A Hybrid Control Framework for Robust Maneuver-based Motion Planning

Ricardo G. Sanfelice and Emilio Frazzoli

***Abstract*— We introduce a modeling framework for robustness of maneuver-based motion planning algorithms for nonlinear systems with symmetries. Our framework implements a hybrid controller that robustly combines motion primitives, which consist of trim trajectories and maneuvers, from a predefined library. The closed-loop system is viewed as a hybrid system with flows given by a differential equation, jumps given by a difference equation, and two sets where these dynamics are allowed. We show that our hybrid controller for implementation of motion planning algorithms confers to the closed-loop system robustness properties to a large class of perturbations.**

I. Introduction

Motion planning algorithms are commonly applied in robotics as a method to solve steering problems. In a real-world scenario, the motion planning task needs to be accomplished in the presence of obstacles, measurement error, exogenous disturbances, and unmodeled dynamics. To guarantee some degree of robustness, motion planning algorithms are usually blended with feedback control algorithms, which track the output of the motion planner; see, e.g., [1]–[5].

The motion planning problem itself is typically recast as an optimal control problem with cost function and constraints stemming from the given task to be accomplished along with its specifications. In complex motion planning problems, online computation of optimal control policies is not always feasible. A motion planning technique suitable in such cases was proposed in [6] for general nonlinear systems with symmetries. A motion plan in [6] is given by a concatenation of a finite number of *motion primitives* selected from a pre-defined library and implemented in a *maneuver automaton*. Motion primitives were defined in [6] as equivalence classes of trajectories, induced by symmetries in the system's dynamics, e.g., invariance with respect to time, translations, and rotations.

One of the main features of the maneuver-motion based approach is that each element in the motion primitives library can be designed off-line subject to particular specifications, like optimality, state constraints, etc., relaxing in this way on-line computation requirements; see, e.g., its applications to robotics in [4], [7], [8]. However, this method combines motion primitives in an open-loop manner, which restricts its application to nominal scenarios, that is, those without perturbations. Moreover, the fact that the trajectories resulting from this approach are not necessarily smooth, renders the task of robustifying motion plans via feedback control challenging since standard trajectory tracking control design techniques are not applicable.

In this paper, we propose a hybrid control algorithm that executes maneuver-based motion plans and combines state feedback control laws for nonlinear systems with symmetries. The purpose of our hybrid controller is to provide a control framework for maneuver-based motion planning featuring robustness properties to perturbations. We show that this framework results in a hybrid system with implementable semantics, and hence, useful experimental setups. This class of hybrid systems has been recently introduced in [9], [10] motivated by the pursue of robustness of asymptotic stability. Our control framework for maneuver-based motion planning also borrows ideas from the techniques in [11] for robust combination of state feedback and open-loop controllers, and also from the invariant constructions in [12].

The paper is organized as follows. Section II introduces notation and basic definitions regarding nonlinear systems with symmetries, motion primitives and plans, and hybrid systems. Section III introduces our hybrid control framework for motion planning, while Section IV states its main properties.

II. Preliminaries

A. Notation

$\mathbb{R}$ denotes the real numbers. $\mathbb{R}_{\geq 0}$ denotes the nonnegative real numbers, i.e., $\mathbb{R}_{\geq 0} = [0, \infty)$. $\mathbb{N}$ denotes the natural numbers including 0, i.e., $\mathbb{N} = \{0, 1, \ldots\}$. $\mathbb{N}_{<k}$ ($\mathbb{N}_{\leq k}$) denotes numbers in $\mathbb{N}$ from 0 to $k-1$ (from 0 to k, respectively). $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. $\mathbb{B}$ denotes the open unit ball in a Euclidean space. Given a set S, $\overline{S}$ denotes its closure and S° denotes its interior. Given sets S_1, S_2 subsets of $\mathbb{R}^n$, $S_1 + S_2 := \{x_1 + x_2 \mid x_1 \in S_1, x_2 \in S_2\}$. Given a vector $x \in \mathbb{R}^n$, $|x|$ denotes its Euclidean norm. The equivalent notation $[x^\top \ y^\top]^\top$, $[x \ y]^\top$, and (x, y) is used for vectors. Given a function $f : \mathbb{R}^m \to \mathbb{R}^n$, its domain of definition is denoted by $\operatorname{dom} f$; i.e., $\operatorname{dom} f := \{x \in \mathbb{R}^m \mid f(x) \text{ is defined}\}$. A function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to class $\mathcal{K}_\infty$ if it is continuous, zero at zero, strictly increasing, and unbounded. $\mathcal{PC}^0(\mathbb{R}_{\geq 0}, \mathbb{R}^m)$ is the set of all piecewise continuous signals $\beta : \operatorname{dom} \beta \to \mathbb{R}^m$, $\operatorname{dom} \beta \subset \mathbb{R}_{\geq 0}$.

B. Motion planning for nonlinear systems with symmetries

We consider nonlinear control systems of the form

$$\mathcal{P} : \qquad \dot{x} = f(x, u) \tag{1}$$

Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, MA, 02139. {sricardo,frazzoli}@mit.edu.
Research partially supported by ARO through grant W911NF-07-1-0499, and by NSF through grant 0715025. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the supporting organizations.

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a locally Lipschitz function, $x \in \mathbb{R}^n$ is the state, and $u \in \mathbb{R}^m$ is the control input. We focus on a particular subclass of nonlinear systems $\mathcal{P}$, those satisfying certain symmetry properties. Next, we review and adapt some of the concepts in [6] for the purposes of this paper.

1) Nonlinear systems with symmetries: A large class of mechanical systems are invariant under certain transformations of their state. These include mobile robots as well as more general autonomous vehicles, like several helicopters and airplanes models, among others. General invariant transformations can be characterized with the theory of Lie groups (see [13] for an introduction to Lie groups and [14] for applications to mechanics).

Let $\mathcal{G}$ be a finite-dimensional Lie group, and let e be its identity element. It is said that Ψ is a left action of the group $\mathcal{G}$ on $\mathbb{R}^n$ if $\Psi : \mathcal{G} \times \mathbb{R}^n \to \mathbb{R}^n$ is a smooth map such that $\Psi(e, x) = x$ for all $x \in \mathbb{R}^n$ and $\Psi(g, \Psi(h, x)) = \Psi(gh, x)$ for all $g, h \in \mathcal{G}$, $x \in \mathbb{R}^n$. Let $\mathfrak{g}$ be the Lie algebra of $\mathcal{G}$.

Definition 2.1: (symmetry of $\mathcal{P}$) The nonlinear system $\mathcal{P}$ is invariant with respect to the left group action Ψ if for all $g \in \mathcal{G}$, $x^0 \in \mathbb{R}^n$, and $\mu \in \mathcal{PC}^0(\mathbb{R}_{\geq 0}, \mathbb{R}^m)$, each solution (in the appropriate sense[1]) to $\mathcal{P}$ starting from x^0 with $u(t) = \mu(t)$, denoted by $t \mapsto \phi(x^0, \mu; t)$, is such that $\Psi(g, \phi(x^0, \mu; t)) = \phi(\Psi(g, x^0), \mu; t)$ for all $t \in \operatorname{dom} \phi$. ■

Definition 2.1 states that $\mathcal{P}$ is invariant if the left action Ψ commutes with the map from initial conditions.

2) Library of motion primitives: Trim trajectories and maneuvers define our "library" of primitives for motion planning; see also [6, Section III].

Definition 2.2 (trim): A C^1 function $x : [0, T] \to \mathbb{R}^n$ is a *trim trajectory* for $\mathcal{P}$ if there exists $\xi \in \mathfrak{g}$, called the *trim velocity vector*, and $\mu \in \mathbb{R}^m$, called the *trim input*, such that

$$\begin{aligned} x(t) &= \Psi(\exp(\xi t), x(0)) \qquad \text{for all } t \in [0, T] \ , \qquad (2) \\ \dot{x}(t) &= f(x(t), \mu) \qquad \text{for almost all } t \in [0, T]. \end{aligned}$$ ■

When the right-hand side of $\mathcal{P}$ is locally Lipschitz, every trim trajectory x for $\mathcal{P}$ is uniquely defined by its velocity ξ and initial condition x^0. We shall assume the following property throughout the paper.

Standing Assumption 2.3: The function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is locally Lipschitz continuous. The nonlinear system $\mathcal{P}$ is invariant under the action of Ψ. ■

Then, for the nonlinear system $\mathcal{P}$ with symmetry group $\mathcal{G}$, we store ξ and x^0 in the set of trim trajectories, which is denoted by $\mathcal{T}(\mathcal{P}, \mathcal{G}) \subset \mathfrak{g} \times \mathbb{R}^n$.

Definition 2.4 (maneuver): A C^1 function $x : [0, T] \to \mathbb{R}^n$ is a *maneuver* for $\mathcal{P}$ if there exist a function $\beta \in \mathcal{PC}^0(\mathbb{R}_{\geq 0}, \mathbb{R}^m)$, called the *maneuver input*, such that

$$\dot{x}(t) = f(x(t), \beta(t)) \qquad \text{for almost all } t \in [0, T] \ ;$$

$g \in \mathcal{G}$, called the *maneuver displacement*, satisfying

$$x(T) = \Psi(g, x(0)) \ ;$$

and trim trajectories $x' : [0, T'] \to \mathbb{R}^n, x'' : [0, T''] \to \mathbb{R}^n$ that are *compatible* with x, i.e., there exist *matching displacements* $g', g'' \in \mathcal{G}$ such that

$$x'(T') = \Psi(g', x(0)), \ x(T) = \Psi(g'', x''(0)) \ .$$ ■

Remark 2.5: The matching displacements g' and g'' in Definition 2.4 guarantee that trim trajectories and maneuvers can be concatenated. More precisely, the left action Ψ with displacement g' guarantees that the end point of the (left compatible) trim trajectory x' can be concatenated with the initial point of the maneuver x, while the left action Ψ with displacement g'' guarantees that the initial point of the (right compatible) trim trajectory x'' can be concatenated with the final point of the maneuver x. ■

Maneuver information for $\mathcal{P}$ with symmetry group $\mathcal{G}$ is stored in the set $\mathcal{M}(\mathcal{P}, \mathcal{G})$. By the regularity properties of f, a maneuver x for $\mathcal{P}$ can be generated by only knowing the input β applied to $\mathcal{P}$ and the initial condition x^0. By construction, the application of β at x^0 causes a maneuver displacement given by $g \in \mathcal{G}$.

Following the definitions above, a "library" of motion primitives for $\mathcal{P}$ with symmetry group $\mathcal{G}$ is given by $(\mathcal{T}(\mathcal{P}, \mathcal{G}), \mathcal{M}(\mathcal{P}, \mathcal{G}))$. Let $Q_T, Q_M \subset \mathbb{N}$ be compact and disjoint sets, and define $Q := Q_T \cup Q_M$. The set Q_T (respectively, Q_M) is such that each of its elements is uniquely associated to a trim trajectory (respectively, to a maneuver). More precisely, for each $q \in Q_T$, $(\xi_q, x_q^0) \in \mathcal{T}(\mathcal{P}, \mathcal{G})$ defines the trim trajectory $x_q(t) = \Psi(\exp(\xi_q t), x_q^0)$ with $x_q(0) = x_q^0$, while for each $q \in Q_M$, $(\beta_q, x_q^0, g_q, T_q) \in \mathcal{M}(\mathcal{P}, \mathcal{G}) \subset \mathcal{PC}^0(\mathbb{R}_{\geq 0}, \mathbb{R}^m) \times \mathbb{R}^n \times \mathcal{G} \times \mathbb{R}$ correspond to the input to generate the maneuver x_q from x_q^0, which, after T_q units of time, results in a displacement given by g_q.

3) Motion plan: A motion plan v is denoted by

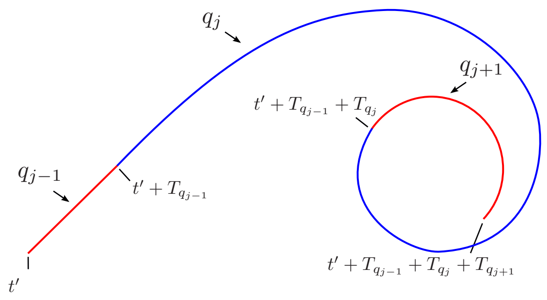
$$\begin{aligned} v \quad := \{ &(q_1, T_{q_1}), (q_2, g_2', g_2''), (q_3, T_{q_3}), \ldots, \\ &\ldots, (q_{k-1}, g_{k-1}', g_{k-1}''), (q_k, T_{q_k}) \} \ , \end{aligned}$$

where $k \in \mathbb{N}_{\geq 3}$ is an odd number and:

- For each odd number $j \in \mathbb{N}_{\leq k}$, $q_j \in Q_T$.
- For each even number $j \in \mathbb{N}_{\leq k}$, $q_j \in Q_M$ and the j-th maneuver is compatible with the $(j-1)$-th trim trajectory with matching displacement g_j' and with the $(j+1)$-th trim trajectory with matching displacement g_j''.
- For each odd number $j \in \mathbb{N}_{<k}$, $T_{q_j} \in \mathbb{R}_{\geq 0}$ defines the time to execute the q_j-th trim trajectory. The nonnegative constant T_{q_k} for the last trim trajectory can be either finite or infinite.

In other words, a motion plan v is given by a sequence $\{v_j\}_{j=1}^k$, where $v_2, v_4, \ldots, v_{k-1}$ are such that $q_2, q_4, \ldots, q_{k-1} \in Q_M$ define maneuvers and $v_1, v_3, \ldots, v_k$ are such that $q_1, q_3, \ldots, q_k \in Q_T$ define (compatible) trim trajectories. (Alternatively, and without affecting the results in this paper, motion plans can be defined as in [6].) We denote by $\mathcal{V}(\mathcal{P}, \mathcal{G})$ the set of motion plans for $\mathcal{P}$ with symmetry group $\mathcal{G}$ generated from $(\mathcal{T}(\mathcal{P}, \mathcal{G}), \mathcal{M}(\mathcal{P}, \mathcal{G}))$. Figure 1 depicts a sample trim-maneuver-trim piece of a motion plan $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$.

[1] This property does not depend on the notion of solution used. It is required to hold for each (perhaps nonunique) solution to $\mathcal{P}$ on its domain.



q_j
q_{j+1}
$t' + T_{q_{j-1}} + T_{q_j}$
q_{j-1}
$t' + T_{q_{j-1}}$
$t' + T_{q_{j-1}} + T_{q_j} + T_{q_{j+1}}$
t'

Fig. 1. Sequence of entries of a motion plan v: $v_{j-1} = (q_{j-1}, T_{q_{j-1}})$ defining trim trajectory $x_{q_{j-1}}$, $v_j = (q_j, g'_{q_j}, g''_{q_j})$ defining maneuver x_{q_j}, and $v_{j+1} = (q_{j+1}, T_{q_{j+1}})$ defining trim trajectory $x_{q_{j+1}}$.

C. Hybrid systems

The hybrid control framework proposed in this paper for maneuver-based motion planning follows the general model for hybrid systems in outlined in [9] (see also [10], [15]). Hybrid systems are dynamical systems with continuous and discrete dynamics. In [9], a hybrid system $\mathcal{H}$ is given by a flow map, a flow set, a jump map, and a jump set. For the purposes of this paper, the state of the hybrid system, denoted by ζ, takes values in $\mathbb{R}^n$, the flow map is given by a function $f : \mathbb{R}^n \to \mathbb{R}^n$ and the flow set, denoted by $C \subset \mathbb{R}^n$, define the flow equation $\dot{x} = f(x), x \in C$; while the jump map is given by a function $g : \mathbb{R}^n \to \mathbb{R}^n$ and the jump set, denoted by $D \subset \mathbb{R}^n$, define the jump equation $x^+ = g(x), x \in D$. Continuous evolution of the solutions (or flows) to $\mathcal{H}$ is permitted only when the solution is in C and discrete evolution (or jumps) is allowed only when the solution is in D. Hence, a hybrid system $\mathcal{H}$ has data (f, C, g, D) and can be written as

$$\mathcal{H}: \qquad x \in \mathbb{R}^n \qquad \left\{ \begin{array}{rcll} \dot{x} &=& f(x), & x \in C \\ x^+ &=& g(x), & x \in D \end{array} \right. .$$

To define solutions to $\mathcal{H}$, the number of jumps is treated as an independent variable j and the state is parametrized by (t, j). A solution is a function defined on subsets of $\mathbb{R}_{\geq 0} \times \mathbb{N}$. A subset $E \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a *compact hybrid time domain* if

$$E = \bigcup_{j=0}^{J-1} ([t_j, t_{j+1}], j)$$

for some finite sequence of times $0 = t_0 \leq t_1 \ldots \leq t_J$. It is a *hybrid time domain* if for all $(T, J) \in E$, $E \cap ([0, T] \times \{0, 1, \ldots J\})$ is a compact hybrid domain. On each hybrid time domain there is a natural ordering of points: $(t, j) \preceq (t', j')$ if $t \leq t'$ and $j \leq j'$. A *hybrid arc* is a function $x : \operatorname{dom} x \to \mathbb{R}^n$ on a hybrid time domain $\operatorname{dom} x$ such that $x(t, j)$ is absolutely continuous in t for a fixed j and $(t, j) \in \operatorname{dom} x$. It is a solution to the hybrid system $\mathcal{H}$ if $x(0, 0) \in \overline{C} \cup D$ and

(S1) For all $j \in \mathbb{N}$ and almost all t such that $(t, j) \in \operatorname{dom} x$,

$$x(t, j) \in C, \quad \dot{x}(t, j) = f(x(t, j))$$

(S2) For all $(t, j) \in \operatorname{dom} x$ such that $(t, j+1) \in \operatorname{dom} x$,

$$x(t, j) \in D, \quad x(t, j+1) = g(x(t, j)).$$

A concept of closeness of solutions to hybrid systems is as follows. Two solutions $x : \operatorname{dom} x \to \mathbb{R}^n$, $y : \operatorname{dom} y \to \mathbb{R}^n$ are (T, J, ε)*-close* if:

(a) for all $(t, j) \in \operatorname{dom} x$ with $t \leq T$, $j \leq J$ there exists s such that $(s, j) \in \operatorname{dom} y$, $|t - s| < \varepsilon$, and

$$|x(t, j) - y(s, j)| < \varepsilon,$$

(b) for all $(t, j) \in \operatorname{dom} y$ with $t \leq T$, $j \leq J$ there exists s such that $(s, j) \in \operatorname{dom} x$, $|t - s| < \varepsilon$, and

$$|y(t, j) - x(s, j)| < \varepsilon.$$

Note that this closeness concept does not require solutions to be close at jumps at the same hybrid instant (t, j). See [9] and [10] for more details.

III. A Hybrid Controller for Motion Planning

Given a motion plan $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$, our goal is to design a controller generating a trajectory of $\mathcal{P}$ that satisfies the motion plan specifications given in terms of a finite sequence of trim trajectories and maneuvers from $(\mathcal{T}(\mathcal{P}, \mathcal{G}), \mathcal{M}(\mathcal{P}, \mathcal{G}))$. We propose a hybrid controller, denoted by $\mathcal{H}_c$, with:

- logic state $q \in Q$ to indicate the system mode: *trim mode* when $q \in Q_T$, *maneuver mode* when $q \in Q_M$.
- logic state $p \in \mathbb{N}$ to select an entry of a given motion plan $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$.
- displacement state $z \in \mathcal{G}$ to store the overall displacement of the trajectory of $\mathcal{P}$.
- timer state $\tau \in \mathbb{R}$ to keep track of the time in maneuver mode and to parametrize the reference trajectory during trim mode.

The output of the controller, that is, the input of $\mathcal{P}$, is

$$u = \kappa_c(x, q, \tau) \tag{3}$$

where $\kappa_c : \mathbb{R}^n \times Q \times \mathbb{R} \to \mathbb{R}^m$. The input to $\mathcal{H}_c$ is the state x of $\mathcal{P}$.

A. Control strategy

Given a motion plan $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$, let $q = q_j \in Q_T$, $j \in \mathbb{N}_{<k}$. The controller $\mathcal{H}_c$ performs the following tasks:

Task 1) *Trim Trajectory Tracking*: Track the trim trajectory x_q, where x_q is defined by $(\xi_q, x_q^0) \in \mathcal{T}(\mathcal{P}, \mathcal{G})$ via (2).

Task 2) *Maneuver Execution Start*: When the state x is such that the maneuver $x_{q_{j+1}}$, which succeeds the trim trajectory x_q, can be executed and the timer elapsed for at least T_q units of time, update q to q_{j+1}, reset timer τ to zero, and execute the $(j+1)$-th maneuver.

Task 3) *Maneuver Execution End*: When the state x is such that the trim trajectory $x_{q_{j+2}}$ can be executed and the timer τ has elapsed for at least T_q units of time, update q to q_{j+2} and perform Task 1) if $j + 2 < k$.

Execution of trim trajectories in Task 1 is performed in closed-loop with a local tracking controller that guarantees $x(t) \to x_q(t)$ asymptotically. Maneuvers are started when: 1) the timer has elapsed for at least the duration planned for the predecessor trim trajectory, and 2) the state reaches a set from where the maneuver can be executed (the latter corresponds to Task 2). The trim trajectory that follows every maneuver is started as soon as the state x is in the set where tracking is possible and the timer has elapsed the specified amount of time for the maneuver.

B. Control design

The following assumption guarantees that Task 1 can be accomplished.

Assumption 3.1: (tracking of trim trajectories) For each $q \in Q_T$, there exists a continuous function $\kappa_q : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^m$, a continuously differentiable function $V_q : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, class-$\mathcal{K}_\infty$ functions α_q^1, α_q^2, and an open neighborhood of the origin $\mathcal{B}_q \subset \mathbb{R}^n$ such that

$$\begin{aligned} \alpha_q^1(|e|) \leq V_q(e) \leq \alpha_q^2(|e|) \qquad & \forall e \in \mathbb{R}^n \ , \\ \langle V_q(e), \tilde{f}(e) \rangle \leq -V_q(e) \qquad & \forall e \in \mathcal{B}_q \ , \end{aligned} \tag{4}$$

where $\tilde{f} : \mathbb{R}^n \to \mathbb{R}^n$ is given by

$$\tilde{f}(e) = f(e + x_q(t), \kappa_q(e + x_q(t), t) - f(x_q(t), \mu_q) \ ,$$

and defines the time-invariant system $\dot{e} = \tilde{f}(e)$ invariant under the action of Ψ, where x_q is the trim trajectory generated by μ_q.

Remark 3.2: In addition to the invariance property, Assumption 3.1 guarantees the existence of a local controller, with basin of attraction $\mathcal{B}_q$, which accomplishes asymptotic tracking of trim trajectories. Additionally, each tracking control law κ_q is such that, when applied to $\mathcal{P}$, result in a time-invariant error system with $e := x - x_q$ having the symmetry property. This assumption holds for nonlinear systems that can be put in feedback linearizable normal form [16], [17] with error system that is invariant under the action of Ψ [18]. ■

The construction of the flow and jumps sets of $\mathcal{H}_c$ follows. By the continuity properties of maneuvers in Definition 2.4, for each maneuver x_q with input β_q and maneuver duration T_q, $q \in Q_M$, there exist disjoint and open sets $S_q, L_q \subset \mathbb{R}^n$ such that for each $x_q(0) \in S_q$, $x_q(T_q) \in L_q$, $\dot{x}_q(t) = f(x_q, \beta_q(t))$. For each $q \in Q_M$, pick compact sets D_q such that $D_q \subset S_q$ and $x_q^0 \in D_q^\circ$, and define $C_q := \overline{\mathbb{R}^n \setminus D_q}$. The set D_q, $q \in Q_M$, corresponds to the maneuver's start set in Task 2.

We now compute the set of points from where tracking of trim trajectories is possible. By construction, there exist $\varepsilon^* > 0$ such that

$$\varepsilon^* := \underset{\varepsilon > 0}{\text{argmax}} \, \{x_q^0 + \varepsilon \overline{\mathbb{B}} \subset S_q, \ \forall q \in Q_M\} \ .$$

Using Assumption 3.1, for each $q \in Q_T$, define

$$D_q := \{e \in \mathbb{R}^n \ | \ V_q(e) \leq c_q\} \ ,$$

where $c_q > 0$ is such that

$$D_q \subset (x_q^0 + \delta_q \overline{\mathbb{B}}) \cap \mathcal{B}_q \ , \ \ \delta_q := (\alpha_1^q)^{-1}(\exp(T_q) \alpha_2^q(\varepsilon^*)) \ ,$$

and $(\alpha_1^q)^{-1}$ is the inverse of the function α_1^q. Define $C_q := \overline{\mathbb{R}^n \setminus D_q}$. This construction yields a constant δ_q such that when the trim trajectory $x_q(t)$ is tracked from initial conditions in D_q, the state x belongs to a subset of the start set of each of the maneuvers in $\mathcal{M}(\mathcal{P}, \mathcal{G})$ after T_q units of time have elapsed (T_q is the execution time of the trim trajectory given in the motion plan).

The following assumption guarantees that maneuvers take trajectories to points where trim trajectories can be executed.

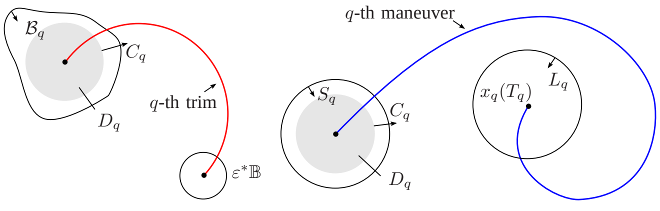
Assumption 3.3 (nested condition): For every motion plan $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$, every maneuver with associated entry v_i in v and input β_{q_i}, its associated set L_{q_i} is such that

$$L_{q_i} \subset D_{q_{i+1}} \ ,$$

where $D_{q_{i+1}}$ is the set associated with tracking of the trim trajectory $x_{q_{i+1}}$, $q_{i+1} \in Q_T$.

Remark 3.4: The condition in Assumption 3.3 assures that, after a maneuver, the state x is in a set from which tracking of the trim trajectory succeeding it is possible. This condition holds by picking small enough landing set L_q when Assumption 3.1 is in place. However, in order to get practical robustness results, the landing sets are usually fixed. In such cases, the tracking law in Assumption 3.1 should be chosen to have large enough set $D_q, q \in Q_T$. ■

Figure 2 illustrates the sets designed above.



(a) Trim sets. (b) Maneuver sets.

Fig. 2. Sets of the hybrid controller for a trim trajectory and maneuver in the motion primitive in Figure 1.

C. Hybrid controller

The control logic outlined above is implemented in the hybrid controller $\mathcal{H}_c$ as follows.

1) Jumps: Jumps occur while in trim mode with $p < k$ (i.e., it is not the last trim trajectory of the motion plan) when the state x reaches the set of points where the maneuver $x_{q_{p+1}}$ can be started and the timer τ has elapsed for T_{q_p} units of time. The set in the first condition is given by $D_{q_{p+1}}$, $q_{p+1} \in Q_M$, after the left action Ψ with displacement given by z multiplied by the nominally expected trim trajectory displacement $\exp(\xi_q T_q)$ and the matching displacement $g'_{q_{p+1}}$. Then, jumps occur when

$$q \in Q_T \text{ and } x \in \Psi(z \exp(\xi_q T_q) g'_{q_{p+1}}, D_{q_{p+1}}) \text{ and } \tau \geq T_q \ , \tag{5}$$

with update law

$$q^+ = q_{p+1}, \ p^+ = p + 1, \ z^+ = z \exp(\xi_q \tau), \ \tau^+ = 0 \ , \tag{6}$$

that is, q is mapped to the next mode in the motion plan v, the motion plan index p is incremented by one, z is updated with the current total displacement of the motion primitive, and τ is reset to zero.

While in maneuver mode, jumps occur when the state reaches the set of points where the trim trajectory $x_{q_{p+1}}$ can be started and the timer state τ has elapsed for at least T_q units of time. As in the case for jumps during trim mode, the set in the former condition is given by D_q, $q \in Q_M$, after the invariant operation Ψ with displacement given by z multiplied by the planned maneuver trajectory displacement, which is given by g_q, and the matching displacement g''_q. Then, jumps in maneuver mode occur when

$$q \in Q_M \text{ and } x \in \Psi(zg_qg''_q, D_{q_{p+1}}) \text{ and } \tau \geq T_q \ , \tag{7}$$

with update law

$$q^+ = q_{p+1}, \quad p^+ = p+1, \quad z^+ = zg_q, \quad \tau^+ = 0 \ . \tag{8}$$

2) Flows: During flows, the controller variables have dynamics given by

$$\dot{q} = 0, \quad \dot{p} = 0, \quad \dot{z} = 0, \quad \dot{\tau} = 1 \ , \tag{9}$$

when

$$q \in Q_T \text{ and } (x \in \Psi(z\exp(\xi_q T_q)g'_{q_{p+1}}, C_{q_{p+1}}) \text{ or } \tau \in [0, T_q]), \tag{10}$$

or

$$q \in Q_M \text{ and } (x \in \Psi(zg_qg''_q, C_{q_{p+1}}) \text{ or } \tau \in [0, T_q]). \tag{11}$$

3) Output: The controller output is the input to $\mathcal{P}$ and is given by $u = \kappa_c(x, q, \tau)$ where

$$\kappa_c(x, q, \tau) = \begin{cases} \beta_q(\tau) & \text{if } q \in Q_M \\ \kappa_q(x, \tau) & \text{if } q \in Q_T \ . \end{cases} \tag{12}$$

The function β_q is the control input that generates the q-th maneuver, $q \in Q_M$. The function κ_q is the tracking control law in Assumption 3.1 for the q-th trim trajectory, $q \in Q_T$, which is designed using trim trajectory information.

4) Closed-loop system: We denote the closed-loop system resulting from controlling $\mathcal{P}$ with $\mathcal{H}_c$ by $\mathcal{H}_{cl}$ and its state by $\varphi := (x, q, p, z, \tau) \in \mathcal{X} := \mathbb{R}^n \times Q \times \mathbb{N}_{\leq k} \times \mathbb{R}^l \times \mathbb{R}$, where the Euclidean space $\mathbb{R}^l$ embeds $\mathcal{G}$. The continuous dynamics are given by closed-loop plant dynamics $\dot{x} = f(x, \kappa_c(x, q, \tau))$ along with (9), with flow set given by the union of the sets defined by (10) and (11). The discrete dynamics are given by the update laws in (6) and (8). The resulting closed-loop system $\mathcal{H}_{cl}$ can be written in the compact form in (II-C) using φ as the state and appropriately defining functions $\tilde{f}, \tilde{g}$ and sets $\tilde{C}$ and $\tilde{D}$.

IV. Motion plan execution: nominal and perturbed case

Given a motion plan v and an initial configuration $(x^0_v, g^0_v) \in \mathbb{R}^n \times \mathcal{G}$ such that $x^0_v = \Psi(g^0_v, x^0_{q_1})$, let $r : \operatorname{dom} r \to \mathbb{R}^n$ describe the desired trajectory of the nominal motion plan v, that is:

$$r(t,j) = \begin{cases} \Psi(g^0_v, x_{q_1}(t)) & \text{if } t \in [0, T_1], \\ & \text{and } j = 0 \\ \Psi(g^0_v \exp(\xi_{q_1} T_1) g'_1, x_{q_2}(t)) & \text{if } t \in [T_1, T_2], \\ & \text{and } j = 1 \\ \Psi(g^0_v \exp(\xi_{q_1} T_1) g''_1, x_{q_3}(t)) & \text{if } t \in [T_2, T_3], \\ & \text{and } j = 2 \\ \vdots & \vdots \\ \Psi(g^0_v \exp(\xi_{q_1} T_1) g''_1 \ldots & \text{if} \\ \ldots \exp(\xi_{q_{k-1}} T_{k-1}) g''_k, x_{q_k}(t)) & t \in [T_{k-1}, T_k], \\ & \text{and } j = k \ , \end{cases}$$

where x_{q_1} is the trim trajectory with $(\xi_{q_1}, x^0_{q_1}) \in \mathcal{T}(\mathcal{P}, \mathcal{G})$, x_{q_2} is the maneuver with $(\beta_{q_2}, x^0_{q_2}, g_{q_2}, T_{q_2}) \in \mathcal{M}(\mathcal{P}, \mathcal{G})$, etc. Note that each jump of r corresponds to a change of motion primitive. For example, for each $(t, j) \in [0, T_1] \times \{0\}$, $r(t, j)$ is given by the q_1-th trim trajectory, and after the jump at $t = T_1, j = 0$, and for all $(t, j) \in [T_1, T_2] \times \{1\}$, $r(t, j)$ is given by the q_2-th maneuver. The duration of the motion plan v is $T_r = \sum_{i=1,3,\ldots,k} T_i + \sum_{i=2,4,\ldots,k-1} T_{q_i}$. When T_r is finite, $\operatorname{dom} r$ is a subset of $[0, T_r] \times \{0, 1, 2, \ldots, k-1\}$, while when T_r is infinite, $\operatorname{dom} r$ is a subset of $[0, \infty) \times \{0, 1, 2, \ldots, k-1\}$.

Theorem 4.1: (nominal execution) Let Assumptions 3.1 and 3.3 hold. For each $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$ with nominal motion plan trajectory r and each $(x^0_v, g^0_v) \in \mathbb{R}^n \times \mathcal{G}$ such that $x^0_v = \Psi(g^0_v, x^0_{q_1})$, $(\xi_{q_1}, x^0_{q_1}) \in \mathcal{T}(\mathcal{P}, \mathcal{G})$, there exists a unique solution φ to $\mathcal{H}_{cl}$ from $\varphi(0,0) = (x^0_v, q_1, 1, g^0_v, 0)$ that is bounded and is such that the x component satisfies $x(t, j) = r(t, j)$ for all $(t, j) \in \operatorname{dom} \varphi$.

Remark 4.2: Theorem 4.1, which follows by construction, states that every motion plan $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$ is properly executed by $\mathcal{H}_{cl}$. This result recovers the nominal motion plan execution property of the hybrid automaton in [6]. ■

In addition to the nominal property in Theorem 4.1, the proposed hybrid control construction guarantees that, under the presence of perturbations, motion plan execution stay close to a nominal one. Note that the presence of perturbations in $\mathcal{H}_{cl}$ on the initial conditions, parameters, and/or the state affects the jump times. In this way, the domain of the resulting trajectory does not need to coincide with the domain of the nominal trajectory r associated to $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$. The (T, J, ε)-closeness notion of distance between hybrid arcs in Section II-C handles such a situation.

Theorem 4.3: (perturbation of initial conditions) Let Assumptions 3.1 and 3.3 hold. For each $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$ with nominal motion plan trajectory r and each $(x^0_v, g^0_v) \in \mathbb{R}^n \times \mathcal{G}$ such that $x^0_v = \Psi(g^0_v, x^0_{q_1})$, $(\xi_{q_1}, x^0_{q_1}) \in \mathcal{T}(\mathcal{P}, \mathcal{G})$, each $\varepsilon > 0$, each compact set $K \subset \mathcal{B}_{q_1}$, and each $(T, J) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, $(T, J) \preceq (T_r, k-1)$, there exists $\delta > 0$ such that every solution φ_δ to $\mathcal{H}_{cl}$ with $\varphi_\delta(0,0) = (x^0_\delta, q_1, 1, g^0_v, 0)$, $x^0_\delta \in K + \delta\mathbb{B}$, is bounded and the x component and r are (T, J, ε)-close.

Remark 4.4: The time horizon (T, J) where the closeness property in Theorem 4.3 holds can be picked to be equal to $(T_r, k-1)$ when T_r is finite. Then, closeness between the component x of the solution and r is guaranteed in the entire duration of the motion plan. The hybrid time domain of each solution to $\mathcal{H}_{cl}$ can be extended to an unbounded one without affecting the behavior of the system up to time (T, J). In addition to the regularity properties of the closed-loop system (guaranteed by the standing assumption and the hybrid controller construction), the proof of Theorem 4.3 extends the hybrid time domain to an unbounded one to enable the application of results in [10] for hybrid systems with perturbations. ■

Under the presence of perturbations, system $\mathcal{P}$ controlled by $\mathcal{H}$ can be written as

$$\dot{x} = f(x, \kappa_c(x + d_1(t), q, \tau)) + d_2(t) \ , \qquad (13)$$

where d_1 corresponds to error in the measurements of x and d_2 models other exogenous disturbances and unmodeled dynamics. The addition of these perturbations in the closed-loop system $\mathcal{H}_{cl}$ results in a perturbed hybrid system, denote as $\tilde{\mathcal{H}}_{cl}$, which can be written as

$$\begin{array}{rclr} \dot{\varphi} & = & \tilde{f}(\varphi + d_1(t)) + d_2(t) & \varphi + d_1 \in \tilde{C} \\ \varphi^+ & = & \tilde{g}(\varphi) & \varphi + d_1 \in \tilde{D} \ , \end{array}$$

The following result asserts that the motion planning is robust to a class of perturbations. [2]

Theorem 4.5: (perturbations) Let Assumptions 3.1 and 3.3 hold. For each $v \in \mathcal{V}(\mathcal{P}, \mathcal{G})$ *with nominal motion plan trajectory* r *and each* $(x_v^0, g_v^0) \in \mathbb{R}^n \times \mathcal{G}$ *such that* $x_v^0 = \Psi(g_v^0, x_{q_1}^0)$, $(\xi_{q_1}, x_{q_1}^0) \in \mathcal{T}(\mathcal{P}, \mathcal{G})$, *each* $\varepsilon > 0$, *each compact set* $K \subset \mathcal{B}_{q_1}$, *and each* $(T, J) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, $(T, J) \preceq (T_r, k-1)$, *there exists* $\delta > 0$ *such that every solution* $\tilde{\varphi}$ *to* $\tilde{\mathcal{H}}_{cl}$ *with* $\tilde{\varphi}(0,0) = (x^0, q_1, 1, g_v^0, 0)$, $x^0 \in K + \delta\mathbb{B}$, $|d_1(t,j)| \leq \delta$ *and* $|d_2(t,j)| \leq \delta$ *for each* $(t, j) \in \operatorname{dom} \varphi$, *is bounded and the* x *component and the motion plan trajectory* r *are* (T, J, ε)*-close.*

Remark 4.6: The proof of this result uses a technique from [10, Section V] in which a perturbed hybrid system $\mathcal{H}_{cl}^{\delta}$ is embedded into a set-valued hybrid system. Using the hybrid time domain extension as in Theorem 4.3, the results follows from [10, Corollary 5.5]. ■

Finally, Figure 3 illustrates a solution to $\mathcal{H}_{cl}$ starting nearby the motion plan in Figure 1. This corresponds to a simulation result from a toolbox for robust maneuver-based motion planning, currently under development.

V. Conclusion

We presented a hybrid systems framework for maneuver-based motion planning algorithms for nonlinear systems with symmetries. We systematically described the construction of a hybrid controller and showed its robustness properties for a large class of perturbations. Our results are built upon recent tools for robustness of stability for hybrid systems. Extensions of the hybrid control strategy to situations where bounds on the perturbations are known beforehand follow from the ideas presented in this manuscript and will be closely explored in the future.

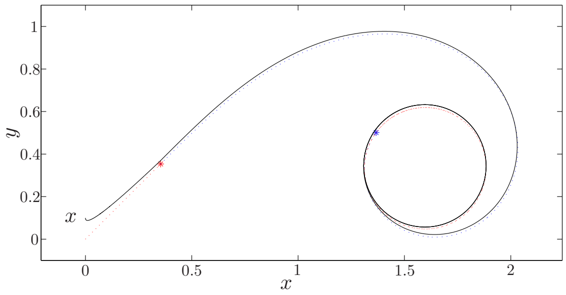


Fig. 3. Motion primitive (dashed) in Figure 1 and simple airplane trajectory resulting from applying our hybrid control strategy for motion planning. Tracking control during trims (red pieces) guarantees that solution and trim trajectory are stay close. Maneuver starts from a point nearby the maneuver (blue piece) in the library and remains close to it.

References

[1] M. Brady. *Robot Motion: Planning and Control*. Mit Press, 1982.

[2] J.-P. Laumond, editor. *Robot Motion Planning and Control*, volume 229 of *Lectures Notes in Control and Information Sciences*. Springer Verlag, 1998.

[3] A. E. Quaid and A. A. Rizzi. Robust and efficient motion planning for a planar robot using hybrid control. In *Proc. ICRA*, 2000.

[4] T. Schouwenaars, B. Mettler, E. Feron, and J.P. How. Robust motion planning using a maneuver automation with built-in uncertainties. In B. Mettler, editor, *Proc. Amer. Control Conf.*, pages 2211–2216, 2003.

[5] S. M. LaValle. *Planning algorithms*. Cambridge University Press, Cambridge, UK, 2006.

[6] E. Frazzoli, M. A. Dahleh, and E. Feron. Maneuver-based motion planning for nonlinear systems with symmetries. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 21:1077–1091, 2005.

[7] V. Gavrilets, E. Frazzoli, B. Mettler, M. Piedmonte, and E. Feron. Aggressive maneuvering of small autonomous helicopters: A human-centered approach. *The International Journal of Robotics Research*, 20:795, 2001.

[8] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25:116–129, 2002.

[9] R. Goebel, J.P. Hespanha, A.R. Teel, C. Cai, and R.G. Sanfelice. Hybrid systems: generalized solutions and robust stability. In *Proc. 6th IFAC Symposium in Nonlinear Control Systems*, pages 1–12, 2004.

[10] R. Goebel and A.R. Teel. Solutions to hybrid inclusions via set and graphical convergence with stability theory applications. *Automatica*, 42(4):573–587, 2006.

[11] R. G. Sanfelice and A. R. Teel. A “throw-and-catch” hybrid control strategy for robust global stabilization of nonlinear systems. In *Proc. 26th American Control Conference*, pages 3470–3475, 2007.

[12] E. Frazzoli, M. A. Dahleh, and E. Feron. Robust hybrid control for autonomous vehicle motion planning. *Proc. 39th IEEE Conference on Decision and Control*, 1:821–826, 2000.

[13] C. C. Chevalley. *Theory of Lie Groups*. Princeton Univ Press, 1946.

[14] J. E. Marsden and T. S. Ratiu. *Introduction to mechanics and symmetry*. Springer New York, 1999.

[15] R.G. Sanfelice, R. Goebel, and A.R. Teel. Generalized solutions to hybrid dynamical systems. *ESAIM: Control, Optimisation and Calculus of Variations*, 2008.

[16] A. Isidori. *Nonlinear control systems: an introduction*. Springer-Verlag New York, Inc. New York, NY, USA, 1989.

[17] F. Esfandiari and H. K. Khalil. Output feedback stabilization of fully linearizable systems. *International Journal of Control*, 56:1007–1037, 1992.

[18] P. Rouchon and J. Rudolph. Invariant tracking and stabilization: Problem formulation and examples. *Lecture notes in control and information sciences*, 246:261–273, 1999.

[2] The exogenous signals d_1 and d_2 are given on hybrid time domains (given a hybrid time domain S and an exogenous signal $d_1(t)$, we can define, with some abuse of notation, $d_1(t,j) := d_1(t)$ for each $(t,j) \in S$.) Solutions to hybrid systems with the perturbations above is understood similarly to the notion of solution outlined in Section II-C.