Reconfiguration in Hierarchical Control of Piecewise-Affine Systems

Tal Pasternak

Institute for Software Integrated Systems,
Department of Electrical Engineering and Computer Science
Vanderbilt University, P.O. Box 36, Peabody, Nashville, TN 37203
Tel: 1-615- 343-7472; Fax: 1-615- 343-7440
Tal.Pasternak@vanderbilt.edu

Abstract. The In this paper the problem of reconfiguration in hierarchical control of piecewise-affine systems in discrete time is considered as the choice of input constraints applied to the low-level control. It is shown how such reconfiguration can provide fault-tolerance to actuator faults while reducing the computational complexity of low-level control. The approach is based on partitioning the state space while taking into account multiple possibilities for the inputs available to low-level control. A so-called "reconfiguration database" is computed at design-time which determines the input constraints that provide for reachability between regions of a state-space partition. This database is used as a basis for reconfiguration decisions at runtime.

1 Introduction

The problem of control reconfiguration in fault-tolerant control is concerned with changing the input-output relation between a plant and its controller in such a way that ensures the achievement of a control objective [1]. Consider for example, the three-tank system in Figure 1. Valves and pumps are used by a controller in order to achieve a set-point of fluid levels. The choice of which valves and pumps are to be used by the controller is a reconfiguration decision. The system in this example is can be approximated as a piecewise-affine system in discrete-time.

Piecewise-affine systems have been receiving increasing attention by the control community because they provide a useful modeling framework for hybrid systems. Discrete-time piecewise-affine systems are equivalent to interconnections of linear systems and finite automata [2] and to a number of other hybrid models [3]. In particular, model predictive control can be applied to piecewise-affine systems by converting them to the equivalent mixed-logic dynamic form [4]. Another approach to control of piecewise affine systems, which is adopted in this paper, is hierarchical control.

Hierarchical control [5] includes low-level control, which may be implemented by model-predictive control, for example, and supervisory control, which operates on a discrete-event abstraction of the hybrid system. The discrete-event abstraction of the

closed-loop system, which includes the low-level control and the plant, is obtained by reachability calculations that take into account the available plant inputs, which the low-level control manipulates.

In this paper, the reconfiguration problem in hierarchical control of hybrid systems, modeled as piecewise-affine systems in discrete time, is formulated as the problem of selecting input constraints that guarantee reachability. The main contribution of this paper is the reduction of complexity for low-level control, which is achieved by selecting from configurations which each use a limited number of actuators.

In relation to the problem of control reconfiguration, hierarchical control can provide fault tolerance at both the supervisory control level and at the low level. Consider again the three-tank system in Figure 1. The objective of the control system is to regulate the fluid level in tank 3. If a leak occurs in Tank 1, the supervisory controller supervises a phased process by which tank 1 is emptied and tank 2 is filled until a configuration is achieved which mirrors the original configuration. In such a multi-phased process the supervisory controller determines set points to be achieved by the low-level control while low-level control achieves these set-points using the pumps and valves. In case of a leak in tank 1, fault-tolerance is achieved by the supervisory controller at a high level by commanding the shut-down of tank 1 and its replacement by tank 2. The low-level control reconfiguration provides fault-tolerance by choosing which pumps and valves to use at each phase in such a way that set-points are reached. For example, if valve V_2 is faulty, low level control will be implemented using valve V_{23}.

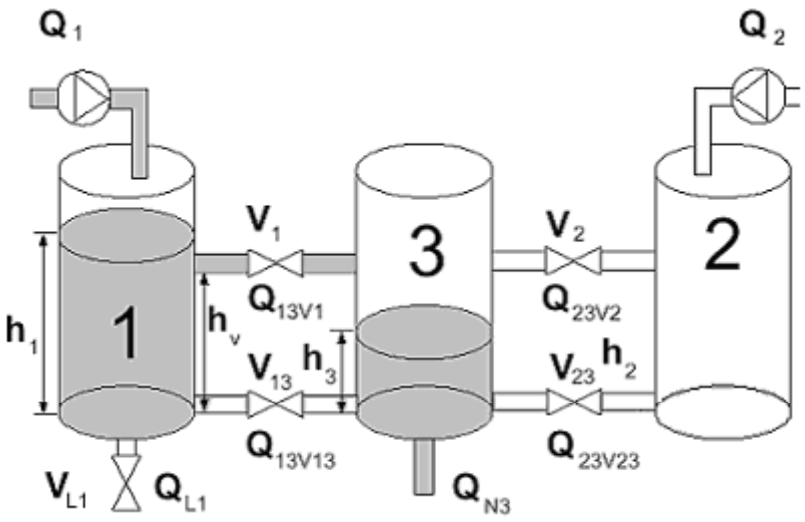

Fig. 1. Three Tank System with Tank 2 Empty

In this paper hierarchical control of piecewise-affine systems is proposed, based on partitioning the state and input space. The significance of considering the inputs when

generating the discrete abstraction of the hybrid system is twofold: it is important both for reconfiguration and for limiting the complexity of the low-level control. With respect to fault-tolerant control reconfiguration, the input constraints can be interpreted as control configurations (i.e. which actuators may be used and in what range) as well as fault conditions (i.e. which actuators are fixed in position or limited in range due to fault). With respect to the implementation of the low-level control, the constraints imposed on the inputs affect the complexity of the problem by determining the number of control variables that can be manipulated by the low-level controller [6]. For example, in the three-tank system there are four valves and two pumps, but as will be shown in the next section, only two of these six actuators need to be used at any given time. By not having to consider the operation of the other four "stand-by" actuators, the complexity of the low-level control is reduced.

The next section outlines the proposed architecture. Section 3, shows the application of the method to the problem of control reconfiguration of the three-tank system shown in Figure 1. Section 4 explains the reconfiguration process and section 5 describes the design of the supervisory controller. Finally, section 6 concludes with a discussion and survey of related work.

2 Architecture Overview

The plant is modeled as a piecewise-affine system with continuous states $X \subseteq R^n$, a finite set of discrete states Q, and inputs $U \subseteq R^m$ operating in a hybrid state space $Q \times X$. An additive state disturbance is assumed, taking values in a polyhedral region $D \subseteq R^l$. The system is described by a set of $|Q|$ affine state-space difference equations of the form (1),

$$x(t+1)= A^q x(t)+ B^q u(t)+f^q+d(t) \quad \text{if} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \chi_q . \tag{1}$$

where $\chi_q \subseteq X \times U$ are convex polyhedra (i.e. given by a finite number of linear inequalities) in the state and input space. The variables $x(t) \in X$, $u(t) \in U$, and $d(t) \in D$ denote state, in put and disturbance respectively at time t. Actuator faults are manifested as limitations which constrain the input values to a reduced input set $U_f \in U$. The control architecture is shown in Figure 2.

The fault and state detector identifies the plant state as a set $X_e \subseteq X$ which determines the possible values of the state vector $x(t)$. It also determines the disturbance set D and the fault-induced input constraints U_f. The implementation of the fault and state detector is beyond the scope of this paper. The sets X_e, D, U_f are assumed to be available correct conservative approximations, which are continually updated at each time step. All the sets are assumed to be convex polyhedra.

The system is designed with respect to some global control objective, as will be detailed in section 5. Based on the global control objective, the supervisory controller determines a control objective for the lower level control and the configuration manager.

Definition 1 (Control Objective). For a system operating in state space X at time t_0, a control objective (T, Ω, t) with T, $\Omega \subseteq X$, is to reach a state $x(t_0+k) \in T$, with $x(t_0+j) \in \Omega$, $\forall\ 1 \leq j \leq k\text{-}1$, for some $1 \leq k \leq t$.

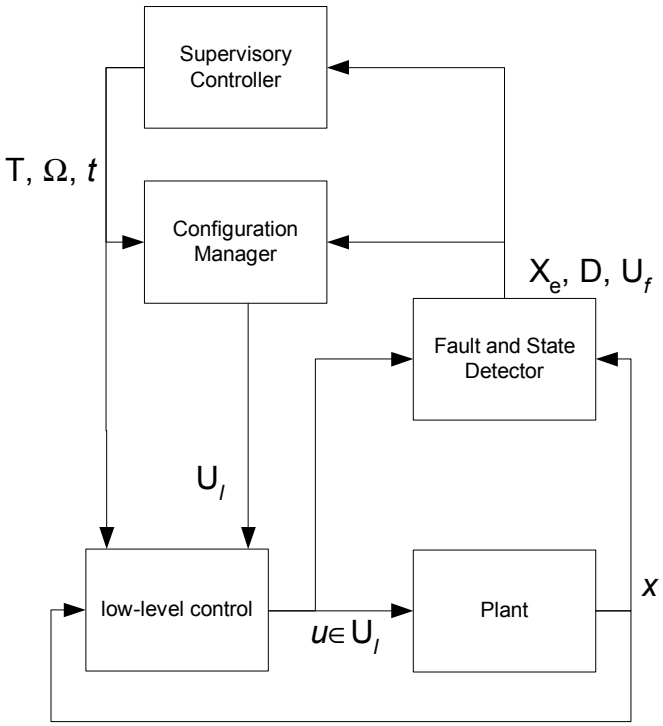

Fig. 2. Architecture

The supervisory controller specifies a set of alternate control objectives. The problem of achieving one of the control objectives is broken down into two levels.

Problem 1 (Reconfiguration). Given system (1), a set of control objectives O, a state and fault detection X_e, D, U_f, and a set of possible input constraints, $\bar{U} \subseteq 2^U$ determine input constraints $U_l \in \bar{U}$ and a control objective (T, Ω, t) $\in$ O, such that system (1) with constraints $u \in U_l$ and disturbance set D can be driven to target set T, within k time steps, with $1 \leq k \leq t$ while staying in Ω for the first k-1 time steps and that $u \in U_l \Rightarrow u \in U_f$.

Problem 2 (Low-Level Control). Determine inputs $u(t) \in U_l$ needed to reach T within k time steps, with $1 \leq k \leq t$ while staying in Ω for the first k-1 time steps.

The low-level control problem is solved continuously by the low-level control module. When a control objective is achieved, the supervisory controller sets a new set of control objectives. Reconfiguration occurs when either of the following happens:

- The set of control objectives specified by the supervisory controller is changed, and no longer includes the current objective.
- The fault-induced input constraints become more restrictive and violate the current configuration. (i.e. $U_l \not\subseteq U_f$.)
- The disturbance set becomes larger and violates the current configuration.

When reconfiguration occurs, the configuration selects one of the control objectives from the set specified by the supervisory controller, and selects input constraints $u \in U_l$ for reconfiguration.

3 Example

In the three-tank system shown in Figure 1 the objective is to regulate the level of fluid in tank 3. The nonlinear continuous-time hybrid model is detailed in [7]. An approximation of this hybrid system as a mixed-logical dynamic system is given in [8]. In the nominal case, Tank 1 serves as a buffer tank, and tank 3 is regulated by controlling the flow between tanks 1 and 3 using valve V_1. One of the possible faults in the system is a leak in Tank 1. The scenario for control and reconfiguration of this system is shown in Figure 3. The control objectives (Ω, T, t) and input constraints U_l for this scenario are shown in Table 1.

The results in Figure 3 and Figure 4 were obtained using the model in [7]. The low-level control was a implemented using PI controllers on the pumps, hysteresis switches on the valves, and additional simple switching elements. The scenario comprises of four phases:

1. The system starts with all tanks empty. Tank 1 and tank 3 are filled to their nominal levels.
2. The system is regulated at the nominal levels around h_1=0.5, h_2=0, h_3=0.1.
3. Following the detection of a leak in tank 1, the supervisory controller sets the control objective to filling tank 2, while regulating tank 3 and emptying tank 1.
4. The system is regulated around the set-point h_1=0, h_2=0.5, h_3=0.1, which mirrors the regulation of phase 2.

For the valves, a value of 1 is interpreted as the valve open, and a value of 0 as closed. Note that V_{23} is never opened in the configurations detailed in Table 1. This means that the configurations are tolerant to faults which cause V_{23} to be permanently closed. Note also that only two actuators are used in each phase.

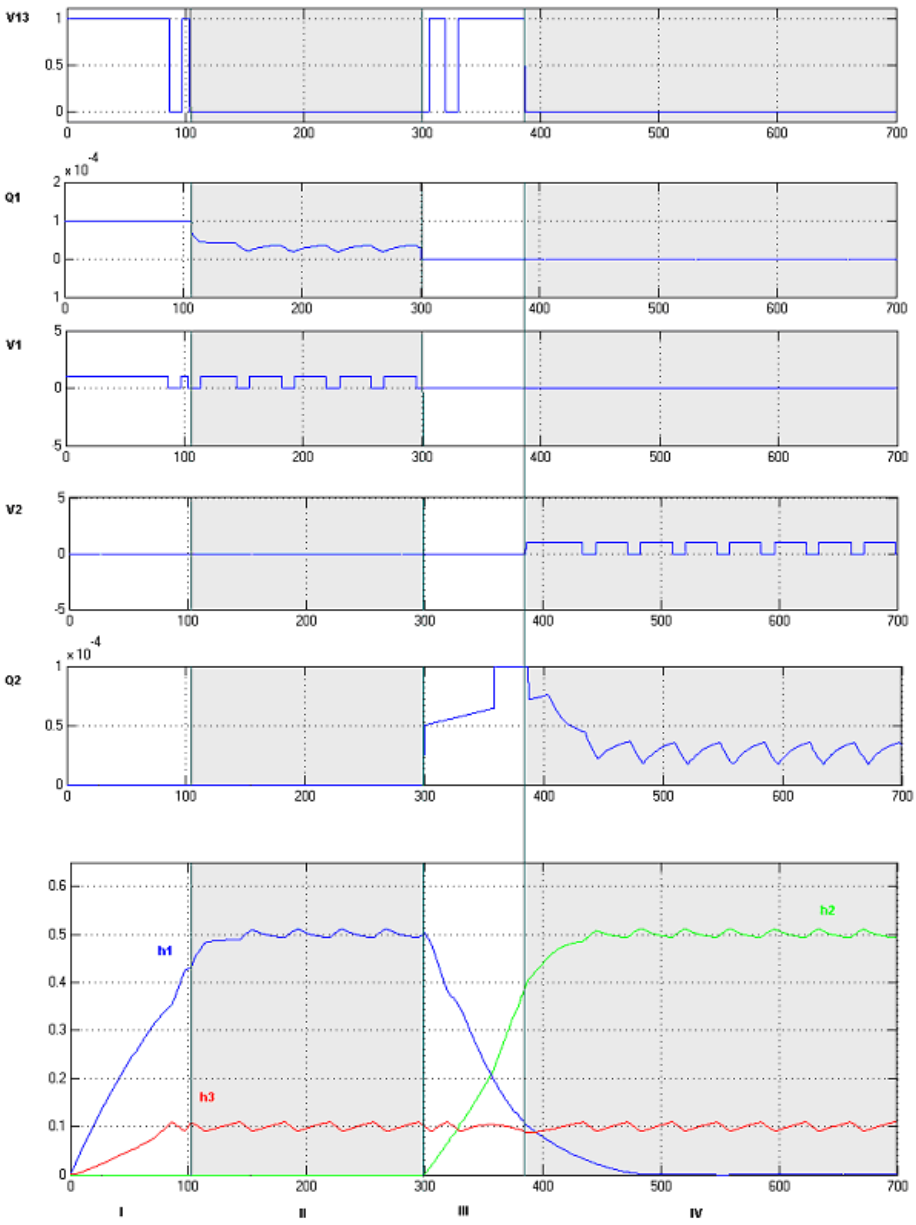

Fig. 3. Reconfiguration scenario for leak in tank 1

Table 1. Control Objectives and Configurations. For each phase the target set T must be acheived within t=200 time steps.

	Ω	T	U_l
1	$\{h_1, h_2, h_3 \mid$ $0 \le h_1 \le 0.6,$ $0 \le h_2 \le 0,$ $0 \le h_3 \le 0.11\}$	$\{h_1, h_2, h_3 \mid$ $0.45 \le h_1 \le 0.55, 0$ $\le h_2 \le 0,$ $0.09 \le h_3 \le 0.11\}$	$\{ V_{13}, V_1, V_2, V_{23}, Q_1, Q_2 \mid$ $0 \le V_{13} \le 1,$ $0 \le V_1 \le 1,$ $Q_1 = 10^{-4},$ $V_{23}= V_2=Q_2=0\}$
2	$\{h_1, h_2, h_3 \mid$ $0.45 \le h_1 \le 0.55,$ $0 \le h_2 \le 0,$ $0.09 \le h_3 \le 0.11\}$	$\{h_1, h_2, h_3 \mid$ $0.45 \le h_1 \le 0.55,$ $0 \le h_2 \le 0,$ $0.0905 \le h_3 \le$ $0.105\}$	$\{ V_{13}, V_1, V_2, V_{23}, Q_1, Q_2 \mid$ $0 \le V_1 \le 1,$ $0 \le Q_1 \le 10^{-4},$ $V_{13}= V_{23}= V_2=Q_2=0\}$
3	$\{h_1, h_2, h_3 \mid$ $0 \le h_1 \le 0.55,$ $0 \le h_2 \le 0.6,$ $0.09 \le h_3 \le 0.11\}$	$\{h_1, h_2, h_3 \mid$ $0 \le h_1 \le 0.2,$ $0.4 \le h_2 \le 0.6,$ $0.09 \le h_3 \le 0.11\}$	$\{ V_{13}, V_1, V_2, V_{23}, Q_1, Q_2 \mid$ $0 \le V_1 \le 1,$ $0 \le Q_1 \le 10^{-4},$ $V_{13}= V_{23}= V_2=Q_2=0\}$
4	$\{h_1, h_2, h_3 \mid$ $0 \le h_1 \le 0.2,$ $0.4 \le h_2 \le 0.6$ $0.09 \le h_3 \le 0.11\}$	$\{h_1, h_2, h_3 \mid$ $0 \le h_1 \le 0,$ $0.45 \le h_2 \le 0.55,$ $0.09 \le h_3 \le 0.11 \}$	$\{ V_{13}, V_1, V_2, V_{23}, Q_1, Q_2 \mid$ $0 \le V_2 \le 1,$ $0 \le Q_2 \le 10^{-4},$ $V_1= V_{13}= V_{23}=Q_1=0\}$

In phase three, shown in Table 1, the supervisory controller specifies a control objective of reaching the neighborhood of h_3=0.1, h_2=0.5, and for phase 4, regulation around that point. Three alternate points specified in order of descending priority are

- h_3=0.1, h_2=0.3,
- h_3=0.1, h_2=0.2,
- h_3=0.0, h_2=0.0.

The last option is a shutdown, a safe state, which covers the case where no other objective is achievable. Consider two cases where reconfiguration is necessary:

1. Valve V_2 is faulty. The configuration shown in Table 1, phase 4, is no longer valid as it requires V_2 to be manipulable. The configuration manager selects a configuration which uses V_{23} instead of V_2 to achieve the setpoint of h_3=0.1, h_2=0.3
2. From time t=380 sec onwards valve V_{23} is permanently open. The configuration shown in Table 1, phase 4, is no longer valid as it requires V_{23} to be permanently closed. In this case the same target set can be achieved, with different input constraints. Figure 4 shows this scenario. The system can still be controlled using pump Q_2 alone. The difference is that when using Q_2 alone, the disturbance that can be tolerated is smaller.

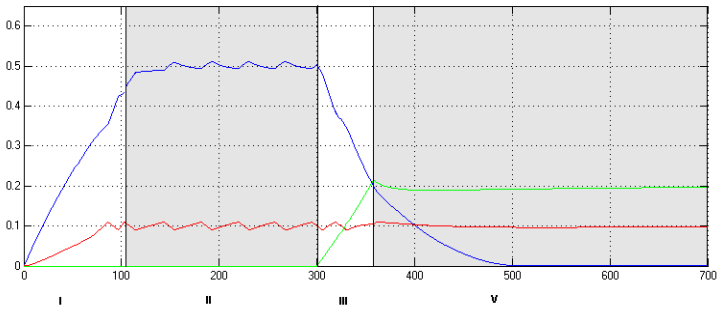

Fig. 4. Alternative ending to the leak scenario

4 Reconfiguration

The purpose of imposing constraints on the inputs is to reduce the number of manipulable input variables for the low-level control. If the low-level control is implemented by model-predictive control (MPC) – which is possible for piecewise-affine systems – the manipulable input variables are decision variables for the MPC optimization problem and reducing their number reduces the computational complexity [6]. Clearly, the reconfiguration task is required to have less computational cost than what is saved by not allowing all input variables to be manipulable by the low-level control. For this reason, the approach taken here is to perform the reachability calculations required for reconfiguration at design-time.

The configuration manager's task is to select input constraints which will guarantee reachability from the current state $x(t)$ to a target state set $T \subseteq X$ in t time steps without leaving $\Omega \subseteq X$ for the first t-1 time-steps. In this paper this process is called *reconfiguration*. It is proposed to perform the necessary reachability calculations at design time and store the results of these calculations in a reconfiguration database. A necessary condition for reachability is that a sequence of input vectors which satisfies the input constraints and the control objective exist. The following definitions will be instrumental in constructing the reconfiguration database.

Definition 2 (Robust One-Step Set). [9, section 2.3] For the system (1), with inputs $u(t) \subseteq U_l$ and disturbances $d(t) \in D$, the *robust one-step set* $Q(\Omega)$ is the set of states in X for which an admissible control input exists which will drive the system to Ω in one step, for any disturbance $d(t) \in D$ i.e

$$Q(\Omega) = \{ x \in X \mid \exists u \in U \; \exists q \in Q : (x, u) \in \chi_q, \forall d \in D \;\; A_q x + B_q u + f_q + d) \in \Omega \}.$$

Definition 3 (Robust Controllable [*i,j*]-step Set). For the system (1), with inputs u(t) $\in$ U_l and disturbances $d(t) \in$ D, the *robust controllable [i,j]-step set* $K_i^j(\Omega, T)$ is the largest set of states in Ω for which an integer $i \le k \le j$ exists for which there exists an admissible control input which will drive the system to T in exactly k steps, while keeping the evolution of the state inside Ω for the first k-1 steps, for any time-varying disturbance $d(t) \in$ D, i.e.

$$K_i^j(\Omega,T) = \left\{x_0 \in R^n \mid \exists i \le k \le j \quad \exists \{u(t) \in U\}_0^{k-1} : \{x(t) \in \Omega\}_0^{k-1}, x(k) \in T, \forall \{d(t) \in D\}_0^{k-1}\right\}$$

Theorem 1. The robust controllable [i,j]-step set can be computed by the following recursive formula:

$$\tilde{K}_{\mathrm{i}}^{\mathrm{j}}(\Omega,T) = \begin{cases} T & i = j = 0 \\ \tilde{Q}\left(\tilde{K}_{\mathrm{j\text{-}1}}^{\mathrm{j\text{-}1}}(\Omega,T)\right) \cap \Omega & 0 < i = j \\ \tilde{Q}(\tilde{K}_{\mathrm{i}}^{\mathrm{j\text{-}1}}(\Omega,T)) \cap \Omega \cup \tilde{K}_{\mathrm{i}}^{\mathrm{j\text{-}1}}(\Omega,T) & i < j \end{cases} \qquad (\mathbf{2})$$

Proof. For $i=j$, the algorithm and proof is shown in [9, section 2.6]. For $i < j$, by definition $K_{\mathrm{i}}^{\mathrm{j+1}}(\Omega,T) = K_i^j(\Omega,T) \cup K_{i+1}^{j+1}(\Omega,T)$. Also, $K_i^j(\Omega,T) = \bigcup_{i \le k \le j} K_k^k(\Omega,T)$ and $K_{i+1}^j(\Omega,T) = \tilde{Q}(K_{\mathrm{i}}^{\mathrm{j\text{-}1}}(\Omega,T)) \cap \Omega$. Therefore,

$$K_{i+1}^{j+1}(\Omega,T) = \bigcup_{i+1 \le k \le j+1} K_k^k(\Omega,T) = \bigcup_{i \le k \le j} \tilde{Q}(K_k^k(\Omega,T)) \cap \Omega = \tilde{Q}(K_i^j(\Omega,T)) \cap \Omega$$

The robust controllable set for LTI systems can be computed using the invariant set toolbox [11]. The robust controllable set for PWA systems can be computed in an iterative way based on the one-step robust controllable set for each mode of the system. One such method is for computing robust controllable sets for piecewise affine systems is described in [9, section 4.5].

Reconfiguration provides fault tolerance by choosing input constraints, which are compatible with fault conditions. For example, if valve V_1 in the three tank system is fixed in position V_1=0, then any configuration constraint which is satisfied by V_1=0 is compatible with this fault.

The reconfiguration database consists of six-tuples ($\tilde{X}$, $\tilde{D}$, $\tilde{U}$, $\tilde{T}$,$\tilde{\Omega}$, $\tilde{t}$) for which it has been determined that $\tilde{X}$ is a robust controllable [1,t]-step set $K_1^t(\tilde{\Omega},\tilde{T})$ for the system with disturbance $\tilde{D}$ and input constraints $\tilde{U}$. At runtime, the configuration manager's task is to find a six-tuple from the database, for which $X_e \subseteq \tilde{X}$, D $\subseteq \tilde{D}$, $\tilde{U} \subseteq U_f$, $\tilde{T} \subseteq$ T, $t \le \tilde{t}$, $\tilde{\Omega} \subseteq \Omega$, based on X_e, D, U_f supplied by the fault and state detector and Ω, T, t supplied by the supervisory controller. The sets are all assumed to be convex polyhedral sets, so the computation of the set inclusions amount to the solution of linear programs. In general the robust controllable set for a piecewise affine system is not convex; however it is sufficient for the purpose of reconfiguration to use an inner

approximation of the robust controllable set, which is convex, for the value of $\tilde{X}$ in the database.

By removing all reconfiguration options, which do not satisfy the necessary conditions for reachability, the search space for the low-level control is reduced, while ensuring the existence of appropriate control inputs to satisfy a control objective. The problem of designing the low-level control to select the optimal control inputs is beyond the scope of this paper. One possibility is to apply model-predictive control for which necessary and sufficient conditions for robust feasibility are known [10].

The reconfiguration database lists six-tuples ($\tilde{X}$, $\tilde{D}$, $\tilde{U}$, $\tilde{T}$,$\tilde{\Omega}$, $\tilde{t}$) for possible combinations of state and fault identification and control objectives given by the state and fault detector and the supervisory controller, respectively. The task of partitioning the state and input sets to determine these sets is the subject of the next section.

5 Supervisory Control

The supervisory control of a hybrid system can be approached as a discrete-event control problem, by abstracting the plant into a discrete event system preserving all properties of interest. In hierarchical control, this is done by forming a partition of the state space, for which it can be guaranteed that the system can be forced to reach a desired region by choosing appropriate controls. In this section the subject of partitioning the continuous state space will be considered. The control specification that forms the primary partition is given by the following definition.

Definition 4. (Global Control Objective) Given a set $Bad \subseteq X$ and a finite collection of sets $A_k \subseteq X$, $k \in K$, that includes an initial set $A_0 \subseteq X$, with $A_k \cap Bad = \varnothing$, a set valued map *next*: $K \rightarrow 2^K$ and a function time *time*: $K \times K \rightarrow Z^+$ the global control objective is that for the system with initial conditions $x \in A_0$ the continuous state will remain in any set A_k for at most *time*(k,k') time steps and then cross into $A_{k'}$ for some $k' \in$ next(k).

Remark 1. Definition 4 applies for the nominal case. In case of a fault, which necessitates reconfiguration, a degraded performance is assumed to be acceptable in which time constraints do not apply. In this case the global control objective requires an event sequence specified by the *next* relation, while the constraints specified by the function *time* do not apply.

Assume a given disturbance set D_k for each region A_k. Let $U_L \subseteq 2^U$ be the set of admissible input sets. The choice of input constraints is based on two considerations: fault-tolerance and reducing the number of manipulated variables. A configuration $U_l \in U$ will be tolerant to a fault if the configuration admits only input vectors which are not precluded by the fault. The additive state disturbance can also be used to model certain input faults (e.g. a leak in the tank).

Let $\Psi = \{\Omega_k\}$ be a collection of sets, which appear as invariant sets Ω in the reconfiguration database, and let $\Omega_0 = A_0$; $\Omega_k \subseteq A_k \; \forall k \in K$. It is required that at any trajec-

tory starting in Ω_0 can be driven to follow the global control objective. This can be assured if

$$\forall x \in \Omega_k, \exists l \in \text{next}(k), \exists u \in U_L : x \in K_1^{\text{time}(k,l)}(\Omega_k, \Omega_l). \quad (3)$$

In the nominal case reconfiguration occurs when the system crosses into a target set from which reachability to the next target set is assured within the required time. When a fault occurs, the time constraint is not necessarily satisfied; however, the condition of Equation 3 ensures that the next state is reachable when reconfiguration occurs at any point along the trajectory. The collection Ψ can be calculated recursively by Algorithm 1.

Algorithm 1. (Compute Collection of Invariant Sets Ψ)

INPUT:
- partition π defining regions A_k, $k \in K$
- input constraints U_L
- disturbance set D_k for each A_k

BEGIN
FOR each $k \in K$
$\Omega'_k = A_k$.
REPEAT
FOR each $k \in K$,
$\Omega_k = \Omega'_k$
FOR each $k \in K$,
$$\Omega'_k = \bigcup_{U_l \in U_L} \bigcup_{l \in next(k)} K_1^{time(k,l)}(\Omega_k, \Omega_l);$$
UNTIL $\Omega_k = \Omega'_k, \forall k \in K$
END
OUTPUT
- Collection $\Psi = \{\Omega_k\}_{k \in K}$ of invariant sets.

The algorithm succeeds if it terminates and $\Omega_0 = A_0$. If the algorithm terminates successfully Equation 3 is satisfied. What remains is to partition the sets $\{\Omega_k\}_{k \in K}$ into regions such that from each region, it can be determined which next target set can be reached and by what configuration. This is performed by Algorithm 2.

Algorithm 2. (Partition sets Ψ, with configurations U_L).

INPUT:
- State space X
- Global control objective: K, $\{A_k\}$, *time*, *next*

– input constraints U_L
– disturbance set D_k for each A_k
BEGIN
$\pi_f := (X \setminus \bigcup_{k \in K} A_k , A_0, \Omega_1, A_1 \setminus \Omega_1, \Omega_2, A_2 \setminus \Omega_2, \ldots)$
FOR each $k \in$ K, $U_l \in U_L$, $l \in next(k)$,
compute partition: $\pi = (X \cap K_1^{time(k,l)}(\Omega_k, \Omega_l) , X \setminus K_1^{time(k,l)}(\Omega_k, \Omega_l))$
Refine: $\pi_f := \pi_f \cdot \pi$
END
END
OUTPUT:
– Refined partition π_f.

After finding the final partition, the supervisory controller and the state detector can be designed to specify their outputs in terms of the refined partition. When the system is in region A_k, the supervisory controller sets the control objective for the configuration manager and low level control as all the 3-tuples (Ω, T, t) with $\Omega = \Omega_k$, $T = \Omega_l$, $l \in next(k)$, $t = time(k,l)$. The state detector must detect partition crossing in the final partition so that when the system crosses into Ω_l it can be determined in which region of the final partition the current state is, so that reconfiguration can proceed. The reconfiguration database is also constructed based on the final partition and the possible control objectives. Throughout this section the disturbance set D_k was assumed to be given for each region A_k of the global control objective. The disturbance set provides another design parameter, which can be relaxed or tightened to enable the global control objective to be achieved or to increase robustness.

6 Conclusions and Related Work

In this paper, the subject of control reconfiguration in hierarchical control of piecewise affine systems was considered as the problem of choosing input constraints for the low-level control to satisfy reachability requirements given by the supervisory controller. This approach provides fault-tolerance to actuator faults, while allowing the supervisory controller to handle major component faults. The computational complexity of low-level control is reduced by limiting the number of available inputs. The additional run-time computational cost due to reconfiguration is kept low by performing reachability analysis at design-time, and limiting run-time calculations to set inclusions.

The problem of reconfiguration in the control of piecewise affine systems as a choice of manipulated inputs, and the need to reduce the number candidate inputs in the process was considered in [6]. A number of methods were proposed in for reducing the number of candidate inputs, using the mixed-logic dynamic representation of the system. This paper adds to the work in [6] by considering the model-predictive control of a piecewise-affine system in the context of hierarchical control. In the con-

text of hierarchical control, the option of changing set-points is considered simultaneously with the option of changing the set of manipulated actuators. As is shown in the example in section 3, when hierarchical control is used, reconfiguration can be handled at the supervisory control level to handle component faults (e.g. emptying the leaking tank and filling the redundant tank) while providing additional fault tolerance by low-level control reconfiguration (e.g. using the lower valve, if the upper valve is faulty). An approach to reconfiguration based on model-predictive control alone is less suitable than one based on hierarchical control when the prediction horizon which is needed for reconfiguration is much larger than the prediction needed for nominal control, because the complexity of the optimization problem solved by the model-predictive control algorithm grows with the prediction horizon.

In [5], hierarchical control of piecewise-linear systems (piecewise affine systems with an offset vector of zero) was investigated. This paper extends the approach of [5] by considering the possibility of selecting the input constraints at runtime. The introduction of faults which require reconfiguration, adds an additional requirement for partitioning the state space that when the system moves from an initial set to a target, the target set must be reachable, even if the input constraints changes before the target set is reached and after the initial set is left. In this case, the target set can be reached in a finite time providing a degraded - but safe - performance in the event of faults.

In [12, section 12.4] a hybrid control strategy is shown for the three-tank system, in which the system is reconfigured in steps when faults occur, while employing low-level control to manipulate the actuators and supervisory control triggered by crossing partitions in the state-space to coordinate the low-level control. This paper provides the foundation for verifying such a control strategy.

In [12, section 12.3] reconfiguration is performed based on a qualitative mode obtained by quantizing the system with a rectangular state space partition. The approach is applied to the three-tank benchmark problem. The approach presented in this paper differs in that it is hierarchical, and includes a low-level control component, which has a continuous range of values for the input variables available to it. This results better quality of control, while not sacrificing the robustness, which is provided by the supervisory controller that operates on a discrete level. In addition, the partition of the state space for supervisory control, as presented in this paper is based on the specification of a global control objective, and the ability of the low-level control to achieve intermediate objectives. Also note that the reconfiguration database proposed in this paper requires enumeration of configurations, but it does not require enumeration of faults.

The theory of invariant sets, the computation of invariant sets for piecewise affine systems, and the applicability of invariant sets to the feasibility of model-predictive control and are studied in [9] and recent results are published in [10]. These results are applicable to the computation of the robust controllable sets in the reconfiguration database.

Current work includes computation of convex approximations of robust controllable sets for piecewise affine systems and using it to generate the partition refinement and the reconfiguration database described in section 4 and section 5 of this paper.

An open problem in the method presented in this paper is how to partition the state space in such a way that time constraints of the global control objectives can be satisfied in the event of faults.

Acknowledgements. The work is supported by DARPA under F33615-99-C-3611 as part of the Software Enabled Control program, and by a Vanderbilt University Graduate Fellowship. The author wishes to thank anonymous reviewers for their constructive comments.

References

1. M. Blanke, C. Frei, F. Kraus, R.J. Patton, and Staroswiecki M "What is Fault-tolerant Control.", Plenary address, In *Proceedings of the IFAC Symposium SAFEPROCESS*, Budapest, pages 40-51, 2000
2. E.D. Sontag. Interconnected automata and linear systems: A theoretical framework in discrete time. In *Hybrid Systems III – Verification and Control*, R. Alur, T.A. Henzinger and E.D. Sontag eds. Lecture Notes in Computer Science, Vol. 1066. Springer-Verlag, Pittsburgh, USA, (1996), 436-448
3. W.P.M.H. Heemels, B. De Schutter and A. Bemporad. Equivalence of Hybrid Dynamical Models, *Automatica* 37(7), July 2001
4. A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints, *Automatica,* 35(3), March 1999
5. X.D. Koutsoukos and P.J. Antsaklis. Hierarchical Control of Piecewise Linear Hybrid Dynamical Systems Based on Discrete Abstractions, *Interdisciplinary Studies of Intelligent Systems, Notre Dam University, Technical Report ISIS-2001-001*, February 2001
6. K. Tsuda, D. Mignone, G. Ferrari-Trecate and M. Morari. Reconfiguration Strategies for Hybrid Systems. In *Proceedings of the American Control Conference*, Arlington Virginia, 2001
7. J. Lunze. Laboratory Three Tanks System Benchmark for the Reconfiguration Problem. Technical report, Tech. Univ.of Hamburg-Harburg, Inst. of Control. Eng., Germany, 1998.
8. D. Mignone., *Moving Horizon Estimation and Fault Detection of Mixed Logic Dynamical Systems*, Postdiploma Thesis, Automatic Control Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland, August 1999
9. E.C. Kerrigan. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. PhD thesis, University of Cambridge, UK, November 2000.
10. E.C. Kerrigan and J.M. Maciejowski. Robust Feasibility in Model Predictive Control: Necessary and Sufficient Conditions. In *Proceedings of the 40th Conference on Decision and Control*, Orlando, Florida, USA, December 2001
11. E.C. Kerrigan, MATLAB Invariant Set Toolbox downloadable from http://www-control.eng.cam.ac.uk/eck21/.
12. J. Lunze, J. Askari-Maranani, A. Cela, P.M. Frank, A.L. Gehin, B. Heiming, M. Lemos, T. Marcu, L. Rato, M. Starosweicki. Three Tank Control Reconfiguration. In *Control of Complex Systems.* K. Åstrom et al (eds.) Springer, London, pp 241-283, 2001.