HIERARCHICALLY CONSISTENT CONTROL SYSTEMS

by

George J. Pappas, Gerardo Lafferriere, and Shankar Sastry

Memorandum No. UCB/ERL M98/16

6 April 1998

HIERARCHICALLY CONSISTENT CONTROL SYSTEMS

by

George J. Pappas, Gerardo Lafferriere, and Shankar Sastry

Memorandum No. UCB/ERL M98/16

6 April 1998

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

HIERARCHICALLY CONSISTENT CONTROL SYSTEMS

GEORGE J. PAPPAS, GERARDO LAFFERRIERE, AND SHANKAR SASTRY

ABSTRACT. Large scale control systems typically possess a hierarchical architecture in order to manage complexity. Higher levels of the hierarchy utilize coarser models of the system resulting by aggregating the detailed lower level models. In this layered control paradigm, the notion of hierarchical consistency is important as it ensures the implementation of high level objectives by the lower level system. In this paper, we define a notion of modeling hierarchy for continuous control systems and obtain characterizations for hierarchically consistent linear systems with respect to controllability objectives. As an interesting byproduct, we obtain a hierarchical controllability criterion for linear systems from which we recover the best known controllability algorithm from numerical linear algebra.

1. INTRODUCTION

Large scale systems such as Intelligent Vehicle Highway Systems [36, 37] and Air Traffic Management Systems [30] are systems of very high complexity. Both the design and the analysis of such systems may be formidable due to the complexity and magnitude of the system. Complexity is typically reduced by imposing a hierarchical structure on the system architecture. In such a structure, systems of higher functionality reside at higher levels of the hierarchy and are therefore unaware of unnecessary lower level details. The main types of hierarchical structures are nicely classified and described in the visionary work of [23].

Consider as a motivating example, Air Traffic Management Systems, where the hierarchical structure shown in Figure 1 has been proposed in [34]. Each aircraft has on board a Flight Management System (FMS) which contains various different planners at different levels of functionality. The Strategic Planner negotiates via points with Air Traffic Control and nearby aircraft based on scheduling, fuel and safety issues. These via points are then directed down the hierarchy to the Tactical Planner which uses a kinematic model to generate output trajectories for the aircraft connecting the desired via points. The desired output trajectories are then passed to the Trajectory Planner which uses a more detailed dynamic model and generates suitable control inputs and state trajectories. Finally, the Regulation Layer utilizes a much more detailed model which considers engine dynamics, wind conditions, actuator saturation and tries to track the trajectories produced by the Trajectory Planner. The structure of Figure 1 is a multi-layered version of the quite common two-level planning and control hierarchies.

In the structure of Figure 1, each level has different objectives with higher levels having higher objectives. The Strategic Planner is interested in optimality, the Tactical planner is interested in controllability whereas the Trajectory Planner and Regulation layers deal with exact and approximate trajectory tracking respectively. In performing their tasks, higher planning levels use coarser aircraft models than the lower levels. The Strategic Planner could be using simple geometric models, the Tactical Planner could be using kinematic models while the Trajectory Planner could be using a much more detailed dynamic model. One of the main challenges in hierarchical systems is the extraction

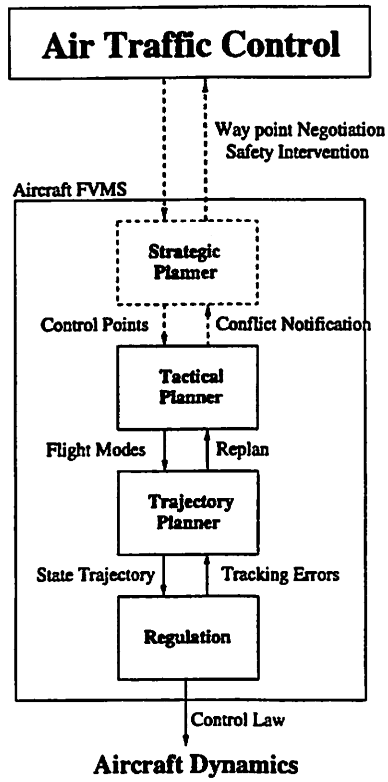

FIGURE 1. Hierarchical Structure for Air Traffic Management Systems

of a hierarchy of models at various levels of abstraction which are compatible with the functionality and objectives of each layer.

Abstraction or *aggregation* typically refers to grouping the system states into equivalence classes. Depending on the cardinality of the quotient space we may have *discrete* or *continuous* abstractions. With this notion of abstraction, the abstracted system will be defined as the induced quotient dynamics. Discrete abstractions of continuous systems have been considered in [10, 11] as well as [5, 26, 32]. Hierarchical systems for discrete event systems have been formally considered in [9, 38, 39, 41]. In this paper, we focus on continuous abstractions and obtain continuous analogues of their results. Therefore, our first priority is to have a formal notion of quotient control systems. More precisely,

Problem 1.1. *Given a control system*

$$\dot{x} = f(x,u) \qquad x \in \mathbb{R}^n \;\; u \in \mathbb{R}^m \tag{1.1}$$

and some map $y = h(x)$, where $h : \mathbb{R}^n \longrightarrow \mathbb{R}^p$, we would like to define a control system

$$\dot{y} = g(y,v) \qquad y \in \mathbb{R}^p \;\; v \in \mathbb{R}^k \tag{1.2}$$

which can produce as trajectories all functions of the form $y(t) = h(x(t))$, where $x(t)$ is a trajectory of system (1.1). That is, h maps trajectories of system (1.1) to trajectories of system (1.2).

The function h will be our "quotient map" which performs the state aggregation. System (1.2) will be referred to as the abstraction [29] or *macromodel* of the finer *micromodel* (1.1). In the ATMS example shown in Figure 1, one can think of system (1.1) as a detailed dynamical model residing in the Trajectory Planner and system (1.2) as a coarser kinematic model of the Tactical Planner. Note that the control input v of coarser model (1.2) is not the same input u of system (1.1) and should be thought of as a macroinput. For example, v can be velocity inputs of a kinematic model whereas u may be force and torque inputs of a dynamic model. This is therefore quite different from model reduction techniques which reduce or aggregate dynamics while using the same control inputs [6, 16, 17, 18, 19].

We will solve Problem 1.1 by first generalizing the geometric notion of Φ-related vector fields to control systems. A notion of Φ-related control systems would allow us to push forward control systems through quotient maps and obtain well defined control systems describing the aggregate dynamics. The notion of Φ-related control systems introduced in this paper is more general than the notion of projectable systems defined in [22] and [19] as we will show that given any control system and any surjective map Φ, there always exists another system that is Φ-related to it. Interestingly enough, our notion of Φ-related control systems mathematically formalizes the concept of *virtual inputs* used in backstepping designs [15]. The fact that the aggregation map sends trajectories of (1.1) to trajectories of (1.2) will enable us to propagate controllability from the micromodel to the macromodel.

Aggregation, however, is not independent of the functionality of the layer at which the abstracted system will be used. In hierarchical systems, each layer has a certain functionality and it is important to ensure that objectives of higher layers have a feasible execution by the lower levels. Therefore, when an abstracted model is extracted from a more detailed model, one would also like to ensure that certain properties propagate from the macromodel to the micromodel. The properties that are of interest at each layer may include optimality, controllability, stabilizability, and trajectory tracking. If one considers the property of controllability, then one would like to determine conditions under which controllability of the abstracted system (1.2) implies controllability of system (1.1). Obtaining such conditions would ensure that the macromodel is a *consistent abstraction* of the micromodel in the sense that controllability requests from the macromodel are *implementable* by the micromodel. Such conditions will serve as good design principles for hierarchical control systems. Different properties may require different conditions. For example, the notions of consistency [23], dynamic consistency [9] and hierarchical consistency [41] have been defined in order to ensure feasible execution of high level objectives for discrete event systems. In this paper, we will focus on controllability of linear control systems and characterize consistent linear abstractions. More precisely, we will solve the following problem:

Problem 1.2. *Given the linear control system*

$$\dot{x} = Ax + Bu \qquad x \in \mathbb{R}^n \quad u \in \mathbb{R}^m \tag{1.3}$$

characterize linear quotient maps $y = Cx$, so that the abstracted linear system

$$\dot{y} = Fy + Gv \qquad y \in \mathbb{R}^p \quad v \in \mathbb{R}^k \tag{1.4}$$

is controllable if and only if system (1.3) is controllable.

In addition to hierarchical control, the above ideas could also be useful in the analysis of complex systems. In order to tackle the complexity involved in verifying that a given large scale system satisfies certain properties, one tries to extract a simpler but qualitatively equivalent abstracted system, shown in Figure 2. Checking the desired property on the abstracted system should be *equivalent* or *sufficient* to checking the property on the original system. The area of computer aided verification, which must be credited with this notion of abstraction, typically faces problems of

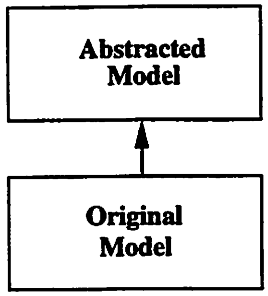

FIGURE 2. System Analysis using Abstractions

exponential complexity and abstractions are frequently used for complexity reduction. Depending on the property, special graph quotients which preserve the property of interest are constructed. For example, verification algorithms of hybrid systems [2, 3, 4, 13, 21], which contain both discrete event and continuous dynamics, are based on abstracting continuous dynamics by constant rectangular differential inclusions [14, 31]. More recently, a methodology for constructing finite graph quotients which have equivalent reachability properties with analytic vector fields is presented in [20]. A similar approach for checking stability of dynamical systems has been developed in [24].

In this spirit, and after having characterized consistent linear abstractions, we propose a hierarchical controllability algorithm which has computational or conceptual advantages over the standard Kalman rank condition or the Popov-Belevitch-Hautus (PBH) tests for large scale systems. Intuitively, instead of checking controllability of a large scale system, we construct a sequence of consistent abstractions and then check the controllability of a system which is much smaller in size. Consistency will then propagate controllability along this sequence of abstractions from the simpler quotient system to the original complex system. It is quite remarkable that a special case of the hierarchical controllability criterion recovers the best known controllability algorithm from numerical linear algebra [12].

The structure of this paper is as follows: In Section 2 we review some standard differential geometric concepts and the notion of Φ-related vector fields. Section 3 generalizes these notions for control systems and establishes the connection between trajectories of Φ-related control systems. In Section 5 we restrict these notions to linear abstractions and characterize consistent linear abstractions. These results are used in Section 6 in order to obtain a hierarchical controllability criterion. Finally, Section 7 discusses many interesting directions for further research.

2. Φ-RELATED VECTOR FIELDS

We first review some basic facts from differential geometry. The reader may wish to consult numerous books on the subject such as [35, 1, 25]. Let M be a differentiable manifold and T_pM be the tangent space of M at $p \in M$. We denote by $TM = \bigcup_{p\in M} T_pM$ the tangent bundle of M and by π the canonical projection map $\pi : TM \longrightarrow M$ taking a tangent vector $X_p \in T_pM \subset TM$ to the point $p \in M$.

Now let M and N be smooth manifolds and $\Phi : M \longrightarrow N$ be a smooth map. Let $p \in M$ and let $q = \Phi(p) \in N$. We push forward tangent vectors from T_pM to T_qN using the induced push forward map $\Phi_* : T_pM \longrightarrow T_qN$. If $f : M \longrightarrow N$ and $g : N \longrightarrow K$ then $(g \circ f)_* = g_* \circ f_*$ which is essentially the chain rule. A vector field or dynamical system on a manifold M is a smooth map $X : M \longrightarrow TM$ which places at each point p of M a tangent vector from T_pM. Let $I \subseteq \mathbb{R}$ be an open interval containing the origin. An integral curve of a vector field is a smooth curve $c : I \longrightarrow M$ whose tangent at each point is identically equal to the vector field at that point. Therefore an integral curve satisfies $c' = c_*(1) = X \circ c(t)$ for all $t \in I$ where $c_*(1)$ denotes $c_*(\frac{d}{dt})$.

An abstraction or aggregation map is a map $\Phi : M \longrightarrow N$ which we will assume to be surjective.[1] Given a vector field X on manifold M and a smooth map $\Phi : M \longrightarrow N$, not necessarily a diffeomorphism, the push forward of X by Φ_* is generally not a well defined vector field on N. This leads to the concept of Φ-related vector fields.

Definition 2.1 (Φ-related Vector Fields). *Let X and Y be vector fields on manifolds M and N respectively and $\Phi : M \longrightarrow N$ be a smooth map. Then X and Y are Φ-related iff the following diagram commutes*

$$\begin{array}{ccc} M & \xrightarrow{\Phi} & N \\ X\big\downarrow & & \big\downarrow Y \\ TM & \xrightarrow{\Phi_*} & TN \end{array} \tag{2.1}$$

or otherwise iff $\Phi_* \circ X = Y \circ \Phi$.

Note that the above definition does not require the map Φ to be surjective. If Φ is not surjective then X may be Φ-related to many vector fields on N. However, if Φ is a smooth surjection from M to N, then given a vector field X on a manifold M, the push forward of X by Φ_* is a well defined vector field on N only if $\Phi_*(X_{p_1}) = \Phi_*(X_{p_2})$ whenever $\Phi(p_1) = \Phi(p_2)$ for any two points $p_1, p_2 \in M$.

Example 2.2. Consider for example the linear vector field

$$\dot{x} = Ax \qquad x \in \mathbb{R}^n \tag{2.2}$$

and the onto, linear quotient map $y = Cx$. Then in order to obtain a well defined quotient vector field,

$$\dot{y} = Fy \qquad y \in \mathbb{R}^m \tag{2.3}$$

by C-relatedness we must have $CAx = FCx$ for all $x \in \mathbb{R}^n$. But for $x \in Ker(C) = \{x \in \mathbb{R}^n \mid Cx = 0\}$ we must have $CAx = F(Cx) = 0$ and thus $Ax \in Ker(C)$. Thus, a necessary condition to obtain a well defined quotient vector field is

$$AKer(C) \subseteq Ker(C) \tag{2.4}$$

It turns out that this is also sufficient for the existence of unique quotient map F [40].

The following well known theorem ([1]) gives us a condition on the integral curves of two Φ-related vector fields. A simple proof is included for completeness.

Theorem 2.3 (Integral Curves of Φ-related Vector Fields). *Let X and Y be vector fields on M and N respectively and let $\Phi : M \longrightarrow N$ be a smooth map. Then vector fields X and Y are Φ-related if and only if for every integral curve c of X, $\Phi \circ c$ is an integral curve of Y.*

[1]Note that any map Φ gives rise to an equivalence relation by defining states x and y equivalent if $\Phi(x) = \Phi(y)$. In order for the resulting quotient space to have a manifold structure, the equivalence relation must be regular [1]

Proof. Assume first that for any integral curve c of X, $\Phi \circ c$ is an integral curve of Y. Then

$$\begin{aligned}
&(\Phi \circ c)' = (\Phi \circ c)_*(1) = Y(\Phi \circ c) \Rightarrow \\
&\Phi_* \circ c_*(1) = Y \circ \Phi \circ c \Rightarrow \\
&\Phi_* \circ X \circ c = Y \circ \Phi \circ c
\end{aligned}$$

But since this is true for any integral curve c it must be true that $\Phi_* \circ X = Y \circ \Phi$ But then, by Definition 2.1, X and Y are Φ-related. Conversely, let X and Y be Φ related. Then for any integral curve c of X,

$$\begin{aligned}
&\Phi_* \circ X = Y \circ \Phi \Rightarrow \\
&\Phi_* \circ X \circ c = Y \circ \Phi \circ c \Rightarrow \\
&\Phi_* \circ c_*(1) = Y(\Phi \circ c) \Rightarrow \\
&(\Phi \circ c)_*(1) = Y(\Phi \circ c)
\end{aligned}$$

and thus $\Phi \circ c$ is an integral curve of Y. This completes the proof. □

If Σ_X and Σ_Y denote all integral curves of vector fields X and Y respectively, then Theorem 2.3 simply states that

$$X \text{ and } Y \text{ are } \Phi\text{-related} \Longleftrightarrow \Phi(\Sigma_X) \subseteq \Sigma_Y$$

Therefore Y overapproximates the collection of curves $\Phi(\Sigma_X)$ and allows redundant evolutions. This is the notion of abstraction defined in [29]. Instead of checking a property, for example reachability, on a vector field X, it is checked on Y, which should be easier to analyze since, in general, N is of lower dimension. If the property is true for Σ_Y then it must be true for all $\Phi(\Sigma_X)$. If however the property fails for some integral curve. in $c \in \Sigma_Y$, then we have no way of telling whether the property fails for $\Phi(\Sigma_X)$ since c may belong in $\Sigma_Y \backslash \Phi(\Sigma_X)$. This procedure therefore is sufficient but not necessary. However,

Corollary 2.4. *Let X and Y be Φ-related vector fields on M and N respectively with respect to a smooth surjective map $\Phi : M \longrightarrow N$. Then $\Phi(\Sigma_X) = \Sigma_Y$.*

Proof. From Theorem 2.1 we have $\Phi(\Sigma_X) \subseteq \Sigma_Y$. Now let $c_Y \in \Sigma_Y$. Let q be any point in c_Y. Since Φ is surjective, let $p \in \Phi^{-1}(q)$ and let c_X be the integral curve passing through p. By Theorem 2.1, $\Phi(c_X)$ is an integral curve passing through $q = \Phi(p)$. By uniqueness of solutions $c_Y = \Phi(c_X)$ and thus $c_Y \in \Phi(\Sigma_X)$. Therefore $\Sigma_Y \subseteq \Phi(\Sigma_X)$ which results in $\Sigma_Y = \Phi(\Sigma_X)$ □

Corollary 2.4 says that checking reachability properties, of vector field X is equivalent to checking reachability on vector field Y. In addition, Corollary 2.4 says that every integral curve c_Y of Y can be writen as $\Phi(c_X)$ for some integral curve c_X of X. Therefore, if one thinks of Y as a coarser model and X as a more detailed model, then every trajectory of the higher model Y can be implemented by a trajectory of the detailed model X.

Even though Φ-relatedness of vector fields is a rather restrictive condition, the above discussion provides the correct conceptual framework for generalizing these concepts to control systems, where due to the freedom of control inputs the equivalent conditions will not be as restrictive.

3. Control System Abstractions

In this section, the notions of Section 2 for vector fields are extended to control systems. We will develop such notions for rather general control systems since it does not require more effort to do so. In addition, generality will ensure that the concepts of this section do not depend on the particular system structure. We first present a global and coordinate-free description of control systems which is due to Brockett [7, 8] and can also be found in [27]. This global description is based on the notion of fiber bundles which are defined first.

Definition 3.1 (Fiber Bundles). *A fiber bundle is a five-tuple $(B, M, \pi, U, \{O_i\}_{i\in I})$ where B, M, U are smooth manifolds called the total space, the base space and the standard fiber respectively. The map $\pi : B \longrightarrow M$ is a surjective submersion and $\{O_i\}_{i\in I}$ is an open cover of M such that for every $i \in I$ there exists a diffeomorphism $\Psi_i : \pi^{-1}(O_i) \longrightarrow O_i \times U$ satisfying*

$$\pi_o \circ \Psi_i = \pi$$

where π_o is the projection from $O_i \times U$ to O_i. The submanifold $\pi^{-1}(p)$ is called the fiber at $p \in M$. If all the fibers are vector spaces of constant dimension, then the fiber bundle is called a vector bundle.

Definition 3.2 (Control Systems). *A control system $S = (B, F)$ consists of a fiber bundle $\pi : B \longrightarrow M$ called the control bundle and a smooth map $F : B \longrightarrow TM$ which is fiber preserving and hence satisfies*

$$\pi' \circ F = \pi$$

where $\pi' : TM \longrightarrow M$ is the tangent bundle projection.

Essentially, the base manifold M of the control bundle is the state space and the fibers $\pi^{-1}(p)$ can be thought of as the state dependent control spaces. Given the state p and the input, the map F selects a tangent vector from T_pM. The notion of trajectories of control systems is now defined.

Definition 3.3 (Trajectories of Control Systems). *A smooth curve $c : I \longrightarrow M$ is called a trajectory of the control system $S = (B, F)$ if there exists a curve $c^B : I \longrightarrow B$ satisfying*

$$\begin{aligned} \pi \circ c^B &= c \\ c' = c_*(1) &= F \circ c^B \end{aligned}$$

In local (bundle) coordinates, Definition 3.3 simply says that a trajectory of a control system is a curve $x: I \to M$ for which there exists a function $u: I \to U$ satisfying, satisfying $\dot{x} = F(x, u)$. Note that even though Definition 3.3 assumes c to be smooth, the bundle curve c^B is not necessarily smooth. The definition therefore allows nonsmooth control inputs as long as the projection $\pi \circ c^B = c$ is smooth. We are now in a position to define Φ-related control systems in a manner similar to Definition 2.1 for vector fields.

Definition 3.4 (Φ-Related Control Systems). *Let $S_M = (B_M, F_M)$ with $\pi_M : B_M \longrightarrow M$ and $S_N = (B_N, F_N)$ with $\pi_N : B_N \longrightarrow N$ be two control systems. Let $\Phi : M \longrightarrow N$ be a smooth map. Then control systems S_M and S_N are Φ-related iff for every $p \in M$*

$$\Phi_* \circ F_M \left(\pi_M^{-1}(p)\right) \subseteq F_N \left(\pi_N^{-1}(\Phi(p))\right) \tag{3.1}$$

Condition (3.1) states that for each $p \in M$ the left hand side of (3.1) first takes the input space available at p, and pushes it through F_M to obtain all possible tangent directions of the control system S_M at p. This set of tangent directions is pushed through Φ_* to obtain a set of tangent vectors in $T_{\Phi(p)}N$. In order for S_M and S_N to be Φ-related, this set must be contained in the image under F_N of the input space available at $\Phi(p)$. Note that many control systems S_N may be Φ-related to S_M as the set of tangent vectors on N that must be captured, can be generated using many control parameterizations.

It is easy to show that Definition 3.4 is transitive. Indeed, if $\Phi_1: M_1 \to M_2$, $\Phi_2: M_2 \to M_3$, S_{M_1} is Φ_1-related to S_{M_2}, and S_{M_2} is Φ_2-related to S_{M_3}, then S_{M_1} is $\Phi_2 \circ \Phi_1$-related to S_{M_3}. It therefore makes sense to consider a sequence of Φ-related systems. In addition, given M, N, a map $\Phi: M \to N$ and a system S_M, one can put a partial order on all possible Φ-related systems S_N, where the partial ordering arises from pointwise subset inclusion of the right hand side of (3.1).

To see that Definition 3.4 is a generalization of Definition 2.1, consider vector fields X_M on M and X_N on N. Then X_M and X_N can be thought of as trivial control systems on M and N respectively by letting $B_M = M$, $B_N = N$, $\pi_M = id_M$, $\pi_N = id_N$, and $F_M = X_M$, $F_N = X_N$. Condition (3.1) requires that for all $p \in M$ we have $\Phi_* \circ X_M(p) \subseteq X_N \circ \Phi(p)$. But since X_N is a vector field on N we can only choose one tangent vector at each point. This forces $\Phi_* \circ X_M(p) = X_N \circ \Phi(p)$, which is Definition 2.1 of Φ-related vector fields.

The following proposition, which is an immediate consequence of Definition 3.4, shows that every control or dynamical system is Φ-related to some control system for any map Φ.

Proposition 3.5. *Given any control system $S_M = (B_M, F_M)$ and any smooth map $\Phi : M \longrightarrow N$, then there exists a control system $S_N = (B_N, F_N)$ which is Φ-related to S_M. In particular, every vector field X on M is Φ-related to some control system S_N.*

Proof. Given S_M, construct S_N by simply letting $B_N = TN$ and $F_N : TN \longrightarrow TN$ equal the identity. Then condition (3.1) is trivially satisfied. Thus $S_N = (B_N, F_N)$ is Φ-related to S_M. □

In local coordinates, Proposition 3.5 simply states that the push forward of a control system or a vector field is a differential inclusion which can be thought of as another control system. Even though Proposition 3.5 is a simple existential result, it is important as it shows that given any control system and any aggregation map, then an abstracted control system always exists. Therefore, Definition 3.4 is a generalization of the notions of *projectable* control systems defined in [19, 22]. A control system is projectable, essentially, when each vector field corresponding to a fixed input value is Φ-related to some vector field. Definition 3.4, instead of globally pushing a vector field for each fixed value of the control input, takes a pointwise approach by pushing forward all possible tangent directions at a state for all possible inputs available at that state. By Proposition 3.5, any projectable system in the sense of [19, 22] is also Φ-related in the sense of Definition 3.4. The following example illustrates that the other direction is not true.

Example 3.6. Consider the double integrator

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= u\end{aligned}$$

with $x_1, x_2, u \in \mathbb{R}$ and the projection $\Phi(x_1, x_2) = x_1$. Using Definition 3.4, we obtain that

$$\dot{x}_1 = x_2$$

is a valid Φ-related system. The double integrator, however, is not projectable in the sense of [22, 19] with respect to this map as for any fixed value of u, the vector field $[x_2\ u]^T$ is not Φ-related to any vector field on $\mathbb{R}$. For the nonlinear control system,

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2)\\ \dot{x}_2 &= f_2(x_1, x_2, u)\end{aligned}$$

with states x_1, x_2, input u, and the projection $\Phi(x_1, x_2) = x_1$, a Φ-related system is

$$\dot{x}_1 = f_1(x_1, x_2)$$

with state x_1 but where x_2 is now thought of as an input. This is the notion of virtual inputs used in backstepping designs [15]. A more constructive methodology for generating abstractions of linear systems will be presented in Section 5.

The following theorem should be thought of as a generalization of Theorem 2.3 for control systems.

Theorem 3.7 (Trajectories of Φ-Related Control Systems). *Let $S_N = (B_N, F_N)$ and $S_M = (B_M, F_M)$ be two control systems and $\Phi : M \longrightarrow N$ be a smooth map. Then S_M and S_N are Φ-related if and only if for every trajectory c_M of S_M, $\Phi \circ c_M$ is a trajectory of S_N.*

Proof. (Sufficiency) Assume that S_M and S_N are Φ-related and thus for all $p \in M$ we have

$$\Phi_* \circ F_M\left(\pi_M^{-1}(p)\right) \subseteq F_N\left(\pi_N^{-1}(\Phi(p))\right) \tag{3.2}$$

Let $c_M : I \longrightarrow M$ be any trajectory of S_M. We must show that $\Phi \circ c_M$ is a trajectory of S_N. We must therefore find a curve $c_N^B : I \longrightarrow B_N$ such that for all $t \in I$ we have $\pi_N \circ c_N^B(t) = \Phi \circ c_M(t)$ and $(\Phi \circ c_M)'(t) = F_N \circ c_N^B(t)$.

Since $c_M : I \longrightarrow M$ is a trajectory of S_M, by Definition 3.3 there exists a curve $c_M^B : I \longrightarrow B_M$ such that for all $t \in I$ we have $\pi_M \circ c_M^B(t) = c_M(t)$ and $c_M'(t) = F_M \circ c_M^B(t)$. By Φ-relatedness of S_M and S_N we obtain that for all $t \in I$,

$$\begin{aligned}\Phi_* \circ F_M\left(\pi_M^{-1}(c_M(t))\right) &\subseteq F_N\left(\pi_N^{-1}(\Phi(c_M(t)))\right) \Longrightarrow\\ \Phi_* \circ F_M \circ c_M^B(t) &\in F_N\left(\pi_N^{-1}(\Phi(c_M(t)))\right)\end{aligned} \tag{3.3}$$

Condition (3.3) implies that for each $t \in I$ there must exist at least one element $c_N^B(t) \in \pi_N^{-1}(\Phi(c_M(t)))$ (and thus $\pi_N \circ c_N^B(t) = \Phi \circ c_M(t)$) such that

$$\begin{aligned}\Phi_* \circ F_M \circ c_M^B(t) &= F_N \circ c_N^B(t)\\ \Phi_* \circ c_M'(t) &= F_N \circ c_N^B(t)\\ (\Phi \circ c_M)'(t) &= F_N \circ c_N^B(t)\end{aligned}$$

Therefore $\Phi \circ c_M$ is a trajectory of S_N.

(Necessity) Assume that for every trajectory $c_M : I \longrightarrow M$ of S_M, $\Phi \circ c_M$ is a trajectory of S_N. Now for any point $p \in M$ let

$$Y_{\Phi(p)} \in \Phi_*\left(F_M(\pi_M^{-1}(p))\right) \tag{3.4}$$

We must show that $Y_{\Phi(p)} \in F_N(\pi_N^{-1}(\Phi(p)))$. We can write $Y_{\Phi(p)} = \Phi_*(X_p)$ for some (not necessarily unique) tangent vector $X_p \in F_M(\pi_M^{-1}(p))$. Then there exists a trajectory $c_M : I \longrightarrow M$ such that at some $t^* \in I$ we have

$$c_M(t^*) = p \tag{3.5}$$

$$c_M'(t^*) = X_p \tag{3.6}$$

Indeed, a curve c_M satisfying (3.5,3.6) always exists by the existence theorems for differential equations. To show that c_M is a trajectory, we need to find $c_M^B \colon I \longrightarrow B_M$ such that $\pi \circ c_M^B = c_M$. Let O be a bundle trivializing neighborhood of p and $\Psi \colon \pi^{-1}(O) \longrightarrow O \times U$ the trivializing map. There exists $u \in U$ such that $X_p = F_M \circ \Psi^{-1}(p,u)$. Restricting I if necessary we may assume $c_M(I) \subset O$. We can then define the desired curve by $c_M^B(t) = F_M \circ \Psi^{-1}(c_M(t), u)$.

Since c_M is a trajectory of S_M satisfying (3.5,3.6), then by assumption we have that $\Phi \circ c_M$ is a trajectory of S_N. Therefore by Definition 3.3, there must exist a curve $c_N^B : I \longrightarrow B_N$ such that for all $t \in I$ we have $\pi_N \circ c_N^B(t) = \Phi \circ c_M(t)$ and $(\Phi \circ c_M)'(t) = F_N \circ c_N^B(t)$. In particular, at $t^* \in I$ we have

$$\begin{aligned} (\Phi \circ c_M)'(t^*) &= F_N \circ c_N^B(t^*) \\ \Phi_* \circ c_M'(t^*) &\in F_N \left(\pi_N^{-1}(\Phi(c_M(t^*)))\right) \\ Y_p = \Phi_*(X_p) &\in F_N \left(\pi_N^{-1}(\Phi(p))\right) \end{aligned}$$

Therefore, at all points $p \in M$ we must have $\Phi_* \circ F_M \left(\pi_M^{-1}(p)\right) \subseteq F_N(\pi_N^{-1}(\Phi(p)))$ and thus S_M and S_N are Φ-related. This completes the proof. □

If Σ_{S_M} and Σ_{S_N} denote all trajectories of control systems S_M and S_N respectively, then Theorem 3.7 simply states that

$$S_M \text{ and } S_N \text{ are } \Phi\text{-related} \Longleftrightarrow \Phi(\Sigma_{S_M}) \subseteq \Sigma_{S_N}$$

This is the notion of control system abstractions defined in [29]. Intuitively, it says that if a state trajectory can be generated by the micromodel using some low level control input, then the abstracted trajectory must also be generated by the macromodel using some high level input. Note again that the quotient system overapproximates the abstracted trajectories of the original system which may result in trajectories that the macrosystem may generate but are infeasible in the micromodel.

Theorem 3.7 does not guarantee that the curve $c_N^B(t)$ is a smooth curve. The main obstacle for generating smooth $c_N^B(t)$ is whether the map $F_N : B_N \longrightarrow TM$ is an embedding. The following variation on a well known example shows that the assumption that F_N be an embedding is necessary.

Example 3.8. Let $M = N = \mathbb{R}^2$, $B_M = B_N = \mathbb{R}^3$, $U_M = U_N = \mathbb{R}$, and let $\Phi : M \to N$ be the identity. Let $F_M : B_M \to TM$ and $F_N : B_N \to TN$ be given by

$$\begin{aligned} F_M(x,y,u) &= (x, y, 2\cos(g(u) - \frac{\pi}{2}), \sin 2(g(u) - \frac{\pi}{2})) \\ F_N(x,y,u) &= (x, y, 2\cos(g(u) - \frac{\pi}{2}), \sin 2(\frac{\pi}{2} - g(u))) \end{aligned}$$

where $g(u) = \pi + 2 \arctan u$. In fact, the analysis below would work with any infinitely differentiable function g which is monotone increasing and satisfies $g(0) = \pi$, $\lim_{u\to\infty} g(u) = 2\pi$, and $\lim_{u\to-\infty} g(u) = 0$. Notice that the difference between F_N and F_M is only in the sign of the last component. For each fixed $(x,y) \in \mathbb{R}^2$, both F_N and F_M embed $\mathbb{R}$ in $\mathbb{R}^2$ as the same set, the figure "eight". However, near the origin of $\mathbb{R}^2$, as u increases the images of $\mathbb{R}$ under F_M and F_N are traversed in different directions: from the fourth to the second quadrant for F_M, and from the first to the third for F_N. As is well known, the maps F_N and F_M so defined are indeed immersions, but not embeddings. The relative topology of their respective images as subsets of $TN = TM = \mathbb{R}^4$ is coarser than the topology induced by F_N and F_M.

As an trajectory on M we take the curve $c_M : \mathbb{R} \to M$ given by

$$c_M(t) = \left(\int_0^t 2\cos(g(\tau) - \frac{\pi}{2})\, d\tau, \int_0^t \sin 2(g(\tau) - \frac{\pi}{2})\, d\tau\right).$$

Then, $c_M(0,0) = 0$ and $c'_M(t)$ as an element of the tangent bundle TM can be written (in global coordinates) as

$$c'_M(t) = \left(\int_0^t 2\cos(g(\tau) - \frac{\pi}{2})\, d\tau, \int_0^t \sin 2(g(\tau) - \frac{\pi}{2})\, d\tau, 2\cos(g(t) - \frac{\pi}{2}), \sin 2(g(t) - \frac{\pi}{2})\right).$$

The desired $c_M^B : \mathbb{R} \to B_M$ (to effectively show that $c_M(t)$ is a trajectory) is defined by

$$c_M^B(t) = \left(\int_0^t 2\cos(g(\tau) - \frac{\pi}{2})\, d\tau, \int_0^t \sin 2(g(\tau) - \frac{\pi}{2})\, d\tau, t\right).$$

We then clearly get

$$c'_M(t) = F_M(c_M^B(t)) \qquad \text{for } t \in \mathbb{R}$$

The induced curve on N is $c_N(t) = \Phi \circ c_M(t)$ and so $c'_N(t) = \Phi_* \circ c'_M(t) = c'_M(t)$, since $\Phi_* = \mathrm{id}_*$. To show that c_N is a trajectory we must find a curve $c_N^B : \mathbb{R} \to B_N$ such that

$$F_N(c_N^B(t)) = c'_N(t) \tag{3.7}$$

for all $t \in \mathbb{R}$. Let's write c_N^B in components as $c_N^B(t) = (\gamma_1(t), \gamma_2(t), s(t))$. By comparing components in Equation 3.7 , we immediately get

$$\gamma_1(t) = \int_0^t 2\cos(g(\tau) - \frac{\pi}{2})\, d\tau$$

$$\gamma_2(t) = \int_0^t \sin 2(g(\tau) - \frac{\pi}{2})\, d\tau$$

Moreover, for each t, the function $s = s(t)$ must solve

$$2\cos(g(s) - \frac{\pi}{2}) = 2\cos(g(t) - \frac{\pi}{2})$$

$$\sin 2(g(s) - \frac{\pi}{2}) = \sin 2(\frac{\pi}{2} - g(t))$$

As a (lengthy and tedious) consideration of all the different possibilities shows, the function $s(t)$ is uniquely defined by

$$s(t) = \begin{cases} 0 & \text{if } t = 0, \\ \tan(\frac{\pi}{2} - \arctan t) & \text{for } t \neq 0. \end{cases}$$

Therefore, $c_N^B(t)$ is clearly not even continuous.

The following theorem shows that F_N being an injective embedding is sufficient to guarantee smoothness of the $c_N^B(t)$. Note that requiring F_N to be an injective embedding implies that the dimension of the input space is less than the dimension of TN and thus there are no redundant inputs (which covers the cases of interest). In particular, if the control system S_N is affine in the controls then this is equivalent to saying that the "controlled" vector fields are linearly independent at each point. That is, if we write the system in local (bundle) coordinates of B_N and local (vector bundle) coordinates of TN as

$$\dot{x} = f(x) + \sum_{i=1}^{k} g_i(x) u_i$$

then for each x the vectors $g_1(x),\dots,g_k(x)$ are linearly independent.

Theorem 3.9 (Control Input Smoothness). *Let $S_N=(B_N,F_N)$ and $S_M=(B_M,F_M)$ be two Φ-related control systems where $F_N: B_N \longrightarrow TN$ is an injective embedding. Let $c_M\colon I \longrightarrow M$ be a trajectory of S_M and assume that the corresponding $c_M^B\colon I \longrightarrow B_M$ is smooth. Then there exists a smooth curve $c_N^B: I \longrightarrow B_N$ such that for all $t\in I$, $\pi_N \circ c_N^B(t) = \Phi\circ c_M(t)$ and $F_N\circ c_N^B(t) = (\Phi\circ c_M)'(t)$.*

Proof. Since S_M and S_N are Φ-related we have $\Phi_*\circ F_M\circ c_M^B(t)\in F_N\left(\pi_N^{-1}\left(\Phi(c_M(t))\right)\right)$ for each $t\in I$. Moreover, since by assumption F_N is an embedding, the space B_N is diffeomorphic to its image under F_N. We can then define

$$c_N^B(t) = F_N^{-1}(\Phi_*\circ F_M\circ c_M^B(t))$$

which is clearly smooth and satisfies the desired properties. □

4. Consistent Control Abstractions

In general, we are not simply interested in abstracting systems but also propagating properties between the original and abstracted model. In this paper, we focus on various notions of controllability.

Definition 4.1 (Controllability). *Let $S=(B,F)$ be a control system on M. For $p\in M$, define $Reach(p,S)$ to be the set of points $q\in M$ for which there exists a trajectory $c: I\longrightarrow M$ of S such that for some $t_1,t_2\in I$ we have $c(t_1)=p$ and $c(t_2)=q$. The control system S is called controllable iff for all $p\in M$, $Reach(p,S)=M$.*

Theorem 3.7 allows us to always propagate the property of controllability from the micromodel to the macromodel for any aggregation map.

Theorem 4.2 (Controllability Propagation). *Let control systems $S_M=(B_M,F_M)$ and $S_N=(B_N,F_N)$ be Φ-related with respect to some smooth surjection $\Phi: M\longrightarrow N$. Then for all $p\in M$,*

$$\Phi\left(Reach(p,S_M)\right)\subseteq Reach(\Phi(p),S_N)$$

Thus, if S_M is controllable then S_N is controllable.

Proof. Consider any $p\in M$ and let $q\in\Phi(Reach(p,S_M))$. Then there exists $p_1\in\Phi^{-1}(q)$ with $p_1\in Reach(p,S_M)$. Thus there exists a trajectory c_M of S_M such that $c_M(t_1)=p$ and $c_M(t_2)=p_1$. By Φ-relatedness, the curve $\Phi\circ c_M$ is a trajectory of S_N which connects $\Phi(c_M(t_1))=\Phi(p)$ and $\Phi(c_M(t_2))=\Phi(p_1)=q$. Therefore $q\in Reach(\Phi(p),S_N)$.

If S_M is controllable, then for all $p\in M$ we have $Reach(p,S_M)=M$. But then $\Phi(Reach(p,S_M)) = \Phi(M)=N=Reach(\Phi(p),S_N)$. Thus S_N is controllable. □

Note that Theorem 4.2 is true regardless of the structure of the aggregation map Φ. From a hierarchical perspective, the reverse question is a lot more interesting since it would guarantee that controllability requests are implementable by the lower level system. In order to arrive at this goal, we define the notions of implementability and consistency. We also give descriptions of those concepts in terms of reachable sets.

Definition 4.3 (Controllability Implementation). *Let $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ be two control systems and $\Phi : M \longrightarrow N$ be a smooth surjection. Then S_N is implementable[2] by S_M iff whenever there is a trajectory of S_N connecting $q_1 \in N$ and $q_2 \in N$, then there exist $p_1 \in \Phi^{-1}(q_1)$ and $p_2 \in \Phi^{-1}(q_2)$ and a trajectory of S_M connecting p_1 and p_2*

Implementability is therefore an existential property. If one thinks of the map Φ as a quotient map, then implementability requires that a reachability request is implementable by at least one member of the equivalence class. It is clear from Definition 4.3 that implementability is transitive, that is if S_{M_1} is implementable by S_{M_2} with respect to Φ_1, and S_{M_2} is implementable by S_{M_3} with respect to Φ_2, then S_{M_1} is implementable by S_{M_3} with respect to $\Phi_1 \circ \Phi_2$. This is important in hierarchical systems which should consist of a sequence of implementable abstractions. It should be noted that the notion of implementability defined above is related to the notion of between-block controllability for discrete event systems, defined in [9, 11].

Proposition 4.4 (Implementation Condition). *Consider control systems $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ and a smooth surjection $\Phi : M \longrightarrow N$. Then S_N is implementable by S_M if and only if for all $q \in N$,*

$$Reach(q, S_N) \subseteq \Phi(Reach(\Phi^{-1}(q), S_M)) \tag{4.1}$$

where $Reach(\Phi^{-1}(q), S_M) = \cup_{p \in \Phi^{-1}(q)} Reach(p, S_M)$.

Proof. Let $q' \in Reach(q, S_N)$. By implementability, there exists a trajectory of S_M connecting some $p \in \Phi^{-1}(q)$ to some $p' \in \Phi^{-1}(q')$ and thus $p' \in Reach(p, S_M)$. But then $q' = \Phi(p') \in \Phi(Reach(p, S_M)) \subseteq \Phi(Reach(\Phi^{-1}(q), S_M))$.

Conversely, let $q_2 \in Reach(q_1, S_N)$ for some $q_1 \in N$. By assumption,

$$q_2 \in \Phi(Reach(\Phi^{-1}(q_1), S_M)) = \Phi(\cup_{p_1 \in \Phi^{-1}(q_1)} Reach(p_1, S_M)) = \cup_{p_1 \in \Phi^{-1}(q_1)} \Phi(Reach(p_1, S_M))$$

But then there must exist at least one $p_1' \in \Phi^{-1}(q_1)$ such that $q_2 \in \Phi(Reach(p_1', S_M))$ which in turn implies that there exists $p_2' \in Reach(p_1', S_M)$ with $\Phi(p_2) = q_2$ and thus S_N is implementable by S_M. This completes the proof. □

We will mostly be interested in implementability of Φ-related systems, in which case the above inclusion becomes an equality, by Theorem 4.2.

Implementability may depend on the particular element chosen from the equivalence class $\Phi^{-1}(q)$. In order to make the controllability request well defined, it would have to be independent of the particular element chosen from the equivalence class. This leads to the important notion of consistency.

Definition 4.5 (Controllability Consistency). *Let $S_M = (B_M, F_M)$ be a control system on M and let $\Phi : M \longrightarrow N$ be a smooth surjection. Then S_M is called consistent with respect to Φ whenever the following holds: if there exists a trajectory of S_M connecting p and q, then for all p' such that $\Phi(p) = \Phi(p')$ there exists a trajectory of S_M connecting p' to some q' with $\Phi(q) = \Phi(q')$.*

Note that while implementability is a condition between two systems S_M and S_N, consistency is a condition on a single system with respect to some quotient map Φ. Consistency requires that the ability to reach a particular equivalence class is independent of the chosen element from the initial equivalence class. Notice that $\Phi^{-1}(\Phi(p))$ is the equivalence class of p with respect to Φ.

[2]In this paper, we only consider implementation of controllability requests. Thus implementability will refer to controllability implementation.

Proposition 4.6 (Consistency Condition). *Consider a control system $S = (B, F)$ on M and a smooth surjection $\Phi : M \longrightarrow N$. Then S is consistent with respect to Φ if and only if for all $p \in M$,*

$$\Phi(Reach(\Phi^{-1}(\Phi(p)), S)) = \Phi(Reach(p, S)). \tag{4.2}$$

Proof. Clearly $\Phi(Reach(p, S)) \subseteq \Phi(Reach(\Phi^{-1}(\Phi(p)), S))$ for any $p \in M$. Let $q = \Phi(p')$ with $p' \in Reach(\Phi^{-1}(\Phi(p)), S)$. There exists $p_0 \in \Phi^{-1}(\Phi(p))$ such that $p' \in Reach(p_0, S)$. By consistency, since $\Phi(p_0) = \Phi(p)$, there exists $p'' \in Reach(p, S)$ with $\Phi(p'') = \Phi(p')$. But then $q = \Phi(p'') \in \Phi(Reach(p, S))$.

Conversely, assume (4.2) holds. Let $q \in Reach(p, S)$ and $\Phi(p') = \Phi(p)$. Then $\Phi(q) \in \Phi(Reach(\Phi^{-1}(\Phi(p)), S)) = \Phi(Reach(p', S))$ and there exists $q' \in Reach(p', S)$ with $\Phi(q) = \Phi(q')$. □

Consistency does not place any conditions on which element of the final equivalence class the system will be steered to. In some hierarchical systems, this may be acceptable as the high level system S_N may be interested in its command having a feasible execution by S_M without being interested about the particular state of S_M as long as it steers it to the correct equivalence class. This form of generalized output controllability is now defined.

Definition 4.7 (Macrocontrollability). *Let $S = (B, F)$ be a control system on M and let $\Phi : M \longrightarrow N$ be a smooth surjection. Then S is called macrocontrollable if for all $p \in M$ and any $q \in N$ there exists an trajectory of S connecting p to some $p' \in M$ with $\Phi(p') = q$.*

By combining the notions of implementability and consistency, we can propagate some controllability information from the coarser system S_N to the more detailed system S_M.

Proposition 4.8 (Macrocontrollability Propagation). *Consider control systems $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ which are Φ-related with respect to the smooth surjection $\Phi : M \longrightarrow N$. Assume that S_M is an implementation of S_N, and S_M is consistent. Then S_M is macrocontrollable if and only if S_N is controllable.*

Proof. Let $p \in M$ and $q \in N$ be any points. Let $q_0 = \Phi(p)$. Since S_N is controllable, there exists a trajectory of S_N connecting q_0 and q. Since S_M is an implementation of S_N, there exists a trajectory of S_M connecting some $p_1 \in \Phi^{-1}(q_0)$ and some $p_2 \in \Phi^{-1}(q)$. Moreover, since S_M is also consistent, there is a trajectory of S_M connecting p to some p' with $\Phi(p') = \Phi(p_2) = q$. Therefore, S_M is macrocontrollable. The other direction follows easily from Theorem 4.2. □

In order to propagate full controllability from S_M to S_N, we need a stronger notion of consistency which would be independent from the elements chosen from both the initial and final equivalence class.

Definition 4.9 (Strong Controllability Consistency). *Let $S_M = (B_M, F_M)$ be a control system on M and $\Phi : M \longrightarrow N$ a smooth surjection. Then S_M is called strongly consistent with respect to Φ whenever the following holds: if there exists a trajectory of S_M connecting p and q, then for all p' and for all q' such that $\Phi(p) = \Phi(p')$, $\Phi(q) = \Phi(q')$ there exists a trajectory connecting p' to q'.*

Definition 4.9 is weaker than the notion of in-block controllability of [9, 11] as it does not restrict the system to remain within the equivalence class in order to steer from one element to another in the same class.

Proposition 4.10 (Strong Consistency Condition). *Consider control system $S = (B, F)$ on M and the smooth surjection $\Phi : M \longrightarrow N$. Then S is strongly consistent with respect to Φ if and only if for all $p \in M$,*

$$Reach(p, S) = \Phi^{-1}(\Phi(Reach(\Phi^{-1}(\Phi(p)), S))). \tag{4.3}$$

Proof. The inclusion $Reach(p, S) \subseteq \Phi^{-1}(\Phi(Reach(\Phi^{-1}(\Phi(p)), S)))$ always holds. Let $q \in \Phi^{-1}(\Phi(Reach(\Phi^{-1}(\Phi(p)), S)))$. Then there exists $q' \in Reach(\Phi^{-1}(\Phi(p)), S)$ with $\Phi(q') = \Phi(q)$. Let $p' \in \Phi^{-1}(\Phi(p))$ be such that $q' \in Reach(p', S)$. Since $\Phi(q) = \Phi(q')$ and $\Phi(p) = \Phi(p')$, strong consistency implies $q \in Reach(p, S)$.

Conversely, assume (4.3) holds. Let $q \in Reach(p, S)$ and p', q' be such that $\Phi(p') = \Phi(p)$, $\Phi(q') = \Phi(q)$. Then

$$\begin{aligned} q' \in \Phi^{-1}(\Phi(q)) &\subseteq \Phi^{-1}(\Phi(Reach(p, S))) \\ &\subseteq \Phi^{-1}(\Phi(Reach(\Phi^{-1}(\Phi(p)), S))) \\ &= \Phi^{-1}(\Phi(Reach(\Phi^{-1}(\Phi(p')), S))) \\ &= Reach(p', S) \end{aligned}$$

Therefore, S is strongly consistent. ☐

Since strong consistency is a more restrictive notion, it is natural that condition (4.3) is stronger than condition (4.2) for consistency.

Proposition 4.11 (Controllability Equivalence). *Consider control systems $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ which are Φ-related with respect to smooth surjection $\Phi : M \longrightarrow N$. Assume that S_M is an implementation of S_N, and S_M is strongly consistent. Then S_N is controllable if and only if S_M is controllable.*

Proof. Let $p_1, p_2 \in M$ any points. Let $q_1 = \Phi(p_1)$ and $q_2 = \Phi(p_2)$. Since S_N is controllable, there exists a trajectory of S_N connecting q_1 and q_2. Since S_M is an implementation of S_N, there exists a trajectory of S_M connecting some $p_1' \in \Phi^{-1}(q_1)$ and some $p_2' \in \Phi^{-1}(q_2)$. Then, since S_M is strongly consistent, there is a trajectory of S_M connecting p_1 to p_2. The other direction is given by Theorem 4.2. ☐

In this section we identified the relevant notions for the study of controllability in Φ-related systems. We also described them for arbitrary systems in terms of reachable sets. In the following sections we give concrete characterizations of these concepts for linear systems. Moreover, we show how to use them to construct explicit Φ-related systems.

5. Consistent Linear Abstractions

The notion of Φ-related control systems is now specialized for the case of linear, time invariant systems with linear aggregation maps. Consider the linear control systems

$$\begin{aligned} (\Sigma_1) \quad & \dot{x} = Ax + Bu \\ (\Sigma_2) \quad & \dot{y} = Fy + Gv \end{aligned}$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^k$, $y \in \mathbb{R}^m$, $v \in \mathbb{R}^l$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times k}$, $F \in \mathbb{R}^{m\times m}$, $G \in \mathbb{R}^{m\times l}$, and the onto, linear aggregation map $y = Cx$. Then by Definition 3.4, Σ_1 and Σ_2 are C-related if for all $x \in \mathbb{R}^n$, and $u \in \mathbb{R}^k$ there exists $v \in \mathbb{R}^l$ such that

$$C(Ax + Bu) = FCx + Gv \tag{5.1}$$

By Proposition 3.5, given any control system and any map Φ, there always exists another control system which is Φ-related to it. It is clear from the proof that in the linear case the new system can also be chosen linear. We are interested, however, in a constructive methodology for generating Φ-related systems. The following proposition gives us a systematic way to generate C-related linear abstractions of a linear system with respect to a linear aggregation map $y = Cx$.

Proposition 5.1 (Construction of Linear Abstractions). *Consider the linear system*

$$(\Sigma_1) \qquad \dot{x} = Ax + Bu$$

and a surjective map $y = Cx$. Let

$$(\Sigma_2) \qquad \dot{y} = Fy + Gv$$

be the system where

$$\begin{aligned} F &= CAC^+ \\ G &= [CB \;\; CAv_1 \;\; \ldots \;\; CAv_r] \end{aligned}$$

with C^+ the pseudoinverse of C and $v_1, \ldots, v_r$ spanning $Ker(C)$. Then Σ_1 and Σ_2 are C-related.

Proof. We need to show that for all $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^k$, there exists $v \in \mathbb{R}^l$ such that

$$\begin{aligned} C(Ax + Bu) &= Fy + Gv \qquad \text{or equivalently} \\ Gv &= CBu + (CA - FC)x \end{aligned}$$

Clearly, CBu belongs in the range of G for all u. Decompose $\mathbb{R}^n = Ker(C) \oplus Ker(C)^\perp$. If $x \in Ker(C)^\perp$ then $C^+Cx = x$ and thus

$$(CA - FC)x = (CA - CAC^+C)x = 0$$

If $x \in Ker(C)$ then $(CA - FC)x = CAx$ which also belongs in the range of G. $\square$

Proposition 5.1 is already interesting as it constructively generates for linear systems the so called virtual inputs used in backstepping designs. As a special case suppose that $Ker(C) = Im(B)$. Then we can take as $v_1, \ldots, v_r$ the columns of B. The input vectors for Σ_2 are the images under C of the vectors Av_i, which correspond to the next r vectors in the controllability matrix of Σ_1. That is, the image under C of the first order Lie brackets of Σ_1 become the new input vectors for Σ_2. The following example illustrates the proposition.

Example 5.2. Consider again the double integrator

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned}$$

and the projection $y = x_1$. So here $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = [0 \quad 1]^T$, and $C = [1 \quad 0]$. Then $Ker(C) = span\{[0 \quad 1]^T\}$ and the procedure of Proposition 5.1 results in $F = 0$, $G = 1$, so

$$\dot{y} = v.$$

Now consider the dynamics of the oscillating vector field

$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -x_1\end{aligned}$$

with the same projection map $y = x_1$. Here $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. Then Proposition 5.1 results in the same control system (or better, differential inclusion)

$$\dot{y} = v$$

The fact that the coarser system may have control inputs, even though the original one did not, is clearly undesirable. However, as will be shown, this will be taken care of by the notion of consistency.

From linear systems theory we know that for the linear system

$$(\Sigma_1) \qquad \dot{x} = Ax + Bu$$

the reachable space from any $x_0 \in \mathbb{R}^n$ is given by

$$Reach(x_0, \Sigma_1) = \bigcup_{T \geq 0} e^{AT}x_0 + Reach(0, \Sigma_1) = \bigcup_{T \geq 0} e^{AT}x_0 + \mathcal{R}(A, B) \tag{5.2}$$

where

$$\mathcal{R}(A, B) = Im[B \; AB \; \ldots \; A^{n-1}B]$$

is the reachable space from the origin. In particular, system Σ_1 is controllable if and only if $\mathcal{R}(A, B) = \mathbb{R}^n$. As a corollary of Theorem 4.2 we obtain the following result.

Theorem 5.3 (Controllability Propagation for Linear Abstractions). *Consider the linear systems*

$$\begin{aligned}(\Sigma_1) \qquad \dot{x} &= Ax + Bu\\ (\Sigma_2) \qquad \dot{y} &= Fy + Gv\end{aligned}$$

which are C-related which respect to the surjective map $y = Cx$. Then

$$C\mathcal{R}(A, B) \subseteq \mathcal{R}(F, G)$$

In particular, if Σ_1 is controllable then Σ_2 is controllable.

Proof. Simple application of Theorem 4.2. □

In order to propagate controllability from the linear system Σ_2 to Σ_1, the notions of implementability and consistency where defined in Section 4.

Proposition 5.4 (Implementability Characterization for Linear Systems). *Consider two linear systems*

$$\begin{aligned}(\Sigma_1) \qquad \dot{x} &= Ax + Bu\\ (\Sigma_2) \qquad \dot{y} &= Fy + Gv\end{aligned}$$

and surjective map $y = Cx$. Then Σ_2 is implementable by Σ_1 if and only if for all y we have

$$\bigcup_{T \geq 0} e^{FT}y + \mathcal{R}(F, G) \subseteq \bigcup_{T \geq 0} \bigcup_{x \in C^{-1}(y)} Ce^{AT}x + C\mathcal{R}(A, B) \tag{5.3}$$

Proof. Follows from Proposition 4.4 and Equation (5.2). □

By Theorem 3.7, C-relatedness of Σ_1 and Σ_2 is equivalent to the fact that trajectories of Σ_1 map to trajectories of Σ_2. We now take a look at the converse problem, namely when do trajectories of Σ_2 come from trajectories of Σ_1? This can be viewed as the trajectory implementation problem. It is essentially a (pseudo) invertibility problem or an exact tracking problem where the system generating the desired trajectory is C-related to the original system.

Theorem 5.5 (Trajectory Implementation). *Consider two linear systems*

$$(\Sigma_1) \qquad \dot{x} = Ax + Bu$$

$$(\Sigma_2) \qquad \dot{y} = Fy + Gv$$

and the surjective map $y = Cx$. Assume $x \in \mathbb{R}^n$ $y \in \mathbb{R}^m$ with $m \leq n$, and $u \in \mathbb{R}^k$ with $k \leq n$. Let B, G, and C be of full rank. Let $\mathcal{K} = Ker\ C$. We make the following two assumptions:

1. *$CAx = FCx$ for all $x \in \mathcal{K}^{\perp}$ (the orthogonal complement of $\mathcal{K}$).*
2. *$\{x\colon Cx \in (CA\mathcal{K})^{\perp}\} \subseteq \mathcal{B} = Im[B]$.*

Then for every trajectory $y(\cdot)$ of Σ_2 corresponding to a differentiable control there exists a trajectory $x(\cdot)$ of Σ_1 such that $y(t) = Cx(t)$ for all t in the domain of $y(\cdot)$.

Proof. Let $y(\cdot)$ be a trajectory of Σ_2 corresponding to the control v. First we define $x_a(t) = C^+y(t)$ where C^+ is the Moore-Penrose pseudo-inverse of C ($C^+ = C^T(CC^T)^{-1}$). If $z \in \mathcal{K}$ then

$$z^T x_a(t) = z^T C^T (CC^T)^{-1} y(t) = (Cz)^T (CC^T)^{-1} y(t) = 0.$$

Therefore, $x_a(t) \in \mathcal{K}^{\perp}$ for all t. Moreover, $\dot{x}_a(t) = C^+\dot{y}(t) = Fy(t) + Gv(t)$.

Let P denote the orthogonal projection from $\mathbb{R}^m$ onto $CA\mathcal{K}$. Let H^+ be the pseudo-inverse of CA considered as a map from $\mathcal{K}$ onto $CA\mathcal{K}$. Define $x_b(t)$ by $x_b(t) = H^+P(Gv(t))$. Notice that by construction, $x_b(t) \in \mathcal{K}$ and $Gv(t) - CAx_b(t)$ is orthogonal to $CA\mathcal{K}$ for all t. Since $v(\cdot)$ is differentiable, so is $x_b(\cdot)$. We then get

$$C(\dot{x}_a + \dot{x}_b) = C\dot{x}_a = \dot{y} = Fy + Gv = FCx_a + Gv = CAx_a + Gv$$

where the last equality holds by Assumption 1. Set $z(t) = \dot{x}_a(t) + \dot{x}_b(t) - Ax_a(t) - Ax_b(t)$. Then for all t, $Cz(t) = Gv(t) - CAx_b(t)$ is orthogonal to $CA\mathcal{K}$. By Assumption 2, for each t there is $u(t) \in \mathbb{R}^k$ such that $z(t) = Bu(t)$. In fact, we can take $u(t) = B^+z(t)$ so $u(\cdot)$ is continuous (here $B^+ = (B^TB)^{-1}B^T$ since $k \leq n$). Then if we let $x(t) = x_a(t) + x_b(t)$ we get $\dot{x}(t) = Ax(t) + Bu(t)$ and $Cx(t) = Cx_a(t) = y(t)$ for all t. □

The following theorem gives a simple characterization of consistency for linear systems in terms of subspace invariance.

Theorem 5.6 (Consistency Characterization for Linear Systems). *The linear system*

$$(\Sigma_1) \qquad \dot{x} = Ax + Bu$$

is consistent with respect to the map $y = Cx$ if and only if

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B) \tag{5.4}$$

Proof. First notice that for any set $\mathcal{V} \subseteq \mathbb{R}^n$ we have $C^{-1}(C\mathcal{V}) = \mathcal{V} + Ker(C)$.

Assume (5.4) holds. We must show consistency condition (4.2), which for linear systems requires, for all x that $C(Reach(x+Ker(C),\Sigma_1)) = C(Reach(x,\Sigma_1))$, or, equivalently

$$C\left(\bigcup_{T\geq 0} e^{AT}(x+Ker(C)) + \mathcal{R}(A,B)\right) = C\left(\bigcup_{T\geq 0} e^{AT}x + \mathcal{R}(A,B)\right). \tag{5.5}$$

Clearly, $CReach(x,\Sigma_1) \subseteq C(Reach(x+Ker(C),\Sigma_1)$. Condition (5.4) and A-invariance of $\mathcal{R}(A,B)$ imply that for all $T\geq 0$ we have

$$\begin{aligned} e^{AT}Ker(C) &\subseteq Ker(C)+\mathcal{R}(A,B) \qquad \text{and therefore} \\ Ce^{AT}Ker(C) &\subseteq C\mathcal{R}(A,B). \end{aligned}$$

This gives the other inclusion, proving consistency.

Conversely, assume that Σ_1 is consistent. Let $x_0 \in Ker(C)$. From (5.5) with $x=0$ we get for any $T>0$ there exists $r\in\mathcal{R}(A,B)$ such that $Ce^{AT}x_0 = Cr$. Therefore, $e^{AT}x_0 = x_0' + r$ for some $x_0' \in Ker(C)$.

We have therefore shown that for all $T>0$, $e^{TA}x_0 \in Ker(C)+\mathcal{R}(A,B)$. By using $\frac{de^{tA}}{dt} = Ae^{tA}$ and taking limits as $t\to 0$ we conclude that $Ax_0 \in Ker(C)+\mathcal{R}(A,B)$. □

Note that condition (5.4) requires $Ker(C)$ to satisfy

$$AKer(C) \subseteq Ker(C)+\mathcal{R}(A,B)$$

which is clearly weaker than the well known condition

$$AKer(C) \subseteq Ker(C)+\mathcal{R}(B)$$

for $Ker(C)$ to be a controlled invariant (or (A,B)-invariant) subspace.

Theorem 5.7 (Strong Consistency Characterization for Linear Systems). *The linear system*

$$(\Sigma_1) \qquad \dot{x} = Ax+Bu$$

is strongly consistent with respect to the map $y=Cx$ if and only if

$$Ker(C) \subseteq \mathcal{R}(A,B) \tag{5.6}$$

Proof. Assume Σ_1 is strongly consistent. Condition 4.3 for linear systems becomes

$$\bigcup_{T\geq 0} e^{AT}x + \mathcal{R}(A,B) = \bigcup_{T\geq 0} e^{AT}(x+Ker(C)) + \mathcal{R}(A,B) + Ker(C). \tag{5.7}$$

Using (5.7) with $x=0$ gives $\mathcal{R}(A,B) \supseteq Ker(C)$.

Conversely, assume (5.6) holds. By A-invariance of $\mathcal{R}(A,B)$ we get, for all $T\geq 0$,

$$e^{AT}Ker(C) \subseteq \mathcal{R}(A,B).$$

This gives the inclusion

$$\bigcup_{T\geq 0} e^{AT}x + \mathcal{R}(A,B) \supseteq \bigcup_{T\geq 0} e^{AT}(x+Ker(C)) + \mathcal{R}(A,B) + Ker(C).$$

The other inclusion always holds. □

Note that by the A-invariance of $\mathcal{R}(A,B)$, condition (5.6) is indeed stronger than condition (5.4). Consistency conditions (5.4) and (5.6) are rather intuitive. Condition (5.4) essentially says that whatever piece of $Ker(C)$ is not A-invariant can be compensated by controls and their Lie brackets. On the other hand, condition (5.6) is a form of controllability within the equivalence classes. The trajectories of the system which connect two points of the same equivalence class (as defined by C) are not, however, restricted to remain within the equivalence class. This is less restrictive than the notion of in-block controllability defined in [9, 11]. The following example illustrates the notions of implementability and consistency.

Example 5.8. Consider the linear system (without controls) $\dot{x} = Ax$, where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

and the C-related (one-dimensional) system $\dot{y} = Fy + Gv$, where

$$F = 0 \qquad G = 1.$$

The systems are obviously C-related. We also have

$$Ker(C) = \text{span}\{\begin{bmatrix} 0 \\ 1 \end{bmatrix}\} \quad AKer(C) = \text{span}\{\begin{bmatrix} 1 \\ 0 \end{bmatrix}\} \not\subseteq Ker(C).$$

Therefore, the system Σ_1 is not consistent. To show it is implementable we simply solve the system explicitly. Notice that since $\dot{y} = v$, any two points (of $\mathbb{R}$) can be connected by a trajectory of Σ_2 in arbitrary positive time. Let $y_0, y_f \in \mathbb{R}$. The curve

$$x_1(t) = \frac{y_f - y_0}{T}t + y_0$$
$$x_2(t) = \frac{y_f - y_0}{T}$$

is a trajectory of Σ_1 from $\begin{bmatrix} y_0 \\ \frac{y_f - y_0}{T} \end{bmatrix}$ to $\begin{bmatrix} y_f \\ \frac{y_f - y_0}{T} \end{bmatrix}$ at time T. Therefore, Σ_2 is implementable by Σ_1. Notice, that if $y_f \neq y_0$ there is not trajectory of Σ_1 connecting $\begin{bmatrix} y_0 \\ 0 \end{bmatrix}$ to any point x with $Cx = y_f$. The reason is that all the points $\begin{bmatrix} x_1 \\ 0 \end{bmatrix}$ are equilibria of Σ_1.

In order to propagate some form of controllability from Σ_2 to Σ_1, we need to check two properties, namely implementability and (strong) consistency. Unfortunately, Condition (5.3) is not easy to check since it involves the explicit integration of the differential equation. However, condition (5.3) in conjunction with consistency conditions (5.4) or (5.6) results in checkable characterizations of implementations which are also (strongly) consistent. To achieve this, we will need the following lemma.

Lemma 5.9. *Let A $(n \times n)$, C $(m \times m)$, F $(m \times m)$ and G $(m \times l)$ be matrices with $l \leq m$ and G of full rank. If for all $x \in \mathbb{R}^n$ $(CA - FC)x \in \mathcal{R}(F,G)$, then for all $t \geq 0$,*

$$(Ce^{tA} - e^{tF}C)x \in \mathcal{R}(F,G) \ .$$

In particular, the conclusion holds if A, F, are G are the corresponding matrices for the C-related systems Σ_1 and Σ_2.

Proof. We have the following identity for all $t \geq 0$

$$Ce^{tA} - e^{tF}C = \sum_{j=0}^{\infty}(CA^j - F^jC)\frac{t^j}{j!} \ . \tag{5.8}$$

We prove by induction the statement

$$(\mathrm{P}_j) \qquad \forall x \in \mathbb{R}^n \ \ (CA^j - F^jC)x \in \mathcal{R}(F, G)$$

It is clearly true for $j = 0$ and by hypothesis it is also true for $j = 1$. Assume P_i holds for $i \leq j$. We can write,

$$(CA^{j+1} - F^{j+1}C)x = (CA^j - F^jC)Ax + F^j(CA - FC)x \ .$$

By the inductive hypothesis applied to x and Ax, $(CA^j - F^jC)Ax \in \mathcal{R}(F, G)$ and $(CA - FC)x \in \mathcal{R}(F, G)$. But then $F^j(CA - FC)x \in \mathcal{R}(F, G)$ for all j since $\mathcal{R}(F, G)$ is F-invariant. Therefore,

$$(CA^j - F^jC)Ax + F^j(CA - FC)x \in \mathcal{R}(F, G) \ .$$

By taking the limit in (5.8) we conclude the proof. □

Theorem 5.10 (Implementability and Consistency Characterization). *Consider the linear systems*

$$(\Sigma_1) \qquad \dot{x} = Ax + Bu$$

$$(\Sigma_2) \qquad \dot{y} = Fy + Gv$$

which are C-related which respect to the surjective map $y = Cx$. Then Σ_2 is implementable by Σ_1 and Σ_1 is consistent if and only if

$$C\mathcal{R}(A, B) = \mathcal{R}(F, G) \tag{5.9}$$

In addition, Σ_2 is implementable by Σ_1 and Σ_1 is strongly consistent if and only if

$$\mathcal{R}(A, B) = C^{-1}(\mathcal{R}(F, G)) \tag{5.10}$$

Proof. Assume $C\mathcal{R}(A, B) = \mathcal{R}(F, G)$ and thus $\mathcal{R}(F, G) \subseteq C\mathcal{R}(A, B)$. Now let $x \in Ker(C)$. By C-relatedness, there exists $v \in \mathbb{R}^l$ such that $CAx = FCx + Gv = Gv$ (using $u = 0$ and since $Cx = 0$). So, $CAx \in R(F, G)$ and by assumption, there is $x_1 \in R(A, B)$ such that $Cx_1 = CAx$. Therefore, $Ax - x_1 \in Ker(C)$ and $Ax = Ax - x_1 + x_1 \in Ker(C) + R(A, B)$. Thus $AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B)$ and Σ_1 is consistent. We must now show that condition (5.3) holds. Consider any

$$y_f = e^{FT}y_0 + r_F^1 \in Reach(y_0, \Sigma_2) = \bigcup_{T \geq 0} e^{FT}y_0 + \mathcal{R}(F, G)$$

with $r_F^1 \in \mathcal{R}(F, G)$. By Lemma 5.9, we have that $e^{FT}y_0 = Ce^{AT}x_0 + Cr_F^2$ for some $r_F^2 \in \mathcal{R}(A, B)$, and for any x_0 with $y_0 = Cx_0$. But then

$$y_f = Ce^{AT}x_0 + r_F^1 + r_F^2 = Ce^{AT}x_0 + Cr_A \in \bigcup_{T \geq 0} \bigcup_{x \in C^{-1}(y_0)} Ce^{AT}x + C\mathcal{R}(A, B) = C(Reach(C^{-1}(y_0), \Sigma_1))$$

for some $r_A \in \mathcal{R}(A, B)$ since $\mathcal{R}(F, G) \subseteq C\mathcal{R}(A, B)$. Therefore Σ_2 is implementable by Σ_1.

For the converse notice that, since the systems are C-related, Proposition 5.3 implies $\mathcal{R}(F, G) \supseteq C\mathcal{R}(A, B)$. Moreover, the implementability condition (5.3) with $y = 0$ gives

$$\mathcal{R}(F, G) \subseteq \bigcup_{T \geq 0} Ce^{AT}Ker(C) + C\mathcal{R}(A, B).$$

And the consistency condition (5.5) with $x = 0$ gives

$$\bigcup_{T \geq 0} Ce^{AT} Ker(C) \subseteq C\mathcal{R}(A, B).$$

These two combined give $\mathcal{R}(F, G) \subseteq C\mathcal{R}(A, B)$. This concludes the proof of the first equivalence.

Now assume that $\mathcal{R}(A, B) = C^{-1}(\mathcal{R}(F, G))$. Then $C\mathcal{R}(A, B) = \mathcal{R}(F, G)$ and therefore Σ_1 implements Σ_2. Since $0 \in \mathcal{R}(F, G)$ we also have $Ker(C) \subseteq \mathcal{R}(A, B)$. Therefore Σ_1 is strongly consistent.

If Σ_1 is strongly consistent and implements Σ_2 then Σ_1 is also consistent and therefore must satisfy $C\mathcal{R}(A, B) = \mathcal{R}(F, G)$. Therefore, $\mathcal{R}(A, B) \subseteq C^{-1}(\mathcal{R}(F, G)) = \mathcal{R}(A, B) + Ker(C)$. By strong consistency $Ker(C) \subseteq \mathcal{R}(A, B)$, and thus $C^{-1}(\mathcal{R}(F, G)) \subseteq \mathcal{R}(A, B)$. Therefore $C^{-1}(\mathcal{R}(F, G)) = \mathcal{R}(A, B)$. □

We now have the main ingredients for propagating controllability from the coarser to the more complex model.

Theorem 5.11 (Consistency and Implementability imply Controllability). *Consider the linear systems*

$$(\Sigma_1) \qquad \dot{x} = Ax + Bu$$
$$(\Sigma_2) \qquad \dot{y} = Fy + Gv$$

which are C-related system with respect to the surjection $y = Cx$. Assume that Σ_1 implements Σ_2, and Σ_1 is consistent that is

$$C\mathcal{R}(A, B) = \mathcal{R}(F, G)$$

Then Σ_2 is controllable if and only if Σ_1 is macrocontrollable. If in addition Σ_1 is strongly consistent,

$$\mathcal{R}(A, B) = C^{-1}(\mathcal{R}(F, G))$$

then Σ_1 is controllable if and only if Σ_2 is controllable.

Proof. Same as the proof of Propositions 4.8 and 4.11. □

Thus in order to propagate controllability between two linear systems, we have to ensure that the systems are C-related and check either condition (5.9) or (5.10) depending on the notion of controllability that is needed. It is desirable to have a methodology for constructing C related systems with the desirable properties. Fortunately, for the C-related system constructed in Proposition 5.1, (strong) consistency implies implementability. In order to show this, we will need the following lemma.

Lemma 5.12. *Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times k}$, and full rank $C \in \mathbb{R}^{m \times n}$, be such that*

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B)$$

and let $F = CAC^+$. Then $C\mathcal{R}(A, B)$ is F-invariant, that is

$$FC\mathcal{R}(A, B) \subseteq C\mathcal{R}(A, B)$$

Proof. Let $y = Cx$ for $x \in \mathcal{R}(A, B)$ and consider

$$Fy = CAC^+y = CAC^+Cx$$

Decompose $x = x^c + x^n$ where $x^c \in Ker(C)$ and $x^n \in Ker(C)^{\perp}$. Then

$$Fy = CAC^+C(x^c + x^n) = CAx^n = CA(x - x^c)$$

Since $x \in \mathcal{R}(A,B)$ and $\mathcal{R}(A,B)$ is A-invariant, we get that $CAx \in C\mathcal{R}(A,B)$. By consistency, there exist $z^c \in Ker(C)$ and $z^r \in \mathcal{R}(A,B)$ such that

$$CAx^c = C(z^c + z^r) = Cz^r \tag{5.11}$$

Thus CAx^c also belongs in $C\mathcal{R}(A,B)$ resulting in $Fy \in C\mathcal{R}(A,B)$. □

Theorem 5.13 (Consistency implies Implementability). *Consider the linear system*

$$(\Sigma_1) \qquad \dot{x} = Ax + Bu$$

which is consistent with respect to the surjective map $y = Cx$. Let

$$(\Sigma_2) \qquad \dot{y} = Fy + Gv$$

be the system where

$$\begin{aligned} F &= CAC^+ \\ G &= [CB \; CAv_1 \; \dots \; CAv_r] \end{aligned}$$

with C^+ the pseudoinverse of C and $v_1, \dots, v_r$ spanning $Ker(C)$. Then Σ_2 is implementable by Σ_1.

Proof. By Theorem 5.3 we have that $\mathcal{R}(F,G) \supseteq C\mathcal{R}(A,B)$ and thus we only need to show that $\mathcal{R}(F,G) \subseteq C\mathcal{R}(A,B)$. Let $y_f \in \mathcal{R}(F,G)$. Then

$$y_f = \left[G \; FG \; \dots \; F^{m-1}G\right] x \tag{5.12}$$

for some $x \in \mathbb{R}^{ml}$. By an appropriate partition of $x = [x_1 \; x_2 \; \dots \; x_m]^T$ we get

$$y_f = Gx_1 + FGx_2 + \dots + F^{m-1}Gx_m \tag{5.13}$$

It suffices to show that $\mathcal{R}(G) \subseteq C\mathcal{R}(A,B)$ since then, by Lemma 5.12, we get that $\mathcal{R}(FG) \subseteq C\mathcal{R}(A,B), \dots, \mathcal{R}(F^{m-1}G) \subseteq C\mathcal{R}(A,B)$. Now consider

$$y_1 = Gx_1 = [CB \; CAv_1 \; \dots \; CAv_k] \begin{bmatrix} x_1^1 \\ x_1^2 \end{bmatrix} = CBx_1^1 + [CAv_1 \; \dots \; CAv_k]\, x_1^2 \tag{5.14}$$

Clearly, $CBx_1^1 \in C\mathcal{R}(A,B)$. By consistency we have

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(A,B) \tag{5.15}$$

and therefore for $i = 1, \dots, k$

$$Av_i = v_i^c + v_i^r \tag{5.16}$$

for some $v_i^c \in Ker(C)$ and $v_i^r \in \mathcal{R}(A,B)$. Thus

$$\begin{aligned} CAv_i &= C(v_i^c + v_i^r) = Cv_i^r \\ &= C\left[B \; AB \; \dots \; A^{n-1}B\right] q_i \end{aligned} \tag{5.17}$$

for some vectors q_i of appropriate dimension. But then

$$\begin{aligned} [CAv_1 \; \dots \; CAv_k]\, x_1^2 &= C\left[B \; AB \; \dots \; A^{n-1}B\right] [q_1 \; \dots \; q_k]\, x_1^2 \\ &= C\left[B \; AB \; \dots \; A^{n-1}B\right] X_1^2 \end{aligned} \tag{5.18}$$

and thus $\mathcal{R}(G) \in C\mathcal{R}(A,B)$. □

As a result of the above theorem, if we use Proposition 5.1 to construct our abstracted models, then consistency (or strong consistency) is the only condition on the aggregation map that is needed to propagate controllability.

Theorem 5.14 (Consistency Implies Controllability). *Consider the linear system*

$$(\Sigma_1) \qquad \dot{x} = Ax + Bu$$

and surjective map $y = Cx$. Let

$$(\Sigma_2) \qquad \dot{y} = Fy + Gv$$

be the C-related system where

$$\begin{aligned} F &= CAC^+ \\ G &= [CB \ CAv_1 \ \ldots \ CAv_r] \end{aligned}$$

with C^+ the pseudoinverse of C and $v_1, \ldots, v_r$ spanning $Ker(C)$. If

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B)$$

then Σ_2 is macrocontrollable if and only if Σ_1 is controllable. In particular, if

$$Ker(C) \subseteq \mathcal{R}(A, B)$$

then Σ_1 is controllable if and only if Σ_2 is controllable.

Proof. Follows from Theorems 5.11 and 5.13. □

It is interesting to notice what happens to conditions (5.6) and (5.4) when the linear system is a linear vector field and thus $B = 0$. In that case, condition (5.4) reduces to

$$AKer(C) \subseteq Ker(C)$$

which, recall from Section 2, is the necessary and sufficient condition to obtain a well defined quotient vector field. Therefore a consistent abstraction of a linear vector field cannot have any control inputs (or cannot be a differential inclusion). Condition (5.6) reduces to

$$Ker(C) = \{0\}$$

and thus $y = Cx$ must be an invertible linear transformation (since it is already surjective). We will be typically interested in consistent abstractions which are *nontrivial*, in the sense that some state space reduction is performed (thus $Ker(C) \neq \{0\}$), but the abstracted system is also nontrivial ($Ker(C) \neq \mathbb{R}^n$).

Corollary 5.15. *Consider the assumptions of Theorem 5.14 and assume that $0 < rank(B) < n$. Then a nontrivial, strongly consistent abstraction always exists.*

Proof. If $rank(B) > 0$ then we can always find a linear map C such that $Ker(C) = Im[B]$. □

Theorem 5.14 and Corollary 5.15 are important as they show that *a consistent abstraction always exists as long as there are control inputs.* If $B = 0$, then we are left with a linear vector field, and in order to abstract a vector field we must satisfy the restrictive Φ-related conditions of Section 2. Therefore, modeling hierarchies are more meaningful for control systems than differential equations since the existence of control always allows us to have a coarser, higher level model. In addition, the notions of consistency are important from a hierarchical perspective as they provide good design principles for constructing valid hierarchies. For example, the condition for strong consistency, $Ker(C) \subseteq \mathcal{R}(A, B)$, suggests that in order to ignore dynamics at a higher level (captured by Ker(C)) then one would have to ensure the ignored dynamics can be accommodated at the lower level.

6. Hierarchical Controllability Algorithm

In this section, we will take advantage of the results of Section 5 in order to analyze the controllability of large scale linear systems. Theorem 5.14 enables us to have a hierarchical controllability criterion which decomposes the controllability problem into a sequence of smaller problems. Such an approach is numerically more efficient or robust than the standard Kalman rank and Popov-Belevitch-Hautus (PBH) eigenvalue tests.

Conceptually the algorithm, starts with the linear system in question, and determines the number of linearly independent input vector fields. If this number is zero, then the system is uncontrollable and the algorithm terminates. If the number of linearly independent inputs is equal to the number of states, then the system is trivially controllable and the algorithm terminates as well. If the number of linearly independent vector fields is less than the number of states but greater than zero, then by Corollary 5.15 we can always find an aggregation matrix C satisfying the strong consistency condition $Ker(C) \subseteq \mathcal{R}(A, B)$. Since $Im[B\ AB\ \ldots\ A^kB] \subseteq Im[B\ AB\ \ldots\ A^{n-1}B]$ for any $0 \leq k \leq n-1$, from a computational standpoint, we can actually choose any matrix C satisfying $Ker(C) = Im[B\ AB\ \ldots\ A^kB]$ for $0 \leq k \leq n-1$. If $k = 0$, then the abstracted system essentially ignores the directions spanned by the input vector fields (which are trivially controllable). As k goes up, we not only ignore the directions of the input vector fields, but also their Lie brackets with the drift dynamics. If $k = n-1$ then the matrix C will ignore the whole reachable space.

After a consistent C matrix is determined, the construction of Theorem 5.14 is used in order to obtain a system of smaller dimension with equivalent controllability properties. We recursively apply the same procedure to this new abstracted system. Eventually, by dimension count, either there will be no inputs left and the system will be trivially uncontrollable, or there should be as many linearly independent inputs as number of states in which case controllability follows trivially. Since at each step, the abstractions that are constructed are consistent, then by Theorem 5.14, the outcome of the algorithm at the coarsest level will propagate along this sequence of consistent abstractions to the original complex model.

Algorithm 6.1. (Hierarchical Controllability Algorithm)

1. Start with system $\dot{x} = Ax + Bu$, $A \in \mathbb{R}^{n \times n}$, $0 \leq k \leq n-1$
2. If $rank(B)$ is
   - 0 : System is uncontrollable. Algorithm Terminates
   - n : System is controllable. Algorithm Terminates
3. Find matrix C such that $Ker(C) = Im[B\ AB\ \ldots A^kB]$
4. Obtain new system matrices A, B of the abstracted system using Theorem 5.14
5. Return to 2

The higher the order of the Lie brackets (the larger k is), the fewer steps the algorithm will need to terminate. On the other hand, as k increases, the amount of computation per step will be higher. Before we discuss computational and implementation aspects of the above algorithm, we will demonstrate its application on various examples.

Example 6.2. Consider the linear system

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u = A_1 x + B_1 u \tag{6.1}$$

Since there is one linearly independent input field, we can find a consistent abstraction satisfying

$$Ker(C_1) = Im[B_1] \subseteq Im[B_1\ A_1B_1\ A_1^2B_1]$$

For example, we can choose

$$C_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The construction of Theorem 5.14, then results in

$$A_2 = C_1A_1C_1^+ = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{6.2}$$

Since B_2 is nonzero and the number of linearly independent inputs is strictly less than the number of states, we can obtain another consistent abstraction by choosing $C_2 = [1\ \ 0]$. The resulting abstraction is

$$A_3 = C_2A_2C_2^+ = 0 \qquad B_3 = 1 \tag{6.3}$$

At this point, the number of inputs is equal to the number of states and thus the pair (A_3, B_3) is trivially controllable. By consistency, the pairs (A_2, B_2) and (A_1, B_1) are also controllable.

There is a much more intuitive explanation of the sequence of steps taken above. Note that the system started with the pair (A_1, B_1) and in the first iteration, we essentially removed the dynamics of x_2 (second row) from equation (6.1) since they have direct connection to the input u. This results in the pair (A_2, B_2). We re-apply the above procedure by now removing the dynamics of x_3 (second row of (6.2)) since they can be directly controlled by the new controls. This results in the pair (A_3, B_3) which is trivially controllable.

Example 6.3. Consider the linear system

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u = A_1x + B_1u \tag{6.4}$$

A consistent abstraction results by choosing the aggregation matrix

$$C_1 = \begin{bmatrix} -1 & 1 \end{bmatrix}$$

resulting in

$$A_2 = C_1A_1C_1^+ = 0 \qquad B_2 = 0 \tag{6.5}$$

Therefore, by Theorem 5.14, the pairs (A_2, B_2) and (A_1, B_1) are both uncontrollable.

In the case where we select $k = 0$ in Algorithm 6.1, then we choose matrices C satisfying $Ker(C) = Im[B]$. In this particular case $CB = 0$, and in addition the columns of B span $Ker(C)$. From a compuatational standpoint, it is advantageous to actually choose a matrix C which not only satisfies $Ker(C) = Im[B]$ but is also a projection to $Im[B]^\perp$. This reduces some of the computations of Theorem 5.14 and results in the following variation of Algorithm 6.1.

Algorithm 6.4. (Hierarchical Controllability Algorithm)

1. Start with system $\dot{x} = Ax + Bu$, $A \in \mathbb{R}^{n\times n}$.
2. If $rank(B)$ is
   - 0 : System is uncontrollable. Algorithm Terminates
   - n : System is controllable. Algorithm Terminates
3. Find matrix C such that $Ker(C) = Im[B]$
4. Let $A := CAC^+$, $B := CAB$

5. Return to 2

It is quite interesting to obtain some intuition of Algorithm 6.4. The algorithm starts with the system in question and, since $Im[B]$ is in the controllable region, chooses an abstraction matrix C which essentially projects the system in a direction which is orthogonal to space spanned by B. Thus the macroinputs of the first abstraction are spanned by CAB, which are the first order Lie brackets, *projected on the orthogonal complement of* $Im[B]$. Similarly, the second abstraction will have as input vector fields the second order Lie brackets projected on the orthogonal complement of both $Im[B]$ and $Im[AB]$. Because of this smart selection of inputs at each abstraction layer, we simply have to add the dimension of the span of the input vector fields at each abstraction layer in order to obtain the dimension of the controllability subspace. From the above discussion, it is also clear that, if the system is uncontrollable, then the algorithm computes the uncontrollable part of the system since at each iteration we are projecting on space orthogonal to the reachable space. The sequence of abstracting maps can then be used in a straightforward manner in order to decompose the system to controllable and uncontrollable subsystems.

We now focus on the implementation issues of Algorithms 6.1 and 6.4. For simplicity, we consider Algorithm 6.4 ; Algorithm 6.1 can be treated in a similar manner. From a computational perspective, the two main problems for implementing Algorithm 6.4 are: first, the construction of a consistent aggregation matrix C satisfying $Ker(C) = Im[B]$, and second, given such a matrix, to perform the computations required for the construction of a consistent abstraction. In order to tackle the first problem, we perform a singular value decomposition decomposition on the matrix B. The $n \times m (n \geq m)$ matrix B with rank r is decomposed as

$$B = U\Sigma V^T = [U_1 \; U_2] \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_r V_1^T \tag{6.6}$$

where Σ_r is the $r \times r$ matrix of nonzero singular values. From the above decomposition we immediately obtain that $Ker(C) = Im[B] = Im[U_1]$ and we can therefore choose the abstracting map $C = U_2^T$. In addition, $C^+ = U_2$ and therefore the singular value decomposition gives us for free the pseudoinverse calculation. Similar constructions are used in the implementation of Algorithm 6.1. The Matlab code that implements Algorithms 6.1 and 6.4 can be found in Appendix A.

It is quite remarkable that the implementation of Algorithm 6.4 is identical to the controllability algorithm of [12], derived from a purely numerical analysis perspective. In [12], the above algorithm is shown to be numerically stable and is a stabilized version of the realization algorithm of [33] (Matlab command CTRBF). This can be seen by the fact that the main operations of the algorithm are the singular value decomposition and multiplication by orthogonal matrices which are very well conditioned. Of course, singular value decompositions are computationally expensive. If speed of computation is of great interest, then QR type decompositions could be used instead of singular value decompositions in order to accelerate the algorithm. However, as is typical in such cases, this may result in less robust algorithm.

Various experimental, comparative studies were performed on a Matlab platform. Given the dimension of the state and input space, random A, B matrices were generated, and their controllability was checked using the Kalman rank condition, the PBH test and Algorithm 6.4. Floating point operations were measured for each test, and the following ratios

$$\text{Ratio} = \frac{\text{Floating Point Operations of Kalman or PBH Test}}{\text{Floating Point Operations of Algorithm 6.4}}$$

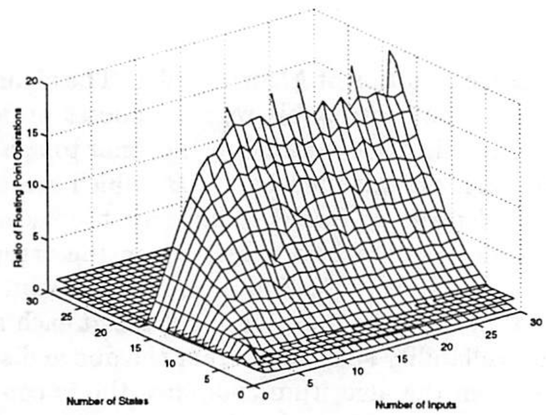

FIGURE 3. Comparison of Algorithm 6.4 and the Kalman rank condition

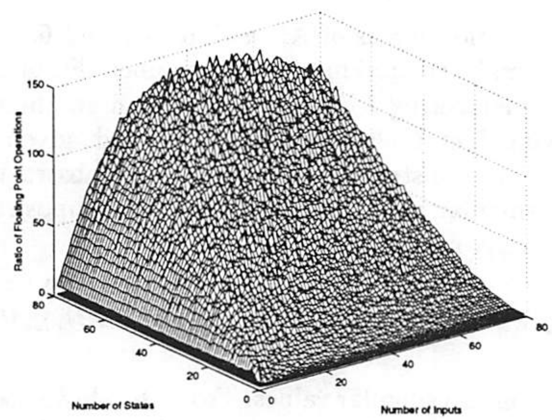

FIGURE 4. Comparison of Algorithm 6.4 and the Popov-Belevitch-Hautus test

are plotted as a function on state and input dimension in Figures 3 and 4. The plane with ratio equal to one is also plotted. Whenever the unreliable Kalman rank test fails to recognize a controllable system, the ratio is set to zero. Note from Figure 3, that the Kalman rank test is more efficient for very low dimensional systems but Algorithm 6.4 is up to 15 times faster for most systems. In addition, the Kalman condition fails to be reliable for systems with more than approximately 15 states. Figure 4 compares the PBH test with Algorithm 6.4. Even though the PBH test is more reliable than the Kalman rank condition, it is significantly slower than Algorithm 6.4 (up to 150 times for some systems). In addition, it is well known (see [28]) that the PBH test is very sensitive to parameter perturbations due to eigenvalue calculations. Finally, Figure 5 compares Algorithm 6.4 and Algorithm 6.1 with $k = 1$. Figure 5 clearly shows that it may be advantageous to use Algorithm 6.1 with $k = 1$ only in cases where the state dimension is much larger than the input dimension. Similar experiments wih higher values of k did not result in significant accelerations of the algorithms.

The fact that the implementation of a particular case of Algorithm 6.1 (Algorithm 6.4) coincides with the best known algorithm from numerical linear algebra, is strong evidence that the research direction presented in this paper is indeed reducing the complexity of control algorithms and is worthwhile pursuing for more general classes of systems (nonlinear) as well as for other properties of interest (stabilizability, optimality, trajectory tracking).

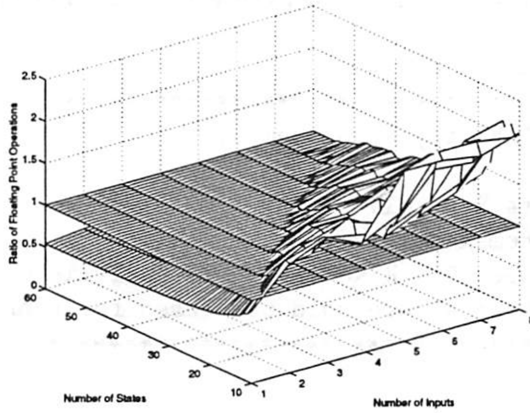

FIGURE 5. Comparison of Algorithm 6.4 and Algorithm 6.1 with $k = 1$

7. Issues for Further Research

In this paper, we considered a notion of control system abstractions which are typically used in hierarchical and multi-layered systems. This was achieved by generalizing the notion of Φ-related vector fields to control systems. This notion is more general than the notion of projectable control systems [19, 22] and, in addition, mathematically formalizes the concept of virtual inputs used in backstepping designs [15]. The notions of implementability and consistency were then defined in order to propagate controllability from the abstracted system to the more detailed one. These notions were completely characterized for linear systems, and the easily checkable conditions allowed us to construct a hierarchical controllability algorithm for linear systems.

There are many directions for further future research. The results of Section 5 enable the development of an open loop backstepping methodology which, given a sequence of consistent abstractions would recursively generate the actual control input, by first generating a control input for the abstracted system and then recursively refine it as one adds more modeling detail. Nonlinearizing the results of Section 5, will result in a hierarchical controllability algorithm for nonlinear system which may be more efficient and robust from a symbolic computation point of view. Many other properties are also of interest and will be investigated both for linear and nonlinear control systems. For example, obtaining consistent abstractions for nonlinear systems with respect to stabilizability would essentially classify all backsteppable systems. Other properties of interest include trajectory tracking, optimality as well as the proper propagation of state and input constraints. The framework presented in this paper provides a suitable platform for such studies.

Finally, another direction which is of great interest from a hybrid systems perspective, is to obtain consistent, discrete and hybrid abstractions of continuous systems. A very interesting problem, however, remains the construction of finite and consistent state space partitions, given a continuous control system. An algorithm for constructing finite, reachability-preserving quotients of analytic vector fields is proposed in [20].

Acknowledgments: This work is supported by the Army Research Office under grants DAAH 04-95-1-0588 and DAAH 04-96-1-0341.

APPENDIX A. MATLAB IMPLEMETATION OF ALGORITHMS 6.1 AND 6.4

```
function [controllable]=HCA(A,B,k,tol)
%**************************************************************
%   Hierarchical Controllability Algorithm 6.1
%
%  Function Call    : HCA(A,B,k,tol)
%  Required Inputs  : System Matrices A,B
%  Optional Inputs  : Integer 0<= k <= n-1     (default is 0)
%                   : Tolerance threshold tol (used for rank computation)
%**************************************************************
n=size(A,1);
if nargin == 2
    k   = 0;
    tol = n*norm(A,1)*eps;
elseif nargin == 3
    tol = n*norm(A,1)*eps;
end
r = rank(B,tol);                %*** Dimension of input space

while ( (n>r) & (r>0) ),        %*** If inputs exist and are less than states
        l = min(k,n-1);         %*** Ignore Lie brackets higher than n-1
        W = B;                  %*** Compute [B AB ...A^kB]
        for j=1:l,
           W = [B A*W];
        end
        [U,S,V] = svd(W);       %*** Obtain consistent matrix C
        m  = rank(S,tol);
        U1 = U(:,1:m) ;
        U2 = U(:,(m+1):n) ;
        C  = U2';
        B  = C*A*U1;            %*** Obtain consistent abstraction
        A  = C*A*C';
        n  = size(A,1)          %*** Dimension of abstracted system
        r  = rank(B,tol);       %*** Dimension of macroinputs
end

if (n==r)
        controllable=1;
elseif (r==0)
        controllable=0;
end
```

```
function [controllable]=HCA(A,B,tol)
%*************************************************
% Hierarchical Controllability Algorithm 6.4
%
%  Function Call    : HCA(A,B,tol)
%  Required Inputs : System Matrices A,B
%  Optional Input   : Tolerance threshold tol
%*************************************************

n=size(A,1);
if nargin == 2
    tol = n*norm(A,1)*eps;
end

[U,S,V] = svd(B);              %*** Dimension of input space
r = rank(S,tol);

while ( (n>r) & (r>0) ),       %*** If inputs exist and are less than states
        U1 = U(:,1:r) ;        %*** Obtain consistent matrix C
        U2 = U(:,(r+1):n) ;
        C = U2';
        B = C*A*U1;            %*** Obtain consistent abstracted system
        A = C*A*C';
        n = size(A,1);         %*** Dimension of abstracted system
        [U,S,V] = svd(B);
        r = rank(S,tol);       %*** Dimension of macroinputs
end

if (n==r)
        controllable=1;
elseif (r==0)
        controllable=0;
end
```

REFERENCES

[1] R. Abraham, J. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis and Applications*. Applied Mathematical Sciences. Springer-Verlag, 1988.

[2] R. Alur, T.A. Henzinger, and E.D. Sontag, editors. *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.

[3] P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors. *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.

[4] P Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors. *Hybrid Systems IV*, volume 1273 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[5] P.J. Antsaklis, J.A. Stiver, and M. Lemmon. Hybrid system modeling and autonomous control systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 366–392. Springer-Verlag, 1993.

[6] M. Aoki. Control of large scale dynamic systems by aggregation. *IEEE Transactions on Automatic Control*, 13(3):246–253, June 1968.

[7] R. Brockett. Control theory and analytical mechanics. In C. Martin and R. Hermann, editors, *Geometric Control Theory*, Lie Groups: History, Frontiers and Applications, pages 1–46. Math. Sci. Press, 1977.

[8] R. Brockett. Global descriptions of nonlinear control problems; vector bundles and nonlinear control theory. manuscript, 1980.

[9] P. Caines and Y.J. Wei. The hierarchical lattices of a finite state machine. *Systems and Control Letters*, 25:257–263, 1995.

[10] P. Caines and Y.J. Wei. Hierarchical hybrid control systems. In S. Morse, editor, *Control Using Logic Based Switching*, volume 222 of *Lecture Notes in Control and Information Sciences*, pages 39–48. Springer Verlag, 1996.

[11] P. Caines and Y.J. Wei. Hierarchical hybrid control systems: A lattice theoretic formulation. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, April 1998. To appear.

[12] P. M. Van Dooren. The generalized eigenstructure problem in linear system theory. *IEEE Transactions on Automatic Control*, AC-26(1):111–129, 1981.

[13] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors. *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.

[14] T.A. Henzinger and H. Wong-Toi. Linear phase-portrait approximations for nonlinear hybrid systems. In R. Alur, T.A. Henzinger, and E.D. Sontag, editors, *Hybrid Systems III*, Lecture Notes in Computer Science 1066, pages 377–388. Springer-Verlag, 1996.

[15] M. Kristic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and Adaptive Control Design*. Adaptive and Learning systems for signal processing, communications and control. John Wiley & Sons, New York, 1995.

[16] C.P. Kwong. Optimal chained aggregation for reduced order modeling. *International Journal of Control*, 35(6):965–982, 1982.

[17] C.P. Kwong. Disaggregation, approximate disaggregation, and design of suboptimal control. *International Journal of Control*, 37(4):843–854, 1983.

[18] C.P. Kwong and C.F. Chen. A quotient space analysis of aggregated models. *IEEE Transactions on Automatic Control*, 27(1):203–205, February 1982.

[19] C.P. Kwong and Y.K. Zheng. Aggregation on manifolds. *International Journal of Systems Science*, 17(4):581–589, 1986.

[20] G. Lafferriere, G.J. Pappas, and S. Sastry. Subanalytic stratifications and bisimulations. In T. Henzinger and S. Sastry, editors, *Hybrid Systems : Computation and Control*, Lecture Notes in Computer Science. Springer Verlag, 1998. To appear.

[21] O. Maler, editor. *Hybrid and Real-Time Systems*, volume 1201 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[22] L.S. Martin and P.E. Crouch. Controllability on principal fibre bundles with compact structure group. *Systems and Control Letters*, 5(1):35–40, 1984.

[23] M.D. Mesarovic. *Theory of hierarchical, multilevel, systems*, volume 68 of *Mathematics in Science and Engineering*. Academic Press, New York, 1970.

[24] A.N. Michel and K. Wang. *Qualitative Theory of Dynamical Systems: The Role of Stability Preserving Mappings*. Monographs and Textbooks in Pure and Applied Mathematics. Marcel-Dekker, 1995.

[25] J.R. Munkres. *Analysis on Manifolds*. Addison-Wesley, 1991.

[26] T. Niinomi, B.H. Krogh, and J.E.R. Cury. Synthesis of supervisory controllers for hybrid systems based on approximating automata. In *Proceedings of the 1995 IEEE Conference on Decision and Control*, pages 1461–1466, New Orleans, LA, December 1995.

[27] H. Nijmeijer and A.J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, 1990.

[28] C. C. Paige. Properties of numerical algorithms related to computing controllability. *IEEE Transactions on Automatic Control*, AC-26(1):111–129, 1981.

[29] G. J. Pappas and S. Sastry. Towards continuous abstractions of dynamical and control systems. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems IV*, volume 1273 of *Lecture Notes in Computer Science*, pages 329–341. Springer Verlag, Berlin, Germany, 1997.

[30] G. J. Pappas, C. Tomlin, J. Lygeros, D. N. Godbole, and S. Sastry. A next generation architecture for air traffic management systems. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 2405–2410, San Diego, CA, December 1997.

[31] A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusions. In *Computer Aided Verification*, pages 95–104, 1994.

[32] J Raisch and S.D. O'Young. Discrete approximations and supervisory control of continuous systems. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, April 1998. To appear.

[33] H.H. Rosenbrock. *State Space and Multivariable Theory*. Jon Wiley, New York, 1970.

[34] S. Sastry, G. Meyer, C. Tomlin, J. Lygeros, D. Godbole, and G. Pappas. Hybrid control in air traffic management systems. In *Proceedings of the 1995 IEEE Conference in Decision and Control*, pages 1478–1483, New Orleans, LA, December 1995.

[35] M. Spivak. *A Comprehensive Introduction to Differential Geometry*. Publish or Perish, 1979.

[36] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, AC-38(2):195–207, 1993.

[37] P. Varayia. Towards a layered view of control. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 1187–1190, San Diego, CA, December 1998.

[38] K.C. Wong and W.M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems*, 6:241–273, 1995.

[39] K.C. Wong and W.M. Wonham. Hierarchical control of timed discrete-event systems. *Discrete Event Dynamic Systems*, 6:275–306, 1995.

[40] W.M. Wonham. *Linear Multivariable Control : A Geometric Approach*, volume 10 of *Applications of Mathematics*. Springer-Verlag, New York, 1985.

[41] H. Zhong and W.M. Wonham. On the consistency of hierarchical supervision in discrete-event systems. *IEEE Transactions on Automatic Control*, 35(10):1125–1134, 1990.

April 6, 1998

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES, UNIVERSITY OF CALIFORNIA AT BERKELEY, BERKELEY, CA 94720

E-mail address: gpappas@eecs.berkeley.edu

DEPARTMENT OF MATHEMATICAL SCIENCES, PORTLAND STATE UNIVERSITY, PORTLAND, OR 97207

E-mail address: gerardo@mth.pdx.edu

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES, UNIVERSITY OF CALIFORNIA AT BERKELEY, BERKELEY, CA 94720

E-mail address: sastry@eecs.berkeley.edu