Global Asymptotic Stabilisation on Lie Groups and Applications

João Carlos Patrício Canas
joaocarloscanas@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal
July 2019

Abstract

Due to recent technological developments, autonomous vehicles have seen an increasing growth in their applications and in their potential to help society. These vehicles, such as mobile robots, underwater vehicles and, more recently, unmanned aerial vehicles (UAVs) are being used to perform a variety of missions. With the need for autonomous systems such as the ones described, comes a need for control systems that allow them to perform their missions in an efficient and safe way while also being robust to perturbations from their environment of operation. One solution to this problem, is the use of hybrid control. In this paper, we propose a control strategy that tackles the problem of global asymptotic stabilisation of a setpoint on Lie groups by means of hybrid control. This can be used for setpoint stabilisation on the special orthogonal group, the special Euclidean group and the circle. These particular examples find applications in the problems of 3-axis gimbal stabilisation, rigid-body stabilisation and orientation tracking for a quadrotor vehicle.

Keywords: Hybrid Control Systems, Geometric Control, Lie Groups, Aerospace

1. Introduction

With the technological developments happening nowadays, autonomous vehicles have seen an increasing growth in their applications. With the need for autonomous systems such as the ones described, comes a need for control systems that allow them to perform their missions in an efficient and safe way while also being robust to perturbations from their environment of operation. Several approaches to the control problem of stabilisation and trajectory tracking have been developed. The most common way is through continuous feedback controllers. Those, however, are limited as it has been shown that if the underlying configuration space is not diffeomorphic to $\mathbb{R}^n$, there exists no continuous state feedback that globally asymptotically stabilises a given configuration. One solution to this problem, is the use of hybrid control as it is proposed in [1]. These controllers allow for closed loop systems which are globally asymptotically stable.

In this paper, we propose a control strategy that tackles the problem of global asymptotic stabilisation of a setpoint on Lie groups by means of hybrid control. In particular, the dynamical model of the system considers that the system state belongs to a Lie Group that is embedded in a higher dimensional Euclidean space, so that the results provided in [2] can be used. To globally asymptotically stabilise a setpoint for this system, we follow a synergistic hybrid feedback approach, which consists on the development of a collection of proper indicators on the Lie group, each of which is associated with a feedback law. Radial unboundedness of each proper indicator and the fact the union of their domains covers the group implies that there exist points in each domain where controller switching leads to a decrease of the associated Lyapunov function. We show that the proposed controller can be applied not only to the problem of globally asymptotically stabilising the position and velocity of the extended dynamical system, but also to the problem of trajectory tracking since Lie group properties allow for the definition of the tracking error as an element of the group itself.

We also show that it is possible to meet the controller design assumptions using proper indicators that are constructed from a smooth atlas of the group. We derive two feedback laws from this construction, the first of which grows unbounded near the boundary of each domain, while the second is bounded during flows of the closed-loop system, possibly allowing to meet actuation constraints.

We show that the proposed controller can be used for setpoint stabilisation on the special orthogonal group, the special Euclidean group and the circle and that these particular examples find applications in the problems of gimbal stabilisation, rigid-body stabilisation and orientation tracking for a quadrotor vehicle.

This work was supported by Fundao para a Cin-
This work is organised as follows. In Section 1.1 we introduce some notation and definitions that will be used throughout the paper.

In Section 2 a description of problems that will be studied is presented. Those are problem structures that may be applied to any Lie group. In Section 3, a solution to those problems is proposed. First, a more standard approach will be proposed which provides stability to the system but does not take actuator limitations into account in the design. Then, a second approach will be introduced now with saturation of the input included.

In the following Sections, examples of real applications of the proposed controllers will be presented. Each section will first introduce the system to be studied, followed by the integration of the hybrid controller. Finally, some results will be presented based on computational simulations. In Section 5, the controller will implemented on a gimbal. Then, in Section 6, we adapt a quad-copter to a vehicle moving in an horizontal plane. In Section 7, a solution for the tracking of a set of landmarks using a fully actuated vehicle will be presented using a Lie Group structure based on [3].

In Chapter 8 we end the paper with some conclusions regarding the obtained results.

2. Problem Setup

In this work, we consider the problem of globally asymptotically stabilising dynamical systems, evolving on an n-dimensional Lie Group $G$, that are properly embedded in $\mathbb{R}^m$, meaning $G$ is a subset of $\mathbb{R}^m$.

A dynamical system evolving on a Lie group $G$ can be first described by

$$\dot{x} = \Pi(x)\xi$$  \hfill (1)

where $x \in G$ is the state of system, $\xi \in \mathbb{R}^n$ is the input and $\Pi : G \mapsto \mathbb{R}^{m \times n}$ is a smooth function whose image at $x$ is the tangent space to $G$. Since $G$ has dimension $n$, it is possible to define a set of parametrizations $\beta_q : U_q \mapsto \mathbb{R}^n$ that map subsets $U_q$ of $G$ to $\mathbb{R}^n$. These maps will be useful more further when defining the hybrid controllers.

**Remark 1.** The dynamical system inherits the Euclidean structure of its ambient space $\mathbb{R}^m$, making it amenable to the analysis tools in [2]. However, these dynamics can be formulated without this additional structure as in [4, Eq. (10)].

Without loss of generality, we assume that the setpoint to stabilise for the closed-loop system is the identity element of the given Lie group $G$, which we denote by $e \in G$. Since global asymptotic stabilisation of a setpoint on a Lie group by means of continuous feedback is not possible in general due to topological obstructions, we resort to a hybrid controller as shown in the following problem statement.

**Problem 1.** Given the dynamical system (1), design a hybrid controller with state $q \in Q$ satisfying

$$q' \in F_q(q, x) \quad (q, x) \in C_1$$

$$q^+ \in G_q(q, x) \quad (q, x) \in D_1$$

and a feedback law $(q, x) \mapsto \kappa(q, x)$ that renders the set

$$A_1 := \{(q, x) \in Q \times G : x = e\}$$

globally asymptotically stable for the closed-loop hybrid system.

In addition to Problem 1, we also address the more complex problem of globally asymptotically stabilising a setpoint in a Lie group considering than the input is the velocity, rather than the velocity, as described next. The second model is a dynamical extension of (1) with a different input and its dynamics are described by

$$\dot{x} = \Pi(x)\xi$$

$$\dot{\xi} = u$$  \hfill (4)

where $x \in G$, $\xi \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ is the input. In this case, we reformulate Problem 1 as follows.
Problem 2. Given the dynamical system (4), design a hybrid controller with state $q \in Q$ satisfying

\begin{align*}
\dot{q} & \in F_{q}(q, x, \xi) \quad (q, x, \xi) \in C_{2} \\
q^{+} & \in G_{q}(q, x, \xi) \quad (q, x, \xi) \in D_{2}
\end{align*}

and a feedback law $(q, x, \xi) \mapsto \kappa_{2}(q, x, \xi)$ that renders the set

$$
A_{2} := \{(q, x, \xi) \in Q \times G \times \mathbb{R}^{n} : x = e, \, \xi = 0, n \}\}
$$

globally asymptotically stable for the closed-loop hybrid system.

In the next section, we present hybrid controllers that solve both Problems 1 and 2.

3. Controller Design

In this section, we present hybrid controllers for global asymptotic stabilisation of systems (1) and (4). To achieve the desired goal, we follow a synergistic hybrid feedback approach. More specifically, we devise a collection of radially unbounded functions on subsets of $G$ and show that it is possible to globally asymptotically stabilise the desired set-point through appropriate switching between different gradient-based vector fields of the functions in the given collection. To construct this collection of functions we resort to the concept of proper indicator that is presented next.

Definition 1 ([2, Definition 7.9]). Let $U$ be an open subset of $G$. A function $V : U \to \mathbb{R}_{\geq 0}$ is a proper indicator on $U$ if it is continuous and, for a sequence $\{x_{i}\}_{i \in \mathbb{N}}, V(x_{i}) \to \infty$ when $i \to \infty$ if either $|x_{i}| \to \infty$ or the sequence $\{x_{i}\}_{i \in \mathbb{N}}$ approaches the boundary of $U$. Let $A \subset U$ be a compact set. A function $V : U \to \mathbb{R}_{\geq 0}$ is a proper indicator of $A$ on $U$ if it is a proper indicator on $U$ and $V(x) = 0$ if and only if $x \in A$.

Assumption 1. Let $Q \subset \mathbb{N}$ be a finite set and \{U_{q}\}_{q \in Q} be a collection of open subsets of $G$ satisfying

$$
\bigcup_{q \in Q} U_{q} = G.
$$

For each $q \in Q$, there exists a continuously differentiable proper indicator $V_{q}$ of $e$ on $U_{q}$ such that

$$
\Pi(x)\nabla V_{q}(x) = 0_{k}
$$

if and only if $x = e$.

In Assumption 1, it is required that potential functions on $G$ exist for each one of the charts that covers the group and, additionally, their gradient must be orthogonal to the tangent space to $G$ at $x$ if and only if $x = e$. It is implicitly assumed in Assumption 1 that the identity $e \in G$ belongs to the domain of $V_{q}(x)$ for each $q \in Q$. In the sequel, we make use of Assumption 1 in the design of the hybrid controllers in Problems 1 and 2.

3.1. Controller Design for Position Stabilisation

Under Assumption 1, we are able to define the function

$$
W_{1}(q, x) := \begin{cases} 
V_{q}(x) & \text{if } (q, x) \in \mathcal{W} \\
+\infty & \text{otherwise}
\end{cases}
$$

for each $(q, x) \in Q \times G$ where

$$
\mathcal{W} := \{(q, x) \in Q \times G : x \in U_{q}\}.
$$

Using (9), we define the hybrid controller

$$
\dot{q} = 0,
$$

$$(q, x) \in C_{1} := \{(q, x) \in G \times Q : \mu_{1}(q, x) \leq \delta\},$$

$$(q, x) \in D_{1} := \{(q, x) \in G \times Q : \mu_{1}(q, x) \geq \delta\}$$

where $\delta > 0$ and

$$
\mu_{1}(q, x) := W_{1}(q, x) - \min_{p \in Q} W_{1}(p, x)
$$

for each $(q, x) \in Q \times G$. Under Assumption 1, the relations $\mathcal{W}, \mu_{1}$ and $g_{1}$, given in (9), (11) and (10), respectively, are endowed with some regularity properties that are important to the robustness of the closed-loop hybrid system.

Lemma 1. Let Assumption 1 hold. Then, the following also hold:

1. The function (9) is continuous and positive definite relative to $A_{1}$;
2. $\nu_{1}(x) := \min_{p \in Q} W_{1}(p, x) < +\infty$ for each $x \in G$;
3. The function $\mu_{1}$ in (11) is continuous;
4. The function $g_{1}$ in (10) is outer semicontinuous.

Defining the feedback law

$$
\xi = \kappa_{1}(q, x) := -k_{x}\Pi(x)^{\top}\nabla V_{q}(x)
$$

for each $(q, x) \in \mathcal{W}$, with $k_{x} > 0$, the interconnection between the system (1) and the controller defined in (10) is the closed-loop hybrid system $\mathcal{H}_{1} := (C_{1}, F_{1}, D_{1}, G_{1})$ given by

$$
(q, x) \in F_{1}(q, x), \quad (q, x) \in C_{1}
$$

$$(q^{+}, x^{+}) \in G_{1}(q, x), \quad (q, x) \in D_{1}
$$

where $C_{1}$ and $D_{1}$ are given in (10) and

$$
F_{1}(q, x) := (0, \Pi(x)\kappa_{1}(q, x)) \quad \forall (q, x) \in C_{1}
$$

$$(g_{1}(q, x), x) \quad \forall (q, x) \in D_{1}
$$

We make use of Lemma 1 to show that the closed-loop hybrid system satisfies the so-called hybrid basic conditions which are pivotal for well-posedness and asymptotic stability of $A_{1}$.
Lemma 2. Let Assumption 1 hold. Then the following hold:

(A1) $C_1$ and $D_1$ are closed;

(A2) $F_1$ is outer semicontinuous relative to $C_1$, locally bounded relative to $C_1$ and $F_1(q, x)$ is convex for each $(q, x) \in C_1$;

(A3) $G_1$ is outer semicontinuous relative to $D_1$ and locally bounded relative to $D_1$.

The hybrid basic conditions (A1)-(A3) as defined in [2, Assumption 6.5] ensure that the closed-loop hybrid system is endowed with robustness to measurement noise and that one may resort to invariance principles for hybrid system such as the ones invoked in the next result.

Theorem 1. Let Assumption 1 hold. Then, the set $A_1$ in (3) is globally asymptotically stable for the closed-loop hybrid system (13).

3.2. Controller Design for Position and Velocity Stabilisation

In order to tackle Problem 2, we follow the same principles of synergistic hybrid feedback that were used in Section 3.1. In this direction, we make use of (9) to define

$$W_2(q, x, \xi) := k_x W_1(x, q) + \frac{1}{2} x^T \xi$$

for each $(q, x, \xi) \in W_2$, and with $k_x > 0$, where

$$W_2 := W_1 \times \mathbb{R}^n.$$

To globally asymptotically stabilise $(e, 0)$ for (4), we define the hybrid controller

$$
\begin{align*}
\dot{q} &= 0, \\
(q, x, \xi) &\in C_2 \\
q^+ &\in g_2(x, \xi) \coloneqq \arg \min \{W_2(q, x, \xi) | q \in Q\}, \\
(q, x, \xi) &\in D_2
\end{align*}
$$

where

$$
\begin{align*}
C_2 &:= \{(q, x, \xi) \in Q \times G \times \mathbb{R}^n : \mu_2(q, x, \xi) \leq \delta\} \\
D_2 &:= \{(q, x, \xi) \in Q \times G \times \mathbb{R}^n : \mu_2(q, x, \xi) \geq \delta\}
\end{align*}
$$

with $\delta > 0$ and

$$
\mu_2(q, x, \xi, p) := W_2(q, x, \xi) - \min_{p \in Q} W_2(q, x, \xi, p)
$$

for each $(q, x, \xi) \in Q \times G \times \mathbb{R}^n$.

Similarly to Lemma 1, we show that the functions $W_2$, $\mu_2$ and $g_2$ given in (15), (18) and (16), respectively, are endowed with the same regularity properties that also hold for $W_1$, $\mu_1$ and $g_1$, as shown next.

Lemma 3. Let Assumption 1 hold. Then, the following also hold:

1. The function (15) is continuous and positive definite relative to $A_2$;

2. $\min_{p \in Q} W_2(x, \xi, p) < +\infty$ for each $(x, \xi) \in G \times \mathbb{R}^n$;

3. The function $\mu_2$ in (11) is continuous;

4. The function $g_2$ in (10) is outer semicontinuous.

Using the feedback law

$$u = \kappa_2(q, x, \xi) := -k_x \Pi(x)^T \nabla V_q(x) - k_\xi \xi$$

for each $(q, x, \xi) \in W_2$ with $k_\xi > 0$, the interconnection between the system (4) with the controller defined in (16) and (19) is the closed-loop hybrid system $H_2 := (C_2, F_2, D_2, G_2)$ given by

$$
\begin{align*}
(q, \dot{x}, \dot{\xi}) &\in F_2(q, x, \xi) \quad (q, x, \xi) \in C_2 \\
(q^+, \dot{x}^+, \dot{\xi}^+) &\in G_2(q, x, \xi) \quad (q, x, \xi) \in D_2
\end{align*}
$$

where $C_2$ and $D_2$ are given in (16) and

$$
\begin{align*}
F_2(q, x, \xi) &:= (0, \Pi(x) x, \kappa_2(q, x, \xi)) \quad \forall (q, x, \xi) \in C_2 \\
G_2(q, x, \xi) &:= (g(x, \xi)), \forall (q, x, \xi) \in D_2.
\end{align*}
$$

The closed-loop hybrid system (20) also satisfies the hybrid basic conditions, as shown next.

Lemma 4. Suppose that Assumption 1 holds. Then, the following are verified:

(B1) $C_2$ and $D_2$ are closed;

(B2) $F_2$ is outer semicontinuous relative to $C_2$, locally bounded relative to $C_2$ and $F_2(q, x, \xi)$ is convex for each $(q, x) \in C_2$;

(B3) $G_2$ is outer semicontinuous relative to $D_2$ and locally bounded relative to $D_2$.

From the previous lemma, we are able to derive the following result.

Theorem 2. Let Assumption 1 hold. Then, the set $A_2$ in (6) is globally asymptotically stable for the closed-loop hybrid system (20).

4. Saturated Feedback Controller Design

In this section, we present hybrid controllers for global asymptotic stabilisation of system (1). Unlike the first approach, in the controller defined in this section the output will always be contained within predefined boundaries. Those boundaries can be tuned to fit the limitations of the system to be controlled.
To achieve the desired goal, we again follow a synergistic hybrid feedback approach. More specifically, we devise a collection of radially unbounded functions on subsets of $G$ and show that it is possible to globally asymptotically stabilise the desired setpoint through appropriate switching between different gradient-based vector fields of the functions in the given collection.

Let $\mathcal{G} = \{\beta_q, U_q \mid q \in Q\}$ denote a smooth atlas of $G$. This set of functions $\beta_q : U_q \to \mathbb{R}^n$ map subsets $U_q$ of $G$ which together cover the whole Lie group. Under Assumption 1, we define the function

$$V_q(x) = \frac{1}{2}|\beta_q(x) - \beta_q(e)|^2 \quad (22)$$

for each $q \in Q$ and $x \in U_q$. Taking (9), we are able to rewrite $W_1$ as

$$W_1(q, x) := \begin{cases} \frac{1}{2}|\beta_q(x) - \beta_q(e)|^2 & \text{if } (q, x) \in \mathcal{W} \\ +\infty & \text{otherwise} \end{cases} \quad (23)$$

for each $(q, x) \in Q \times G$.

With this defined, we resort on the following result to define the control law.

**Theorem 3.** Let Assumption 1 hold and let $V_q$ be defined by (22). Then, $\mathcal{D}_x \beta_q(x) \Pi(x)$ is a non-singular matrix in $\mathbb{R}^{n \times n}$ for all $(q, x) \in \mathcal{W}$.

To guarantee that the controller to be presented is bounded, we make the following definition.

**Definition 2.** We define a scalar saturation function $l_b(x) : \mathbb{R} \to [-b, b], b \in \mathbb{R}^+$ with the following properties:

1. $l_b(x)$ is continuously differentiable for each $x \in \mathbb{R}$;
2. $l_b(0) = 0$;
3. $l_b(x) > 0$ for each $x \in \mathbb{R} \setminus \{0\}$.

Using this definition, we define the function $\sigma_b(x) : \mathbb{R}^k \to \mathbb{R}^k$ as

$$\sigma_b(x)_i = l_b(x_i) \quad (24)$$

for $i \in \{1, \ldots, k\}$ and $x \in \mathbb{R}^k$.

Under assumption 1, and we define the feedback law

$$\xi = s_b(q, x) := -k_x(\mathcal{D}_x \beta_q(x) \Pi(x))^{-1} \sigma_b(\beta_q(x) - \beta_q(e))$$

for each $(q, x) \in \mathcal{W}$ and with $k_x > 0$, the interconnection between the system (1) and the controller defined in (10) is the closed-loop hybrid system $H_b := (C_1, F_b, D_1, G_1)$ given by

$$(\dot{q}, \dot{x}) \in F_b(q, x) \quad (q, x) \in C_1$$

$$q^+, x^+ \in G_1(q, x) \quad (q, x) \in D_1$$

where $C_1$ and $D_1$ are given in (10) and

$$F_b(q, x) := (0, \Pi(x) \kappa_b(q, x)) \quad \forall (q, x) \in C_1$$

$$G_1(q, x) := (g_1(x), x) \quad \forall (q, x) \in D_1$$

(27)

Similar to the previous chapter, we make use of Lemma 2 to show that the closed-loop hybrid system satisfies the so-called hybrid basic conditions which are pivotal for well-posedness and asymptotic stability of $A_1$. From this Lemma, we conclude that the system verifies conditions (A1) and (A3). The following result guarantees that this system also verifies the second hybrid condition.

**Lemma 5.** Let Assumption 1 hold. Then, $F_1$ is outer semicontinuous relative to $C_1$, locally bounded relative to $C_1$ and $F_b(q, x)$ is convex for each $(q, x) \in C_1$.

Under Lemmas 2 and 5, we will be able to derive the next result. The theorem that follows is very similar to Theorem 1 so on its proof we will focus on the main differences.

**Theorem 4.** Let Assumption 1 hold. Then, the set $A_1$ in (3) is globally asymptotically stable for the closed-loop hybrid system (26).

With the stability of the closed-loop guaranteed, the limitations of $\kappa_b$ will be shown for each application that is considered.

In following sections, we demonstrate how the controllers presented in sections 3 and 4 can be used in applications of interest.

5. **Gimbal Stabilisation**

In this section, we address the problem of globally asymptotically stabilising a trajectory for a gimbal using the controllers presented in Chapters 3 and 4. The dynamics of this system can be written as

$$\dot{R} = RS(\omega_b)$$

where $R \in SO(3) := \{R \in \mathbb{R}^{3 \times 3} : R^T R = I_3, \det(R) = 1\}$ is a rotation matrix that maps vectors in a body-fixed frame to the inertial frame and represents the state of the gimbal and $S(v)$ is given by

$$S(v) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

(29)

for each $v = [v_1 \ v_2 \ v_3]^T \in \mathbb{R}^3$. The vector $\omega_b \in \mathbb{R}^3$ is the angular velocity vector in the body frame, which is considered as an input. Let $\omega_d(t) \in \mathbb{R}^3$ be the desired angular velocity defined for all $t \geq 0$, $R_d(t) \in SO(3)$ be a trajectory to be followed according to

$$\dot{R}_d(t) = R_d(t)S(\omega_d(t))$$

(30)
and $R_e = R_d^T R$ be the tracking error. With this definition, and using (28), (30) and the property
\[ RS(\omega) = S(R_e)R \]
we write the dynamics of this error as
\[ \dot{R}_e = R_e S(\omega_b - R_e^T \omega_d) = R_e S(\omega_c) \]
with $\omega_c = \omega_b - R_e^T \omega_d$.

Under the previous assumptions, the system (32) can be written in the form (1) with $x := \text{vec}(R_e)$, $\xi = \omega_c$ and
\[ \Pi(x) = - \begin{bmatrix} R_e(x) S(e_1) \\ R_e(x) S(e_2) \\ R_e(x) S(e_3) \end{bmatrix} \]
for each $R_e(x) \in SO(3)$. We define the identity as $e = \text{vec}(I_3)$.

To meet Assumption 1, we construct two different atlas of $SO(3)$ in the next section.

5.1. Finite Atlas of $SO(3)$

In order to meet Assumption 1, we first define a potential function $V_q$ as
\[ V_q(x) = \frac{1}{2} \beta_q(x) - \beta_q(\text{vec}(I_3)) \]
for each $x \in U_q$, with $q \in Q := \{0, 1, 2, 3\}$ and such that $V_q$ is a proper indicator of $I_3$ on $U_q$. The function $\beta_q$ together with $U_q$ form a maximal atlas of $SO(3)$. To complete this maximal atlas of $SO(3)$, let $U_q$ be defined by:
\[ U_0 = U \]
\[ U_q = \{ R_e(x) \in SO(3) : R(e_q, \pi/2) R_e(x) \in U \}, \quad q \in \{1; 2; 3\} \]
where $U := \{ R_e(x) \in SO(3) : (I_3 + R_e(x)) \text{ is non-singular} \}$ and $R(v, \alpha) \in SO(3)$ is a rotation matrix of an angle $\alpha$ around the axis defined by the unity vector $v$.

Two functions $\beta_{1,q}$ and $\beta_{2,q}$ will be defined in the following sections.

5.1.1 Cayley Transform

For each $x \in U_q$ we define $\beta_{1,q}(x) : SO(3) \rightarrow \mathbb{R}^3$ as
\[ \beta_{1,q}(x) = \begin{cases} S^{-1}(C^{-1}(R_e(x))), & \text{if } q = 0 \\ S^{-1}(C^{-1}(R(e_q, \pi/2) R_e(x))), & \text{if } q \in \{1; 2; 3\} \end{cases} \]
where $S^{-1}$ is the inverse of (29) and $C^{-1}(M)$ is the inverse Cayley transformation given by
\[ C^{-1}(M) = (I_3 + M)^{-1}(I_3 - M) \]
for $M \in U$.

5.1.2 Logarithmic Transform

For each $x \in U_q$, we define the function $S_L : Q \times SO(3) \rightarrow \mathbb{R}^3$ as
\[ S_L(q, R_e) = \begin{cases} S^{-1}(\ln R_e), & \text{if } q = 0 \\ S^{-1}(\ln (R(e_q, \pi/2) R_e))), & \text{if } q \in \{1; 2; 3\} \end{cases} \]
for each $R_e \in SO(3)$ and $q \in Q$ where $\ln R_e$ represents the natural logarithm of the matrix $R_e$.

We also define the function $M_\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ as
\[ M_\pi(v) = \frac{v}{\pi - |v|} \]
for $v \in \{v \in \mathbb{R}^3 : |v| < \pi\}$. Using this function, we then define $\beta_{2,q} : SO(3) \times Q \rightarrow \mathbb{R}^3$ as
\[ \beta_{2,q}(x) = M_\pi(S_L(q, R_e(x))) \]
for $(q, x) \in W$.

This closed-loop system can be applied to a gimbal mechanism both in simulation and real environments. In the following section, we show how this controller can be implemented.

5.2. System adaptation

The control laws previously described previously can be tested using an iFlight G40 3 Axis Aerial Gimbal available in the Dynamic Systems and Oceanics Laboratory of the Institute for Systems and Robotics.

Each motor controls one Euler angle in the order $\lambda = [\theta \ \phi \ \psi]^T$ (pitch, roll, yaw). The rotation matrix $R \in SO(3)$ used as the state of the system, is defined by
\[ R(\lambda) = R(e_3, \psi) R(e_2, \phi) R(e_1, \theta) \]
as a function of the Euler angles.

The orientation of this gimbal can be controlled by angular velocity commands $\lambda \in \mathbb{R}^3$ to three motors. However, the angular velocity of the gimbal motors $\lambda$ is not always coincident with $\omega_b$ thus requiring a transformation before being applied which is given by
\[ \dot{\lambda} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ \sin \theta \tan \phi & 1 & \cos \theta \tan \phi \\ \cos \phi & 0 & \cos \phi \end{bmatrix} \omega_b \]
for all $\lambda : \cos \phi \neq 0$.

6. Trajectory Tracking for a Quadrotor

The application of the controllers proposed in Sections 3 and 4 is the stabilisation of the orientation of a quadrotor while it is tracking a trajectory. In
such vehicle, as it will be explained later, if we want to stabilise its position, there is only one degree of freedom, expressed as rotation around a fixed axis. For this reason, the hybrid controller presented will assume a system evolving in

\[ SO(2) \]

for this reason, the hybrid controller presented will assume a system evolving in \( SO(2) \) with the following dynamics

\[ \dot{R}_z = R_z S_{2D}(\omega) \]  

(43)

where \( R_z \in SO(2) := \{ R_z \in \mathbb{R}^{2 \times 2} : R_z R_z^\top = I_2, \det(R) = 1 \} \) and we define \( S_{2D} : \mathbb{R} \rightarrow \mathbb{R}^{2 \times 2} \) as

\[ S_{2D}(\omega) = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}. \]  

(44)

With this dynamics, the system (43) can be written in the form (1) with \( x := \text{vec}(R_z) \) and

\[ \Pi(x) = [R_{12}(x) \ R_{22}(x) \ -R_{11}(x) \ -R_{21}(x)]^\top \]  

(45)

for each \( R_z \in SO(2) \). We define the identity as the matrix \( I_2 \) leading to \( A_1 = \{ I_2 \} \).

To meet Assumption 1, we construct an atlas of \( SO(2) \) in the sequel.

6.1. Finite Maximal Atlas of \( SO(2) \)

We define a set of functions \( \beta_q \) that for each \( q \in Q := \{0, 1\} \), map subsets \( U_q \) of \( SO(2) \) to \( \mathbb{R} \). This set of functions \( \beta_q \) together with \( U_q \) form a maximal atlas of \( SO(2) \) mapping the entire group.

A rotation matrix \( R_z \in SO(2) \) can be written as function of the angle of rotation \( \alpha \in \mathbb{R} \) according to

\[ R_z(\alpha) = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \]  

(46)

which allows us to write

\[ \tan \alpha = \frac{\sin \alpha}{\cos \alpha} = \frac{R_{21}}{R_{11}} \]

which is a function of the values of \( R_z \). To get a value of \( \alpha_q \) from this, we use the arc-tangent function \( \text{atan} : \mathbb{R} \rightarrow \begin{bmatrix} -\pi/2, \pi/2 \end{bmatrix} \), together with the signals of \( \sin \alpha \) and \( \cos \alpha \) to determine \( \alpha \) such that

\[ \begin{cases} \alpha_q \in \begin{bmatrix} -\pi/2, \pi/2 \end{bmatrix}, & q = 0 \\ \alpha_q \in \begin{bmatrix} -3\pi/2, \pi/2 \end{bmatrix}, & q = 1 \end{cases} \]  

(47)

This leads to the function \( \text{atan2}(M, q) : SO(2) \times Q \rightarrow \mathbb{R} \) defined as

\[ \text{atan2}(R_z, 0) = \begin{cases} \text{atan}(\tan \alpha) & , R_{11} > 0 \\ \text{atan}(\tan \alpha) + \pi & , R_{11} < 0 \end{cases} \]  

\[ \frac{\pi}{2} & , R_{11} = 0 \]

\[ \text{atan2}(R_z, 1) = \begin{cases} \text{atan}(\tan \alpha) & , R_{11} > 0 \\ \text{atan}(\tan \alpha) - \pi & , R_{11} < 0 \end{cases} \]  

\[ -\frac{\pi}{2} & , R_{11} = 0 \]

Based on this function, using \( \alpha_q(R_z) = \text{atan2}(R_z, q) \), for each \( (x, q) \in U_q \times Q \), \( \beta_q(x) : SO(2) \rightarrow \mathbb{R} \) is given by

\[ \beta_q(\alpha_q(x)) = \begin{cases} \frac{\alpha_0(x)}{(\alpha_0(x) + \frac{\pi}{2})(\frac{\pi}{2} - \alpha_0(x))}, & q = 0 \\ \frac{\alpha_1(x)}{(\alpha_1(x) + \frac{\pi}{2})(\frac{\pi}{2} - \alpha_1(x))}, & q = 1 \end{cases} \]  

(48)

with \( U_q \) being defined as

\[ U_0 = \{ R_z \in SO(2) : R_{12} \neq 1 \} \]  

\[ U_1 = \{ R_z \in SO(2) : R_{21} \neq 1 \}. \]  

(49)

In the following section we demonstrate how the controllers from Sections 3 and 4 can be used to stabilise a quadrotor.

6.2. System Adaptation

We first start by presenting the dynamics of a quadrotor. A quadrotor is an aerial vehicle consisting on four motors placed in square around a central platform. The last, is coincident with the vehicle’s center of mass. By controlling the four motors separately, it is possible to generate torques in all directions along with a total thrust in the vertical axis. This vehicle was modeled by the following equations of motion

\[ \begin{cases} \dot{p} = v_i \\ \dot{v}_i = g - \frac{T}{m} R e_3 \end{cases} \]  

(50)

according to [5] where \( p \in \mathbb{R}^3 \) is the position of the vehicle, \( v_i \in \mathbb{R}^3 \) is its velocity both in the inertial frame, \( R \) is the rotation matrix that rotates vectors from the body fixed frame to the inertial frame, \( g \) is the gravity acceleration vector in the inertial frame and \( m \in \mathbb{R}^+ \) is the mass of the vehicle. In this model, we assume that aerodynamic forces are not significant.

The four inputs that the system receives are the total thrust \( T \) and the angular velocity \( \omega_b = [\omega_1 \ \omega_2 \ \omega_3]^\top \), represented in the body fixed frame.

To control the position of the vehicle, a continuous saturated was implemented to compute the total thrust \( T \) and the angular speeds \( \omega_1 \) and \( \omega_2 \) required to grant accelerations \( a \) in the inertial frame. With the trajectory tracking done by this control law, we need to write the dynamics of the rotation around the thrust axis so that we can control it with \( \omega_3 \).

We define \( r_d \in \mathbb{R}^3 : |r_d| = 1, e^\top r_d = 0 \) as the desired orientation of the front of the quadrotor in the horizontal plane of the inertial frame.
To make the conversion from (50) to an angular movement around the thrust axis controlled by $n_z$, let us define $r_1 = Re_1$ as the unit vector pointing to the front of the quadrotor, represented in the inertial frame.

We also define a reference $r_{1d}$ to be stabilised as

$$r_{1d} = -\frac{S^2(r_3)r_d}{|S(r_3)r_d|}$$

which is a projection of the vector $r_d$ in the plane orthogonal to the thrust vector.

Stabilising $r_1$ to $r_{1d}$ is equivalent to stabilising the vector $e_1$ to $e_{1d} = R^T r_{1d}$, both represented in the body fixed frame. From (51), we can write $e_{1d}$ as

$$e_{1d} = R^T - \frac{S^2(r_3)r_d}{|S^2(r_3)r_d|} = -\frac{S^2(e_3)R^T r_d}{|S^2(e_3)R^T r_d|}$$

resulting in its dynamics being

$$\dot{e}_{1d} = S(e_{1d}) \frac{-S(e_{1d})S(e_3)S(R^T r_d) - S(\bar{\omega}h)R^T r_d e_3}{|S^2(e_3)R^T r_d|}$$

From (52), we can derive that $e_3$ and $e_{1d}$ are normal resulting in $e_3 e_{1d} = 0$. With this property, we can write

$$S(e_{1d})S(e_3) = e_3 e_{1d}^T - e_{1d}^T e_1 I_3 = e_3 e_{1d}^T$$

and then simplify (53) as

$$\dot{e}_{1d} = S(e_{1d})e_3 e_{1d}^T \bar{\omega}$$

thus having the form

$$\dot{\bar{\omega}} = S(e_{1d})e_3 e_{1d}^T$$

with $\bar{\omega}$ given by

$$\bar{\omega} = \frac{\bar{\omega}h^T (S(r_d)\bar{r}_d) + S^2(r_d)R\bar{v}_b}{r_d S^2(r_3)r_d}$$

as a function of the system state (through $R$ and $\bar{\omega}h$) and the reference $r_d$ and its derivative $\bar{r}_d$.

With this reference, we define the matrix $R_z \in SO(2)$ as

$$R_z = \begin{bmatrix} c_z & -s_z \\ s_z & c_z \end{bmatrix}$$

with $c_z$ and $s_z$ defined as

$$c_z = e_{1d}^T e_{1d}$$

$$s_z = -e_{1d}^T e_{1d}$$

which represent the cosine ($c_z$) and sine ($s_z$) of the oriented angle from $e_1$ to $e_{1d}$. The identity of this space corresponds to states where $e_1 = e_{1d}$. The derivatives of those two parameters can be obtained as a function of $\dot{e}_{1d}$ being given by

$$\dot{c}_z = e_{1d}^T \dot{e}_{1d} = -s_z \bar{\omega}$$

$$\dot{s}_z = -e_{1d}^T \dot{e}_{1d} = c_z \bar{\omega}$$

after using (56), resulting in the following dynamics for $R_z$

$$\dot{R}_z = R_z S_{2D}(\bar{\omega})$$

as proposed in (50). Finally, we can rearrange $\bar{\omega}$ as a function of $\omega_3$ by taking (57) and rewrite it as

$$\bar{\omega} = \frac{e_{1d}^T S(e_3)R^T \bar{r}_d - e_{1d}^T S(e_3)S(\bar{\omega}h)R^T r_d}{|S^2(e_3)R^T r_d|}$$

which leads to

$$\bar{\omega} = \frac{r_d^T S(r_3)r_d - r_d^T \bar{\omega}_12 r_d r_d}{r_d^T S^2(r_3)r_d} + \omega_3$$

having a component dependent on the reference $r_d$ and the system state through $R$ (and $r_3$) and the angular velocity component $\omega_12$ controlled by the inner loop defined previously and a component $\omega_3$ which is where the hybrid controller will actuate thus concluding the definition of the system dynamics in $SO(2)$. When computing the input $\omega_3$ of the system, we do so according to

$$\omega_3 = \bar{\omega} - \frac{r_d^T S(r_3)r_d - r_d^T \bar{\omega}_12 r_d r_d}{r_d^T S^2(r_3)r_d}$$

using the results from the $\omega_12$ controller together with $\bar{\omega}$.

### 7. Trajectory tracking based on Landmarks

The third situation considered in this work, is the application of the controllers proposed in sections 3 and 4 to trajectory tracking based on a set of landmarks. For this case, we consider a generic, fully actuated vehicle moving in $SE(3) := (R, p) \in SO(3) \times \mathbb{R}^3$. Its position and orientation are described by

$$\dot{R} = RS(\bar{\omega}h)$$

$$\dot{p} = R\bar{v}_b$$

where $R \in SO(3)$ is the rotation matrix that rotates vectors from the body fixed frame to the inertial frame, $p \in \mathbb{R}^3$ is the position of the vehicle and $\bar{\omega}_h, v_b \in \mathbb{R}^3$ are its angular and linear velocities respectively in body fixed coordinates. It will also be useful to write the position and orientation and their dynamics as

$$M = \begin{bmatrix} R & p \\ 0_3 & 1 \end{bmatrix}, \quad \dot{M} = M \begin{bmatrix} S(\bar{\omega}h) \\ v_b \end{bmatrix}$$
to have a similar structure to the landmarks definition as it be developed forward. After an appropriate transformation that will be presented forward, we define and error matrix and respective dynamics with time as

\[ M_e = \begin{bmatrix} R_e & 0_e \\ 0_e^T & 1 \end{bmatrix}, \quad \dot{M}_e = M_e \begin{bmatrix} S(\omega_e) & v_e \\ 0_3 & 0 \end{bmatrix} \]

(69)

with \( \omega_e, v_e \in \mathbb{R}^3 \).

With these dynamics, such system can be written in the form depicted in (1) with \( x = [\text{vec}(R_e)^T \ p_e^T]^T \) and \( \xi = [\omega_e^T \ v_e^T]^T \). With this definition, \( \Pi(x) \) is given by

\[ \Pi(x) = \begin{bmatrix} -R_e(x)S(e_1) & 0_{3 \times 3} \\ -R_e(x)S(e_2) & 0_{3 \times 3} \\ -R_e(x)S(e_3) & 0_{3 \times 3} \end{bmatrix} \]

(70)

allowing us to apply the controllers presented in Sections 3 and 4.

7.1. Finite Atlas of \( SE(3) \)

We first define a potential function \( V_q : \mathbb{R}^{12} \mapsto \mathbb{R} \) as

\[ V_q(x) = \frac{1}{2} | \beta_q(x) - \beta_q(e) |^2 \]

(71)

where \( e = [\text{vec}(I_3)^T \ 0_3^T]^T \) for each \( x \in U_q \), with \( q \in Q := \{0, 1, 2, 3\} \) and such that \( V_q \) is a proper indicator of \( x_0 \) on \( U_q \). The function \( \beta_q \) together with \( U_q \) form a finite atlas of \( SO(3) \). To complete this atlas of \( SE(3) \), let \( U_q \) be defined by:

\[
\begin{cases}
U_0 = U \\
U_q = \{x \in SE(3) : R(e_q, \pi/2)R_e(x) \in U\}, \\
q \in \{1; 2; 3\}
\end{cases}
\]

(72)

where \( U := \{x \in SE(3) : (I_3 + R_e(x)) \text{ is non-singular}\} \) and \( R(v, \alpha) \in SO(3) \) is a rotation matrix of an angle \( \alpha \) around the axis defined by the unity vector \( v \).

Similarly to Chapter 5, two functions \( \beta_{1,q} \) and \( \beta_{2,q} \) will be defined based on the parametrizations used for \( SO(3) \).

The function \( \beta_{1,q} \) was based on the Cayley transform from (37) and is defined as

\[
\beta_{1,q}(x) = \begin{cases} 
S^{-1}(C^{-1}(R_e(x)))^T p_e(x)^T, & \text{if } q = 0 \\
S^{-1}(C^{-1}(R(e_q, \pi/2)R_e(x)))^T p_e(x)^T, & \text{if } q \in \{1; 2; 3\}
\end{cases}
\]

(73)

for each \( (x, q) \in U_q \times Q \). As for \( \beta_{2,q} \), we use

\[
\beta_{2,q}(x) = [M_x(S_L(q, R_e(x)))^T \ p_e(x)^T]^T
\]

(74)

for each \( (x, q) \in U_q \times Q \), with \( M_x \) and \( S_L \) defined in (39) and (38) respectively.

Those parametrizations have a structure that is very similar to the ones used for \( SO(3) \) in Section 5 which makes it easier to prove that it fulfills Assumption 1.

7.2. System Adaptation with a structure similar to Problem 1

First, we define the landmarks position in an inertial frame, with center and orientation chosen arbitrarily.

Assumption 2. There exist \( l \) landmarks in the free space, for \( l \in \mathbb{N} : l \geq 4 \), with position \( p_{L_i} \in \mathbb{R}^3 \) for the \( i \)-th landmark and such that

1. \[ P := [p_{L1} \ldots p_{Li}] \]

(75)

2. \[ \text{rank} \left( \begin{bmatrix} P \\ 1_L \end{bmatrix} \right) = 4 \]

(76)

We will consider that the vehicle can obtain the position of each landmark represented in its body fixed frame, defined as

\[ \begin{bmatrix} L_m \\ 1_L \end{bmatrix} = M^{-1}L = M^{-1}M_0 \begin{bmatrix} P \\ 1_L \end{bmatrix} \]

(77)

where \( M^{-1} \) can be determined by

\[ M^{-1} = \begin{bmatrix} R^\top & -R^\top p \\ 0_3 & 1 \end{bmatrix}. \]

(78)

The trajectory to be followed by the vehicle, is written relatively to the set of landmarks such that the desired measurements \( L_d \in \mathbb{R}^{3 \times l} \) are obtained through

\[ \begin{bmatrix} L_d(t) \\ 1_L \end{bmatrix} = M_d^{-1}(t) \begin{bmatrix} P \\ 1_L \end{bmatrix} \]

(79)

with \( M_d \in SE(3) \) and its dynamics defined as follows

\[ M_d(t) = \begin{bmatrix} R_d & p_d \\ 0_3 & 1 \end{bmatrix}, \quad \dot{M}_d = \begin{bmatrix} S(\omega_d) & v_d \\ 0_3 & 0 \end{bmatrix} \]

(80)

where \( R_d \in SO(3) \) represents the desired attitude of the vehicle relative to the set of landmarks, \( p_d \in \mathbb{R}^3 \) is the desired position relative to the same set and \( \omega_d \in \mathbb{R}^3 \) and \( v_d \in \mathbb{R}^3 \) being the desired angular and linear velocities respectively, written in the the desired body fixed frame.

By comparing (77) and (79), it is easy to conclude that tracking \( L_m \) to \( L_d \) is equivalent to track \( M^{-1}M_0 \) to \( M_d^{-1} \). From this conclusion, we define the tracking error

\[ M_e := M^{-1}M_0M_d = \begin{bmatrix} R_e & p_e \\ 0_3 & 1 \end{bmatrix} \]

(81)
which we want to stabilise in $I_4$.

Under assumption 2, we can define 
\[
\left(\begin{bmatrix} P^\top & 1_L \end{bmatrix}\right)^\dagger
\] as the right inverse of 
\[
\begin{bmatrix} P^\top & 1_L \end{bmatrix}
\] which allows to write $M_e$ as
\[
M_e = \begin{bmatrix} I_{rv} \\ 1_L \end{bmatrix} \begin{bmatrix} P^\top \\ 1_L \end{bmatrix}^\dagger M_d
\] (82)
which is function of the desired position and orientation $M_d$ relative to the landmarks, the measurements $L_{rv}$ made by the vehicle and the known shape of the set of landmarks $P$.

For a first application, we assume the dynamics of the vehicle to be modelled by (67) with inputs $v_b$ and $\omega_b$. Using the structure from (68), we can write $M^{-1}$ and its dynamics as
\[
M^{-1} = \begin{bmatrix} R_e^\top & -R_e^\top p_b \\ 0_3 & 1 \end{bmatrix},
\] (83)
\[
(M^{-1})' = \begin{bmatrix} S(-\omega_h) - v_b \\ 0_3 \\ 1 \end{bmatrix} M^{-1}
\] (84)
which allows to write the dynamics of the error $M_e$ from (81) as
\[
\dot{M}_e = M^{-1}(\dot{M}_0 M_d + M^{-1}(M_0 \dot{M}_d) = M_e \begin{bmatrix} S(\omega_e) & v_e \\ 0_3 & 0 \end{bmatrix}
\] (85)
where $\omega_e = \omega_d - R_e^\top \omega_b$ and $v_e = v_d - R_e^\top v_b - R_e^\top S(\omega_d)p_e$ are the new inputs.

7.3. System Adaptation with a structure similar to Problem 2

In a second application, we assume the dynamics of the vehicle to be modelled by (67) with the additional two equations
\[
\dot{\omega}_h = -J^{-1}S(\omega_h)J\omega_h + n
\]
\[
\dot{v}_h = -S(\omega_h)v_b + f
\] (86)
with news inputs $f \in \mathbb{R}^3$ and $\tau \in \mathbb{R}^3$ both represented in a body fixed frame. Picking the dynamics of $M_e$ from (85), we derive $\omega_e$ and $v_e$ as
\[
\dot{\omega}_e = \dot{\omega}_d + S(\omega_d)R_e^\top \omega_b + R_e^\top J^{-1}S(\omega_h)J\omega_h + R_e^\top n
\] (87)
for $\dot{\omega}_e$ and
\[
\dot{v}_e = \dot{v}_d + R_e^\top S(\omega_h)(v_b - R_e v_d)
\]
\[
+ S(\omega_d)R_e^\top (v + S(\omega_b)p_e)
\]
\[
+ R_e^\top S(\omega_h)J^{-1}\omega_b + R_e^\top S(p_e)n - R_e^\top f
\] (88)
for $\dot{v}_e$ as functions of only the state $M_e$ and the desired trajectory and attitude of the vehicle.

8. Conclusions

With this work, it was possible to study and implement control laws that asymptotically stabilise systems evolving on Lie groups. The proposed controllers were able to so, even if the underlying configuration space is not diffeomorphic to the n-dimensional Euclidean space they were embedded in. Besides that, it was possible to develop a saturated feedback control law for the problem of position stabilisation.

Those controllers were then successfully adapted to three systems evolving in different Lie groups. After appropriate transformations were developed to write those systems in the form proposed in Section 2, all the required assumptions were verified and then simulations were performed to validate the theory. As expected, the closed hybrid systems which resulted from the implementation of the control laws all shown asymptotic stability for all situations that were simulated. Furthermore, the system with the saturated feedback control law, revealed to be bounded, not increasing beyond a set of values that can be tuned using the available parameters. This control law also shown a quicker response for the same parameters when compared with the non-saturated feedback law, while also leading to a more smoother evolution of the system state.

References


