based on a quantization of the system dynamics, leading to a control architecture based on a hybrid automaton, the states of which represent feasible trajectory primitives for the vehicle. This paper focuses on the feasibility of this approach, in the presence of disturbances and uncertainties in the plant and/or in the environment: the structure of a Robust Hybrid Automaton is defined and its properties are analyzed. In particular, we address the issues of well-posedness, consistency and reachability. For the case of autonomous vehicles, we provide sufficient conditions to guarantee reachability of the automaton.

1 Introduction

The operation of an autonomous vehicle in an unknown, dynamic and potentially hostile environment is a very complex problem, especially when the autonomous vehicle is required to use its full maneuvering capabilities, and to react in real time to changes in the operational environment. A common way of dealing with highly complex systems is via a hierarchical decomposition of the activities to be performed by the autonomous vehicles, and consequently the introduction of a hierarchy of control and decision layers.

While the control layers that interact directly with the plant operate on a continuous state space, higher control layers are most often designed as logical decision-making agents, operating on a discrete state space. Systems that include both discrete and continuous dynamics are usually referred to in the literature as hybrid systems. Hybrid control systems have been the object of a very intense and productive research effort in the recent years, which has resulted in the definition of very general frameworks (for example, see [1, 2] and references therein). General hybrid systems, derived from arbitrary hierarchical decompositions, however, can be extremely hard to analyze and verify, and only limited results can be obtained [3, 4]. On the other hand, it

may be convenient to design the hybrid system in such a way that it offers safety and performance guarantees by construction, at least in an idealized situation. This reflects the designer's insight into the nominal behavior of the system [5, 6]. Consequently, the analysis and verification problems of the hybrid system are translated to a robustness analysis problem, which can be solved using the relevant tools from systems and control theory [7].

In the present framework, the hybrid controller is responsible for both the generation and the execution of a feasible trajectory, or flight plan, satisfying the mission requirements while optimizing some performance criterion. Our main objective is the definition of a robust control architecture, and algorithms, to address the motion planning problem for an autonomous vehicle, exploiting to the maximum extent the vehicle's dynamics. Even though our main focus is control of autonomous vehicles, the concepts that we will introduce can be used profitably for control of a large class of nonlinear systems: the presentation of the control architecture will be kept at a general level. In this paper, due to space constraints, we will present only the definition of the automaton and the analysis of some of its fundamental properties. Algorithms for motion planning based on the framework presented in this paper are available in [8, 9].

2 Hybrid Automaton

A possible approach to reduce the computational complexity of the motion planning problem for a nonlinear, high dimensional system, is based on a quantization of the system dynamics, in the sense that we restrict the feasible nominal system trajectories to the family of time-parametrized curves that can be obtained by the interconnection of appropriately defined primitives. These primitives will then constitute a *maneuver library* from which the nominal trajectory will be constructed. Instead of solving an optimal control problem over a high-dimensional, continuous space, we will solve a mixed integer programming problem, over a much smaller space. In addition to the reduction in computational complexity, one of the objectives of this approach is the ability to provide a mathematical foundation for generating a provably stable hierarchical system, and for developing the tools to analyze robustness in the presence of uncertainty in the process as well as in the environment.

We want to characterize trajectory primitives in order to: (i) capture the relevant characteristics of the vehicle dynamics; (ii) allow for the creation of complex behaviors from the interconnection of primitives (we want to obtain "good" approximations to optimal solutions) (iii) determine the *minimal* set of key parameters identifying the state of the system: this is even more important for extension to multi-vehicle operations, or more complex systems. In the following we will define the class of systems we want to control, and, ac-

¹Research Assistant, Laboratory for Information and Decision Systems, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, email: frazzoli@mit.edu

²Professor, Laboratory for Information and Decision Systems, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, email: dahleh@lids.mit.edu

³Associate Professor, Laboratory for Information and Decision Systems, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, email: feron@mit.edu

cordingly, we will give a characterization of the trajectory primitives we will consider. This will be used to present the Robust Hybrid Automaton structure.

2.1 System dynamics

We will consider a time-invariant nonlinear system the dynamics of which are described by the differential equation:

$$\frac{dx}{dt} = f(x, u, w) \quad (1)$$

where $x \in X$ is the state, belonging to an n -dimensional manifold X , and u and w represent respectively the control and disturbance input signals, taking values in the sets $U \subseteq \mathbb{R}^m$, $W \subseteq \mathbb{R}^l$; both signals are assumed to be bounded with respect to some norm p , i.e. $u(\cdot) \in \mathcal{U} \subseteq \mathcal{L}_p^m$, $w(\cdot) \in \mathcal{W} \subseteq \mathcal{L}_p^l$. Finally, the function $f : X \times U \times W \rightarrow TX$ is assumed to be locally Lipschitz in its arguments. Let us concentrate first on the nominal system:

$$\frac{dx}{dt} = f_0(x, u) = f(x, u, 0) \quad (2)$$

that is the system obtained from (1) when the disturbance input is identically zero. We can define equilibrium points for the nominal system (2) as the points (\bar{x}, \bar{u}) for which $f_0(\bar{x}, \bar{u}) = 0$. In general, we will be more interested in systems that have symmetries, for which we can define the notion of relative equilibrium.

2.2 Symmetries and relative equilibria

Roughly speaking, a symmetry on the system (2) is a group action on the state that leaves the dynamics invariant. A simple definition that is enough for the purpose of this paper is given in the following: for further details and a precise definition in a Hamiltonian mechanics framework, see [10, 11]. Assume that the state space has the structure of a Lie group, and that the manifold X can be written *at least locally* as $X = Y \times Z$, so that the state of the system can be written as $x = (y, z) \in Y \times Z$. Consider the map $\Psi_h : X \rightarrow X$, parametrized by an element $h \in Y$, such that the state $x = (y, z)$ is transformed into $\Psi_h(y, z) = (hy, z)$. Given the initial conditions x_0 at $t = 0$, let the time-parametrized curve $\phi_u(\cdot, x_0)$ be the resulting trajectory of the nominal system (2) when the control input is any given signal u . We will indicate with \mathfrak{y} the Lie algebra associated with Y . At this point we can define the following:

Definition 1 (Symmetry Group) *If for all initial conditions $x_0 \in X$, $y \in Y$, $t \in \mathbb{R}$ we have that $\Psi_y \circ \phi_u(t, x_0) = \phi_u(t, \Psi_y(x_0))$, then Y is a **symmetry group** for the system (2).*

Definition 2 (Relative equilibria) *Assume that Y is a symmetry group for the system (2), and that it is possible to find constants $\bar{z} \in Z$, $\bar{u} \in U$ and $\hat{\eta} \in \mathfrak{y}$ such that: $\phi_{\bar{u}}(t, (id_Y, \bar{z})) = (\exp(\hat{\eta}t), \bar{z})$. The resulting class of trajectories of the system $\phi_{\bar{u}}(t, (y_0, \bar{z}))$ is called a **relative equilibrium**, or **trim trajectory**.*

The collection of all possible trim trajectories defines a manifold $\mathcal{S} \subset X \times U$, denoted as the **trim surface**. It is also clear that relative equilibria include trivially all the equilibrium points of the system. A consequence of the symmetry of the dynamics is that we can treat all trajectory primitives as equivalence classes, and choose a prototype for each primitive, starting at a reference position on the symmetry group. Without loss of generality, we can define all trajectory primitives as starting at the identity element id_Y of the symmetry group Y .

Example: A very simple example of a system with symmetries is a system with integrators:

$$\begin{aligned} \dot{y} &= z \\ \dot{z} &= f_z(z, u) \end{aligned} \quad (3)$$

where $(y, z) \in \mathbb{R}^n$, and the group operation we are interested in is the usual vector addition. It is evident that a translation $\psi_{\Delta y} : (y, z) \mapsto (y + \Delta y, z)$ does not change the dynamics of the system. Relative equilibria are all trajectories for which we can find \bar{z} and \bar{u} such that $\dot{z} = f_z(\bar{z}, \bar{u}) = 0$. The momentum can be identified in the constant vector \bar{z} , and the dynamics of y on a trim trajectory are given by $y(t) = y(t_0) + \bar{z}(t - t_0)$.

A more interesting kind of symmetry, that is invariance to translation and rotation about a vertical axis, is exhibited by a large class of mechanical systems. This class includes, in particular, most human built vehicles. In the following we will focus on the definition of a control architecture for autonomous vehicles, however the concepts and methods are valid and can be used for systems with multiple equilibria, and possibly with relative equilibria.

2.3 Autonomous vehicle dynamics

The dynamics of a large class of small autonomous vehicles can be adequately described by the rigid body equations [12]. The configuration of the vehicle will be described by an element g of the Special Euclidean group in the three-dimensional space, usually denoted by $SE(3)$. Using homogeneous coordinates, a matrix representation of $g \in SE(3)$ is

the following: $g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$ where $R \in SO(3)$ is a rotation

matrix and $p \in \mathbb{R}^3$ is a translation vector. The kinematics of the rigid body are determined by $\dot{g} = g\hat{\xi}$ where $\hat{\xi}$, denoted as *twist*, is an element of the Lie algebra $\mathfrak{se}(3)$ associated with $SE(3)$. A matrix representation of an element $\hat{\xi} \in \mathfrak{se}(3)$ is $\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}$ where ω and v are respectively the angular and translational velocities in body axes, and the skew matrix $\hat{\omega}$ is the unique matrix such that $\hat{\omega}u = \omega \times u$, for all $u \in \mathbb{R}^3$. The full state of the vehicle as a rigid body will then be represented by $x = (g, \hat{\xi})$, with $X = SE(3) \times \mathfrak{se}(3)$. The dynamics equations, in matrix notation, will be given by:

$$J_b \dot{\omega} = -\omega \times J_b \omega + M_b(g, \hat{\xi}, u, w) \quad (4)$$

$$m \dot{v} = -\omega \times mv + F_b(g, \hat{\xi}, u, w) \quad (5)$$

where J_b and m are the vehicle's inertia tensor and mass, and M_b and F_b represent the torques and forces in body axes, which are in general a function of the vehicle state $x = (g, \hat{\xi})$, of the control inputs u , and of the disturbances w . Note that in the above we have no assumption on the characteristics of the forces acting on the vehicle (i.e. we do not require potential forces).

The dynamics of a vehicle, including cars, aircraft, ships, etc., under fairly reasonable assumptions, (such as homogeneous and isotropic atmosphere, and constant gravity acceleration for an aircraft) are invariant to translation and rotation about a vertical axis, i.e. an axis parallel to the local gravitational acceleration. If this is the case, the subgroup $H \subset SE(3)$, composed of translation and rotations about the vertical axis is a symmetry group for the vehicle dynamics. An element $h \in H = \mathbb{R}^3 \times S^1$ is completely described by the translation vector $p \in \mathbb{R}^3$ and the heading angle $\psi \in [0; 2\pi)$.

2.4 Equilibrium points and trim trajectories

The simplest possible motion primitive is trivially represented by equilibrium points. In a system with multiple equilibrium points each equilibrium point can be chosen as a trajectory primitive. A closely related and more interesting class of primitives is given by trim trajectories. In an autonomous vehicle setting, these can be seen as those trajectories along which the velocities in body axes (the twist) and the control inputs are constant.

From the above discussion of the symmetry properties, it follows that all trim trajectories will be the composition of a constant rotation \bar{g} and a screw motion $h(t) \in H$, given by the exponential of an element $\hat{\eta}$ of the Lie sub-algebra $\mathfrak{h} \subset \mathfrak{se}(3)$. This screw motion corresponds in the physical space to a helix traversed at a constant speed and sideslip angle. For aerial vehicles, such helices are usually described by the parameter vector $\bar{T} := [V, \gamma, \dot{\psi}, \beta]$, where V is the magnitude of the velocity vector, γ is the flight path angle, $\dot{\psi}$ is the turning rate and finally β is the sideslip angle. To make the above clearer, we will give some details on the matrix representation. The elements of H can be represented in matrix notation as:

$$h = \begin{bmatrix} \cos \psi & -\sin \psi & 0 & x \\ \sin \psi & \cos \psi & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

The identity element id_H will be represented by the identity matrix I_4 . Trim trajectories are described by the corresponding element $\hat{\eta}$ in the Lie algebra \mathfrak{h} , which can be represented as:

$$\hat{\eta} = \begin{bmatrix} 0 & -\dot{\psi} & 0 & V \cos \beta \cos \gamma \\ \dot{\psi} & 0 & 0 & V \sin \beta \cos \gamma \\ 0 & 0 & 0 & V \sin \gamma \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (7)$$

Following a trim trajectory for a time interval Δt results in a displacement: $h_{trim}(\Delta t) = \exp(\hat{\eta}\Delta t)$

The first step in the design of our control architecture is the selection of a number of trim trajectories. The selection of trim trajectories can be carried out by gridding the set of attainable values of \bar{T} ; this set is compact, and can be identified with the flight envelope in the case of aerial vehicles.

This class of trajectory primitives has been used widely to construct switching control systems, in which point stabilization is achieved by switching through a sequence of controllers progressively taking the system closer to the desired equilibrium [13, 14]. The ideas of gain scheduling and of Linear Parameter Varying (LPV) system control can also be brought into this class [15], as well as other integrated guidance and control systems for UAV applications [16]. However, such a design choice generally results in relatively poor performance, and in “slow” transitions, as the system is required to stay in some sense close to the trim surface. Moreover, the absence of any information on the transient behavior can lead to undesirable effects, such as limit cycles.

2.5 Maneuvers

For more aggressive maneuvering it is deemed necessary to better characterize trajectories that move “far” from the trim surface. Even though ours can be seen as a reductive

definition of what is considered a maneuver in the common language, it leads to significant simplifications in the design of the control architecture.

Definition 3 (Maneuver) *In this paper, a maneuver is defined as a (finite time) transition between two trim trajectories, for the nominal system (2).*

Note that the transition can also be from and to the same trim trajectory (e.g. in the case of aircraft acrobatic maneuvers like loops and barrel rolls can be considered as transitions from and back to straight and level flight, and in the case of cars a lane change is a transition from and back to forward motion). The execution of the maneuver results in a total configuration change g_m , that is $g(t_m) = g_m g(0)$ if t_m indicates the maneuver duration. For reasons that will be made clear in the following, we are more interested in the evolution on the subgroup H , and from the properties of trim trajectories we have that $g_m = (h_m, \bar{g}_{end} \bar{g}_{start}^{-1})$. We will not discuss the details of how to generate the nominal state and control trajectories describing maneuvers: several methods can be used depending on the application at hand, the desired performance, and the available computing, simulation and experimental resources. Among these methods we can mention actual tests or simulations with human pilots, off-line solutions to optimal control problems, or real-time trajectory generation.

3 Extensions for robustness

In the preceding sections we defined the trajectory primitives as feasible, possibly but not necessarily optimal (for some cost) trajectories for the *nominal* system, that is when the disturbance signal w is identically zero. In real applications, we will not be able to achieve exactly the reference trajectories defined by the primitive library, because of deviations in the initial conditions, noise in the measurements, unmodeled dynamics and modeling errors, and exogenous inputs. We will therefore need to examine the behavior of the system at non-nominal conditions, and make sure that the resulting system trajectories are in some sense “close” to the trajectories of the nominal system. In general, this requires some form of feedback control, complementing the feed-forward open-loop reference input trajectory stored along with the reference state trajectory.

The reference trajectory will be completely determined, in terms of nominal state and control histories, by the primitive being executed, its inception time, and initial H configuration. A feedback control policy will then be a function $v : \mathbb{R} \times X \times H \rightarrow U$, designed to track (or regulate to) the nominal trajectory. Once a feedback control law is associated with the system (1) it is transformed into the closed-loop form: $\dot{x} = f(x, v(t - t_0, x, h_0), w)$, in which the only exogenous input is the disturbance input w .

The robustness characteristics of equilibrium points and of trim trajectories can be expressed in terms of invariant sets.

Definition 4 (Invariant set) *A set $M \subseteq X$ is said to be a (right)-invariant set if for all $x_0 \in M$, $w \in \mathcal{W}$, and $t > t_0$: $\Psi_{\exp(-\hat{\eta}(t-t_0))} \circ \phi_{v,w}(t - t_0, x_0) \in M$ where $\phi_{v,w}(\cdot, x_0)$ describes the trajectory of the system under the action of the control policy v and disturbance w , and with initial conditions $x(0) = x_0$.*

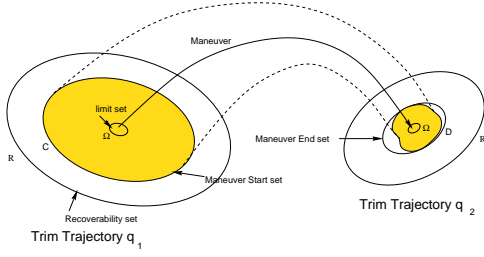


Figure 1: Invariant set definitions

Invariant sets are “tubes” centered on trim trajectories. However, in the following we will refer to the section at $h = \text{id}_H$.

Definition 5 (Limit set) We will call the limit set of a trim trajectory q the smallest invariant set $\bar{\Omega}_q$ associated with that trajectory.

Definition 6 (Recoverability set) We will call the recoverability set $\bar{\mathcal{R}}_q$ of a trim trajectory the largest set for which there exists a finite time \bar{t} such that for all initial conditions $x_0 \in \bar{\mathcal{R}}_q$, and for all disturbance signals $w \in \mathcal{W}$, the system enters the limit set, that is if: $\Psi_{\exp(-\bar{\eta}(t-t_0))} \circ \phi_{v,w}(t, x_0, t_0) \in \bar{\Omega}_q, \forall t > t_0 + \bar{t}$

In general, the exact determination of the sets $\bar{\mathcal{R}}_q$ and $\bar{\Omega}_q$ presents a very difficult challenge. However, often it is possible to compute conservative approximations, in the sense that we can compute a set $\mathcal{R}_q \subseteq \bar{\mathcal{R}}_q$ such that for all initial conditions in \mathcal{R}_q the system trajectory will enter a set $\Omega_q \supseteq \bar{\Omega}_q$ after a finite time, and stay in Ω_q thereafter. It is obviously of interest to design a control law in such a way to have a large \mathcal{R}_q , and a small Ω_q . In the case in which the control law provides global stability, \mathcal{R}_q will coincide with X , and in the case in which we have asymptotic stability, Ω_q will collapse to the trajectory itself.

Similar concepts, close in nature to Lyapunov stability theory, cannot be defined for the maneuvers, since these are by definition objects with a finite time horizon. Instead we will use a concept more closely related to Poincaré maps.

Definition 7 (Image of a set) We will define the image of a set C under the maneuver q the smallest set $D \in X$ such that for all $x_0 \in C$, and for all disturbance signals $w \in \mathcal{W}$ (supported on $[t_0, t_0 + \Delta t_{m_q}]$), we have that: $\phi_{v,w}(t_0 + \Delta t_{m_q}, x_0, t_0) \in D$

The objective of the control law in this case is to make the ending set D as small as possible for a given starting set C . Notice that we are not directly interested in the transient behavior of the system during the execution of the maneuver, as long as we can ensure that at the end of the maneuver the state enters the set D .

4 Robust Hybrid Automaton definition

We are now ready to discuss the details of the control architecture. It should be clear by now that the controlled system will include both continuous and discrete dynamics, thus belonging to the realm of hybrid control. In the following we will present the definition of a hybrid system that is based on the general model in [1]. The Robust Hybrid Automaton we are concerned with is described by the n-tuple: $\text{RHA} = \{Q, H, T, X, f, \eta, v, \mathbf{C}, \mathbf{D}, \mathcal{R}, \Omega, \mathbf{A}, V\}$, where:

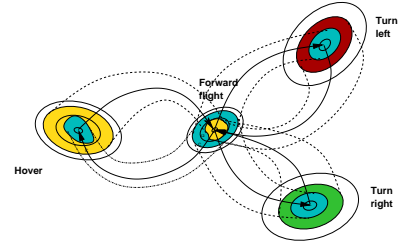


Figure 2: Robust Hybrid Automaton (simplified)

- $Q := Q_M \cup Q_T$ is the discrete set of the index state. The values of Q identify the trajectory primitive being executed; as such, Q is assumed to be a finite set. The subscript T and M indicate, respectively, trim trajectories and maneuvers;
- H is the symmetry group, identifying the position of the current trajectory primitive;
- $T = \mathbb{R}$: we augment the reference state by a clock, or timer state;
- X is the state space of the continuous system;
- f is the Lipschitz function describing the continuous system dynamics in the usual ODE form;
- $\eta := \{\eta_q, q \in Q_T\}$, where each η_q describes the motion on the trim trajectory q ;
- $v := \{v_q, q \in Q\}$ is the set of control laws designed for each trajectory primitive; we assume that each v_q is a Lipschitz function;
- $\mathbf{C} := \{C_q, q \in Q_M\}$: the collection of sets from which we can initiate maneuvers (controlled jump set);
- $\mathbf{D} := \{D_q, q \in Q_M\}$: the collection of sets at which maneuvers are terminated;
- $\mathcal{R} := \{\mathcal{R}_q, q \in Q_T\}$: the collection of recoverability sets for each trim trajectory;
- $\Omega := \{\Omega_q, q \in Q_T\}$: the collection of limit sets for each trim trajectory;
- $\mathbf{A} = \{A_q = (q_{new}, \Delta t, \Delta h), q \in Q_M\}$: autonomous jumps occur during maneuver execution, when the timer state $\tau \in T$ reaches the value ΔT . The state is reset such that $q \leftarrow q_{new}, h \leftarrow h_m h, \tau \leftarrow 0$;
- $V := \mathbb{R} \times Q_M$: is the hybrid control set, determining the controlled jump execution. Controlled jumps can only be executed from trim trajectories; given a hybrid control $(\Delta t_{coast}, q_{new})$, the jump occurs when $\tau \in T$ reaches Δt_{coast} , and the state is reset such that: $q \leftarrow q_{new}, h \leftarrow \exp(\eta_q \Delta t_{coast})h, \tau \leftarrow 0$.

We can graphically depict the hybrid automaton as a directed graph, where the nodes represent the trim trajectories, and the edges represent the maneuvers. Each edge can be labeled with a cost corresponding to the maneuver duration.

The RHA architecture that we have just defined can be considered as both a design paradigm and a modeling tool for nonlinear systems. The selection of the trajectory primitives, and the design of the tracking control law have to be carried out according to some specific requirements that are directly derived from the RHA structure. In particular, the

conditions for the automaton consistency and controllability have to be satisfied. On the other hand, a *RHA* can be seen as a powerful modeling tool, encoding all the relevant information on the dynamics of the system in a reduced set of state variables. The design of control laws for “higher-level” tasks to be performed by the system will then be substantially simplified: as it has been shown in [6, 8, 9], it will be possible to operate in a relatively small “maneuver space”, as opposed to the full state space. Motion planning on this maneuver space will be completely free from all the stability concerns because these have been already addressed in the construction of the *RHA*.

4.1 Well-posedness, consistency, and controllability

When dealing with systems of the form (1), where the right-hand side is not continuous, care must be taken to ensure that the system is well-posed, that is, a unique solution exists. In our case well-posedness is ensured by the fact that the system is piecewise continuous, and the maneuvers are a finite set of primitives with a finite time duration (by definition). As a consequence in every finite time interval there will be a finite number of switches, or discontinuities in the feedback map.

In order to decouple stability and motion planning concerns we have to ensure that the automaton is *consistent*, which means that any sequence of hybrid controls will generate a trajectory which remains “close” to the nominal trajectory, in the sense of the invariant sets defined in the previous sections.

Definition 8 (Consistency) *We say that the automaton is consistent if for all $q \in Q_T$ the following conditions hold: (i) $\Omega_q \subseteq C_l, \forall l \in L_q$; (ii) $\bigcup_{p \in P_q} D_p \subseteq \mathcal{R}_q$; (iii) $\bigcup_{p \in P_q} D_p \subseteq C_l, \forall l \in L_q$*

where $L_q, P_q \subset Q_M$ are respectively the set of indices of maneuvers leaving and arriving at the trim trajectory q .

Remark (Maneuver Recovery) *If the first and second conditions are satisfied, the third one can always be satisfied by adequately extending the maneuvers with a recovery phase at the new trim trajectory.*

Definition 9 (Hybrid State) *We will say that the system is in the hybrid state (q, h, τ) if $\Psi_{h-1}x \in \bigcap_{l \in L_q} C_l \subseteq R_q$, that is if the continuous state is inside the starting sets for all the maneuvers leaving the current trim trajectory.*

The full state of the system will then be described by (q, h, τ, x) , where $q \in Q, h \in H, \tau \in \mathbb{R}, x \in X$. Once we have established that the hybrid system is well posed and consistent, we have to ensure that it is controllable:

Definition 10 (Reachability) *We say that the *RHA* is reachable if it is possible to find an admissible sequence of primitives such that we can steer the system from any initial condition (q, h) to any desired location $\bar{h} \in H$, and at any desired operating condition $\bar{q} \in Q_T$, in finite time.*

It is clear that a necessary condition for reachability is that the directed graph describing the automaton be fully connected. Moreover, the set of trim trajectories must be rich enough that by interconnecting an appropriate sequence of them we cover the group H . In the case of a system with integrators, this translates simply to the requirements that the set $\{\eta_q : q \in Q_T\}$ is a complete basis for \mathbb{R}^n . The problem

of assessing the minimum set of trim trajectories based on which we can build a reachable automaton for autonomous vehicles recalls some classical problems both in the nonlinear control and robotics literature. For example, it is known that for models of car-like robots optimal trajectories are indeed composed by straight lines and by arcs of minimum radius circles (trim trajectories in the plane)[17, 18]. We are not aware of results in the literature which are applicable to systems switching between trim trajectories in a three-dimensional space, with arbitrary transients (maneuvers) during the switches. Below we will state and prove the main result of this paper, giving a sufficient characterization of minimal sets of trim trajectories for reachability of the hybrid automaton. It is clear that this would be just a minimum set of trim trajectories to ensure reachability: for practical applications, the set of trim trajectories will be much richer.

Proposition 4.1 (Minimum set of trim trajectories)

Assume that the system dynamics are invariant to translation and rotations about a vertical axis, i.e. that H is a symmetry group for the system. Then two trim trajectories q_1 and q_2 , described by the parameters $\{V_1, \psi_1, \gamma_1, \beta_1\}$, $\{V_2, \psi_2, \gamma_2, \beta_2\}$, along with any two maneuvers connecting them, are sufficient for reachability in a three-dimensional space if: $V_1\psi_2 \cos \gamma_1 \neq V_2\psi_1 \cos \gamma_2$, and $V_1 \sin \gamma_1 < 0 < V_2 \sin \gamma_2$. If the motion of the vehicle is restricted to a horizontal plane then the system is reachable if just the first condition is satisfied.

Proof: To prove that two trim trajectories, along with two maneuvers connecting them, are sufficient for reachability, we will show how to construct a sequence connecting any desired starting and ending hybrid states. Without loss of generality, assume that the system starts in the hybrid state (q_1, id_H) , and that we want to move it to (q_1, \bar{h}) .

Since either ψ_1 or ψ_2 is non-zero, as a first step we can steer the heading angle ψ to any desired value. We have to show that we can change independently the translational coordinates of the vehicle, without affecting the heading thus obtained.

We have to examine the two cases, in which $\psi_1\psi_2$ is either zero or non-zero. Assume first that $\psi_1, \psi_2 \neq 0$, and indicate with h_{12}, h_{21} and ψ_{12}, ψ_{21} respectively the maneuvers connecting the two trim trajectories and the resulting heading changes. We can define the following maneuver sequence: $h_S(t_1, t_2) = h_{trim_1}(t_S(t_1, t_2)) \cdot h_{12} \cdot h_{trim_2}(t_2) \cdot h_{21} \cdot h_{trim_1}(t_1)$ with $t_1, t_2 > 0$, and $t_S(t_1, t_2) := \psi_1^{-1} \left[(2k+1)\pi - \psi_1 t_1 - \psi_2 t_2 - \psi_{12} - \psi_{21} \right]$, where k is the smallest integer such that the right hand side is strictly positive. If we combine two h_S sequences we obtain: $h_0(t_1, t_2, \tau_1, \tau_2) := h_S(t_1 + \tau_1, t_2 + \tau_2) \cdot h_S(t_1, t_2)$ Simple geometric considerations indicate that h_0 preserves the heading. Moreover we have that that $h_0(t_1, t_2, \tau_1, \tau_2)$ results in a translation $p = [p_x, p_y, p_z]^T$, where $p_x = p_y = 0$ when $\tau_1 = \tau_2 = 0$. Straightforward calculations show that the Jacobian matrix $\partial(p_x, p_y)/\partial(\tau_1, \tau_2)$ has full rank for some (in fact almost all) choice of t_1 and t_2 . From continuity we can conclude that it is possible to steer the system to any horizontal position, while maintaining the heading, by iterating h_0 with appropriate values of $(t_1, t_2, \tau_1, \tau_2)$.

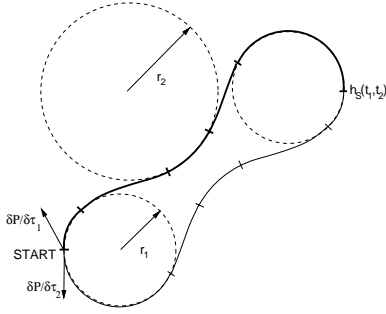


Figure 3: Construction of h_0

In the case in which the vehicle is moving on a horizontal plane, the proof is complete. In the three-dimensional case, we still have to adjust the altitude. As already mentioned, if $\tau_1 = \tau_2 = 0$ then the execution of h_0 results only in a change of altitude. We also have that $\partial p_z / \partial t_1 = -\partial p_z / \partial t_2 = V_1 \sin \gamma_1 - V_2 \sin \gamma_2 \neq 0$. Hence, maneuver sequences of the type $h_0(t_1, t_2, 0, 0)$ can be used to arbitrarily change the altitude of the vehicle.

In the case in which one of the trim trajectories (assume the first one) is characterized by a zero turning rate, then the above arguments have to be repeated. In this case we consider the following basic maneuver sequence (straight segment plus a 90 degree turn): $h_q(t_1) := h_{12} \cdot h_{trim_2} \left(\frac{(2k-1)\pi/2 - \psi_{12} - \psi_{21}}{\psi_2} \right) \cdot h_{21} \cdot h_{trim_1}(t_1)$, which can be repeated four times to obtain: $h'_0(t_1, t_2, \tau_1, \tau_2) = h_q(t_2 + \tau_2) \cdot h_q(t_1 + \tau_1) \cdot h_q(t_2) \cdot h_q(t_1)$. The considerations we made for the previous case are valid also in this case when considering h'_0 instead of h_0 , that is maneuver sequences of the type $h'_0(t_1, t_2, \tau_1, \tau_2)$ can be used to arbitrarily change the horizontal position first, and then the altitude (by setting $\tau_1 = \tau_2 = 0$). ■

5 Conclusions

In this paper a Robust Hybrid Automaton architecture, applicable to autonomous vehicles has been presented and discussed. Its structure has been defined and motivated, and some of its fundamental properties have been analyzed. Specifically, we have established well-posedness, and given sufficient conditions for consistency and reachability. The architecture presented in this paper has been used to develop computationally efficient, real-time motion planning algorithms for agile autonomous vehicles [8, 9].

6 Acknowledgments

This research was supported by the C. S. Draper Laboratory through the IR&D grant DL-H-505334, by the AFOSR, under grant F49620-99-1-0320, and by the ONR, under Young Investigator Award N-00014-99-1-0668.

References

[1] M. S. Branicky. *Studies in Hybrid Systems: Modeling, Analysis, and Control*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[2] N. Lynch, R. Segala, F. Vandraager, and H.B. Weinberg. Hybrid I/O automata. In R. Alur, T. Henzinger, and E. D. Sontag, editors, *Hybrid Systems III: Verification and Control*, volume 1066 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[3] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[4] T. Henzinger, P. H. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Trans. Aut. Control*, 43(4):540–554, 1998.

[5] T. J. Koo, F. Hoffmann, B.Sinopoli, and S. Sastry. Hybrid control of an autonomous helicopter. In *IFAC Workshop on Motion Control*, 1998.

[6] E. Frazzoli, M.A. Dahleh, and E. Feron. A hybrid control architecture for aggressive maneuvering of autonomous helicopters. In *IEEE Conf. on Decision and Control*, December 1999.

[7] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3), March 1999.

[8] E. Frazzoli, M.A. Dahleh, and E. Feron. Robust hybrid control for autonomous vehicle motion planning. Technical Report LIDS-P-2468, Massachusetts Institute of Technology, 1999. Revised version submitted to IEEE Trans. on Automatic Control.

[9] E. Frazzoli, M.A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. In *AIAA Conf. on Guidance, Navigation and Control*, Denver, CO, August 2000.

[10] J. E. Marsden and T.S. Ratiu. *Introduction to Mechanics and Symmetry*. Springer Verlag, New York, NY, 1999. Second edition.

[11] V.I. Arnold. *Mathematical Methods of Classical Mechanics*, volume 60 of *GTM*. Springer Verlag, New York, NY, second edition edition, 1989.

[12] R.M. Murray, Z.Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[13] M.W. McConley, B.D. Appleby, M.A. Dahleh, and E.Feron. A computationally efficient Lyapunov-based scheduling procedure for control of nonlinear systems with stability guarantees. *IEEE Transactions on Automatic Control*, December 1999.

[14] R. R. Burrige, A. A. Rizzi, and D.E. Koditscheck. Sequential decomposition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, 18(6):534–555, June 1999.

[15] J.S. Shamma and J.R. Cloutier. Gain-scheduled missile autopilot design using linear parameter varying transformations. *AIAA J. on Guidance, Control and Dynamics*, 16(2):256–263, March-April 1993.

[16] I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre. Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control. *AIAA Journal of Guidance, Control, and Dynamics*, 21(1):29–38, January-February 1998.

[17] L.E. Dubins. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[18] J.A. Reeds and R.A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2), 1990.