The Composition and Validation of Heterogeneous Control Laws*

Benjamin Kuipers†
University of Texas at Austin
and
Karl Åström‡
Lund Institute of Technology

January 22, 1993

Abstract

We present a method for creating and validating a nonlinear controller by the composition of heterogeneous local control laws appropriate to different operating regions. Like fuzzy logic control, these methods apply even in the presence of incomplete knowledge of the structure of the system, the boundaries of the operating regions, or even the control action to take. Unlike fuzzy logic control, these methods can be analyzed by a combination of classical and qualitative methods. Each operating region of the system has a classical control law, which provides high-resolution control and can be analyzed by classical methods. Operating regions are defined by fuzzy set membership functions. The global control law is the weighted average of the local control laws, where the weights are provided by the operating region membership functions. A heterogeneous control law can be analyzed, even in the presence of incomplete knowledge, by representing it as a qualitative differential equation and using qualitative simulation to predict the set of possible behaviors of the system. By expressing the desired guarantee as a statement in a modal temporal logic, the validity of the guarantee can be automatically checked against the set of possible behaviors. We demonstrate heterogeneous controllers and our qualitative methods for proving their properties, first for a simple level controller for a water tank, and second for a highly nonlinear chemical reactor.

---

*The citation for this paper is: B. J. Kuipers and K. Åström. 1994. The composition and validation of heterogeneous control laws. *Automatica* **30**(2): 233–249. Portions of this paper were presented at the 1991 American Control Conference, Boston, Massachusetts, 26-28 June 1991. Reprinted by permission.

†Department of Computer Sciences, University of Texas at Austin, Austin, Texas 78712 USA. Email: kuipers@cs.utexas.edu.

‡Department of Automatic Control, Lund Institute of Technology, Lund, Sweden. Email: kja@control.lth.se.

1 Introduction

Much control theory is based on linear models. This works very well for steady state regulation at a fixed operating point. To make a control system that can operate over wide regions it is however necessary to introduce nonlinearities. There are several ways to do this. Linear feedback control can be combined with logic for switching between several linear feedback laws. Selectors that choose between different control laws depending on signal levels can be introduced. Systems of these types are common in industry, where their design is based on engineering experience combined with extensive simulation. Classical control theory (e.g., [Franklin, et al, 1986]) provides a rich set of methods for local analysis of the individual control laws and for describing their behavior, but theoretical analysis of combined laws has proved to be much more difficult.

Fuzzy logic control [Zadeh, 1973; Mamdani, 1974] is another approach to obtain nonlinear control systems, especially in the presence of incomplete knowledge of the plant or even of the precise control action appropriate to a given situation. In this approach the measured variables are represented as fuzzy variables. A representation of the control signal as a fuzzy variable is computed from the measurements using fuzzy logic. The fuzzy variable is converted to a real variable using some type of "defuzzification." Again, design and validation of these control laws is based primarily on experience and extensive simulation.

In this paper, we take a new look at the problem of specifying, analyzing, and verifying the behavior of nonlinear control laws, especially in the presence of incomplete knowledge. We focus our attention on *heterogeneous* control laws, which are composed of classical (typically linear) control laws defined over different operating regions (possibly with fuzzy boundaries). The control signal from a heterogeneous controller is the average of the signals from the local control laws, each weighted by the value of its operating region membership function. This approach to fuzzy control was pioneered by Takagi and Sugeno [1985] and Sugeno and Kang [1986].

The analysis and verification of a heterogeneous control law is complicated by the fact that such a controller is normally designed to cope with incomplete knowledge of the structure of the plant, the boundaries of the operating regions, and even the desired control action. Qualitative simulation [Kuipers, 1986, 1989] addresses this issue by making it possible to predict the behaviors consistent with qualitative knowledge about a dynamical system and its initial state. Specifically, given a qualitative differential equation (QDE) and a qualitative description of an initial state ($QState(t_0)$), the QSIM algorithm predicts a set of possible behaviors,

$$QDE \wedge QState(t_0) \rightarrow or(Beh_1, \ldots Beh_n),$$

that is guaranteed to include a description of the solution to any ordinary differential equation and initial state matching the qualitative description. Finally, using a model-checking algorithm for statements in modal temporal logic [Emerson, 1990], we can automatically determine whether a given specification is guaranteed to hold within the predicted set of qualitative behaviors.

The basic concepts of heterogeneous control will be introduced with a simple level controller for a water tank. We then discuss qualitative simulation and modal temporal logic, leading up to semi-automated proofs of propteries of the heterogeneous controller. Finally, we present a heterogeneous controller for a highly nonlinear chemical reactor [Economou, et al, 1986].

2 Fuzzy Control and Other Heterogeneous Methods

Fuzzy sets were originally developed by Zadeh [Zadeh, 1965; cf. Yager, et al, 1987] to formalize qualitative concepts without precise boundaries. For example, when describing values of a continuous scalar quantity such as the amount of water in a tank, there are no meaningful landmark values representing the boundaries between *low* and *normal*, or between *normal* and *high*.

Zadeh [1965] formalizes linguistic terms such as these as referring to *fuzzy sets* of numbers. A fuzzy set, S, within a domain, D, is represented by a *membership function*, $s : D \to [0, 1]$. We interpret the value of $s(x)$, for $x \in D$, as a measure of the *appropriateness* of describing x with the descriptor S. Figure 2 includes three membership functions defining the appropriateness of applying the qualitative descriptors $\{low, medium, high\}$ to quantitatively-defined levels.[1]

Fuzzy control is a family of methods for expressing and applying control laws, using fuzzy sets to provide several benefits. First, they provide the ability to express and use incomplete knowledge of the system being controlled and of the control law itself. Second, they allow one to specify a complex control law as the composition of simple components. Third, fuzzy set membership functions provide smooth transitions from region to region.

There are at least two distinct approaches to fuzzy control:

- *Fuzzy logic control* determines the control action by a combination of fuzzy logic rules.
- *Heterogeneous control* determines the control action as the weighted average of classical control laws.

A *fuzzy logic controller* [Zadeh, 1973; Mamdani, 1974; cf. Michie & Chambers, 1968] consists of a collection of simple control laws whose inputs and outputs are both fuzzy values. For example,

If water level is $high$, then set drain opening to $wide$;

where $high$ and $wide$ are qualitative terms described by fuzzy sets over their quantitative domains.

All controller rules are fired in parallel, and the recommended actions are combined according to fuzzy value combination rules, weighted (or bounded) by the degree of satisfaction of the antecedent. Some process of "defuzzification" is required to convert the resulting fuzzy set description of an action into a scalar value for a control variable.

A *heterogeneous controller* decomposes the state space into multiple, possibly overlapping, operating regions. The domain of each operating region is characterized by a fuzzy set membership function. This makes it possible to express smooth transitions between adjacent regions. Each operating region is associated with a qualitative description of the system state, e.g. the low, normal, or high level of water in a tank. The fuzzy set membership functions may be regarded as a measure of the *appropriateness* of applying a given qualitative description to the system state. It may be assumed that, for any given system state, the appropriateness measures sum to 1.0.

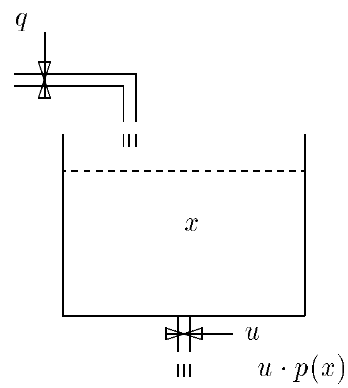
Each region is associated with a control law. The control signal applied to the plant is a weighted average of the control signals for each region, where the weights are provided by the membership functions of each region.

[1] *Appropriateness measure* is technically synonymous with the terms *membership function* and *possibility measure* as used in the fuzzy research community. However, for our purposes, the English connotation of *appropriateness measure* seems better to capture the relationship between a linguistic term and a scalar quantity.

The idea of combining simple linear feedback units with operations such as average, min, max, etc, is widely used industrially. The intent of this paper is to provide a mathematical basis for the local and global analysis of these systems. The heterogeneous control approach decomposes the design of a controller into two relatively independent decisions: (1) the specification of natural, qualitatively distinct operating regions, and (2) the specification of a control law for each region. The weighted sum combination method provides smooth transitions from one region to another, and facilitates local and global analysis.

Heterogeneous control is also related to gain scheduling. There are however some differences. In gain scheduling a specific control law is selected for a given operating region and the parameters of the controller are changed with the region. In heterogeneous control the values of the control signal for different regions are computed and averaged.

Sliding control [Utkin, 1977, 1991; Slotine & Li, 1991] is another method for constructing nonlinear controllers by combining the effects of simpler controllers. In its pure form, the sliding control law changes discontinuously at a boundary, possibly leading to rapid "chattering" between the two control surfaces but yielding extremely good ("perfect") tracking of disturbances. To avoid chattering, the discontinuity can be smoothed by taking a weighted average of the two control laws in a narrow band near the boundary, at some cost in tracking error. Essentially, this replaces "crisp" by "fuzzy" operating regions for two classical control laws, and thus is in the spirit of heterogeneous fuzzy control. However, the design philosophy behind fuzzy control typically treats overlap regions as transitory, existing primarily to provide smooth transitions between pure control regions. In contrast, sliding control works exactly by exploiting the joint action of two control laws to confine the system to a very small overlap region. Cross-fertilization between these two approaches should be of value.



$$\dot{x} = f(x, u) = q - u \cdot p(x).$$

Figure 1: The water tank

3 A Heterogeneous Controller for the Water Tank

3.1 The Water Tank

Consider control of the amount x of water in a tank, where the inflow rate q may vary, and the area u of the drain opening is the control variable. The function $p(x)$ is a monotonically increasing function of x whose exact form is not known to the designer. In this case, $p(x)$ is approximately proportional to the square root of the pressure, whose relation to x depends on the geometry of the tank.

$$\begin{aligned} x &= \text{amount in tank} \\ q &= \text{inflow into tank} \\ u &= \text{drain area} \\ p(x) &= \text{influence of pressure at drain} \end{aligned}$$

The dynamic behavior of the system is described by:

$$\dot{x} = f(x, u) = q - u \cdot p(x).$$

3.2 Overlapping Operating Regions

The system has separate control laws in three operating regions, Low, $Normal$, and $High$, with overlapping appropriateness measures, as shown in figure 2.

For an implemented controller, the operating region appropriateness measures must be specified as real-valued functions of a real variable. However, for purposes of analysis, the appropriateness measures $l(x)$, $n(x)$, and $h(x)$, for the three operating regions need not be completely specified. All

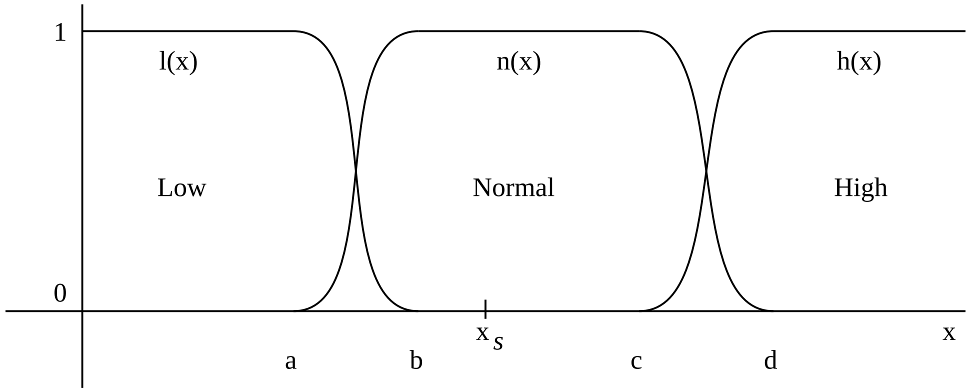


Figure 2: Three operating regions and their appropriateness measures.

that is known is that they rise or fall smoothly and monotonically between their plateaus, where the boundaries of the plateaus are characterized by the landmark values, a, b, c, and d. They are normalized, so that $l(x) + n(x) + h(x) = 1$. Note that there is a "pure" region over each interval $[0, a]$, $[b, c]$, and $[d, \infty)$, and overlapping regions on (a, b) and (c, d). We assume that the setpoint x_s is in (b, c).

Because we specify the appropriateness measures qualitatively, and depend only on properties of the qualitative class, the conclusions we derive apply to every member of the class. The remaining degrees of freedom are available to the designer to meet other implementation requirements.

3.3 Heterogeneous Control Laws

The control laws for the three regions are:

$$\begin{aligned} x \in Low &\Rightarrow u_l(x) = 0 \\ x \in Normal &\Rightarrow u_n(x) = k(x - x_s) + u_s \\ x \in High &\Rightarrow u_h(x) = u_{max} \end{aligned}$$

where $0 \leq u \leq u_{max}$, and the bias term u_s is adjusted to give the desired set point x_s for a nominal inflow q_s. We are assuming for this example that the state variable x is directly observable, rather than separating out measurements, $y = g(x, u)$.

The global heterogenous control law is the average of the individual control laws, weighted by the appropriateness measures of their regions:

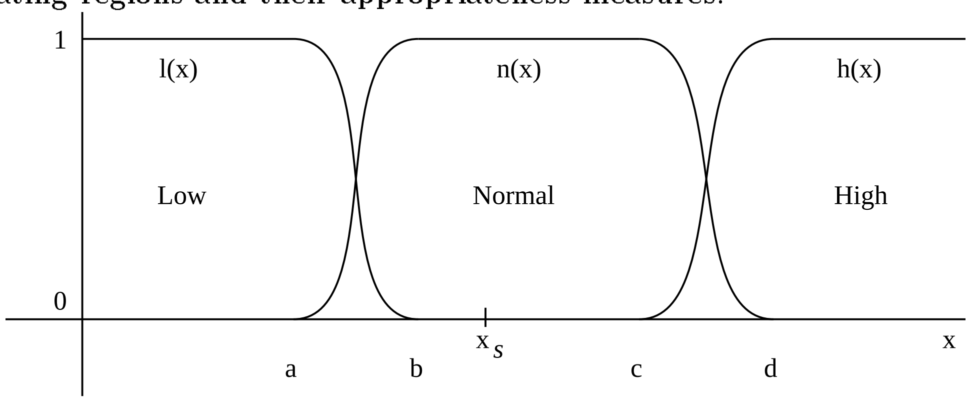
$$u(x) = l(x) \cdot 0 + n(x) \cdot [k(x - x_s) + u_s] + h(x) \cdot u_{max}$$

The design goal for the controller is a simple discrete abstraction: the system will move from the High or Low region into the Normal region, and reach equilibrium there. Our qualitative analysis, described below, helps us identify the additional constraints necessary to provide this guarantee. Figure 3 summarizes the heterogeneous controller for the water tank.

The water tank:

$$
\begin{array}{rcl}
\dot{x} & = & q - u \cdot p(x),\ p \in {M_0}^+ \\
q & : & \text{the flow into the tank (exogenous)} \\
x & : & \text{the level in the tank (sensed)} \\
u & : & \text{the drain opening (controlled)}
\end{array}
$$

The operating regions and their appropriateness measures:



The local control laws:

$$
\begin{array}{rcl}
x \in Low & \Rightarrow & u_l(x) = 0 \\
x \in Normal & \Rightarrow & u_n(x) = k(x - x_s) + u_s \\
x \in High & \Rightarrow & u_h(x) = u_{max}
\end{array}
$$

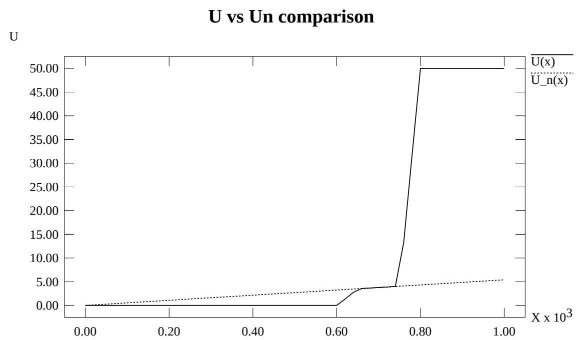
The global control law:

$$
u(x) = l(x)u_l(x) + n(x)u_n(x) + h(x)u_h(x).
$$

The discrete abstraction:

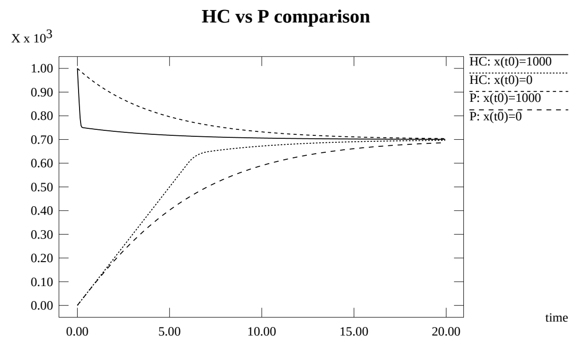
$$
\boxed{\text{Low}} \longrightarrow \boxed{\boxed{\text{Normal}}} \longleftarrow \boxed{\text{High}}.
$$

Figure 3: A heterogenous controller for the water tank.



$u(x)$ and $u_n(x)$

(a)



$x(t)$

(b)

Figure 4: Comparison between P and HC controllers.

(a) The heterogeneous control law $u(x)$, and the proportional controller $u_n(x)$ are identical in the Normal region.

(b) The behaviors, $x(t)$, of the P- and HC-controllers, starting with the tank empty or full, with constant q at the nominal rate, so that steady state is at the setpoint.

3.4 Simulation Results

By numerical simulation, we can illustrate the performance of this heterogeneous controller on a water tank, in comparison with a proportional controller.

The capacity of the tank is 1000 liters of water. The nominal inflow rate is 100 liters/minute. The setpoint, x_s, is 700 liters. The offset u_s in the $Normal$ control law u_n is set so that the steady state is at the setpoint when inflow is nominal. The gain k is set so that $u_n(0) = 0$. The proportional controller simply uses u_n as the global control law. The comparison is for illustration only, since the proportional controller has an unrealistically low gain. With a higher gain, however, the physical limits on the valve make the proportional controller behave like a heterogeneous controller, but without smooth transitions or explicit design and validation.

The operating regions for the heterogeneous (HC) controller are specified as in figure 3, with $a = 600$, $b = 650$, $c = 750$, $d = 800$, and $u_{max} = 50$.

Figure 4(a) compares the two control laws $u(x)$ and $u_n(x)$. Figure 4(b) contrasts the behavior of the two controllers at constant nominal inflow, starting from the initial states $x(t_0) = 0$ and $x(t_0) = 1000$. Figure 5 shows the responses of the two controllers to random variation in inflow between zero, nominal, and twice nominal.

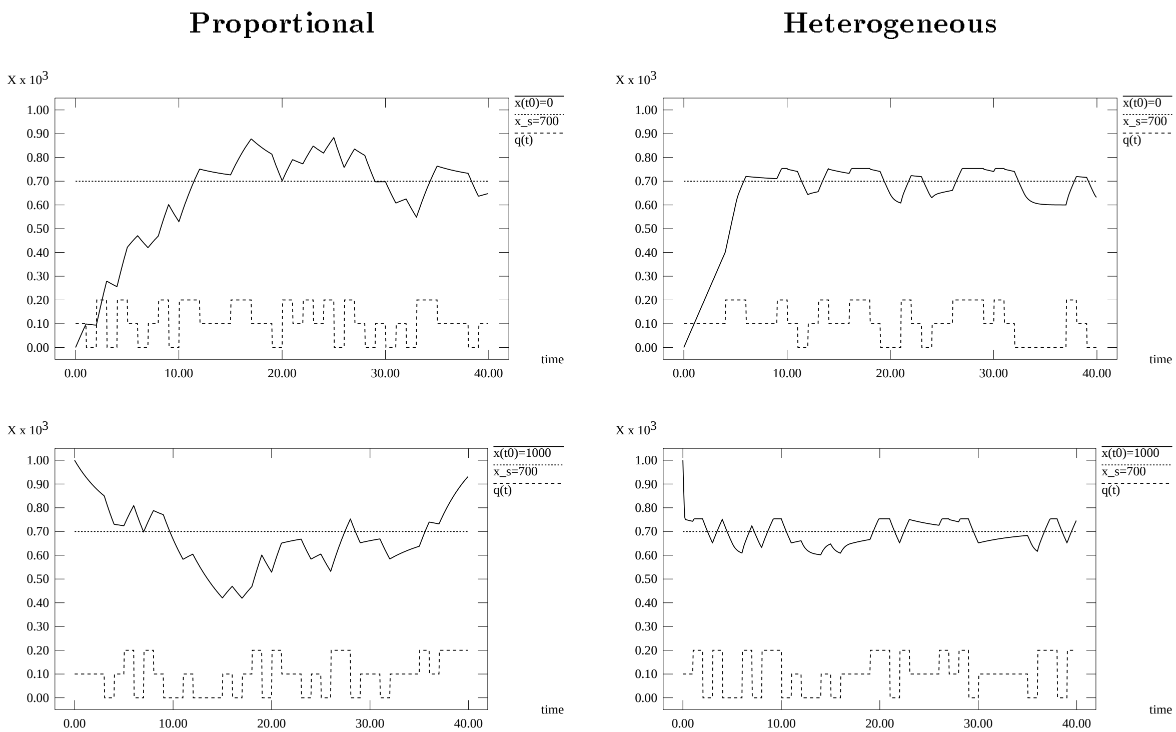


Figure 5: The effect of random inflow variation on P and HC controllers.
Inflow, q, plotted at the bottom of each graph, varies randomly between zero, nominal (100 liters/minute), and twice nominal. This figure shows the proportional controller (left column) and the heterogeneous controller (right column), with the tank initially empty (top row) or full (bottom row).

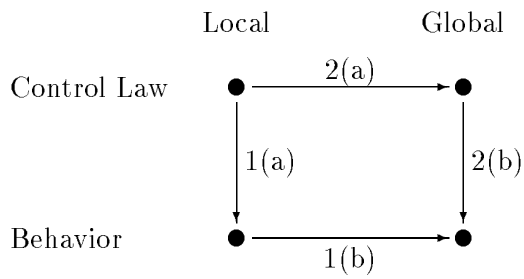


Figure 6: Two approaches to analyzing a heterogeneous controller.

4 Methods for Providing Guarantees

We want to prove that the heterogeneous controller brings the system back to the $Normal$ operating region under some range of disturbances, and that an equilibrium in the region is obtained for constant disturbances. More importantly, we want to determine any quantitative constraints on the design of the controller (e.g. the value for u_{max}), and the range of possible disturbances on q that the controller can handle.

There are two methods for doing this (figure 6), which are elaborated on below.

1. (a) Determine the qualitative behavior of the system within each operating region.
   (b) Combine the qualitative descriptions.
2. (a) Combine the local laws into a global law using the weighted average combination rule.
   (b) Determine the qualitative behavior of the global system.

First, however, we must introduce the ordinal, landmark-based methods of qualitative representation and the QSIM algorithm for qualitative simulation [Kuipers, 1986, 1989].

Qualitative categories may be described by fuzzy set membership functions where they lack meaningful boundaries, or by ordinal relations with landmark values where precise boundaries are meaningful. By using the landmark-based representation to describe a restricted class of fuzzy set membership functions, we can combine the performance benefits of smooth fuzzy-set transitions with the analytic power of landmark-based qualitative simulation.

4.1 Landmark-Based Representation

Qualitative categories may be defined by *landmark values*: precise boundary points separating qualitatively distinct regions of a continuum. For example, angles in a triangle can be described in the following qualitative terms:

$$\begin{array}{ccccc} Zero & \cdots\cdots & Right & \cdots\cdots & Straight \\ & (acute) & & (obtuse) & \end{array}$$

A value can be described qualitatively either as equal to a landmark value or in an open interval bounded by two landmark values, even when numerical information is unavailable. It is often easier to obtain and justify the qualitative description of a quantity than its numerical value, particularly when knowledge is incomplete. Human perception, memory, and similarity judgments often reflect underlying landmark-based qualitative representations [Goldmeier, 1972]. Fortunately, landmark-based descriptions support *qualitative simulation*, to derive qualitative descriptions of the possible behaviors of a system from a qualitative description of its structure [Kuipers, 1986, 1989].

An ordinary differential equation describes a system in terms of a set of variables which vary continuously over time, along with constraints among those variables such as addition, multiplication, and differentiation. A *qualitative differential equation* (QDE) describes a system in much the same terms, except that (1) the values of variables are described qualitatively, and (2) certain functional relationships between variables may be incompletely known. For example, air resistance on a moving body increases monotonically with velocity, and flow of water through an orifice increases monotonically with pressure. Both of these relations are non-linear, but useful qualitative conclusions can be drawn purely from monotonicity. It is useful to define the class M^+ of monotonic functions, and the class S^+ of monotonic functions with saturation.

- A *reasonable function* is a continuously differentiable function defined on a closed interval (including the extended real number line $\Re^* = [-\infty, +\infty]$), with only isolated critical points, and derivatives continuous at the endpoints of the domain. (See Kuipers [1986].)

- M^+ is the set of reasonable functions f such that $f' > 0$ on the interior of its domain. In a QDE, we may write $M^+(pressure, outflow)$ or $outflow = M^+(pressure)$ to mean that there is some $f \in M^+$ such that $outflow = f(pressure)$. M_0^+ is the subset of M^+ such that $f(0) = 0$, and M^- is the set of f such that $-f \in M^+$.

- S^+ is the set of reasonable functions f such that, for specified pairs of landmark values (x_1, y_1) and (x_2, y_2),

  - $f(x) = y_1$ for all $x \leq x_1$,
  - $f(x) = y_2$ for all $x \geq x_2$,
  - $f'(x) > 0$ for all $x_1 < x < x_2$.

  The turning points (x_1, y_1) and (x_2, y_2) must be specified as landmark values whenever the S^+ constraint is used. The subset of S^+ with turning points at (a, c) and (b, d) is called $S^+_{(a,c),(b,d)}$. S^- is the set of f such that $-f \in S^+$.

The qualitative structure of the appropriateness measures in Figure 2 can be expressed in terms of the S^+ constraint by introducing the two functions,

$$s_1(x) \in S^+_{(a,0),(b,1)} \qquad\qquad s_2(x) \in S^+_{(c,0),(d,1)}$$

such that

$$\begin{aligned} l(x) &= 1 - s_1(x) \\ n(x) &= s_1(x)(1 - s_2(x)) \\ h(x) &= s_2(x) \end{aligned}$$

This qualitative description expresses a state of incomplete knowledge about the operating regions and their boundaries. We represent our knowledge of the system, the operating regions, and the local control laws, as a qualitative differential equation.

QSIM [Kuipers, 1986, 1989] allows us to specify a qualitative differential equation (QDE) and a description of an initial state ($QState(t_0)$), and predicts a set of possible behaviors, such that

$$QDE \wedge QState(t_0) \rightarrow or(Beh_1, \ldots Beh_n).$$

The inference done by QSIM is sound, so the set of behaviors $\{Beh_1, \ldots Beh_n\}$ includes all possible behaviors of systems described by QDE and $QState(t_0)$.

Each behavior Beh_i is a sequence of qualitative states representing alternating time-points and qualitatively uniform time-intervals:

$$QState(t_0), QState(t_0, t_1), QState(t_1), \ldots QState(t_{k-1}, t_k), QState(t_k).$$

The set of behaviors is represented by a tree of qualitative states linked by successor relations, so each time-point state $QState(t_i)$ is followed by all possible immediately succeeding time-interval states $QState(t_i, t_{i+1})$, and each time-interval state $QState(t_i, t_{i+1})$ is followed by all possible immediately succeeding time-point states $QState(t_{i+1})$.[2] This tree of states can be viewed as a branching-time description of a set of possible futures.

The tree of possible behaviors of a qualitatively described system can be a powerful analytical tool. In particular, if a qualitative property (e.g. stability or zero-offset) holds on every branch of the tree, it must hold for every behavior of every fully-specified instance of the system. The importance of the *qualitative* level of description is that the tree of behaviors for a given QDE may be finite, whereas the corresponding set of ordinary differential equations and their solutions is infinite.

4.2 QSIM and Temporal Logic

In order to state and prove properties of a continuous dynamic system, we use QSIM to create a symbolic description of the set of all possible behaviors in the form of a tree of qualitative states, then state the desired properties as propositions in a modal temporal logic, and finally check that the propositions are true of the tree of states.

Temporal and modal logics have been developed for expressing and inferring properties of time-varying systems [Emerson, 1990]. They have primarily been applied to discrete-time rather than continuous-time models of the world, such as might arise in verification of computing systems. However, as we have seen, QSIM provides a discrete tree of qualitative states that describes the behaviors of a continuous system. Therefore, we define a modal temporal logic (an instance of Computational Tree Logic (CTL) [Emerson, 1990]) customized for application to QSIM behavior trees.

Following Emerson [1990], we define:

- A *temporal structure* is $M = \langle S, R, L \rangle$, where
  - S is a set of states;

[2] Provisions exist in QSIM for identifying and representing cycles and other repeated states, so the behavior tree may actually be a transition graph.

  - R is a binary successor relation defined on $S \times S$;
  - L is a labelling, associating with each state $s \in S$ a set of atomic propositions.

- The behavior tree resulting from QSIM simulation of QDE and $QState(t_0)$ can be viewed as a temporal structure $M = \langle S, R, L \rangle$, where S is the set of states in the tree, R is the union of the QSIM successor and transition relations, and L labels states in S with atomic propositions as follows.
  - $status(s, tag)$, where s is a state, and tag is one of $\{quiescent, stable, unstable, transition\}$. The proposition is true of a state s in M if tag is an element of `s.status` (an element of the qualitative state description).
  - $qval(s, v, qmag, qdir)$, where s is a state, v is a variable appearing in QDE, $qmag$ is a landmark or interval defined by a pair of landmarks in the quantity space for v, and $qdir$ is one of $\{inc, std, dec\}$ representing the sign of the derivative of v. The proposition is true of a state s in M if the value of v in s satisfies the description $\langle qmag, qdir \rangle$.

- *State formulae* and *path formulae* are built up from atomic propositions. A *path* is a sequence of states in M linked by the successor relation R.

- State formulae have the following syntax and semantics.
  - (S1) Each atomic proposition is a state formula. Its truth conditions have already been defined.
  - (S2) If p, q are state formulae, then so are $p \wedge q$, $p \vee q$, $\neg p$, and $p \rightarrow q$, with the usual rules defining the truth values of composed expressions.
  - (S3) If p is a path formula, then (`necessarily` p) is a state formula, which is true of a state s in M if p is true of *every* path starting at s. Similarly, (`possibly` p) is a state formula, which is true of a state s in M if p is true of *any* path starting at s.

- Path formulae have the following syntax and semantics.
  - (P0) If p, q are state formulae, then (`until` p q) is a path formula, which is true of a path if q is true for some state in the path, and p is true of every previous state in that path. Similarly, (`next` p) is a path formula, which is true of p if it is true of the path starting with the second element of p.
  - We can define the useful forms (`eventually` p) as (`until true` p), and (`always` p) as (`not (eventually (not` p`)))`.

  Basically, CTL allows a path quantifier (`possibly` or `necessarily`) to be followed by a single temporal operator (`until`, `next`, `always`, or `eventually`).

- An extended language, CTL*, allows arbitrary boolean combinations or nestings of the temporal operators. It can be defined by replacing (P0) with the following syntax and semantics for path formulae.
  - (P1) Each state formula is also a path formula, and is true of the path if it is true of the first state in the path.

- (P2) If p, q are path formulae, then so are $p \wedge q$, $p \vee q$, $\neg p$, and $p \rightarrow q$, with the usual rules defining the truth values of composed expressions.
- (P3) If p, q are path formulae, then so is `(until` p q`)`, which is true of a path x if q is true of some suffix path x^i of x, and p is true of every longer suffix path x^j containing x^i. Similarly, `(next` p`)` is a path formula, which is true of p if it is true of the path starting with the second element of p.

- Based on the above definitions, we have written algorithms to check whether a state s in the temporal structure M, derived from a QSIM behavioral prediction, satisfies a given temporal logic proposition P in CTL or CTL*.

- Emerson [1990] provides the following complexity results for the slightly more difficult *model checking* problem, which determines the satisfiability of P for every s in M.
  - The model-checking problem for CTL is in deterministic polynomial time.
  - The model-checking problem for CTL* is PSPACE-complete.

  Therefore, while CTL* is a more expressive language than CTL, model-checking for CTL* is potentially vastly more computationally expensive. CTL appears to be sufficiently expressive for our purposes.

- Interesting statements are naturally expressible in modal temporal logic, and hence can be automatically checked against a qualitative behavior tree. For example,
  - `(necessarily (always` P`))` means that P is true throughout the dynamic behavior of the system.
  - `(necessarily (eventually (and (status quiescent) (status stable))))` and `(necessarily (always (implies (status stable)` P`)))` together imply that the dynamical system has a quasi-equilibrium view, and that P is true in that view.

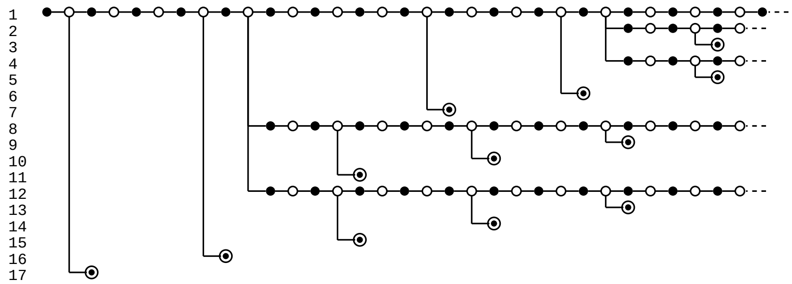
Figure 7 illustrates the checking of statements in CTL against a partial behavior tree produced by QSIM simulation of a simple non-linear PI controller.

QSIM guarantees coverage of every possible behavior by the predictions in the behavior tree. However, its inference capabilities may not be powerful enough to refute every impossible behavior. Therefore, caution is required before concluding that a proposition is true of the behavior of a dynamical system, given the fact that it is true of a QSIM behavior tree.

- If `(necessarily` P`)` is true of the behavior tree, then it is true of every dynamical system matching the QDE and initial state description.

- If `(possibly` P`)` is true of the behavior tree, it may *not* be true of corresponding dynamical systems, since P might have been true only of spurious behaviors in the tree.

Fortunately, in all of our examples, we have needed only statements of the form `(necessarily` P`)`, so their validity applies both to the QSIM behavior tree and to the dynamical systems being reasoned about.



(a)

modal	temporal	logical	QSIM
necessarily	*until*	*and*	*qval*
possibly	*next*	*or*	*status*
	eventually	*implies*	
	always	*not*	

(b)

```
Simulating KJA PI controller.
Behavior tree rooted at S-0, with 1 initial states and 17 behaviors.

Some behaviors don't terminate...
Checking:  (EVENTUALLY (STATUS QUIESCENT)).
Validity = (NIL NIL T NIL T T T NIL T T T NIL T T T T T).

...but all that terminate have zero error.
Checking:  (NECESSARILY (ALWAYS (IMPLIES (STATUS QUIESCENT) (QVAL E (0 STD))))).
Validity = T.

Every fixed point is stable.
Checking:  (NECESSARILY (ALWAYS (IMPLIES (STATUS QUIESCENT) (STATUS STABLE)))).
Validity = T.
```

(c)

Figure 7: Qualitative behavior trees and temporal logic

(a) The possible behaviors of a QDE model of a simplified non-linear PI controller — $\dot{e} + f(e) + \int e = 0$, where $f \in M^+$ — are described by a tree of qualitative state descriptions.

The tree grows – and time passes – from left to right, with each state linked to its possible direct successors. Filled and open circles represent time points and open intervals, respectively. A circled dot represents a fixed point, not necessarily stable. An ellipsis represents an incomplete branch of the tree.

(b) The operators of modal temporal logic provide an appropriate language for querying and describing the behavior tree.

(c) For this model of a simple PI controller, we can conclude that every fixed-point is stable and represents zero error. (There are methods in QSIM for eliminating the non-terminating behavior descriptions, but they are not yet integrated with the CTL validity-checker.)

5 Guarantees for the Water Tank Controller

5.1 Qualitative Combination of Local Properties

Figure 8 summarizes a qualitative analysis of the water tank controller taking the first approach described in figure 6. The analysis begins by determining the direction of motion of the system as specified by each control law individually. The properties of the appropriateness measures are not required. Then, in the regions of overlap, if the directions of change agree, the global law for the heterogenous controller must give motion in the same direction. If the different control laws give motion in opposite directions in the overlap regions, a deeper analysis combining qualitative simulation with order-of-magnitude [Mavrovouniotis & Stephanopoulos, 1988] or semi-quantitative constraints [Kuipers & Berleant, 1988; Berleant & Kuipers, 1992; Kay & Kuipers, 1992] may be able to reduce ambiguity about the system's behavior.

In order to guarantee that the directions of change agree on the overlap regions (a, b) and (c, d), and therefore that the system always ends up within the "pure" operating region (b, c) of the $Normal$ controller, we need to impose constraints on (1) the range of inflow perturbations to be handled, and (2) the magnitude of the $High$ response.

1. From the $Normal$ model:
$$q_b < q < q_c \tag{1}$$
where q_b (respectively, q_c) is the value of q that results in steady state at $x = b$ (respectively, $x = c$).

2. From the $High$ model:
$$q < u_{max} \cdot p(c). \tag{2}$$

The individual steps of the analysis depicted in figure 8(b) are accomplished by simulating qualitative models of the water tank with each individual local controller, over the region where its appropriateness measure is non-zero. With the constraints (1) and (2) incorporated into the models, the predictions have the desired properties. As shown in Figure 9(top), QSIM simulation of the water tank with control law u_l gives three qualitatively distinct behaviors; u_n gives 34; and u_h gives 21.

Once all possible behaviors have been determined, specifications in CTL of the desired properties of each local controller are checked for validity against the behavior tree. In figure 9(bot), we see that the CTL model-checker determines that each of the given specifications is a valid description of the corresponding tree of possible behaviors.

Finally, since the direction of motion of the global system is the average of the directions of motion of the local systems, with non-negative weights, the global system must show the desired qualitative behavior: motion from any point in the state space to a stable fixed-point in the interval (b, c). Note that this conclusion does not depend on other constraints, in particular on the shapes of the appropriateness measures.

5.2 Abstracting Behavior to a Transition Graph

These qualitative properties of the behavior of the system and its heterogeneous controller can be expressed as a finite transition graph in which the nodes correspond to the operating regions, and the directed edges correspond to transitions between regions.

$$\boxed{\text{Low}} \longrightarrow \boxed{\boxed{\text{Normal}}} \longleftarrow \boxed{\text{High}}$$

where the double box signifies that the $Normal$ region includes a steady state, and so can persist indefinitely, while the other regions can persist only for a finite time.

The abstraction relation is defined as follows:

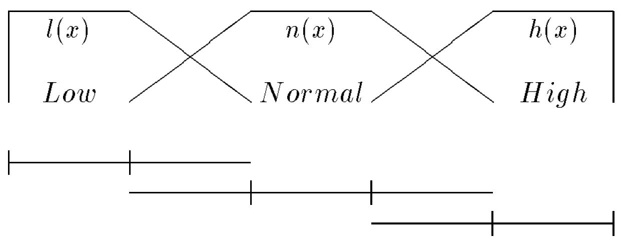
- The state of the system corresponds to a node of the transition graph if it is in the *interior* of the corresponding "pure" operating region, where its appropriateness measure is equal to 1.
- There is a directed link between two nodes in the transition graph if the system state moves monotonically from one node to the other (i.e., there is no quiescent state in the overlap region between the two pure regions). During behavior corresponding to a directed link, the appropriateness measure of one region decreases monotonically, while the other increases monotonically.
- THere may be no other behaviors that intersect the overlap regions.

5.3 Human and Automated Inference

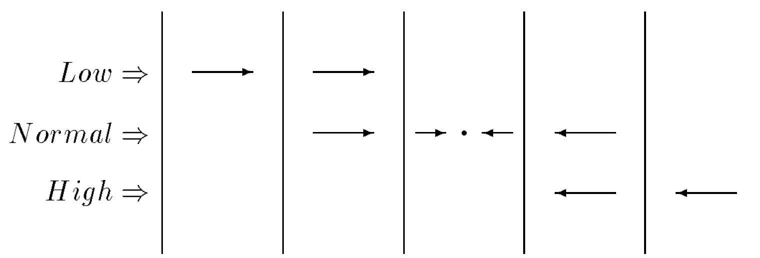
It is important to be clear about which steps in the analysis are and are not automated. Qualitative simulation of the controller models and CTL checking of the specifications are both automated. The specifications themselves are provided by the human analyst. The need for additional constraints is determined automatically, since QSIM simulation produces a behavior tree for which the specifications are not valid. The constraints themselves are determined by the human analyst. Assembly of the local specifications into a guarantee for the heterogeneous controller is done by the human analyst.

In principle, it should be possible to automate these steps as well, but this will require advances to the state of the art in qualitative model-building, symbolic and algebraic manipulation, and temporal logic theorem-proving.

- (a) Overlapping operating regions for the local laws.



- (b) Predict qualitative behaviors; require agreement where local laws overlap.



- (c) Determine constraints to guarantee monotonic behavior in overlap regions.

$$\begin{aligned} Low &\Rightarrow q > 0 \\ Normal &\Rightarrow q_b < q < q_c \\ High &\Rightarrow q < u_{max} \cdot p(c) \end{aligned}$$

- (d) Abstract the control law to a finite transition diagram.

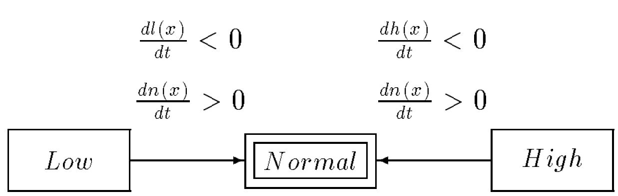
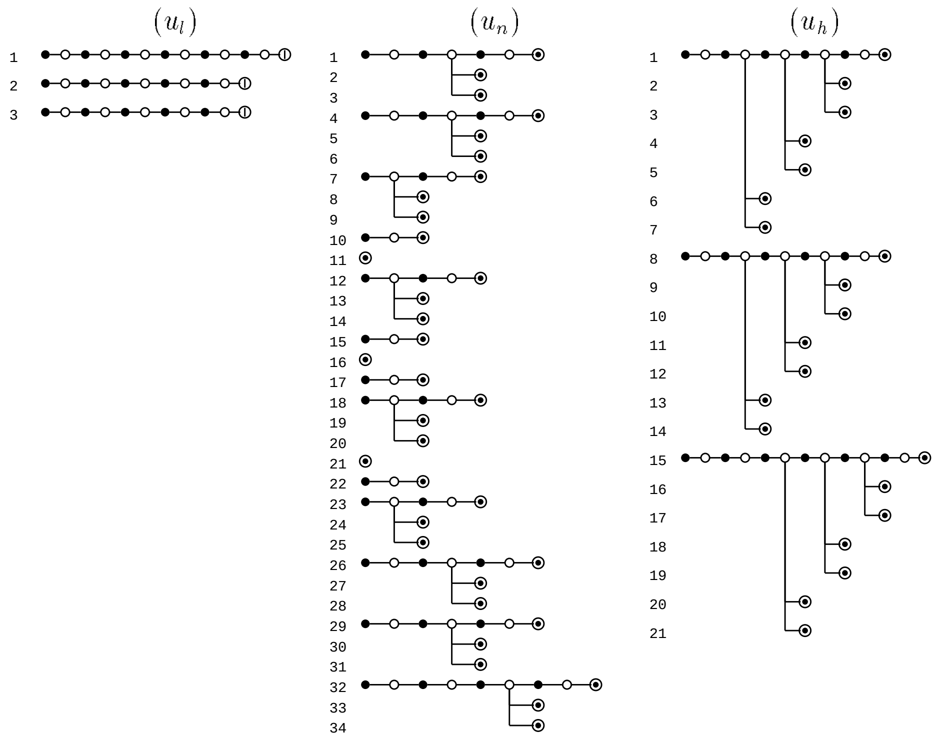


Figure 8: Qualitative combination of properties of local laws.




```
Simulating controller U_l.
Behavior tree rooted at S-0, with 3 initial states and 3 behaviors.
Checking UPWARD-MOTION:  (NECESSARILY (ALWAYS (IMPLIES (QVAL X ((NIL B) NIL))
                                                       (QVAL X (NIL INC))))).
Validity at S-0 = T.
Checking DESTINATION:  (NECESSARILY (EVENTUALLY (QVAL X ((B C) NIL)))).
Validity at S-0 = T.

Simulating controller U_n.
Behavior tree rooted at S-40, with 16 initial states and 34 behaviors.
Checking UPWARD-MOTION:  (NECESSARILY (ALWAYS (IMPLIES (QVAL X ((NIL B) NIL))
                                                       (QVAL X (NIL INC))))).
Validity at S-40 = T.
Checking DOWNWARD-MOTION:  (NECESSARILY (ALWAYS (IMPLIES (QVAL X ((C NIL) NIL))
                                                         (QVAL X (NIL DEC))))).
Validity at S-40 = T.
Checking DESTINATION:  (NECESSARILY (EVENTUALLY (QVAL X ((B C) NIL)))).
Validity at S-40 = T.
Checking STABILITY:  (NECESSARILY (EVENTUALLY (AND (QVAL X ((B C) STD))
                                                   (STATUS QUIESCENT)
                                                   (STATUS STABLE)))).
Validity at S-40 = T.

Simulating controller U_h.
Behavior tree rooted at S-167, with 3 initial states and 21 behaviors.
Checking DOWNWARD-MOTION:  (NECESSARILY (ALWAYS (IMPLIES (QVAL X ((C NIL) NIL))
                                                         (QVAL X (NIL DEC))))).
Validity at S-167 = T.
Checking DESTINATION:  (NECESSARILY (EVENTUALLY (QVAL X ((B C) NIL)))).
Validity at S-167 = T.
```

Figure 9: Local analysis of heterogeneous level-controller

QSIM behavior trees representing the possible behaviors of the water tank controlled by each local law, along with CTL statements implying qualitative agreement among the local laws and justifying the discrete abstraction in Figure 8.

5.4 Qualitative Analysis of the Global Control Law

Following the second path in figure 6, a global analysis of the heterogeneous system is possible when we can establish suitable relations among the individual control laws.

Suppose we can establish that the global control law $u(x)$ is a monotonic function of x. Then the closed-loop system can be described as

$$\dot{x} = q - u(x)\, p(x) = q - f(x), \text{ for some } f \in M^+.$$

Since this is a first-order system, the analysis is straight-forward. An equilibrium exists if q is in the range of f. The solution is unique since f is monotone. The solution is stable because $f' > 0$, since $f \in M^+$.

It is necessary to introduce some compatibility conditions in order to avoid pathological behavior of the system. To see this, consider the case where only two controllers are combined (e.g., the $Normal$ and $High$ controllers over the range (b, ∞) in the water-tank example). The control signal is then

$$u(x) = n(x)\, u_n(x) + h(x)\, u_h(x).$$

It is natural to have controllers such that

$$\frac{du_n}{dx} \geq 0 \text{ and } \frac{du_h}{dx} \geq 0.$$

Unfortunately, these conditions do not guarantee that u is monotone. To obtain this, some auxiliary conditions are required.

Consider

$$u' = n\, u_n' + n'\, u_n + h\, u_h' + h'\, u_h$$

$$\begin{aligned} n + h &= 1 \\ n' + h' &= 0 \end{aligned}$$

The problem is that n' is negative. However, we can conclude:

$$u' = n\, u_n' + h\, u_h' + h'(u_h - u_n)$$

This assures us that $u' > 0$, and hence that $f(x) = u(x)\ p(x)$ is in M^+, if we impose the natural condition

$$u_n(x) \leq u_h(x).$$

This condition needs to hold only for x where the two regions overlap. The argument obviously extends to more complex heterogeneous controllers, such as the water tank, where no more than two regions overlap at any point.

5.5 Integral Action

The bias term in the proportional controller was introduced to make it possible for the controller to keep the level at the set point. Integral action may be viewed as an automatic adjustment of the bias term [See Figure 2.2 in Åström and Hägglund, 1988]. For a simple PI controller the bias is adjusted according to

$$T\frac{du_s}{dt} + u_s = ke + u_s$$

or

$$T\frac{du_s}{dt} = u_p = ke \tag{3}$$

where u_p is the output of the PI controller, e is the error $x - x_s$, k is the proportional gain, and T is the integration time. For a composite controller like the one used in heterogeneous control, u_p should be replaced by the output of the heterogeneous controller.

Analysis of a controller with integral action is more complicated because the closed loop system is described by a second order differential equation and a simple monotonicity argument like the one used previously does not apply directly.

There were two alternative approaches to the qualitative analysis of the heterogeneous "proportional" controller. Similarly, there appear to be three basic approaches to analyzing the "integral" component of a heterogeneous controller.

1. The bias term u_s is adjusted, at a slower time-scale, by a heterogeneous P-controller as a function of the steady-state error, $x_s - x(\infty)$ as discussed below.

2. Local control laws, even with integral action, can be analyzed qualitatively, and associated with overlapping operating regions in the phase plane. If the directions of flow in the overlap regions are compatible, the qualitative descriptions can be combined into a discrete transition-graph representing behavior in the phase plane [Sacks, 1990].

3. The local laws may be combined into a global control law using the weighted average combination rule, which may then be analyzed qualitatively.

One possibility is to exploit the fact that integral action is a slow process. The idea of time scale separation introduced in [Kuipers, 1987] can then be applied. The full details will be given elsewhere. Let us just outline the ideas of the reasoning. Provided that the integration time T is sufficiently small the closed loop system can be decomposed into a fast system, where the bias term is considered constant, and a slow system, where the fast system is considered as a static system. The previous analysis then applies to the fast system. It follows from this analysis that the level goes to an equilibrium which may be different from the set point. At equilibrium the fast system can be described by

$$u_p = -f(u_s) \tag{4}$$

where the function f belongs to M^+. The slow system is described by (4) and (3), i.e.

$$T\frac{du_s}{dt} = u_p = -f(u_s)$$

Since f is monotone this equation has a unique stable equilibrium

$$u_p = ke = 0$$

which implies that the error e must be zero when the slow system reaches equilibrium.

6 Example: A Nonlinear Chemical Reactor

6.1 The System

In order to demonstrate heterogeneous control in a somewhat more realistic context, we consider a simple, but highly nonlinear chemical reactor: a reversible exothermic reaction

$$A \underset{k_{-1}}{\overset{k}{\rightleftharpoons}} R$$

in which the extent of the reaction increases with reactor temperature at lower temperatures and decreases at higher temperatures. This causes the gain of the plant (the dependence of the concentration of R, which is the controlled variable, on the inlet temperature, T_i) to change sign as a function of reactor temperature, making the reaction difficult to control [Economou, et al., 1986].

The process can be modeled by the following system of ordinary differential equations, where A_i and R_i represent the feed species concentrations of the reactant and the product, and the state variables A, R and T represent the outlet species concentrations of the (unreacted) reactant and the product, and the reactor's temperature, respectively. The system is manipulated by varying the inlet feed temperature T_i.

- Reactant mass balance:

$$\frac{dA}{dt} = \frac{1}{\tau}(A_i - A) - k_1 A + k_{-1} R \tag{5}$$

- Product mass balance:

$$\frac{dR}{dt} = \frac{1}{\tau}(R_i - R) + k_1 A - k_{-1} R \tag{6}$$

- Energy balance:

$$\frac{dT}{dt} = \left(\frac{-\Delta H_R}{\rho C_p}\right)(k_1 A - k_{-1} R) + \frac{1}{\tau}(T_i - T) \tag{7}$$

where $k_1 = C_1 \exp(-Q_1/RT)$ and $k_{-1} = C_{-1} \exp(-Q_{-1}/RT)$.

The equilibrium conversion from A to R is a function of temperature with a well-defined maximum, as shown in figure 10. We want to operate the system at the setpoint $R = R_s$ when that is physically possible given the input; otherwise, as close as possible to the maximum conversion. The problem, of course, is that the steady-state gain at the maximum is zero, so manipulation of inlet feed temperature T_i has no effect on the system at that point.

6.2 The Controller

We will cope with the non-linearity of the system by defining three operating regions: Left and Right, having linear control laws with gains of opposite signs; and Dead, a region around the peak of the curve where control actions have no effect, so none is taken.

The heterogeneous controller is described in figure 11.

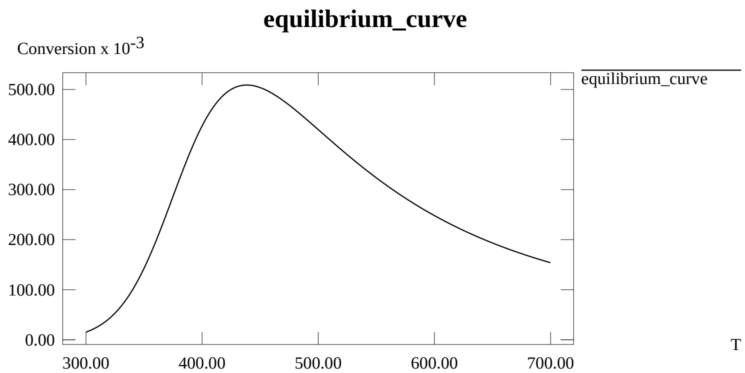


Figure 10: The equilibrium conversion from A to R is a non-monotonic U^- function of temperature, with a well-defined maximum. (Corresponds with [Economou, et al, 1986], figure 7.)

6.3 The Proof

A QSIM model was constructed corresponding to equations (5), (6), and (7), and the heterogeneous controller described in figure 11. The model predicted the response of the system to a change in the inlet reactant concentration, A_i.

A certain amount of effort was required to find a level of abstraction at which the QSIM behavior prediction was tractable. This included a quasi-equilibrium assumption: the controller assumes that the reaction is always near equilibrium. The set of possible behaviors is shown in figure 12.

In response to a change in inlet reactant concentration A_i, the desired behavior is that the output R returns to the setpoint R_s. In case of decreasing A_i this may be impossible, in which case the system should move into the dead zone, $T \in [b, c]$. Within the dead zone, no control action is possible, so we cannot ensure that the system attains the absolute maximum conversion, but since the system's gain is very low in that region, it cannot be very far from the maximum.

We express the desired behavior as the following CTL proposition, stating that the system necessarily moves to a stable quiescent state, either at zero error or in the dead zone.

```
STABILITY:  (:NECESSARILY (:EVENTUALLY (:AND (:STATUS QUIESCENT)
                                             (:STATUS STABLE)
                                             (:OR (:QVAL E (0 STD))
                                                  (:QVAL TO (B STD))
                                                  (:QVAL TO (C STD))
                                                  (:QVAL TO ((B C) STD))))))
```

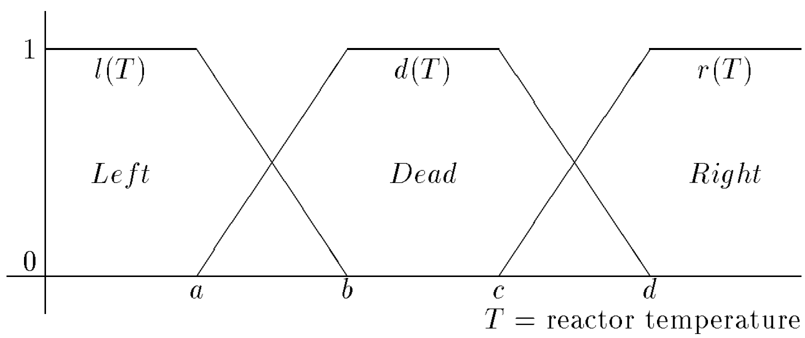
Figure 12 shows the result of using QSIM and CTL to check the validity of this proposition on two scenarios, in which the system starts in equilibrium at optimal conversion and there is an upward or downward perturbation to the inlet reactant concentration A_i.

To illustrate these conclusions by numerical simulation, Figure 13 shows the effect on R and T of temporary but substantial (-20%, -2%, and $+10\%$) perturbations to A_i. The reaction is modeled with the same numerical parameters used in [Economou, et al, 1986], and the heterogeneous controller uses the values $k_l = -1.5$, $k_h = 3.0$, $a = 420.0$, $b = 438.4$, $c = 438.5$, $d = 445.0$.

The non-monotonic reactor:

R	:	the outlet product concentration (controlled)
R_s	:	the setpoint for R
T_i	:	the inlet temperature (manipulated)
T	:	reactor temperature (determines appropriateness)

The operating regions and their appropriateness measures:



The local control laws:

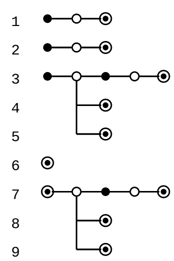
$$\begin{aligned} T \in Left &\Rightarrow u_l(R) = k_l(R - R_s) + u_l, \quad k_l < 0 \\ T \in Dead &\Rightarrow u_d(R) = 0 \\ T \in Right &\Rightarrow u_r(R) = k_r(R - R_s) + u_h, \quad k_r > 0 \end{aligned}$$

The global control law:

$$u(T, R) = l(T)u_l(R) + d(T)u_d(R) + r(T)u_r(R).$$

Figure 11: A heterogenous controller for the non-monotonic reaction.

increase A_i

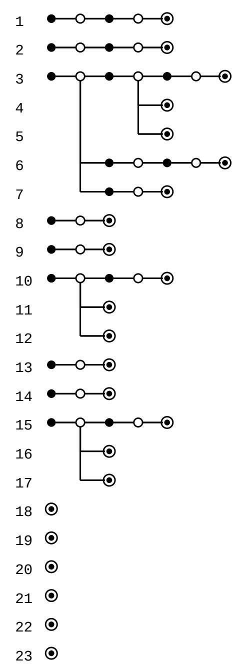



```
Behavior tree rooted at S-2,
  with 3 initial states and 5 behaviors.
 Validity at S-2 = T.

Behavior tree rooted at S-26,
  with 1 initial states and 1 behaviors.
 Validity at S-26 = T.

Behavior tree rooted at S-29,
  with 1 initial states and 3 behaviors.
 Validity at S-29 = T.
```

decrease A_i




```
Behavior tree rooted at S-2,
  with 9 initial states and 17 behaviors.
 Validity at S-2 = T.

Behavior tree rooted at S-312,
  with 3 initial states and 3 behaviors.
 Validity at S-312 = T.

Behavior tree rooted at S-322,
  with 3 initial states and 3 behaviors.
 Validity at S-322 = T.
```

Figure 12: Under both perturbations, the desired stability property holds on all paths. The final states on all paths are quiescent and stable. Either the system reaches its setpoint ($E = R - R_s = 0$) or the dead zone ($T_o \in [b, c]$).

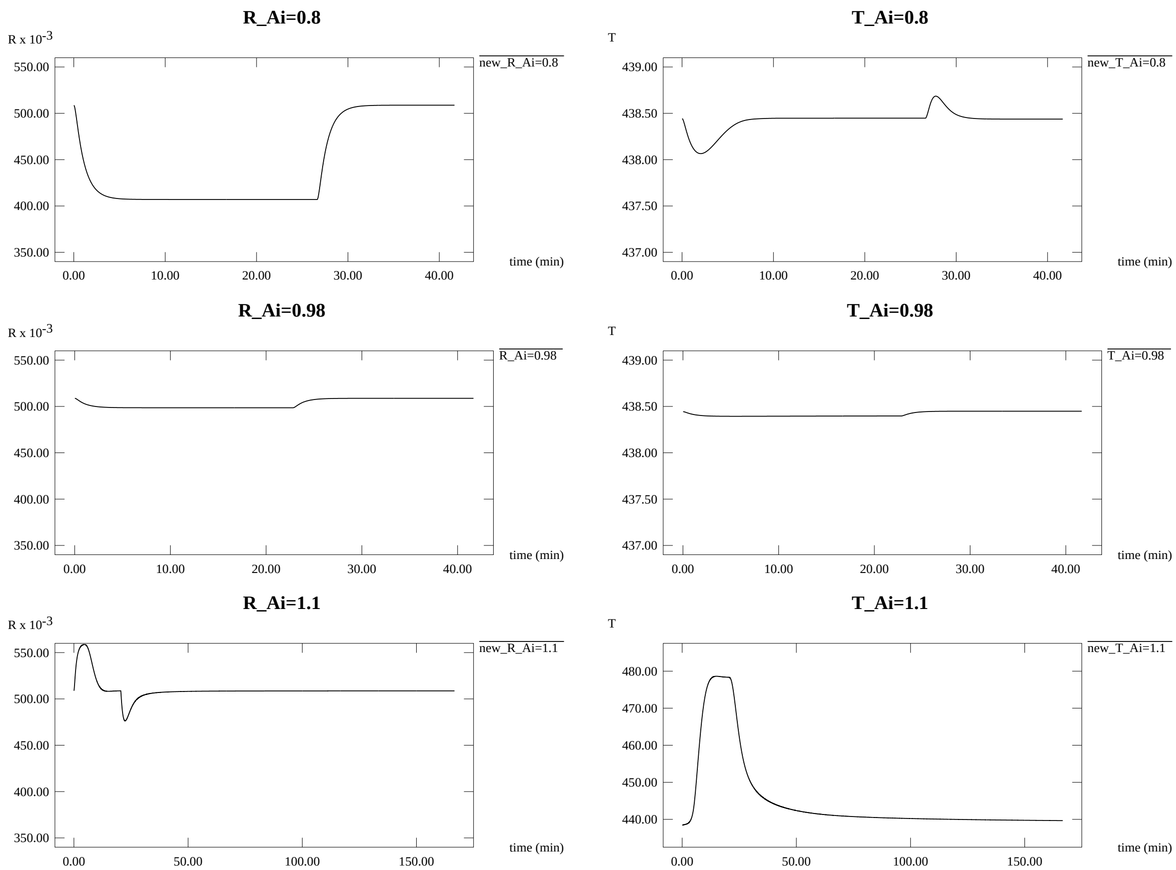


Figure 13: Temporary disturbance (from $t = 0$ to 20 min) of -20%, -2%, and $+10\%$ in A_i, the inlet reactant concentration. ($k_l = -1.5$, $k_h = 3.0$, $a = 420.0$, $b = 438.4$, $c = 438.5$, $d = 445.0$.)

- **-20%**. Temperature at the beginning drops because A_i dropped. Then the controller brings the temperature up, towards the peak, to increase conversion, i.e. keep R high. Of course it fails to keep R to the set point (there is simply not enough input to get that much output), but it keeps it as high as possible keeping conversion near the max. When the disturbance is over ($t = 20$), A_i returns to 1.0, so T starts increasing for a while, till the controller lowers it to get back to the peak of the curve. Note that the total T variation is about 0.5 degree. (Similar to Figure 12 in [Economou, et al, 1986], but for a much larger disturbance).

- **-2%**. Equivalent to Figure 12 in [Economou, et al, 1986].

- **$+10\%$**. Temperature is increased in order to lower R to R_s, in which it succeeds before $t = 20$, and after $t = 20$ the controller drops T to bring the system back to the set point. The time-axis on this graph is extended to demonstrate that the temperature returns to nominal.

7 Conclusion

We have demonstrated a method for composing heterogeneous control laws from simple classical elements. We have also demonstrated a method for validating the composed laws, even in the presence of incomplete knowledge, using qualitative simulation to predict the tree of all possible behaviors of the system and modal temporal logic to check that a desired guarantee holds for that tree.

7.1 Relation to Fuzzy Logic Control

Heterogeneous control is a kind of fuzzy control. It shares many goals with, and draws much inspiration from, fuzzy logic control [Zadeh, 1973; Mamdani, 1974; Kosko, 1992]. However, there are important differences between heterogeneous and fuzzy logic control. Within the framework of fuzzy logic control, it is difficult to exploit, or even relate to, the methods or results of traditional control theory. Our approach uses landmark-based qualitative reasoning to combine the benefits of fuzzy control with the analysis methods of traditional control theory.

Granularity. A fuzzy logic controller is typically specified as a relatively fine-grained set of (fuzzy) regions, with a constant (fuzzy) action associated with each region. In heterogeneous control, the design for a controller specifies a smaller set of possibly overlapping operating regions, but with a classical control law associated with each region.

The net result of these two differences is that a heterogeneous controller requires a simpler specification, while providing the higher-precision control characteristic of classical control laws.

Validation. The concepts underlying fuzzy logic control are relatively difficult to map into the classical framework, making it difficult to exploit existing methods for providing guarantees for the properties of fuzzy logic controllers.

In heterogeneous control, qualitative simulation can be used to analyze the local laws, and to combine their properties to provide guarantees on the global laws.

Ontology. Heterogeneous control does not treat "linguistic values" or "linguistic variables" as objects in either the domain or range of functions. Rather, the fundamental objects in heterogeneous control are real-valued, continuously differentiable functions, and sets of such functions defined by qualitative descriptions.

Linguistic terms are treated simply as names for the operating regions of the mechanism. The specifications for the operating regions are evaluated by qualitative analysis methods.

"Defuzzification." The output of a fuzzy logic controller is an action with a fuzzy magnitude. The fuzzy magnitude must then be mapped to a real value for output. Since heterogeneous control laws are algebraic combinations of classical control laws, they provide real, not fuzzy, outputs, so "defuzzification" is not necessary.

Generality. In the *implementation* of either a heterogeneous or a fuzzy logic controller, fuzzy set membership functions must be represented as specific real-valued functions. However, the *analysis* of a heterogeneous controller relies only on its qualitative description (e.g. S^+, S^-, or $S^+ \cdot S^-$). This makes explicit the fact that a single guarantee applies to a whole class of appropriateness measures, clarifying the degrees of freedom available for implementation decisions. The goal of qualitative analysis is to define the least restrictive description of the controller that provides a given performance guarantee.

8 References

1. Åström, K. T. and T. Hägglund (1988). *Automatic Tuning of PID Controllers.* Research Triangle Park, NC: Instrument Society of America.

2. Berleant, D. and B. Kuipers (1992). Combined qualitative and numerical simulation with Q3. In B. Faltings and P. Struss (Eds.), *Recent Advances in Qualitative Physics*, MIT Press.

3. Economou, C., M. Morari and B. Palsson (1986). Internal Model Control. 5. Extension to Nonlinear Systems. *Ind. Eng. Chem. Proc. Des. Dev.* **25**, 403-411.

4. Emerson, E. A. (1990). Temporal and modal logic. In J. van Leeuwen, (Ed.), *Handbook of Theoretical Computer Science*, Elsevier Science Pub. B. V./MIT Press, pp. 995-1072.

5. Franklin, G. F., J.D. Powell and A. Emami-Naeini (1986). *Feedback Control of Dynamics Systems*, Reading, MA: Addison-Wesley.

6. Goldmeier, E. (1972). *Similarity in Visually Perceived Forms. Psychological Issues 8*: 1; monograph 29, New York: International Universities Press.

7. Kay, H. and B. Kuipers (1992). Numerical behavior envelopes for qualitative models. In *Working Papers of the Sixth International Workshop on Qualitative Reasoning about Physical Systems*, Edinburgh, Scotland.

8. Kosko, B. (1992). *Neural Networks and Fuzzy Systems.* Englewood Cliffs, NJ: Prentice-Hall.

9. Kuipers, B. J. (1986). Qualitative simulation. *Artificial Intelligence* **29**: 289 - 338.

10. Kuipers, B. J. (1987). Abstraction by time-scale in qualitative simulation. *Proceedings of the National Conference on Artificial Intelligence (AAAI-87).* Los Altos, CA: Morgan Kaufman.

11. Kuipers, B. J. (1989). Qualitative reasoning: modeling and simulation with incomplete knowledge. *Automatica* **25**: 571-585.

12. Kuipers, B. and D. Berleant (1988). Using incomplete quantitative knowledge in qualitative reasoning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-88).* Los Altos, CA: Morgan Kaufman.

13. Mamdani, E. H. (1974). Applications of fuzzy algorithms for control of a simple dynamic plant. *Proc. IEEE* **121**: 1585-1588.

14. Mavrovouniotis, M. L. and G. Stephanopoulos (1988). Formal order-of-magnitude reasoning in process engineering. *Computers & Chemical Engineering*, **12**:867-880. Pergamon Press, Ltd.

15. Michie, D. and R.A. Chambers (1968). BOXES: An experiment in adaptive control. In E. Dale and D. Michie (Eds.), *Machine Intelligence 2*, Edinburgh: Oliver and Boyd, 1968, pp. 137-152.

16. Sacks, E. (1990). Automatic qualitative analysis of dynamic systems using piecewise linear approximations. *Artificial Intelligence* **41**: 313-364.

17. Slotine, J.-J. and W. Li (1991). *Applied Nonlinear Control.* Englewood Cliffs NJ: Prentice Hall.

18. Sugeno, M. and G. T. Kang (1986). Fuzzy modelling and control of multilayer incinerator. *Fuzzy Sets and Systems* **18**: 329-346.

19. Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* **15**: 116-132.

20. Weld, D. S. and J. de Kleer (1990). *Readings in Qualitative Reasoning About Physical Systems.* Los Altos, CA: Morgan Kaufman.

21. Utkin, V. I. (1977). Variable structure systems with sliding modes. *IEEE Transactions on Automatic Control* **22**: 212-222.

22. Utkin, V. I. (1991). *Sliding Modes in Control and Optimization.* New York: Springer-Verlag.

23. Yager, R. R., S. Ovchinnikov, R. M. Tong, and H. T. Nguyen (Eds.), (1987). *Fuzzy Sets and Applications: Selected Papers by L. A. Zadeh.* NY: John Wiley & Sons.

24. Zadeh, L. (1965). Fuzzy sets. *Information and Control* **8**: 338-353.

25. Zadeh, L. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems, Man, and Cybernetics* **SMC-3**: 28-44.

9 Acknowledgements

We would like to thank Evangelina Gazi, Lyle Ungar, and the anonymous referees for their help.

This work has taken place in the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Qualitative Reasoning Group is supported in part by NSF grants IRI-8905494, IRI-8904454, and IRI-9017047, and by NASA contract NCC 2-760.