QUALITATIVE SYNTHESIS OF CONFIGURATIONS FOR TWO-TERMINAL SYSTEMS BASED ON DESIRED BEHAVIOR

by

Michel Benaroch
Information Systems Department
Leonard N. Stern School of Business
New York University
New York, NY 10003

and

Vasant Dhar
Information Systems Department
Leonard N. Stern School of Business
New York University
New York, NY 10003

July 1991

Center for Research on Information Systems
Information Systems Department
Leonard N. Stern School of Business
New York University

Working Paper Series

STERN IS-91-18

Abstract

In design, inferring structure from function is a combinatorial generate-and-test problem. Existing methods use prestored domain-specific partial configurations to constrain the generator. We have found that for certain types of economic and physical systems consisting of two-terminal components connected in parallel, it is fruitful to specify function in terms of desired behavior, and to identify sets of components whose resultant behavior matches that desired behavior. In this paper, we present two synthesis operators called *stretch* and *steepen* that operate on qualitatively specified piecewise linear functions that characterize the behavior of components. We are currently applying this model to the domain of financial hedging, where behaviors of the components (stocks, bonds, options, etc.) are specified in terms of two-dimensional piecewise linear relationships, and the goal is to synthesize these to produce a constrained behavior in response to uncontrollable events.

Contents

1 Introduction

Many design problems can be formulated as a process of search in which design parameters are constrained to produce system configurations with some desired functionality (Mittal & Araya, 1986). When the search space of alternative configurations is immense, it is more reasonable to construct configurations rather than prestore them for selection. In such cases, the design problem entails synthesis of configurations.

Synthesis of configurations involves searching the space of permutations of elementary components in a domain. This search problem is combinatorial, and can be shown to be NP-complete. To solve this problem one must therefore use a search process which makes use of good heuristics.

This paper presents a qualitative synthesis technique that we have developed to solve a synthesis problem that is similar to the one above. This problem involves only systems that are configured from two-terminal components (i.e., one input and one output nodes) connected in parallel, where each component has associated with it a number of two-dimensional piecewise linear relationships that characterize its behavior in specified regions. Such relationships are used commonly in economics to model financial instruments, such as bonds, options, etc. More importantly, these relationships are used as a basis for evaluating and managing the risk associated with uncontrollable factors such as interest rates, currency exchange, and so on. Specifically, if the problem goal and constraints are specified in terms of two-dimensional piecewise linear relationships, the search problem is one of permuting these piecewise linear functions in order to satisfy the constraints. Conceptually, the problem could be formulated as a linear programming problem although such a formulation would be difficult to solve. For this reason, heuristics based on the structure of constraints are important for achieving good solutions.

Our qualitative synthesis technique searches a space of two-dimensional piecewise linear functions, each modeling the behavior of one component under its operational regions. The technique constructs configurations by permuting such piecewise linear functions and comparing them against another piecewise linear function that specifies the desired behavior of the system we seek to synthesize.

Our technique uses two means to constrain the search. One is knowledge about algebraic operations on two-dimensional piecewise linear functions (e.g., addition) that are used to create a permutation of such function, and stretching and/or steepening such a function over one of its definitional regions. Such knowledge is used to eliminate permutations associated with configurations that do not satisfy the problem goal. The other means is a qualitative abstraction over the piecewise linear functions modeling the behavior of elementary components with similar functionality. Such an abstraction reduces the number of piecewise linear functions to be permuted.

Unless specified otherwise, the rest of this paper restricts the discussion to systems that are configured from two-terminal components connected in parallel. Moreover, the paper uses the term configuration to refer to the structure of a system.

The rest of the paper is organized as follows. Section 2 provides the theoretical foundation underlying our technique. Section 3 extends Kuipers' (1986) notion of qualitative behavior to define the *complete qualitative behavior* of a two-terminal system and its corresponding *qfunction*, two-dimensional qualitative piecewise linear function, representation. It then defines the terms qualitative configuration of two-terminal systems. Section 4 defines operators on qfunctions which are heuristically used to constrain the search process used by our qualitative synthesis algorithm, which is explained in section 5. Section 6 presents an application of our technique in the domain of financial hedging. Section 7 discusses the scope of our technique, and briefly presents directions for future work.

2 Approach

Much of the work on qualitative reasoning about physical systems is based on modeling the relationship between: *structure* — a collection of components connected as a system; *behavior* — a sequence of states of a system and its components over some time-interval; and *function* — the purpose of structure in producing the behavior of a system. This relationship can be summarized as follows. The behavior of a system results from interactions between the behavior of its components, i.e., the effects of a change in the state of a component propagates locally through structural connections causing a change in the state of other components and the system as a whole. The function of a system, on the other hand, explains in terms of causality why and how structure of a system determines its behavior (De Kleer & Brown, 1984).

The goal of analysis techniques (e.g., qualitative simulation) is to infer behavior from structure. Given a structural description of a system and its initial state, these techniques predict the transitions that a system makes from one state to another, and describe behavior using intuitive terms such as equilibrium and oscillation. Some techniques also describe the transitions of every parameter from one state to another using a number of two-dimensional piecewise linear plots (e.g., Kuipers, 1986).

The primary goal of design techniques is to infer the structure of a system from its function. Inferring structure from function is a search problem involving generate and test, which is harder than inferring behavior from structure (a prediction problem). In order to constrain the generator, many design techniques use prestored knowledge about configurations of systems that are specific to one domain. For example, in the domain of paper transportation, Mittal and Araya (1986) use two types of contextual knowledge to generate configurations. One includes hierarchical decompositions of top-level functional goals (e.g., "design a driver role") into subgoals that are each associated with prestored partial configurations, and the other includes pre-packaged and off-the-shelf configurations. It seems clear that in the lack of such contextual knowledge, inferring structure from function is an overwhelmingly complex task.

For some design situations, however, it makes sense to specify function in terms of the desired behavior of a system over some operational regions, that is, in terms of input values that a system can accept and corresponding output values it should produce. For example, the desired behavior of a hydraulic servo valve is "If the applied

```
   -----------
 --|component 1|--
 |   -----------   |                -----------   -----------
--|   ----------   |--   --|component 1|--|component 2|--
 --|component 2|--                -----------   -----------
   ----------
(a) Parallel components           (b) Serial components
```

Figure 1: Two generic configurations of a system

electrical signal is null the valve should be closed. Otherwise, the valve opening should be proportional to the applied electrical signal." Notice that such desired behavior can be also described by a two-dimensional piecewise linear plot.

This suggests that in some situations it may be possible to reduce the need for pre-storing design configurations by looking at the problem of inferring structure from behavior. Given the desired behavior of a system, the goal is to identify sets of components whose elements are connected in such a way that the overall behavior resulting from interactions between behaviors of those components according to the laws of causality is identical to the desired behavior. However, since the number of configurations (i.e., sets of one or more components and their permutations) is infinite, it is necessary to construct configurations by means of synthesis, rather than select them by means of classification.

2.1 Synthesizing System Configurations

One can construct system configurations by applying algebraic operations on the mathematical functions that model the behavior (i.e., transfer function) of components and systems in a domain (Saucedo & Schiring, 1968). Given two functions that model the behavior of two components, algebraically adding the two functions produces a function that models the behavior of a system with the two components connected in parallel. Multiplying the two functions produces a function that models the behavior of a system with the two components connected in series (see figure 1).

Given that the desired behavior of an objective system (i.e., a system we seek to synthesize) can be specified by a two-dimensional piecewise linear function, configurations can be synthesized by searching the space of combinations of two-dimensional piecewise linear approximations of the mathematical functions describing the behavior of each component (hereafter, elementary function). The goal of that search is to construct a set of elementary functions and a sequence of algebraic operations, such that when we apply that sequence of operations on elements of that set, we produce a two-dimensional piecewise linear function that is identical to the one describing the desired behavior of an objective system.

Two possible piecewise linear approximations for the mathematical function describing the behavior of a two-terminal component that is characterized by parameters

p_1 and p_2 are: $\frac{dp_2}{dp_1} = \begin{cases} slope_1 & p_1 \in region_1 \\ ... & ... \\ slope_n & p_1 \in region_n \end{cases}$ and $\mathrm{sign}(\frac{dp_2}{dp_1}) = \begin{cases} qdir_1 & p_1 \in qval_1 \\ ... & ... \\ qdir_n & p_1 \in qval_n \end{cases}$,
where $slope \in \Re$ is the coefficient of a linear expression, $qdir \in \{-1, 0, 1\}$ is the qualitative direction-of-change of p_2, and $region$ and $qval$ are values bounding a quantitative and qualitative range on the real-line, respectively[1].

3 Behavior and Configuration

This section uses the notion of qualitative behavior of a two-terminal system to define the term *complete qualitative behavior* and the term *qfunction*, which stands for the qualitative piecewise linear representation of a complete behavior. The reader is referred to Kuipers (1986) for a discussion of definitions 3.1 – 3.5. We then define the term *qualitative configuration* of a two-terminal system using the notion of a complete qualitative behavior.

3.1 Qualitative Behavior

A physical system is characterized by multiple real-valued continuously time-varying parameters. A parameter is considered to be a *reasonable function*, $f : [a, b] \rightarrow \Re^*$, where $\Re^* = [-\infty, \infty]$ is the extended real-line[2]. A reasonable function has a finite totally ordered set of *landmark values*, $l_1 < l_2 < ... < l_k$, which must include 0, $f(a)$, $f(b)$, the value of $f(t)$ at every critical point, and may include additional values. It also has a finite totally ordered set of *distinguished time-points*, $a = t_0 < t_1 < ... < t_n = b$, each designating a point where something important happens to the value of f, such as passing a landmark value or reaching an extremum. All functions mentioned from now on should be presumed reasonable.

Definition 3.1 Let $l_1 < ... < l_k$ be the landmark values of $f : [a, b] \rightarrow \Re^*$. For every $t \in [a, b]$ the *qualitative state of f at t*, $\mathrm{QS}(f, t)$, is a pair $\langle qdir, qval \rangle$, where $qdir$ is 1, 0, or -1 for $f'(t) > 0$, $f'(t) = 0$, or $f'(t) < 0$, respectively, and $qval$ is a landmark value, l_i. The *qualitative state of f on an interval* between two adjacent distinguished time-points, $\mathrm{QS}(f, t_i, t_{i+1})$, is the qualitative state of f at any $t \in (t_i, t_{i+1})$.

Definition 3.2 The *qualitative behavior of f on* $[a, b]$, $\mathrm{QB}(f, [a, b])$, is the sequence of qualitative states: $\mathrm{QS}(f, t_0)$,$\mathrm{QS}(f, t_0, t_1)$,...,$\mathrm{QS}(f, t_n)$, alternating between states at distinguished time-points and states on intervals between distinguished time-points.

[1] Since the number of functions necessary to represent the behavior of a system increases with the number of its terminal nodes, it becomes more difficult to use this representation to specify the behavior of a multi-terminal system. For example, while the behavior of a two-terminal system can be represented by one function of the above form (i.e., it can be plotted as one line in a two dimensional space), the behavior of a three-terminal system must be represented by more than one function of the above form (i.e., it can be plotted as a plane in a three dimensional space).

[2] f is continuous on $[a, b]$, continuously differentiable on (a, b), has a finite number of critical values in any interval, and has $f'(a)$ and $f'(b)$ as the left and right limits of $f'(t)$ at a and b, respectively.

Definition 3.3 A *two-terminal system*, $S=\{p_1,p_2\}$, is a pair of functions each with its set of landmark values. The *distinguished time-points of S* are the union of the distinguished time-points of p_1 and p_2. The *qualitative state of S* is the 2-tuple of states: $\mathrm{QS}(S,t_i)=[\mathrm{QS}(p_1,t_i),\mathrm{QS}(p_2,t_i)]$, $\mathrm{QS}(S,t_i,t_{i+1})=[\mathrm{QS}(p_1,t_i,t_{i+1}),\mathrm{QS}(p_2,t_i,t_{i+1})]$.

Definition 3.4 Let l_i and l_j be landmarks of $f_1, f_2:[a,b]\rightarrow\Re^*$, respectively. Landmarks l_i and l_j are *corresponding values*, if there is $t\in[a,b]$ such that $f_1(t)=l_i$ and $f_2(t)=l_j$. Such t is called a *corresponding time-point* of f_1 and f_2.

Definition 3.5 A *qualitative behavior of a two-terminal system S*, $\mathrm{QB}(S,t_0,t_n)$, is a sequence of qualitative states: $\mathrm{QS}(S,t_0)$,$\mathrm{QS}(S,t_0,t_1)$,$\mathrm{QS}(S,t_1)$,...,$\mathrm{QS}(S,t_n)$.

Definition 3.6 Let $S=\{p_1,p_2\}$ be a two-terminal system with distinguished time-points $t_0<...<t_n$. The totally ordered *set of corresponding time-points of S*, T_S, contains t_0, t_n, and every corresponding distinguished time-point of $p_1(t)$ and $p_2(t)$.

3.1.1 Complete Qualitative Behavior

A system can exhibit a number of different qualitative behaviors depending on the initial state. Given that a system has some initial state at t_0 (i.e., it is perturbed from equilibrium), one qualitative behavior is the sequence of qualitative states that system goes through from t_0 to t_n. Since in a two-terminal system only the input parameter can be perturbed, such a system will exhibit all of its possible behaviors if we let the input parameter increase over the operational regions of that system. Based on this, we define the notion of a *complete qualitative behavior* of a two-terminal system.

Definition 3.7 Let $S=\{p_1,p_2\}$ be a two-terminal system with a set of corresponding time-points $T_S=\{t_0,...,t_n\}$. The *complete qualitative behavior of S*, $\mathrm{CQB}(S,t_0,t_n)$, is the sequence of states: $\mathrm{QS}(S,t_0)$,$\mathrm{QS}(S,t_0,t_1)$,...,$\mathrm{QS}(S,t_{n-1},t_n)$,$\mathrm{QS}(S,t_n)$, where $t_i\in T_S$ and $qdir(p_1,t)=1$ (i.e., inc) for every $t\in[t_0,t_n]$.

3.1.2 Qfunction Representation of a Complete Behavior

The complete qualitative behavior of a two-terminal system can be described by a two-dimensional piecewise linear plot (see figure 2). Notice, however, that exclusion of the qualitative states at corresponding distinguished time-points from a complete behavior does not change the shape of such a piecewise linear plot.

Definition 3.8 Let $\mathrm{CQB}(S,t_0,t_n)=\mathrm{QS}(S,t_0)$,$\mathrm{QS}(S,t_0,t_1)$,...,$\mathrm{QS}(S,t_{n-1},t_n)$,$\mathrm{QS}(S,t_n)$ be the complete qualitative behavior of a two-terminal system S. The *qualitative piecewise linear function (hereafter, qfunction) of S*, $\mathrm{QF}(S)$, is the sequence of states on time-intervals: $\mathrm{QS}(S,t_0,t_1)$, $\mathrm{QS}(S,t_1,t_2)$, ..., $\mathrm{QS}(S,t_{n-1},t_n)$.

```
        p2(t)
          |
        y3+  -----
          | /
          |/
--+--+--+--/--+----p1(t)
 x1 x2 x3/|  x4
    ____/ +y2
   /      |
__/      y1+
```

```
QF(S)=(((t0,t1) (p1 1 (-inf,x1))(p2 0 [y1]    ))
       ((t1,t2) (p1 1 (x1,x2)  )(p2 1 (y1,y2)))
       ((t2,t3) (p1 1 (x2,x3)  )(p2 0 [y2]    ))
       ((t3,t4) (p1 1 (x3,x4)  )(p2 1 (y2,y3)))
       ((t4,t5) (p1 1 (x4,inf) )(p2 0 [y3]    )))
```

Figure 2: Qfunction representation of a complete qualitative behavior

3.2 Qualitative Configuration

A physical system is a collection of disjoint components (i.e., systems) that are structurally connected in parallel (and/or series). We define one *configuration constraint* called PARALLEL which is used to represent parallel connection of components. Accordingly, we consider a configuration of a two-terminal system to be a collection of PARALLEL constraints applied on two-terminal components.

Definition 3.9 Let $S_1=\{p,p_1\}$ and $S_2=\{p,p_2\}$ be two-terminal systems connected in parallel to produce a two-terminal system $S_3=\{p,p_3\}$, $p,p_1,p_2,p_3:[a,b]\rightarrow\Re^*$. PARALLEL$(S_1,S_2,S_3)$ is a three-place predicate on two-terminal systems which holds iff $p_1(t)+p_2(t)=p_3(t)$ for every $t\in[a,b]$.

Since the functionality of a component can be modeled using a qfunction in terms of its complete qualitative behavior, we can relate the PARALLEL constraint to equivalent algebraic operations on qfunctions. We define the qfunction of a system configured from two components connected in parallel as the sum of qfunctions of the two components. Thus, we can determine if PARALLEL(S_1,S_2,S_3) hold by comparing QF(S_1)+QF(S_2) against QF(S_3).

Definition 3.10 Let $S_1=\{p,p_1\}$, $S_2=\{p,p_2\}$, $S_3=\{p,p_3\}$ be two-terminal systems with set $T_S=\{t_0,...,t_n\}$ of corresponding time-points. QF(S_1)+QF(S_2)=QF(S_3) iff for every i, $0<i<n$, the following conditions hold:
1) $qdir(\mathrm{QS}(p_1,t_i,t_{i+1}))+qdir(\mathrm{QS}(p_2,t_i,t_{i+1}))=qdir(QS(p_3,t_i,t_{i+1}))$,
2) $qval(\mathrm{QS}(p_1,t_i))+qval(\mathrm{QS}(p_2,t_i))=qval(\mathrm{QS}(p_3,t_i))$.

4 Synthesis Operators on Qfunctions

Suppose we seek to create an electrical analog computer whose desired behavior over $(0,\infty)$ is described by QF(G) in Figure 3a. It seems clear from Figure 3 that our proposed approach can synthesize QF(G) from qfunctions QF(S_1) and QF(S_2), where

```
p2(t)
 |     QF(G)
 |  -----
 | /     \
 |/       \
 |         \
-----+----+\----p1(t)
 |  x1   x2\
```

(a) A goal qfunction

```
Dead-Zone component

p2(t)
 |     QF(S1)
 |
 |--
 |  \
 |   \
--|--+-\---p1(t)
 | x1 \
```

```
Limit component

p2(t)
 |     QF(S2)
 |
 |  -----
 | /
 |/
--|--+-----p1(t)
 | x1
```

(b) Qfunctions of two analog computing components (Chorafas, 1965)

Figure 3: Creating an analog computer by synthesizing qfunction

S_1 is a member of class $\mathcal{D}$ of dead-zone computing components and S_2 is a member of class $\mathcal{L}$ of limit computing components (see Figure 3b). Yet, according to Definition 3.10 QF(S_1)+QF(S_2) $\neq$QF(G). Recall that a qfunction is by definition a qualitative description of the piecewise linear function approximating the behavior of a system. Thus, QF(S_1) and QF(S_2) are each an abstraction of the qfunction of every component in class $\mathcal{D}$ and $\mathcal{L}$, respectively. What is therefore necessary is to identify every one component in $\mathcal{D}$ and every one component in $\mathcal{L}$ for which the sum of their qfunctions is equal to QF(G). To avoid generating every combination of qfunctions for elements in $\mathcal{D}$ and $\mathcal{L}$ and compare it against QF(G), we use heuristic operators to constraint the generator. We define two heuristic operators on qfunctions — STRETCH and STEEPEN — which can change the definition of a two-dimensional piecewise linear function over its definitional regions. The result of applying these operators provides more information about certain characteristics of components in a configuration.

4.1 The *Stretch* Operator

For equality QF(S_1)+QF(S_2) =QF(G) to be valid, Definition 3.10 assumes that systems S_1, S_2, and G have the same set of corresponding time-points. When this assumption is violated, it is necessary to modify QF(S_1) or QF(S_2) so as to discover new corresponding time-points for which this assumption holds. We can do that using operator STRETCH(QF(S),i,l_j) whose parameters are a qfunction, an element number in that qfunction, and a landmark of the input parameter of that system.

Assume that QF(S_1)+QF(S_2) =QF(G) does not hold because the sum of p_2-qdirs in elements i in QF(S_1) and QF(S_2) is not equal to that in element i in QF(G). If the sum of p_2-qdirs in elements $i-1$ and i in QF(S_1) and QF(S_2), respectively, is equal to that in element i in QF(G), element $i-1$ in QF(S_1) can be stretched to hold over a larger time-interval so as to discover a new pair of corresponding landmarks of S_1.

```
p2(t)                                           p2(t)
  |   _____QF(G)                                  |   _____QF(G)
y2+  /      \                                   y2+  /      \
  | /        \                                    | /        \
  |/  ______\_____QF(S2)                          |/  ______\_____QF(S2)
y1+__/_      \                                  y2+__/_____ \
  | /  \      \       STRETCH(QF(S1),2,x2) ==>    | /        \ \
  |/    \QF(S1)                                   |/          \QF'(S1)
-----+--\-+-----p1(t)                           -----+----+-----p1(t)
  |   x1   x2                                     |   x1   x2
   (a) Before stretching                           (b) After Stretching
```

Figure 4: The stretch operator

That operator could be also applied on element $i-1$ in S_2 if the sum of p_2-qdirs in elements i and $i-1$ in QF(S_1) and QF(S_2) is not equal to that in element i in QF(S_3).

Let us look, for example, at figure 4 where we try to check if QF(S_1)+QF(S_2) is equal to QF(G). The second elements in QF(S_1), QF(S_2), and QF(G) are, respectively:

```
((t1,t2) (p1 1 (x1,x2))(p2 -1 (minf,y1)))
((t1,t2) (p1 1 (x1,x2))(p2  0 [y1]     ))
((t1,t2) (p1 1 (x1,x2))(p2  0 [y2]     )).
```

The sum of p_2-qdirs in these elements are inconsistent, but they could be consistent if the second element of QF(S_1) is

```
((t1,t2) (p1 1 (x1,x2))(p2 0 [y1])).
```

The application of operator STRETCH(QF(S_1), $1, x_2$) replaces the first element in QF(S_1) by the two elements

```
((t0,t1) (p1 1 (0,x1) )(p2 0 [y1]))
((t1,t2) (p1 1 (x1,x2))(p2 0 [y1])),
```

to produce QF′(S_1). It leads to the conclusion that the sum of the second elements in QF′(S_1) and QF(S_2) is equal to the second element in QF(G), if landmarks x_2 of p_1 and y_1 of p_2 are corresponding in S_1.

One can interpret the result of operator STRETCH in the context of a specific domain to derive more information about characteristics of the components in a certain configuration. For example, one can interpret Figure 4b in the domain electrical analog computers as follows. QF′(S_1) and QF(S_2) are the qfunctions of a dead-zone component and a limit component. QF′(S_1) indicates, however, that in order for a configuration of the two components connected in parallel to have a qfunction that is identical to QF(G), S_1 must have a dead-zone over the range $(0, x_2)$ rather than over $(0, x_1)$.

4.2 The *Steepen* Operator

Let us assume for the moment that we allow a *qdir* to take its value from quantity space $\Re$ (i.e., *slope*). For equality QF(S_1)+QF(S_2) =QF(G) to be valid, Definition 3.10

```
p2(t)                                          p2(t)
  |   _____QF(S3)                                |   _____QF(S3)
y2+  /     \                                   y2+  /     \
  | /      \                                     | /      \
  |/  ______\_____QF(S2)                         |/  ______\_____QF(S2)
y1+__/_____  \                                 y2+__/_____  \
  | /      \ \       STEEPEN(QF(S1),2,-2) ==>    | /      \ \
  |/        \QF'(s1)                             |/        \QF"(S1)
 -----+----+-\---p1(t)                          -----+----+\----p1(t)
  |   x1   x2                                    |   x1  x2\
    (a) Before steepening                          (b) After Steepening
```

Figure 5: The steepen operator

assumes that for every time-interval between two consecutive corresponding time-points the sum of p_2-qdirs in QF(S_1) and QF(S_2) is equal to that in QF(G). When this assumption is violated for a specific element, it is necessary to modify QF(S_1) or QF(S_2) so as to find a new set of p_2-qdirs for which this assumption holds. We can do that using operator STEEPEN(QF(S_1),i,q), which changes the p_2-qdir in element i in QF(S_1) to be q, the difference between the p_2-qdirs in elements i in QF(G) and QF(S_2). Operator STEEPEN is applicable only when the p_2-qdir in element i in QF(S_1) is not zero and $q \neq 0$ (i.e., which would mean changing the shape of a qfunction).

Consider the example in Figure 5. Applying operator STEEPEN on the second element in QF′(S_1) would change the p_2-qdir in that element to be -2, so as to produce QF″(S_1). Assuming that QF″(S_1) and QF(S_2) are the qfunctions of the same configuration in the previous example, the result of applying operator STEEPEN on QF(S_1) would indicate that the conversion rate (i.e., multiplication coefficient of the dead-zone component) should be greater than 1.

5 Qualitative Synthesis

This section describes a qualitative synthesis algorithm called QSYN, which uses a search process to synthesize all feasible configurations of a system of two-terminal components that are connected in parallel. Given that the functionality of every two-terminal system can be represented by a qfunction (i.e., in terms of behaviors it can produce), QSYN synthesizes configurations by searching the space of sets of qfunctions and their permutations. QSYN's goal is to construct all sets of qfunctions, such that when members of a constructed set are summed up they produce a qfunction that is identical to the one describing the desired behavior of an objective system.

5.1 Input and Output

The QSYN algorithm receives as input: (1) two reasonable functions p_1 and p_2, each with its totally ordered set of landmarks; (2) a qfunction, QF(G), associated with an objective system $G = \{p_1, p_2\}$; and (3) a set $\mathcal{QF}$ of n qfunctions, QF($\mathcal{C}_l$), each associated with a class of components $\mathcal{C}_l = \{p_1, p_2\}$, $1 \leq l \leq n$.

The output of QSYN is a set of all feasible configurations for system G. Each configuration is a two-terminal system whose qfunction is identical to the qfunction of G. A configuration is a set of PARALLEL constraints on two-terminal components in class $\mathcal{C}_l$, $1 \leq l \leq n$. Each component is a associated with its possibly modified qfunctions. A modified qfunction is one on which the STRETCH and/or STEEPEN operators were applied. It provides some information about characteristics of components in a specific configuration.

5.2 Algorithm QSYN

Given we can use Definition 3.10 to compute the sum of two qfunctions and compare it against another qfunction, and given we know the conditions under which it is worthwhile applying operators STRETCH and STEEPEN to avoid search the infinite space of qfunctions exhaustively, algorithm QSYN can be summarized as follows:

1. If there is a pair of qfunctions, QF($\mathcal{C}_i$) and QF($\mathcal{C}_j$), in $\mathcal{QF}$ that have not yet been selected, select it. Otherwise, stop.

2. Compare the sum QF($\mathcal{C}_i$)+QF($\mathcal{C}_j$) =QF($\mathcal{C}_{ij}$) against QF(G). If necessary and appropriate, apply operator STRETCH on QF($\mathcal{C}_i$) and/or QF($\mathcal{C}_j$) to discover new corresponding landmarks of p_1 and p_2, apply operator STEEPEN on QF($\mathcal{C}_i$) and/or QF($\mathcal{C}_j$) to force consistency of *qdirs*.

3. If QF($\mathcal{C}_{ij}$) matches part of QF(G), add QF($\mathcal{C}_{ij}$) to $\mathcal{QF}$ and go to step 1.

4. If QF($\mathcal{C}_{ij}$) matches all of QF(G), report the names of the class of components associated with QF($\mathcal{C}_i$) and QF($\mathcal{C}_j$) (if QF($\mathcal{C}_i$) or QF($\mathcal{C}_i$) were created from two other qfunctions, report the names of the class of components associated with them). Go to step 1.

6 Financial Hedging — An Application

We have applied qualitative synthesis in a domain called financial hedging. Hedging is concerned with the design of financial instruments[3] that provide protection against potential losses due to future uncertain events. The underlying principle of hedging is, given an uncertain event and an asset that is sensitive to that event, to match that asset with a liability (asset) whose sensitivity to the event is similar (opposing).

[3]We can apply qualitative synthesis in hedging because the behavior of a financial instrument is modeled very much like that of a physical system.

```
VHI
 |        $
v2+****** $                  $$$$$$ Buy  one call       (QF(C4))
 |     /*                    ****** Sell two call       (QF(C5))
 |    /$*\                   ______ Goal payoff-profile (QF(G))
 |   /$  *\
v1+__/$   * \
 |$$$      * \               S   - Stock Price
 +--+---+-*--\-->S           VHI - Value of Hedge Instrument
   s1  s2  *  \
```

Figure 6: A payoff-profile

The primary design goal of hedging is to identify or construct financial instruments that provide a certain payoff-profile (see Benaroch and Dhar [1991] for other design goals). A payoff-profile specifies qualitatively what a trader is willing to pay and risk based on his beliefs regarding how the behavior of certain economic factors is likely to change over a specific period, and his assessments of how this change is likely to effect the behavior of financial instruments (i.e., monetary value, annual-return, risk). To derive a payoff from such assessments, a trader can use a hedge instrument that takes into account his predictions.

If a trader believes, for example, that over the next month the price of a stock, S, will increase above s_1 but not above s_2, he can define the payoff-profile in figure 6. One instrument that provides that payoff-profile entails buying one call option[4] and selling more than one call option with exercise prices s_1 and s_2, respectively (i.e., *Ratio Spread*). Buying a call with exercise price s_1 ensures that the trader will not lose money if the price of S moves below s_1, that is, if the stock price moves above s_1, the trader will make money by buying stocks for s_1. Since the trader also believes that the price of S will not move above s_2, he can make a profit by selling two calls to another party that does not share his belief.

A trader can use two types of hedge instruments. One is generic instruments such as options. There is a large number of generic instruments that each provides a different payoff-profile. The other type is compounded instruments such as ratio spread. A compounded instrument is a set of two or more generic instruments that are combined in a certain way. Accordingly, its payoff-profile is a combination of the payoff-profiles of the generic instruments by which it is constructed.

As the number of compounded instruments one can construct is virtually infinite, one can not prestore all their payoff-profiles. Rather one must synthesize them from payoff-profiles of generic instruments. Though the number of payoff-profiles of generic instruments is large, they all fall into a small number of classes of payoff-profile, when they are described qualitatively. These payoff-profiles are presented in figure 7 — profiles C_1, C_2, and C_3 are provided by non option based instruments, C_4 and C_5 by

[4]The buyer of a call option on a stock with exercise price s has the right, but not the obligation, to buy from the call seller that stock at the exercise price s at some future expiration date.

```
  QF(C1)    QF(C2)    QF(C3)    QF(C4)     QF(C5)    QF(C6)    QF(C7)
VHI       VHI       VHI       VHI        VHI       VHI       VHI
 |  /      |         |         |          |         |         |
 | /       |\        |------   |   /      |__       |         |  ----
 | /       | \       |         |  /       | \       |\        | /
 |/        |  \      |         |__/       |  \      | \____   |/
 |         |   \     |         |          |   \     |         |
 +-----S   +----\-S  +------S  +--+---S   +--+---S  +--+---S  +--+---S
                 \                s1         s1        s1        s1
   S   - Stock price
   VHI - Value of Hedge Instrument
   s1  - arbitrary exercise price
```

Figure 7: Generic payoff-profiles

call option based instruments, and $\mathcal{C}_6$ and $\mathcal{C}_7$ by put option based instruments. We shall refer to these as generic payoff-profiles.

The construction of compounded instruments that provide a payoff-profile that matches the one specified by a trader can be solved using qualitative synthesis. Generic instruments are the components used to configure compounded instruments. A class of similar components has associated with it one generic payoff-profile which is the qfunction of every component in that class. For reasons that will be clarified in the next subsection, we let *qdir* take its value from quantity space $\Re$. The qfunction (i.e., desired behavior) of the compounded instrument we seek to synthesize is the trader's payoff-profile.

6.1 Example — Constructing a *Ratio Spread*

Suppose we are trying to construct a compounded instrument that provides the payoff-profile in figure 6. While doing so we also try to see if the combined pair of generic payoff-profiles QF($\mathcal{C}_4$) and QF($\mathcal{C}_5$) in figure 7 matches the trader's payoff-profile (see trace in figure 8).

We are given the two reasonable functions S (stock price)and VHI (value of hedge instrument), each with its set of landmarks, and the following three qfunctions:

```
QF(C4) = (((S 1 (0 ,s1 )) (VHI  0 [v1]       ))
          ((S 1 (s1,inf)) (VHI  1 (v1,inf)))))
QF(C5) = (((S 1 (0 ,s1 )) (VHI  0 [v2]       ))
          ((S 1 (s1,inf)) (VHI -1 (minf,v2))))
QF(G)  = (((S 1 (0 ,s1 )) (VHI  0 [v1]       ))
          ((S 1 (s1,s2 )) (VHI  1 (v1,v2)  ))
          ((S 1 (s2,inf)) (VHI -1 (minf,v2)))),
```

for which we need to check if QF($\mathcal{C}_4$)+QF($\mathcal{C}_5$) = QF(G) hold.

In the first triplet of elements from the three qfunction,

```
QF(C4,1) = ((S 1 (0,s1)) (VHI 0 [v1]))
QF(C5,1) = ((S 1 (0,s1)) (VHI 0 [v2]))
QG(G,1)  = ((S 1 (0,s1)) (VHI 0 [v1])),
```

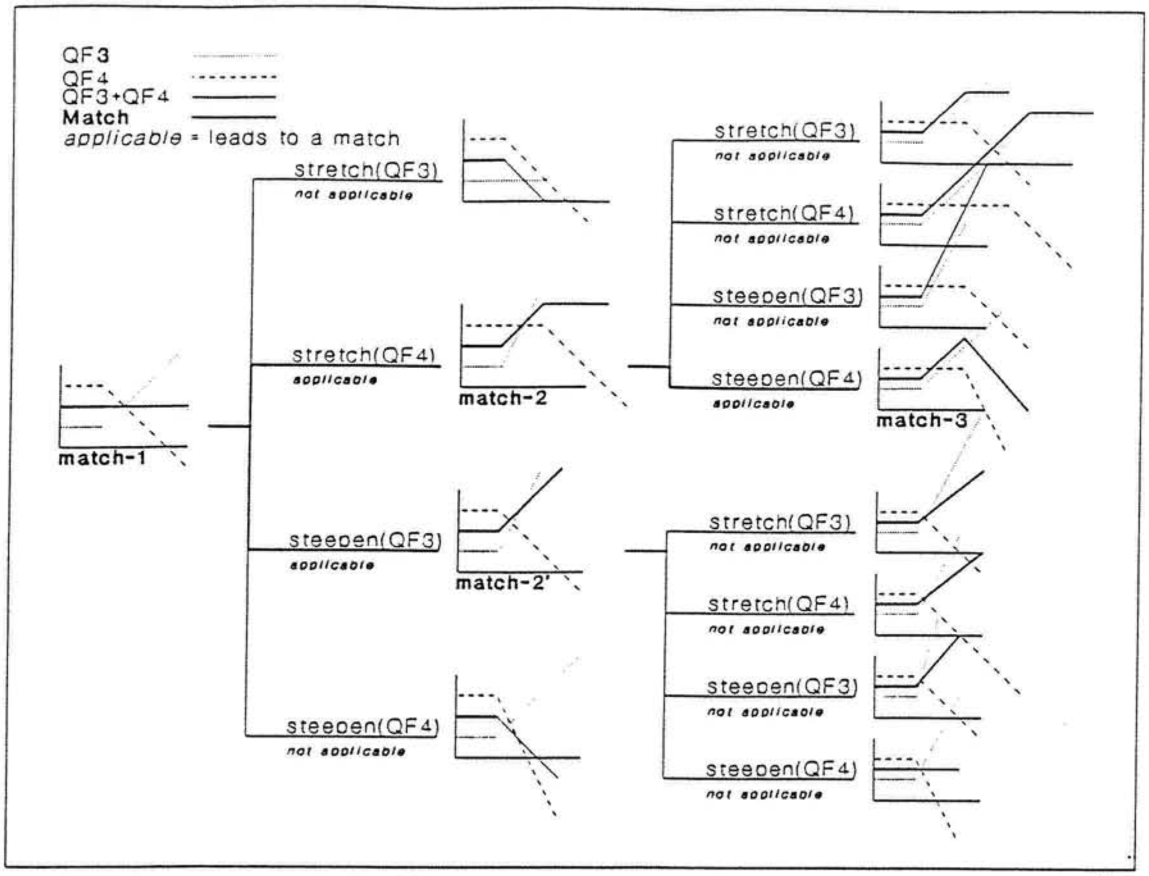

Figure 8: Part of QSYN's search tree for a *Ratio Spread* profile

the *VHI*-qdirs are consistent, and we conclude that the combined first elements of the summed qfunctions match the first element in QF(G).

The second triplet of qfunction elements is:

```
QF(C4,2) = ((S 1 (s1,inf)) (VHI  1 (v1,inf) ))
QF(C5,2) = ((S 1 (s1,inf)) (VHI -1 (minf,v2)))
QF(G,2)  = ((S 1 (s1,s2) ) (VHI  1 (v1,v2)  )),
```

for which the *VHI*-qdirs are inconsistent. We therefore try to apply synthesis operators on qfunctions. In this situation we can apply STRETCH on the first element of QF($\mathcal{C}_5$) (i.e., STRETCH is not applicable on the first element of QF($\mathcal{C}_4$) since the *VHI*-qdirs of the new triplet of second elements would be 0, -1, and 1, which are inconsistent). We can also apply STEEPEN on the second element of QF($\mathcal{C}_4$) to force its qdir to become 2 (i.e., applying STEEPEN on the second element of QF($\mathcal{C}_5$) will force the qdir in that element to become 0, which is not allowed). Let us continue tracing only the case where we STRETCH the first element in QF($\mathcal{C}_5$) such that QF($\mathcal{C}_5$) becomes:

```
QF'(C5) = (((S 1 (0 ,s1 )) (VHI  0 [v1]      ))
           ((S 1 (s1,s2 )) (VHI  0 [v1]      ))
           ((S 1 (s2,inf)) (VHI -1 (minf,v1)))).
```

We are now dealing with a new triplet of second elements,

```
QF(C4,2)  = ((S 1 (s1,inf)) (VHI 1 (v1,inf) ))
QF'(C5,2) = ((S 1 (s1,s2) ) (VHI 0 [v2]     ))
QF(G,2)   = ((S 1 (s1,s2) ) (VHI 1 (v1,v2)  )),
```

for which the *VHI*-qdirs are consistent. Therefore we conclude that the combined second elements in the summed qfunctions match the second element in QF(G).

The next triplet of qfunction elements is:

```
QF(C4,2)  = ((S 1 (s1,inf)) (VHI  1 (v1,inf) ))
QF'(C5,3) = ((S 1 (s2,inf)) (VHI -1 (minf,v2)))
QF(G,2)   = ((S 1 (s2,inf)) (VHI -1 (minf,v2))),
```

for which the *VHI*-qdirs are inconsistent. At this point only operator STEEPEN can be applied on the third element of QF$'(\mathcal{C}_5)$ to change its qdir to be -2. Thus, we conclude that the combined third elements in the summed qfunctions match the third element in QF(G).

Overall, we conclude that QF$(\mathcal{C}_4)$+QF$'(\mathcal{C}_5)$=QF(G), where QF$'(\mathcal{C}_5)$ is the modified qfunction of $\mathcal{C}_5$. This tells us that the payoff-profile in figure 6 is provided by a compounded instrument that is constructed by the purchase of one call option with exercise price s_1 and the sale of more than one call options with exercise price s_2, where $s_1 < s_2$.

7 Limitations and Directions for Future Work

In this paper we have presented a technique that can be used to synthesize the configuration of systems that have the following characteristics: (1) they are configured only from two-terminal components connected in parallel; (2) components are uniform in the sense that they are all modeled using the same input and output parameters; and (3) their functionality and the functionality of each of their components can be specified in terms of qualitative behavior using a two-dimensional piecewise linear function.

Many physical systems, however, involve three complexities which our technique cannot handle. Firstly, they use components with an input node for time (i.e., multi-terminal components) to create the effect of change in behavior over time. Secondly, they are configured of components that are not uniform on the input and output parameters. To handle this complexity we must allow components to be also connected in series (i.e., the output to one component can be the input of another component). This requires matching the input with the output of every two serially connected components according to known relationships (e.g., $resistance = voltage/current$). Finally, they use components that are connected both in series and parallel to create feedback loops.

In order for our technique to be able to synthesize systems with one or more of the above complexities, it is necessary to define additional configuration constraints to represent structural connections other than parallel ones. Accordingly, it is also necessary to modify algorithm QSYN to account for algebraic operations on qfunctions other than addition.

We feel, however, that it is necessary to answer two questions before any extensions to our technique can be developed. One is "how much one needs to know about the structure of an objective system in order to specify its desired behavior in terms of two-dimensional piecewise linear functions?" The other question is "how feasible it is to specify the desired behavior of a system with multi-terminal components using a number of two-dimensional piecewise linear functions?"

References

[1] Chorafa, N., Dimitris, *Systems and Simulation*, Academic Press, New York, 1965.

[2] Benaroch, Michel, and Dhar, Vasant, "An Intelligent Assistant for Financial Hedging," *Proc. of the 7th IEEE Conference on AI Applications*, Miami, 1991.

[3] De Kleer, J., and Brown, J.S., "A Qualitative Physics Based on Confluences," *Artificial Intelligence*, 24:7–83, 1984.

[4] Kuipers, B., "Qualitative Simulation," *Artificial Intelligence*, 29:289–338, 1986.

[5] Mittal, Sanjay, and Araya, Agustin, "A Knowledge-Based Framework for Design," *Proc. of the 5th International Conference on Artificial Intelligence, AAAI-86*, Philadelphia, 1986.

[6] Saucedo, Roberto, and Schiring, E., Earl, *Introduction to Continuous and Digital Control Systems*, The Macmillan Company, 1968.