制約に基づくモデリングの提案と考察

京都情報大学院大学　教授
山縣　敬一

コンピュータを利用して計画型の問題や設計型の問題解決を図ろうとするとき，コンピュータ内部にいろいろな形態のモデルが作られる。機械製品の場合のディジタルモックアップはその典型的な例である。モデル構築における重要なポイントの一つは，単に幾何学的なモデルを表現するに止まらず，多様な設計目的や設計変更に応えられることである。本論文では，とくに製品モデルのように多くの部品からなる複雑なモデルに対して，部品間の制約条件や依存関係をモデルに組み込むことを検討している。具体的には，水圧シリンダー，油圧クラッチ，産業用ロボットのリンク機構の性質の異なる3種類の場合について，ケーススタディを行い，改良解の導出に効果的なモデルのあり方を検討している。

1.　はじめに

ものづくりの現場では，ディジタルモックアップが作られる。CADシステムによる部品設計が行われた後，それらを組み合わせた製品に近いモデルをコンピュータの中に作り出す。その目的とするところは，製品出来上がりの意匠評価，製造組立工程でのシミュレーション，その製品が実際に使われる状況になったときの各種機能の事前評価に利用される。展示会などでは，数千もの部品を乗せた仮想的製品モデルを見ることができる。このとき，これらの多くの部品相互の間には多くの依存関係や拘束関係が存在しており，それは単にCGによるリアルイメージだけからは見えてこない。しかし，ディジタルモックアップは一定の手順で完全なものが作り出せるわけではなく，製造過程から市場に出た後の状況での多面的なシミュレーションを経て，順次不具合を取り除き，適切な材料や部品を選び，CAEによる解析結果を使ってより好ましい物に仕上げて行く。このとき，部品相互の間の寸法・機能・強度などの依存関係がどこまでモデルに記述されているかが重要になる。本稿では，この様な観点からいくつかのケーススタディについて，考察してみる。

2.　依存関係によるモデルの必要性

少し，極端な場合を考えてみる。最近，市場に出回った商品に不具合が見つかるケースがしばしば報告されている。そこで，その原因があきらかになったとして，ある部品を交換するものとしよう。そこで，過去に作ったディジタルモックアップを取り出して修正を加えようとするとき，その部品を単純に差し替えるだけで済むのは極めて稀であろう。その部品の周辺で，寸法・機能・強度の影響が他の部品にも及ぶはずである。しかし，その依存関係がすぐに分かるようになっているだろうかと云う問題である。

実はこの問題は，CADシステムでモデルを作り上げていく過程ですでに存在している。例えば，3次元のプリミティブを配置したり集合演算を行って複雑な形状を作っていく過程で，すでに操作したことの誤りが見つかったとする。素朴な発想からすれば，これまで進めてきた操作をそのまま逆順に元に戻していって間違いの源まで遡り，やりなおすことになる。しかし，このやり方が適切でないことはすぐに分かる。例えば，ワードで文章を作成しているとき，ずっと前の方で間違った文章を入れたことが分かったとしても，その文章を入れたところまで操作をもどしてやり直すようなことはしない。ひとまず間違った文章がすでにあるものとして，すでに入力した文章を改訂するという手順をとる。実は，CADシステムでもこの方法をとりたいのだが，モデルを作るときにすでに定義された部品なり幾何形状を利用して次の形状を定義していくため，後から勝手に一部の形状を修正すると全体としての整合が崩れてしまう可能性がある。このため，CADシステムにおける操作は単純に形状が表現できればよいのではなくて，後からの修正に対しても都合のよいように，形を作り上げていく過程の中で，依存関係のある部分とない部分の切り分けができるように工夫が盛り込まれている。この特徴がある意味でシステムベンダーによる差別化の対象になっているのである。

そこで，このような部品や幾何形状相互の間の依存関係あるいは拘束関係を，最初から表面に出して扱うようなモデリングのプロセスは考えられないだろうか，というのがここでの議論の発端である。

3.　制約による問題解決の簡単な例

設計問題を制約条件に従って解くという簡単な例として，水圧シリンダーの例を図1に示す。もともと設計問題を解くときには，設計パラメータ間の固定された制約（拘束）条件があって，それを充足させながら設計意図を反映させることになる。しばしばその設計意図は多目的であり，いろいろな見方が必

要になる[1]。図1の場合，シリンダーの内径，厚さ，材料応力，内部圧が重要な設計パラメータで，相互に制約条件の拘束を受けている。例えば，全体をコンパクトに作ろうとすれば内圧を高くとらざるを得ず，材料は高価になり材料費も高くなる。もし大きくてもよければ，ピストンの断面積を大きくすれば，内部圧が小さくて構わないので，材料は安価なものでよい。実際にはこのようなバランスを取って設計が進められる。

図1に各変数間の制約条件が書いてあるが，このような表現をそのままプログラミングとして考えようと云う考え方がある。例えば，文献[2]はこの考え方をとっている。ただし，図1のように制約条件を並べただけでは，具体的な解法については何も記述していないので，少なくとも標準的な解法についてはその手法を用意しておく必要がある。しかも，このような考え方では，なるべく一般的な解法をシステムとして用意することが期待される。図1のような構造についての見通しのよい例の場合には，制約条件の数式とそれを解釈するプログラムを用意しておくと，予めどれが独立変数かを決める必要がなく，設計の意図に応じてどれかの変数の値を変更したとき，その影響が他のどの変数に及ぶかを知ることができる。これが制約プログラミングと呼ばれる考え方であり，その効果的な使い方である。

実は，制約をキーワードにして問題解決を図ろうとする試みは非常にたくさんある。ここでその全貌は述べられないが，多少筆者の私見を交えて述べるとすれば，その代表的なものに制約論理プログラミングがある[3]。述語論理を使って与えられた条件を満たす解を見つける問題は制約充足システムとして古くから使われている。第5世代コンピュータプロジェクトの成果としての意味もあって，述語論理を使う問題解決の枠組みに，設計のための条件記述を盛り込んだシステムが1980年代後半から1990年代前半にかけて開発された。また，少し別の観点から，設計問題を解く過程で構成している要素間の依存情報をATMS(Assumption-Based Truth Maintenance System)で管理する研究が多く行われた。ここでは，代表的な文献一つを挙げておく[4]。また，制約条件を充足させる枠組みとして，論理推論と数理計画法を包含した考え方もある[5]。

4. 製品モデルへの適用

これまで述べてきたように，設計対象要素間の依存情報（ここではこれらを総称して制約条件として述べる）を明確に記述し，論理推論を使って制約条件を満たす解を見つけることは，成果をあげてきた。現在の状況で問題になるのは，冒頭の議論にもどって，多数の部品からなる製品モデルで同様の考え方が適用できるかどうかの問題である。一例であるが，図2は油圧クラッチで前進歯車と後進歯車を切り替える機構の概念図で図3はこれをもう少し詳しく書いたものである。

この機構は，もともと動力を伝達する減速歯車に，油圧によ

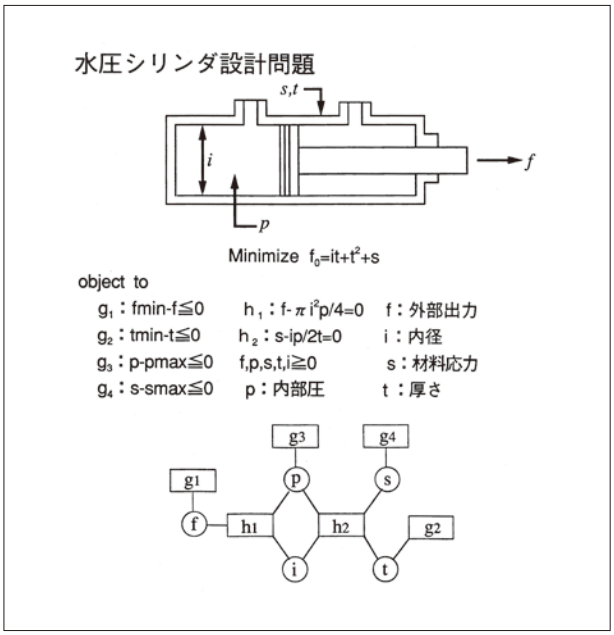

図1　水圧シリンダーにおける制約条件

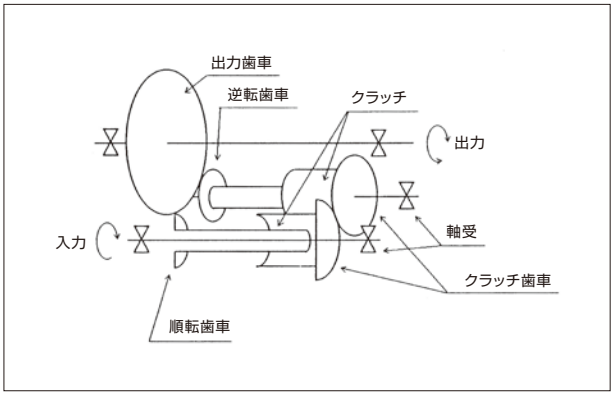

図2　前進後進を切り替える油圧クラッチ

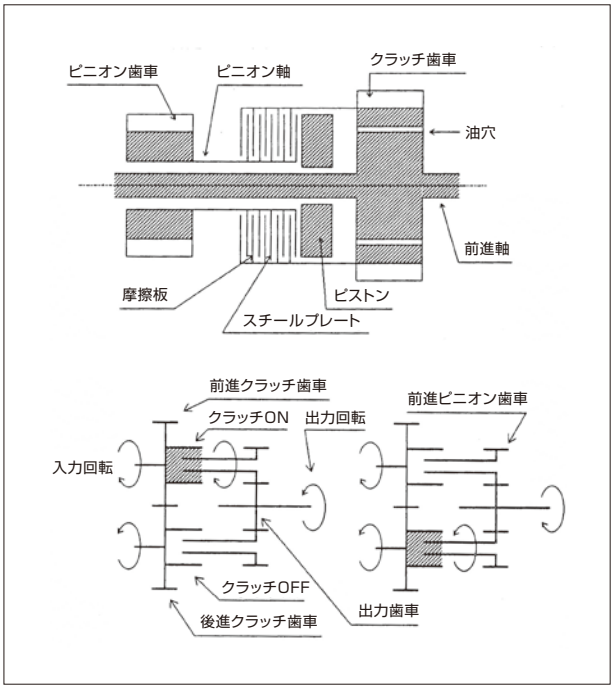

図3　油圧クラッチ詳細図

るクラッチ機構が組み込まれたものである。したがって，軸と歯車の設計が基礎にあって，さらにこれに油圧による摩擦板の切り替え機構が挿入されている。しかも，入力軸にはエンジンが結合され，出力軸にかかる負荷は，この機構をどこに組み込むかによって変化する。実際の設計者にこのような場合の設計パラメータを書き出してもらうと，すぐ数十から数百にも及ぶ。このような複雑なモデルになると，論理式を使ってこの機構の全体を表現することも困難になる。実際に実験的にやってみて，図1の例のように設計パラメータ間の関係記述を与えて，制約伝播の方法で，一部のパラメータの変更が他にどのように影響するかを調べることはできた[6]。しかし，実際に設計問題を解くときには，このような素朴な捉え方だけでなく，内部の構造に踏み込んで制約条件を記述する必要がある。つまりそうでないと，わずかのパラメータ変更が実現できるだけで，大幅な設計の改善が期待できないのである。これが，単に制約条件を記述して，後はシステム側の処理に期待しようとしても思うように結果が得られないことの重要なポイントになっている。

5. リンク機構と制約条件

もう一つ製品モデルの例をあげる。今回の例は制約条件（この例では拘束と呼ぶほうが適切であるが）がリンク機構のいろいろな直動関節や回転関節などのジョイントになっている例である。リンク機構の仕組みは，機械工学では古くてなおかつ新しいテーマである。ここでリンク機構とジョイント（ここでは，いろいろなリンクの結合の仕組みを総称してジョイントと呼ぶ）の全体を説明することは困難なので，後の産業用ロボットの例を見て頂くとして，もともと多様な直動関節や回転関節さらにそれらの組み合わさった複雑な仕組みを，コンピュータのデータ構造としてどのように扱うかが問題になる。これについては，Kramer[7]が提案を行っている。一般に剛体の運動を考えるとき一つの剛体の位置と姿勢をあわせると六つの自由度がある。これに対して，リンクをジョイントで接続していくと，そこで使われたジョイントの種類によって各リンクの自由度が拘束されていく。この条件を組み合わせると，多様な機構のプランニングが可能になる。

この表現法を利用して，各リンクとそのジョイントによる接続関係をモデルに組み込んでモデルを作っていくことを試みた[8]。実際には，過去の経験に基づいて人が頭脳の中にイメージを作り，それに従ってリンク機構のシミュレータを構成して動作の確認をするはずである。本稿で扱っているテーマの目的から云うと，モデルを組み立てる設計の過程でいつでも制約条件が記述されていて，その制約を考慮しながら設計過程を進めたいのである。そのような実験を試みたのが，図4である[8]。各リンクの構造を決めながら，それとあわせてジョイントによる動きの自由度が常に確認できるようになっている。モデルを組み上げる途中で，リンクの動きが作れないような組み合わせを持ち込むと，その段階でその状況が分かるようになっている。

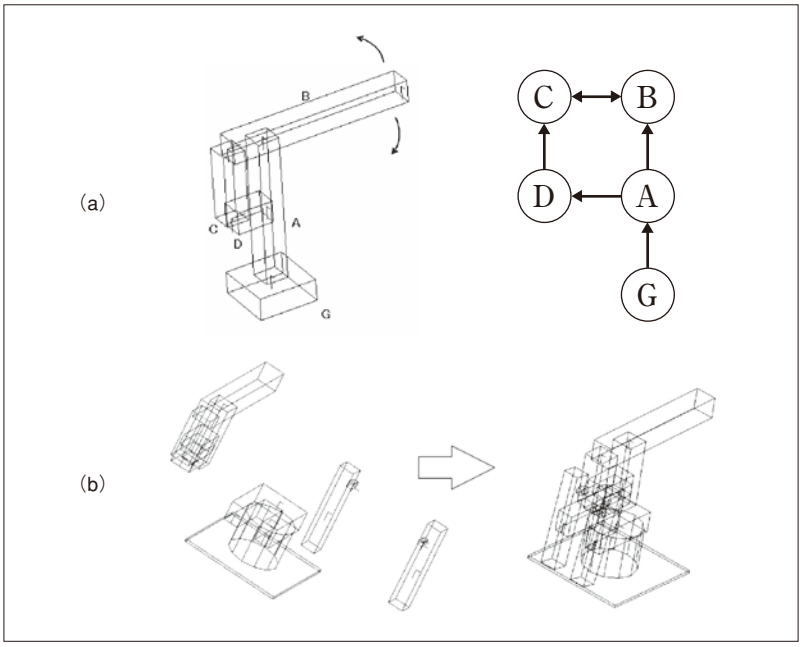

図4　リンク機構の組立

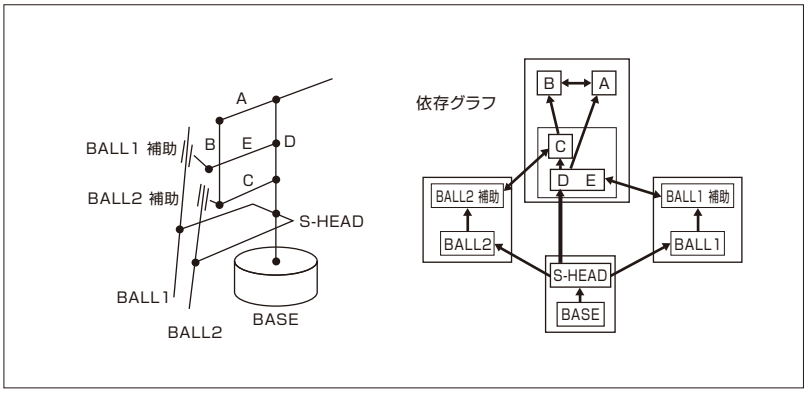

図5　ロボット機構におけるリンクの依存関係

この様子を見ると，制約条件を表したモデルと云っても，図1のように数値計算についての制約条件と，ここで述べたリンク機構の制約条件では，その取扱いが大きく異なっていることがわかる。そして，実際に図2のような一般的な製品モデルを考えるときには，そのような異質な制約条件がお互いに関係しあって結びついていることになる。図4の過程を進めて行って，産業用ロボットの機構に使われるリンクとジョイントの依存関係を記述したものが図5である。この例の場合は，

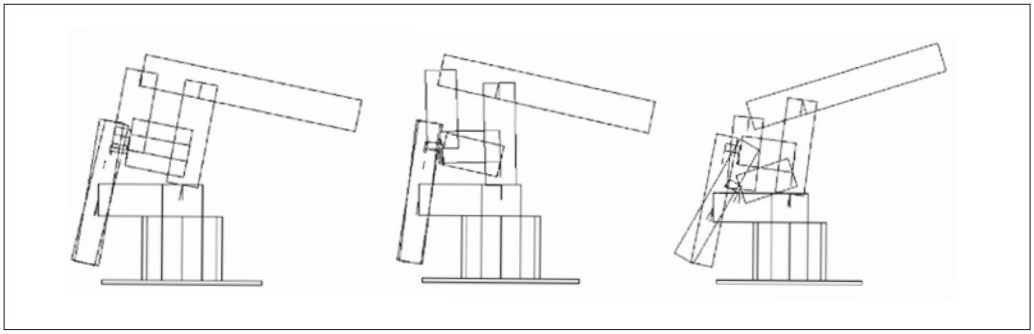

図6
ジョイントの制約条件による動作確認

設計する人の知識としてこのような機構でロボットを構成できることが分かっていてモデルを構築している。無論，この過程でリンクの長さやジョイントの作動範囲をいろいろ変えてモデルを作ってみることはそれほど難しくない。しかし，どのようなリンク機構が適当か全く分からない状態で，このような制約に基づくモデリングがどこまで有効かは明確になっていない。結局，最後のところは，人のもっている知識に依存する面が強いことになる。

出来上がった機構のモデルについて，ジョイントの制約条件だけによる動作確認を行ったのが図6である。ここまでくると，シミュレータを作るのと同じに見えるが，制約に基づくモデリングの役割はモデルを作るまでであり，シミュレータはこのモデルを基礎にして改めて作る必要がある。図6に示したものは，各ジョイントに残された自由度がどこに存在するかという情報だけで，リンクを動かしている。別の言い方をすれば，シミュレータを作る以前に，予めリンクの自由度がどこにあるかを確認できる。制約に基づくモデリングを一般的に考えるとすると，箱の中にいろいろな長さのリンクといろいろな拘束条件を与えるジョイントがあって，それらを組み合わせながら設計者が新しい機構の仕組みを作れるだろうか，ということが大切になる。今回の試みでは，ジョイントの制約を考慮しながら機構を組み立てる可能性は示したが，設計のための思考作業を支援する過程についてはまだ検討ができていない。

6. おわりに

ここでは，水圧シリンダー，油圧クラッチ，産業用ロボットの機構の例について，制約に基づくモデリングの検討を行った。まだ，ケーススタディしか行っておらず，汎用性のある結論は得られていない。現実の問題としては，ここにあげた機械設計にとどまらず，いろいろな計画型問題の解決に制約充足解を見出すことが重要になってきている。しかも，その対象は複雑になり，数理計画法で定式化するにしても容易ではない。ここに挙げた参考文献以外にも，制約の視点から問題を調べる試みは現在も行われている。より複雑化していくグローバルな計画型の問題解決に対して，少なくとも問題を整理することの視点に，制約というキーワードが使えることを期待したい。

参考文献

［1］N. F. Michelena and A. M. Agogino, “Multi-objective Hydraulic Cylinder Design”, Trans. of the ASME Journal of Mechanisms, Transmissions and Automations in Design, Vol. 110, pp.81-87. 1988.

［2］Wm Leler, “Constraint Programming Languages-Their Specification and Generation-”, Addison-Wesley, 1988.

［3］淵　一博監修，溝口文雄，古川康一，J-L. Lassez編, “制約論理プログラミング”，共立出版，1989.

［4］乾　正知，木村文彦，“依存情報を用いたモデル作成過程の表現とその可変ソリッドモデリングへの応用”，情報処理学会論文誌，Vol.31，No. 7, pp. 1038-1050, 1990.

［5］Kim Marriott and Peter J. Stucky, “Programming with Constraints: An Introduction”, The MIT Press, 1998.

［6］長尾嘉祐，“設計支援における制約充足モデルと定性的評価機構”，大阪大学大学院工学研究科修士論文，1991.

［7］Glenn A. Kramer, “Solving Geometric Constraints Systems”, The MIT Press, 1992.

［8］Naoshi Kubo, “A Study on Design Support in Mechanical Systems Using Geometric Constraint Solver and Constraint Propagation”，広島大学大学院工学研究科修士論文，2001.

山縣　敬一　Keiichi Yamagata

京都大学大学院工学研究科数理工学専攻博士課程単位取得満期退学。工学博士（大阪大学）。大阪大学大型計算機センター助手，大阪大学工学部精密工学科助手・助教授を経て広島大学総合科学部教授，広島大学工学研究科情報工学専攻担当を経て，現在，京都情報大学院大学教授。