Hierarchically Consistent Control Systems

George J. Pappas†, Gerardo Lafferriere‡, and Shankar Sastry†
†Department of EECS, University of California, Berkeley, CA 94720
‡Department of Mathematics, Portland State University, Portland, OR 97207
gpappas,sastry@eecs.berkeley.edu, gerardo@mth.pdx.edu

Abstract

In hierachical systems, higher levels of the hierarchy utilize coarser system models by aggregating the detailed lower level models. In this layered control paradigm, the notion of hierarchical consistency 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

The complexity of large scale systems is typically reduced by imposing a hierarchical structure on the system architecture. Each layer of the hierarchy uses a coarser system model than the lower levels. One of the main challenges in hierarchical systems is the extraction of a hierarchy of models at various levels of abstraction which are compatible with the objectives of each layer.

In the literature, the notions of *abstraction* or *aggregation* refer to grouping the system states into equivalence classes. Depending on the cardinality of the resulting quotient space we may have *discrete* or *continuous* abstractions. In this paper, we focus on *continuous abstractions*. More precisely, given a control system

$$\dot{x} = f(x,u) \qquad x \in \mathbb{R}^n \;\; u \in \mathbb{R}^m \tag{1}$$

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

$$\dot{y} = g(y,v) \qquad y \in \mathbb{R}^p \;\; v \in \mathbb{R}^k \tag{2}$$

which can produce as trajectories all functions of the form $y(t) = h(x(t))$, where $x(t)$ is a trajectory of system (1). System (2) will be referred to as the *abstraction* [8] or *macromodel* of the finer *micromodel* (1). Note that the control input v of the coarser model (2) is not the same input u of system (1) and should be thought of as a macroinput. This is therefore quite different from model reduction techniques which reduce or aggregate dynamics while using the same control inputs.

We will solve the abstraction problem by first generalizing the geometric notion of Φ-related vector fields to control systems. The notion of Φ-related control systems introduced in this paper is more general than the notion of projectable systems defined in [4, 5]. Furthermore, our notion of Φ-related control systems mathematically formalizes the concept of *virtual inputs* used in backstepping designs [3].

When an abstracted model is extracted from a more detailed model, one would like to determine conditions under which controllability of the abstracted system (2) implies controllability of system (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 notions of dynamic [1] and hierarchical consistency [10] have been defined for discrete event systems. In this paper, we will focus on controllability of linear control systems and characterize consistent linear abstractions. More precisely, given the linear control system

$$\dot{x} = Ax + Bu \qquad x \in \mathbb{R}^n \;\; u \in \mathbb{R}^m \tag{3}$$

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

$$\dot{y} = Fy + Gv \qquad y \in \mathbb{R}^p \;\; v \in \mathbb{R}^k \tag{4}$$

is controllable if and only if system (3) is controllable.

The above framework is also useful in the analysis of complex systems. In order to verify that a given large scale system satisfies certain properties, one tries to extract a simpler but qualitatively equivalent abstracted system. Checking the desired property on the abstracted system should be *equivalent* to checking the property on the original system. In this spirit, we obtain a hierarchical controllability criterion for large scale systems. The computational advantages of this approach are verified by recovering the best of the known controllability algorithms from numerical linear algebra [2] as a special case of the hierarchical controllability criterion.

The structure of this paper is as follows: In Section 2 we define a notion of control system abstractions. In Section 3 we define consistent abstractions and in Section 4 we restrict and characterize consistent linear abstractions. These results are used in Section 5 to obtain a hierarchical controllability criterion. Finally, Section 6 discusses issues for further research. Due to space limitations, proofs, examples, and algorithm implementations can be found in [7]. All mathematical objects are assumed smooth unless otherwise stated.

2 Control System Abstractions

In this section, we will develop notions of control system abstractions for rather general control systems.

Definition 2.1 *[6] 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, that is $\pi' \circ F = \pi$ 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 2.2 *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*

$$\pi \circ c^B = c$$
$$c' = c_*(1) = F \circ c^B$$

In local (bundle) coordinates, Definition 2.2 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 $\dot{x} = F(x, u)$. Note that even though Definition 2.2 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 now define Φ-related control systems.

Definition 2.3 *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 $p \in M$*

$$\Phi_* \circ F_M \left(\pi_M^{-1}(p)\right) \subseteq F_N \left(\pi_N^{-1}(\Phi(p))\right) \qquad (5)$$

Control system S_N will be referred to as an *abstraction* of control system S_M ([8]). 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 Φ-relatedness is transitive and that Definition 2.3 is a generalization of the notion of Φ-related vector fields. The following proposition, is an immediate consequence of Definition 2.3.

Proposition 2.4 *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.*

Consider, for example, 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 [3]. Other examples which demonstrate that the concept of Φ-related control systems is a generalization of the notion of *projectable* control systems [4, 5] can be found in [7].

The following theorem allows us to propagate trajectories between Φ-related control systems.

Theorem 2.5 *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.*

The quotient system thus overapproximates the abstracted trajectories of the original system. Therefore the macrosystem S_N may generate trajectories that are infeasible in the micromodel S_M.

3 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 3.1 *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 2.5 allows us to always propagate controllability from the micromodel to the macromodel for any aggregation map.

Theorem 3.2 *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.

Note that Theorem 3.2 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 3.3 *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[1] 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. Equivalently S_N is implementable by S_M if and only if for all $q \in N$,*

$$Reach(q, S_N) \subseteq \Phi(Reach(\Phi^{-1}(q), S_M)) \qquad (6)$$

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

We will be interested in implementability of Φ-related systems, in which case the above inclusion becomes an equality, by Theorem 3.2. 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 notion of consistency.

Definition 3.4 *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 $\Phi(q) = \Phi(q')$. Equivalently, S is consistent with respect to Φ if and only if for all $p \in M$,*

$$\Phi(Reach(\Phi^{-1}(\Phi(p)), S)) = \Phi(Reach(p, S)). \qquad (7)$$

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 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 3.5 *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 3.6 *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.*

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 3.7 *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'. Equivalently S is strongly consistent with respect to Φ if and only if for all $p \in M$,*

$$Reach(p, S) = \Phi^{-1}(\Phi(Reach(\Phi^{-1}(\Phi(p)), S))). \qquad (8)$$

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

Since strong consistency is a more restrictive notion, it is natural that condition (8) is stronger than condition (7) for consistency.

Proposition 3.8 *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.*

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 with the desirable properties.

4 Consistent Linear Abstractions

Consider the linear time-invariant control systems

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

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 surjective, linear aggregation map $y = Cx$. Then by Definition 2.3, 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

$$C(Ax + Bu) = FCx + Gv \tag{9}$$

The following proposition gives us a systematic way to generate C-related linear abstractions.

Proposition 4.1 *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 $F = CAC^+$, $G = [CB \quad CAv_1 \quad \ldots \quad CAv_r]$, with C^+ the pseudoinverse of C and $v_1, \ldots, v_r$ spanning $Ker(C)$. Then Σ_1 and Σ_2 are C-related.

It is immediate from Proposition 4.1 that an abstraction of a linear system with respect to a linear aggregation map can be also a linear system. Proposition 4.1 is interesting as it constructively generates for linear systems the so called *virtual inputs* used in backstepping designs [3]. In particular, if the aggregation map is a projection on some of the states, then the states that are ignored appear as inputs at the abstracted system.

From linear systems theory we know that for the system

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

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

$$Reach(x_0, \Sigma_1) = \bigcup_{T\geq 0} e^{AT}x_0 + \mathcal{R}(A, B) \tag{10}$$

where $\mathcal{R}(A, B) = Im[B \; AB \; \ldots \; A^{n-1}B]$. As a corollary of Theorem 3.2 we obtain the following result.

Theorem 4.2 *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 $y = Cx$. Then $C\mathcal{R}(A, B) \subseteq \mathcal{R}(F, G)$. In particular, if Σ_1 is controllable then Σ_2 is controllable.

In order to propagate controllability from the linear system Σ_2 to Σ_1, the notions of implementability and consistency where defined in Section 3. The following theorem gives a simple characterization of consistency for linear systems in terms of subspace invariance.

Theorem 4.3 *The linear system*

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

is consistent with respect to the map $y = Cx$ iff

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B) \tag{11}$$

It is strongly consistent with respect to $y = Cx$ iff

$$Ker(C) \subseteq \mathcal{R}(A, B) \tag{12}$$

Note that by the A-invariance of $\mathcal{R}(A, B)$, condition (12) is indeed stronger than condition (11). Condition (11) 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 (12) is a form of controllability within the equivalence classes. 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. Fortunately, implementability in conjunction with consistency conditions (11) or (12) results in checkable characterizations of implementations which are also (strongly) consistent.

Theorem 4.4 *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 $C\mathcal{R}(A,B) = \mathcal{R}(F,G)$. In addition, Σ_2 is implementable by Σ_1, and Σ_1 is strongly consistent if and only if $\mathcal{R}(A,B) = C^{-1}(\mathcal{R}(F,G))$.

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

Theorem 4.5 *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. In addition, if Σ_1 is strongly consistent, that is $\mathcal{R}(A,B) = C^{-1}(\mathcal{R}(F,G))$, then Σ_1 is controllable if and only if Σ_2 is controllable.

It is desirable to have a methodology for constructing C related systems with the desirable properties. Fortunately, if we use Proposition 4.1 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 4.6 *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 $F = CAC^+$, $G = [CB \;\; CAv_1 \;\; \ldots \;\; CAv_r]$ 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.

We will be 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 4.7 *Consider the assumptions of Theorem 4.6 with $0 < rank(B) < n$. Then a nontrivial, strongly consistent abstraction always exists.*

Theorem 4.6 and Corollary 4.7 are important as they show that *a consistent abstraction always exists as long as there are control inputs.* 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.

5 Hierarchical Controllability Algorithm

Theorem 4.6 enables us to have a hierarchical controllability criterion which decomposes the controllability problem into a sequence of smaller problems. 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 4.7 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 $0 \leq k \leq n-1$. This can be done using singular value decomposition or other (less robust) matrix decomposition techniques. 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 4.6 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 4.6, the outcome of the algorithm at the coarsest level will propagate along this sequence of consistent abstractions to the original complex model.

Algorithm 5.1 (Controllability Algorithm)

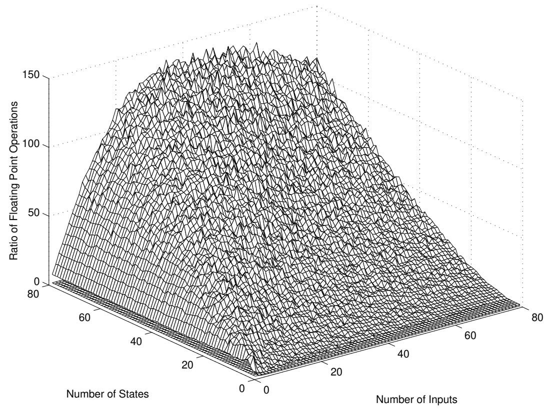

Figure 1: Comparison of Alg. 5.1 and the PBH Criterion

1. *Start with system $\dot{x} = Ax + Bu$, $0 \leq k \leq n-1$*
2. *If $rank(B)$ is*
   - *0 : System is uncontrollable. STOP*
   - *n : System is controllable. STOP*
3. *Find C with $Ker(C) = Im[B\ AB\ \ldots A^k B]$*
4. *Obtain new system matrices A, B of the abstracted system using Theorem 4.6*
5. *Return to 2*

The implementation issues and matlab code that implements Algorithm 5.1 can be found in [7]. Various experimental, comparative studies were performed on a matlab platform. Figure 1 compares the PBH test with Algorithm 5.1 where $k = 0$. Even though the PBH test is more reliable than the Kalman rank condition, it is significantly slower than Algorithm 5.1 (up to 150 times for some systems). The computational and conceptual advantages of Algorithm 5.1 are verified by the fact that Algorithm 5.1 with $k = 0$ is identical to the controllability algorithm of [2], derived from a purely numerical analysis perspective. In [2], the above algorithm is shown to be numerically stable and is a stabilized version of the realization algorithm of [9].

The fact that the hierarchical framework developed in this paper places a geometric and conceptual framework on the best of the known controllability algorithms from numerical linear algebra, is strong evidence that hierarchical decompositions of control problems are indeed reducing the complexity of control algorithms. It is therefore worthwhile pursuing this direction of research for more general classes of systems (nonlinear) as well as for other properties of interest (stabilizability, optimality, trajectory tracking).

6 Issues for Further Research

There are many directions for future research. The results of Section 4 enable the development of an open loop backstepping methodology. Nonlinear analogues of the results of Section 4, will provide a hierarchical controllability algorithm for nonlinear systems which may be more efficient and robust from a symbolic computation point of view. Obtaining consistent abstractions for nonlinear systems with respect to stabilizability would essentially classify backsteppable systems. The framework presented in this paper provides a suitable platform for such studies.

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

References

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

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

[3] 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.

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

[5] 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.

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

[7] G. J. Pappas, G. Lafferriere, and S. Sastry. Hierarchically consistent control systems. Technical Report UCB/ERL M98/16, University of California at Berkeley, Berkeley, CA, April 1998.

[8] 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.

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

[10] 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.

August 12, 1998