



Title	機能構造に基づく油圧回路の合成手法に関する研究 (第2報 : 記号ベース制約伝播による回路検証とシステム構築)
Author(s)	藤田, 喜久雄; 赤木, 新介; 佐々木, 誠
Citation	日本機械学会論文集 C編. 1996, 62(593), p. 383-390
Version Type	VoR
URL	<a href="https://hdl.handle.net/11094/2864">https://hdl.handle.net/11094/2864</a>
rights	
Note	

*The University of Osaka Institutional Knowledge Archive : OUKA*

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

# 機能構造に基づく油圧回路の合成手法に関する研究

## (第2報：記号ベース制約伝播による回路検証とシステム構築)\*

藤田 喜久雄<sup>†</sup> 赤木 新介<sup>†</sup>  
佐々木 誠<sup>‡</sup> 上平 明弘<sup>§</sup>

Hydraulic Circuit Synthesis based on Functional Structure  
(2nd Report: Symbol-Based Logic Simulator and Computer Implementation)

Kikuo FUJITA, Shinsuke AKAGI, Makoto SASAKI and Akihiro UEHIRA

An adaptive synthesis method for hydraulic circuits is proposed using a case-based reasoning technique based on their functional structure. In the first report, we proposed the adaptive synthesis method itself and the strategies used in it. In this report, we present a verification method of the circuits generated by the adaptive synthesis method, computer implementation using an object-oriented programming language, and graphical interface drawing of schematic structures of circuits. Since design verification in the conceptual steps of the design process cannot utilize specified attributes of system elements of designs, we verify the structures of circuits and kinds of components through a symbol-based constraint propagation mechanism. These synthesis and simulation methods are applied to some design problems in order to show their effectiveness and validity.

**Key Words :** Design Engineering, Hydraulic Circuits, Conceptual Design, Logical Simulation, Constraint Propagation, Object-Orientation, Artificial Intelligence

### 1 緒 言

油圧回路は、シリンダ、ポンプ、方向切換弁・流量制御弁・圧力制御弁などの様々な制御弁から構成されるシステムである。そのような回路の機能は構成要素とそれらの接続関係に関する構造に従って実現されるものであり、適切な回路を構成するためには、作動油の流れについての流量や圧力を調整するように、適切な機器を選択するとともに、それらを回路中の適切な箇所に組み込む必要がある。このような設計問題は、典型的なシステム構成についての概念設計における問題であるため、一般に「生成検査型」の処理を必要とし、効率的な合成処理を行なうことに加えて、合成された結果そのものの妥当性を何らかの方法で検証しつつ、設計を進めていく必要がある。

前報<sup>(1)</sup>では、以上のような回路合成の問題に対し、事例ベース推論<sup>(2)(3)</sup>の枠組を基本とした回路事例の適合化による合成手法を提案し、設計仕様をインデックス化したラベルを用いて表現する一方、そのようなラベルの差異に基づいて検索した回路事例を「分割」

・「直列組合せ」・「並列組合せ」の3つの戦略に従って適合化させていくことにより、回路を合成できることを示した。上述のように、概念設計ではこのような方法に対して、検証の方法を組み合わせて用いることが不可欠であり、本報は、そのような手段として、記号処理をベースとした代数的な制約伝播処理による油圧回路の論理動作シミュレーション手法を提案する。加えて、前報<sup>(1)</sup>の合成手法とこのシミュレーション手法を統合化することにより構築した設計システムの構成方法などについても報告する。

### 2 機能構造に基づく油圧回路の合成手法

**2.1 事例からの適合化による回路合成** 緒言でも述べたように、油圧回路は、ポンプから供給される油圧を各種の制御弁により調整し、シリンダにおいて所要の動作や仕事を得ようとするものであり、個々の機器が何らかの機能的な役割を果たすとともに、それらの接続関係がシステムとしての機能を構成している。したがって、そのような回路を合成するためには、与えられた設計仕様に対して、適切な種類の機器を選択するとともに、システムとしての構造を設計する必要がある。前報<sup>(1)</sup>では、このような処理内容は、個別的な知識のみによって行なうことは困難であり、

\*原稿受付 1995年5月31日。

†正員、大阪大学工学部(〒565 吹田市山田丘2-1)。

‡学生員、大阪大学大学院工学研究科(現在、コマツ)。

§正員、大阪大学大学院工学研究科(現在、住友金属工業)。

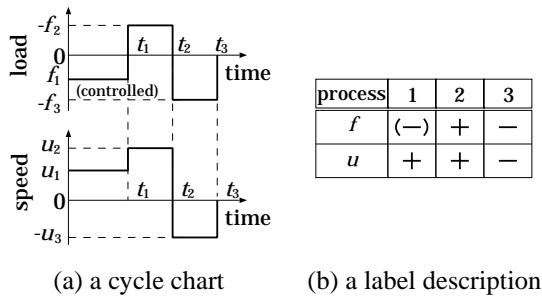


Fig. 1 A cycle chart and a label description

全体の回路構造が個別の部分構造の重ね合わせによって成り立っているとの立場から、事例ベース推論<sup>(2)(3)</sup>の枠組を導入することによって、回路事例に含まれる部分構造を適合的に組み合わせたり、抜き出したりすることによる設計手法を提案した。

一般に、油圧回路の設計仕様は、図1の(a)に示すような負荷と速度についてのサイクル線図として表現されるが、概念設計過程である回路合成においては、その定性的な特徴に従って、設計が行なわれる。前報<sup>(1)</sup>で示した回路合成手法においては、そのような内容を効率的に表現する方法として、シリンダ負荷( $f$ )と速度( $u$ )の個々の状態を、状態量の符号‘+’・‘-’と、それらが設定値に対して制御されているかされていないかを表す括弧‘( )’によって表現するものとして、図1の(b)に示すようなラベルによる表現を導入した。このような表現を導入することにより、与えられた設計仕様と各回路事例の実現している機能との関係をラベルの対応関係に基づいて比較することができるようになる。

図2は、そのような対応関係に基づいた合成手法の概要を示したものであり、ラベルの対応関係により設計目標と回路事例相互の機能的な「過剰(surplus)」あるいは「不足(deficit)」の関係を判断し、前者の場合には、過剰な機能を持った回路事例の不要な部分を除外していくことによる「分割(decomposition)」合成を、後者の場合には、機能的に不足するものの、相互の機能の和が設計目標よりも過剰になるような回路事例の対を組み合わせることによる「組合せ(combination)」合成を行ない、さらに、そのような合成処理によって生じる冗長な部分を除外するなどの「修正(refinement)」操作を行なうことによって、回路を合成することができるようになる。

しかし、緒言でも述べたように、以上のようにして合成された回路が、場合によっては、設計目標である機能を実現しない可能性もあるため、何らかの動作検証の方法を統合化し、それによる検証処理を行なった

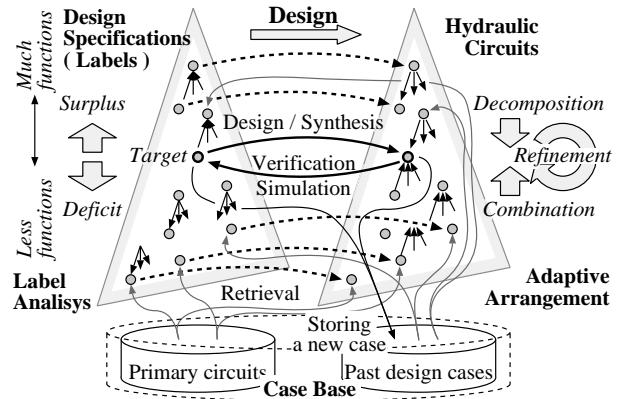


Fig. 2 Outline of adaptive circuit synthesis from design cases

上で、最終的な設計解とする必要がある。

**2.2 概念設計における設計検証** 上述のように、油圧回路の合成問題は、システムを構成する要素の種類や方式を定め、それらの織り成す構造を決定する典型的な概念設計の問題であり、そのような過程においては、構成機器の具体的な要目などは決定されていない。そのため、上述のような目的のもと、動作シミュレーションを行なうとする場合、微分方程式系による数量的なシミュレーションを行なうことはできず、何らかの論理的あるいは定性的な関係に基づいたシミュレーションを行なう必要がある。

そこで、本研究では、油圧回路における状態量を数値ではなく記号を用いて表現し、それらの間で成立する様々な関係式を制約指向型の処理<sup>(4)(5)</sup>により操作することによって、設計されたシステムの構成が妥当であるかどうかを検証する方法を提案し、そのような方法を前報<sup>(1)</sup>の合成手法と統合化することにより、油圧回路の概念設計のための総合的な方法を構成する。

なお、このようなシミュレーションの手法としては、現象の定性的な側面に着目した「定性推論<sup>(6)</sup>」の適用も考えられるが、油圧回路の合成問題においては、動的な過渡現象は付随的であり、代数的な因果関係が本質的であることから、境界標を用いた区間値を導入して、微分方程式系によって表現される動的現象の定性的な解析を行なうのではなく、上記のように、記号による表現を用いた制約指向型の代数処理により、機能シミュレーションを行なう。

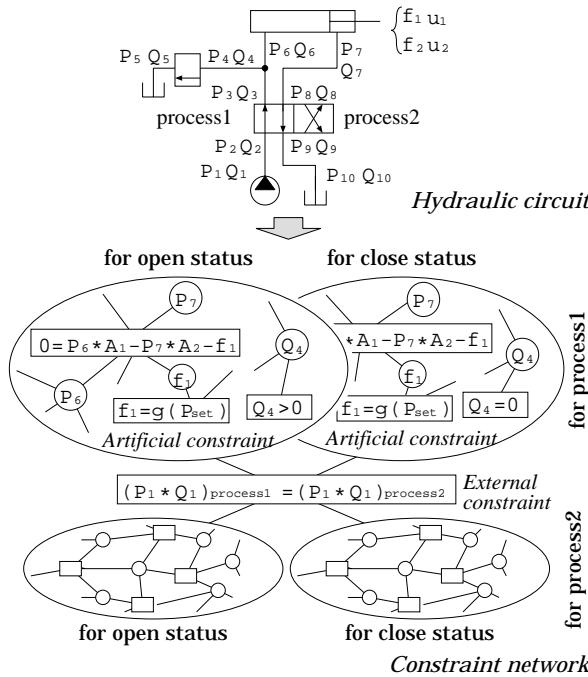


Fig. 3 Constraint network on circuit behavior

### 3 記号ベース制約伝播による回路検証

前節で述べたように、設計仕様に対して生成された回路は論理的かつ定性的な手法を用いて検証する必要があり、本節では、そのような手法として、記号処理をベースとした代数的な制約伝播処理による動作シミュレーション手法について示す。

**3.1 回路の振舞と代数制約** 油圧回路の振舞は、その構造である機器の種類やそれらの間の接続関係によって定まるものであり、振舞を表す状態量の間に成立する代数的な関係式を解くことによって求めることができる。本研究では、概念設計段階においては、個々の機器の属性情報が定量的には定まっておらず、また、制御機器の設定値なども具体的には定まっていないなどの理由により定量的な取り扱いが不可能であることから、このような内容を論理的な因果関係を記号処理を用いて解析することにより、回路の妥当性を検証するようにする。具体的には、状態量の間の代数的な関係式を制約としてとらえて、制約指向の考え方<sup>(4)(5)</sup>を導入する一方、制約の内容を数値ではなく、記号のまま、処理するようにして、回路検証を行なうようにする。なお、以下では、回路を構成する各機器の各ポートにおける圧力や流量などの状態量とそれらの間の関係式とが図3のようなネットワーク状のグラフを構成することから、制約の集合を“制約ネットワーク”と呼ぶことにする。

制約指向の中でも、制約伝播の方法は、電気回路の動作を対話的にシミュレートするシステムであるCONSTRAINTS<sup>(4)</sup>を起源のひとつとするが、その特徴とするところは、対象システムにおける様々な関係を具体的な解法とは独立した形で宣言的に記述するようにし、そのような関係式をどのようにして解くかについては、記述された関係式や解くべき内容に従って動的に定めようとする点にあり、それによって、対象システムの振舞を柔軟に取り扱うことができるようになる。

以下に、このような制約ネットワークにおける構成要素の振舞モデルとその代数表現や、それらを解く方法について示す。

**3.2 油圧回路における状態量** 回路合成の過程では、上述の理由により、各ポートにおける圧力や流量などの状態量を数値的に表現することができないため、それらを記号として表し、それらの記号の間で成立すべき代数的な関係式を解くようにする。このような記号は、大きく、シリンダやパイプの径、負荷や速度の設定値などのように定数として表現されるべきものと、他の解くべき変数に対応するものに類別され、ここでは、前者を「定数記号 (constant symbol)」、後者を「変数記号 (variable symbol)」と呼んで区別することにする。

**3.3 複数の動作プロセス** 一般に、油圧回路は複数の動作プロセスを有しており、それらのプロセスは、方向切換弁を切り換えることにより相互に遷移するようになっている。個々のプロセスにおける回路の振舞はそれ各自独立に生じるものであり、状態量に対する制約ネットワークもそれぞれの動作プロセスに対してそれぞれ定義される。一方、機器そのものはすべての動作プロセスの間で共通のものであり、なかでも、ポンプの出力が共通であることから、独立した個別のサブネットワークの間に「外部制約 (external constraint)」を定義してそのような関係を表現する。

前出の図3は、このような制約ネットワークの状況を示したものであり、全体のネットワークは、個々の動作プロセスや各制御弁の状態についての部分ネットワークから構成され、それらが外部制約により相互に連結されている。回路の動作を検証する処理は、このようなネットワークに対して、制約ネットワークを代数的に解くことができれば、そのような動作プロセスが成立すると言え、外部制約によって相互に接続されたサブネットワークのすべてを解くことができれば、対応する回路が設計仕様に対して正常に動作するものと判断することができる。

**3.4 油圧回路における制約式** 回路の振舞を表す制約式は、回路の構造に従ってそれら記号の間で定義することができ、流量についての Kirchhoff の法則や設定値に関する条件式などから構成される。例えば、配管系の分岐において、 $n$  本のパイプが分岐している場合、 $i$  番目のパイプに接続しているポートの流量を  $Q_i$  と表したとき、次の制約式が成立しなければならない。

$$\sum_{i=1}^n Q_i = 0 \quad (1)$$

また、同様に、各ポートの圧力  $P_i$  についても、分岐における圧力損失を無視した場合、次の制約式が成立しなければならない。

$$P_1 = P_2 = \dots = P_n \quad (2)$$

一方、個別の機器についても、それぞれの機器機能に対応した関係式が諸量の間で成立する必要がある。

**3.5 制御弁の動作** 回路の振舞は、各種の機器の中でも、圧力や流量についての制御弁の状態に依存するものであり、例えば、圧力制御弁に関しては、対応するポートの圧力が設定値である  $P_{set}$  よりも小さければ、弁は閉じているものの、大きければ、弁は開いていることになる。このような制御弁の各動作プロセスにおける開閉状態は、設定値  $P_{set}$  に依存して定まるものである。

設定値  $P_{set}$  自体は、圧力(負荷)を所要の値に設定しようとする動作プロセスにおける回路の振舞から定められるものであり、シリンダにおいて生じる圧力が制御弁の設定値  $P_{set}$  に依存して定まる関係が成り立っているならば、そのような因果関係をもとに設定値  $P_{set}$  を定めることができる。このような関係は、シリンダ圧力が制御弁の設定値  $P_{set}$  の関数になっていることに応じ、図 3 に示すような制約ネットワークにおいては、これを「人為制約 (artificial constraint)」として表現するようにする。これによって、制御弁の設定値も、他の変数記号と同様の方法で定めることができるようになる。

一方、設定プロセス以外の動作プロセスにおける制御弁の状態については、図 3 にも示したように、開閉状態の両方の可能性を制約ネットワークの全体に関連させて検討した上で、それを定める必要がある。開閉状態の両方に関わる可能性は、以下の代数的な関係式により表現することができる。

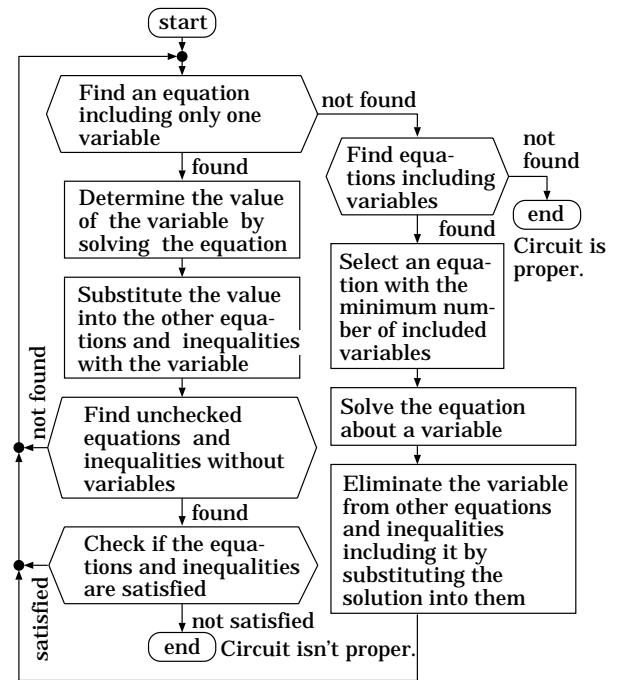


Fig. 4 Algorithm for solving constraints

制御弁が開いている場合：

$$\left. \begin{array}{l} P_{in} > P_{out}, \quad P_{in} = P_{set} \\ Q_{in} > 0, \quad Q_{out} < 0 \\ Q_{in} + Q_{out} = 0 \end{array} \right\} \quad (3)$$

制御弁が閉じている場合：

$$\left. \begin{array}{l} P_{in} < P_{set} \\ Q_{in} = Q_{out} = 0 \end{array} \right\} \quad (4)$$

ここで、 $P$  と  $Q$  はそれぞれ圧力と流量を表し、下付の添字  $in$  と  $out$  は、制御弁の入口側、出口側のポートを指している。また、流量は制御弁に流入してくる方向を正としている。

**3.6 記号ベース制約伝播** 以上の各種の制約により構成される制約ネットワークに対して、記号処理を通じた制約伝播処理<sup>(4)</sup>によりその代数関係を解くようとする。図 4 は、このアルゴリズムを示したものである。アルゴリズムにおいては、まず、唯一の変数記号を含んだ制約式を見つけ出し、その式を変数記号について解くことによって、その変数記号を定数記号のみによって表現する。そのもとで、変数記号を定数記号に置き換えるとともに、解いた式を他の制約式に代入する。以上の処理を、すべての変数記号が定数記号からなる式によって表現されるようになるまで、繰り返した場合に、制約ネットワークを解くことができれば、回路は正常に動作するものと判定し、そうではな

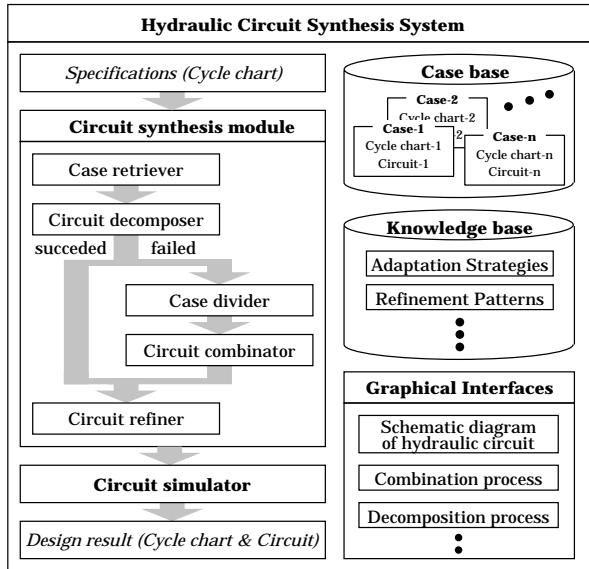


Fig. 5 Architecture of the design system

く、途中で制約違反に陥りネットワーク全体を解くことができなければ、正常に動作しないものと判定するものとする。

以上のアルゴリズムにより制約ネットワークを原理的には解くことができるが、そのための処理は、動作プロセスの数や制御弁の数に対応して多くの制約ネットワークを解く必要を生じる。このような処理を少なくするために、まず、個別の動作プロセスに関わるサブネットワークを解き、可能であれば制御弁の開閉状態を定めた上で、正常に動作すると判定されたサブネットワークを組み合わせて、全体としてのネットワークを解くようとする。これによって解くべきネットワークの組合せの数を少なくすることができる。

#### 4 油圧回路合成システムの構築

前報<sup>(4)</sup>で示した回路事例の適合化による合成手法と、本報で示した記号ベースの制約処理による論理シミュレータによる回路検証手法を統合化した油圧回路の設計システムの構成方法について、以下に示す。

**4.1 システム構成** 図5は、本設計システムの構成を示したものであり、大きく、回路合成を行なう部分とそれに関連した知識や事例ベースの部分、動作検証を行なう部分、さらに各種のインターフェース機能の部分から構成されている。システムにおける設計処理の手順は図中の左部に示す通りであり、そのなかから、右部の知識ベースやインターフェース機能が参照される。具体的なシステムの構築にあたっては、以上のシステムにおける処理が記号処理に基づいていること、様々な複雑なデータを表現する必要がある。

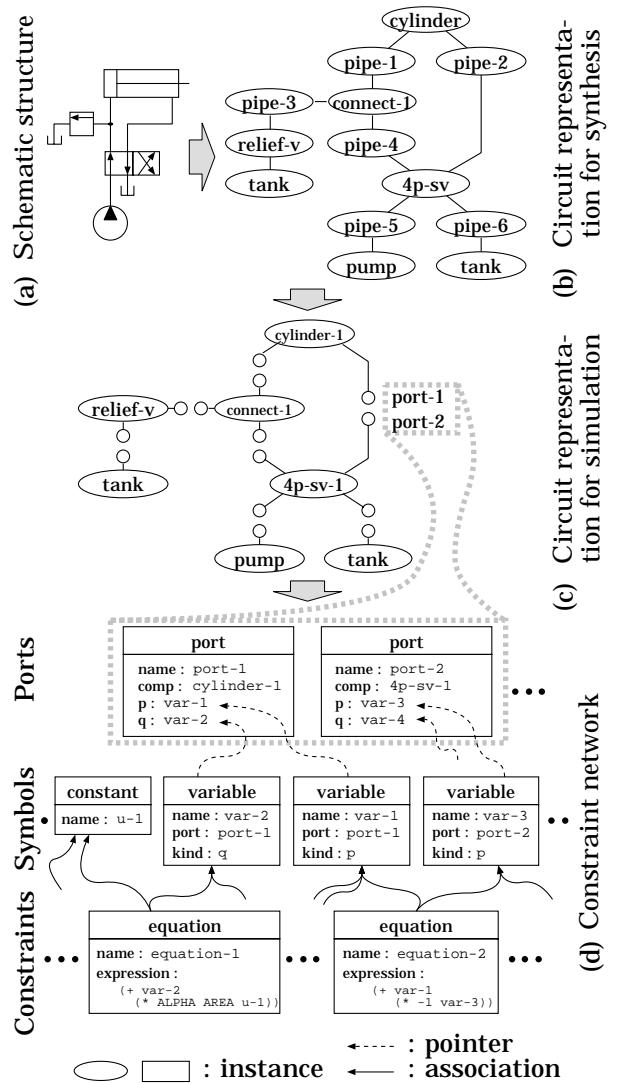


Fig. 6 Object-oriented representation

ことなどの理由から、COMMON LISP<sup>(7)</sup>上のオブジェクト指向プログラミング環境であるCLOSを用いる。また、グラフィカルユーザインターフェース機能の実装にあたっては、X ウィンドウシステム上でCLIM (Common Lisp Interface Manager)を用いる。

**4.2 回路表現と合成処理・検証処理** 油圧回路は機器要素とそれらの間の接続関係により表現され、また、動作検証を行なうための制約ネットワークも変数や制約とそれらの間の関係により表現されることから、そのような内容を効率的に表現するために、オブジェクト指向による表現<sup>(8)(9)</sup>を導入する。

図6は、回路構造と制約ネットワークの表現方法を例にして、そのような表現方法の概略を示したものである。図は(a)に示すような構造の回路を例にしたものであり、合成処理を行なう際には、(b)に示すよ

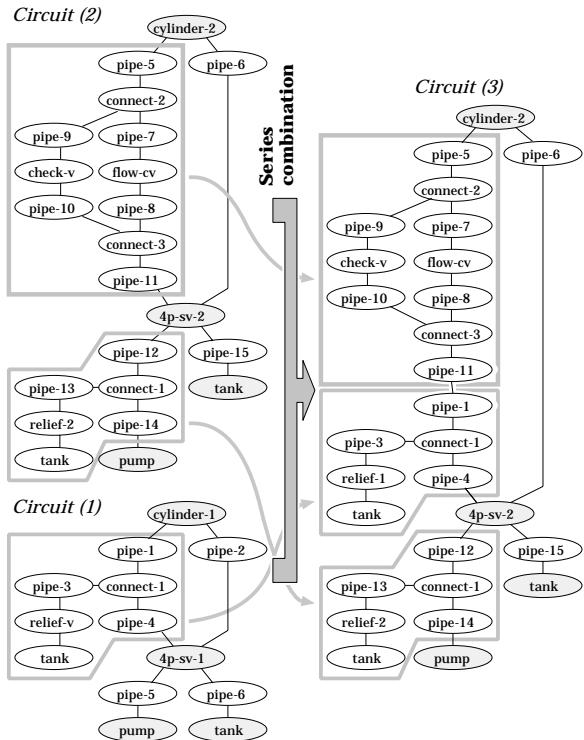


Fig. 7 Series combination

うに、回路中の機器・パイプ・パイプの分岐点のそれをオブジェクトとして表現し、それらの間の接続関係をオブジェクト間の関係として表現する。このような表現のもと、例えば、回路事例を直列に組み合わせて新しい回路を合成する処理は、図7に示すように、組み合わせるモジュールの部分についての関係を切り離した後、両者の部分を組合せの形式にしたがって関係を再定義することにより、組合せ合成後の回路を表現するようにして、行なうことができる。一方、動作検証を行なうに当たっては、個々の機器における各ポートの状態量を取り扱う必要があることから、図6の(b)の表現に対して、ポートに対応したオブジェクトを導入することにより、(c)の表現を新たに生成する。そのもとで、(d)のように、各ポートの状態量である圧力や流量を表す記号のオブジェクトを定義し、さらに、それらの間で成立する制約式をそのような記号オブジェクトの間で定義し、制約ネットワークを表現して、検証の処理を行なう。

なお、事例ベースにおける回路事例、合成処理に関わる戦略や回路修正におけるルールなども、以上の表現方法をもとにして記述し、それらのもとで、回路の合成や検証に関わる様々な処理を行なうようとする。

#### 4.3 回路図の表示アルゴリズム 油圧回路の設計に限らず、システムの構成を慣習的な記法を用いて

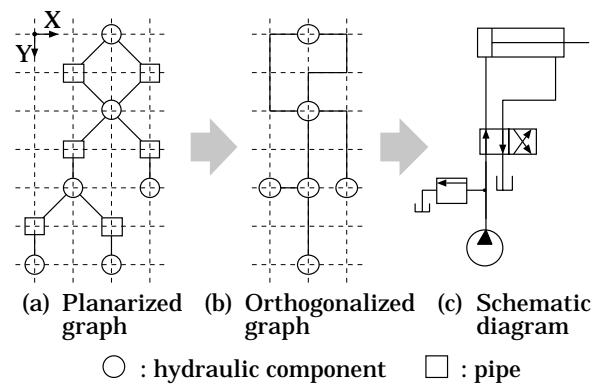


Fig. 8 Drawing schematic diagrams

図的に表現することは多く、それによって設計処理の内容を容易に理解できるようになるため、コンピュータによる設計を行なう場合においても、そのような表示を行なえることが重要である。

図8は、そのような図的表現の自動作図アルゴリズムの概略を示したものである。油圧回路図は、特定のノードが面積を持っており、エッジがグリッド上の折れ線で表される特殊なグラフとしてとらえられることから、フローチャートを描くためにグラフを平面化し直行化するアルゴリズム<sup>(10)~(14)</sup>を基本として様々な拡張を行うことにより、回路図を自動的に作図することができるようになる。

その手順は以下に示す通りである。

- (1) 回路のグラフ化 … 合成された回路中の機器や配管の分岐点、パイプをノード、それらの間の接続関係をエッジとするグラフを定義する。このとき、圧力制御のためのリリーフ弁とタンクの対などのように機能的に関連の強い機器群については、単一のノードとみなしておく。これにより、最終的な回路図においてこれらの機器を隣接して配置することができるようになる。
- (2) グリッド上での平面化処理 … 格子状の直交座標系を考え、まず、シリンダに相当するノードをY軸方向の最上位の配置し、それに接続されている機器から順に、下方に向かって配置していくことにより、すべてのノードのY座標値を定める。このとき、4ポート切換弁などの回路図上で配管の出していく方向についての相対関係に拘束があるノードについては、それが接続されているノードとの上下関係に対応する拘束条件を付与する。その後、横方向のノードの並びをエッジが交差しないようにソートすることにより、各ノードのX座標値を定める(図8(a))。

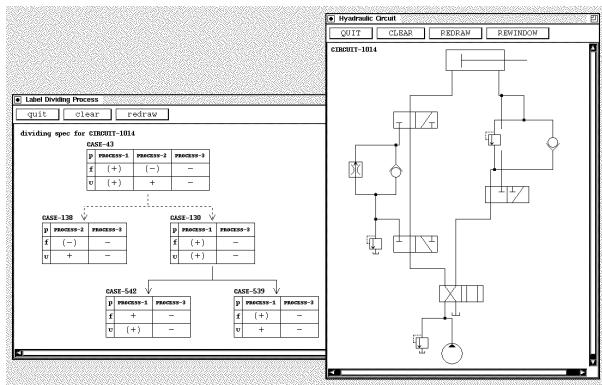


Fig. 9 An example of screen image

- (3) 平面化に対する補正処理 … 上記のようにして得られた位置関係に対して、各ノードに機器のシンボルを対応させた場合、その大きさがエッジとの干渉を生じる場合は、そのようなノードをグリッド上から若干ずらすことにより、そのような可能性を排除しておく。
- (4) 特殊ノードに対する処理 … シリンダや方向切換弁などの大きな面積のシンボルを割り当てる必要のあるノードについては、それらのポートに相当する個別の付加ノードを用意し、元々のノードの周辺に配置する。さらに、各ノードが付加ノードのいずれに接続されるべきであるかを、相手側のノードのある位置をもとにして決定する。
- (5) 直行化処理 … 各エッジを直交座標軸に沿った折れ線として直交化することにより、パイプに相当する折れ線を描く。具体的には、パイプや分岐点に対応するノードに対して、そのY座標に基づいた水平な線分を想定し、それに接続されている他のノードからその線分に対して垂直な足を下ろすことにより、直交化を行なった折れ線を描くようにする。なお、このとき、シリンダとポンプ以外の枝ノードについては、中間部分のパイプの部分を省略しつつその位置を変更することにより、回路図における配管の折れ曲がりを少なくするようになる(図8(b))。
- (6) 機器イメージの張り付け処理 … 直交化されたグラフのうち、機器に対応するノードに対して、その機器種類に対応するシンボルを張り付けることにより、回路図を完成する(図8(c))。

なお、図9は、このようなインターフェースの機能によって作図された、回路合成におけるラベル表現に対する処理内容と油圧回路の回路図の一例である。

## 5 適用事例

最後に、以上の回路検証手法と設計システムを具体的な設計問題に適用した事例を示す。

本回路合成手法における戦略は、大きく、回路事例の組合せと分割に大別できるが、一般に、組合せ合成を行なった場合には、それによって得られた回路は所要の機能を満たす場合が比較的多いものの<sup>1</sup>、分割合成を行なった場合は、設計仕様に対して必要な部分まで削除してしまう場合があり得るため、検証の必要性は大きい。

図10は、事例ベースが8個の基本回路と40個の回路事例を含む場合に、図1のサイクル線図に示すような動作を行なう回路合成を行なった場合の合成結果と検証結果を示している。この場合、設計仕様のラベル表現に対して過剰なラベル表現を持つCase (1)からCase (4)の4つの回路事例が存在し、それぞれにおいて、機能的に不要である部分を除去することにより、図中各部の右側のような適合化による設計解が合成される。これらのうち、Case (1)とCase (2)は、設計仕様を満足することが検証されるが、Case (3)とCase (4)については、それぞれ、制約処理により、第2プロセスにおいて動作条件に対応する回路の状態が存在し得ないことが導出され、設計仕様を満足しないものと判定される。

## 6 結 言

本研究では、前報<sup>(1)</sup>に示した油圧回路の機能的な構造に着目した回路事例からの適合化による回路合成手法に対して、その設計処理過程で必要となる回路の動作検証方法として、記号処理を用いた制約伝播処理による代数的な論理シミュレーション手法を提案し、さらに、両者をあわせた設計システムのCLOSによる構成方法、このような油圧回路の構造を設計するシステムにおいて不可欠となる回路図の自動作図法について報告した。

システムの構成に関する概念設計は、構成要素の要目が確定していない段階で、機器の種類やそれらの接続関係を検証する必要があり、本報で示した検証手法は油圧回路に特化されたものであるものの、その記号処理により代数的な関係式を取り扱って機能検証を行なう方法は、概念設計における検証の一方法としても期待を持つことができると思われる。また、設計システムの構成方法や回路の作図インターフェースなども、様々なシステムの概念設計を取り扱う上で、共通する部分も多いと考えられる。

<sup>1</sup> 組合せ合成の処理事例は、前報<sup>(1)</sup>を参照されたい。

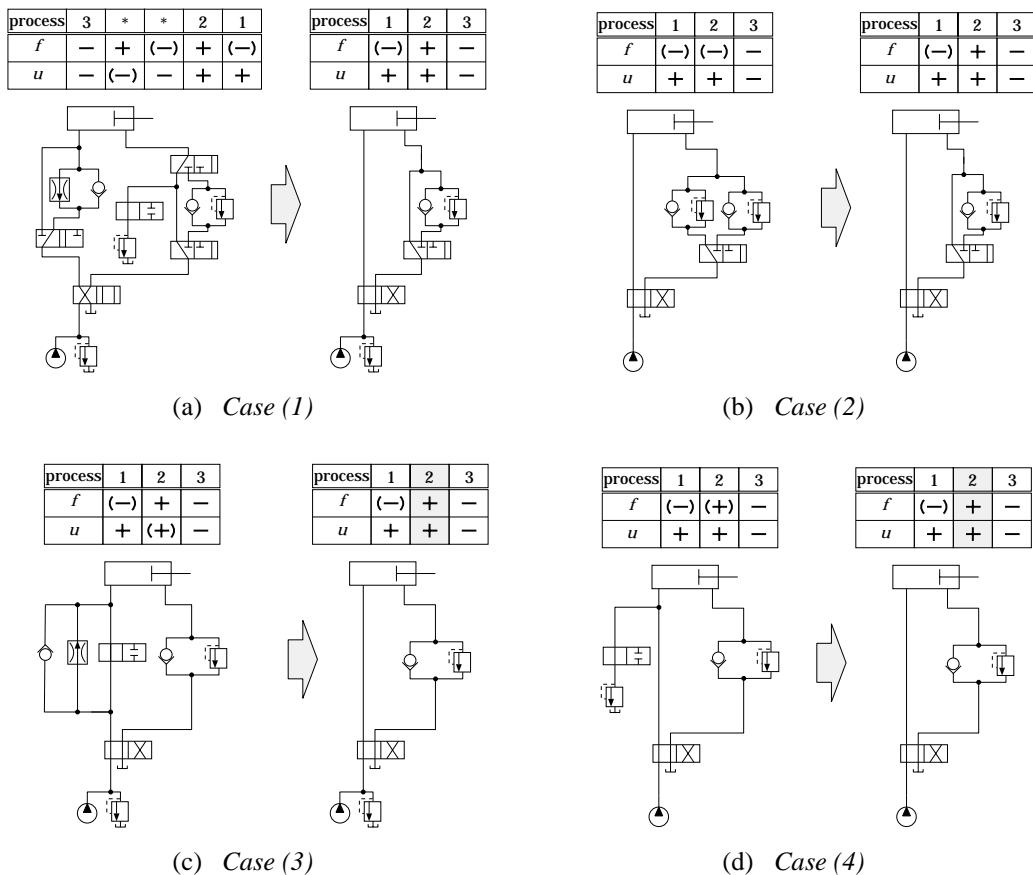


Fig. 10 Example — Synthesis by decomposition

文 献

- (1) 藤田・赤木・佐々木, 機能構造に基づく油圧回路の合成手法に関する研究(第1報:回路事例の適合化による回路合成), 機論, **62**-593, C(1996), 375-382.
  - (2) Riesbeck, C. K. and Schank, R. C., *Inside Case-based Reasoning*, (1989), Lawrence Erlbaum Associates.
  - (3) Slade, S., Case-Based Reasoning: A Research Paradigm, *AI Magazine*, **12**-1, (1991), 42-55.
  - (4) Sussman, G. J. and Steele Jr., G. L., CONSTRAINTS — A language for expressing almost hierarchical descriptions, *Artificial Intelligence*, **14**, (1988), 1-39.
  - (5) Serrano, D. and Gossard, D., Constraint management in MCAE, *Artificial Intelligence in Engineering: Design*, (Gero, J. S., editor), (1988), Elsevier, 217-240.
  - (6) 西田, 定性推論の諸相, (1993), 朝倉書店 .
  - (7) Steer Jr., G. L., *COMMON LISP — The Language, Second Edition*, (1990), Digital Press.
  - (8) 藤田・赤木・辻本, オブジェクト指向モデリングを用いたリンク機構の最適形状設計支援システムの構築, 機論, **58**-545, C(1992), 310-315.
  - (9) 藤田・赤木・天木, 機能モジュール統合化による歯車伝達装置の設計支援システムに関する研究, 機論, **60**-579, C(1994), 3602-3609.
  - (10) 鎌田, グラフ描画アルゴリズム, *bit*, **23**-3, (1991), 共立出版, 23-25.

- (11) Tamassia, R., Battista, G. D. and Batini, C., Automatic Graph Drawing and Readability of Diagrams, *IEEE Transactions of Systems, Man, and Cybernetics*, **18**-1, (1988), 61-79.
  - (12) Kameda, T. and Kawai, S., An algorithm for drawing general undirected graphs, *Information Processing Letters*, **31**-1, (1989), North-Holland, 7-15.
  - (13) Tamassia, R. and Tollis, I. G., A Unified Approach to Visibility Representations of Planar Graphs, *Discrete & Computational Geometry*, **1**-4, (1986), 321-341.
  - (14) Tamassia, R. and Tollis, I. G., Efficient embedding of planer graphs in linear time, *Proceedings of IEEE International Symposium on Circuit and Systems*, (1987), 495-498.