

## 機械の組立・分解に必要な協調作業について

阿久津達也

工業技術院 機械技術研究所

機械の組立・分解においては複数の部品を同時に動かす必要がある場合がある。本稿ではそのような協調作業について考察する。特に、組立・分解において最低何個の部品を協調して動かさないといけないかを求める方法について考察する。具体的には、組立・分解をロボットのプランニング問題としてとらえ、その個数を求める問題をロボットのプランニング・アルゴリズムや Configuration Space 上のある種の探索問題と関連づける。

## On Cooperative Motions in Assembling a Composite Object

Tatsuya AKUTSU

Mechanical Engineering Laboratory  
Namiki 1-2, Tsukuba, Ibaraki 305, Japan

When rigid components are assembled to a composite object or a composite object is disassembled to components, it is sometimes required that several components are moved simultaneously or cooperatively. This paper considers the problem of deciding the minimum number of components which are required to be moved simultaneously or cooperatively.

## 1. はじめに

現在、協調して動作するシステムに関する研究が盛んに行われている。一方、実際問題においては複数人間が協調して作業を行わないと解決できないような問題も数多く存在する。本稿では、特に機械の組立・分解という作業を対象として、その際に必要となる協調作業およびその検出法について考察する。現実問題において必要となる協調作業について考察することは、協調型システムのモデル化のために意味があることと思われる。

機械の組立もしくは分解においては、「ある部分を片手でやさえながら別の部分をもう一方の手で動かす」といった作業が頻繁に行われる。また、ほとんどの部分は一人でも組立・分解が行えるように設計されているが、ある場合には人手までは借りなくても工具で押さえておく、などというように他の助けが必要になる場合がある。Natarajan は文献[3]において機械の分解作業において必要となる手の数はいくらでも多くなりうることを示している。それでは、最低限必要となる手の数はどのようにして計算すれば良いかということが自然な疑問としておこることになる。本稿では、Configuration Space[2]という概念を用いることにより、最低限必要となる手の数を求めることが Configuration Space 上の探索問題としてとらえられることを示す。

## 2. 組立・分解の定式化

実際の機械の組立・分解には様々な要因が含まれるが、本稿では、問題を単純にするために(まさつや重力などは考慮せず)幾何学的要因のみを考えて機械の分解問題を以下のように定式化する。

[機械の分解問題]

入力: 機械部品  $B_1, B_2, \dots, B_n$  の形状と初期配置(各部品が組合わさった状態)  $S$

出力: 各部品が互いに干渉せずに、互いに十分離れた位置に移動するための各部品の動作

上記の定義では、「手」に相当するものがでてこないが、「手」は十分に小さく、部品どうしが干渉しないという条件のもとで、各部品を自由に動かせるものとする。さて、機械の組立問題を同様の枠組みにおいて考えると、機械の組立問題は互いに十分離れた位置にある部品をある定められた配置に移動させるための動作を求める問題となる。つまり、機械の分解問題が解ければそれと逆の軌跡をたどることにより機械の組立問題も解けることになる。よって、以降、本稿では機械の分解問題だけを対象とすることにする。

なお、機械の分解問題はロボットの動作計画問題(Piano Movers Problem)としてとらえることができる。各部品を平行移動の3自由度と回転移動の3自由度の計6つの自由度を持ったロボットと考えると、系全体は6mの自由度を持ったロボット・システムとしてとらえることができる。すると、機械の分解問題は、このロボット・システムにおいて、与えられた初期配置から自分自身と干渉しないようにして目標配置へ至る動作をプランニングする問題としてとらえることができる。ロボットの動作計画問題は実際的にも理論的にも数多くの研究がなされている。特に、理論的に重要な結果としては、ロボットの自由度に制約がない場合にはプランニング問題は PSPACE 困難であることが示されており[4]、また、動作経路を求めるアルゴリズムとして、自由度に関して一重指数時間、ロボットや障害物の形状の記述に関しては多項式時間というものがある[1]。

## 3. 協調作業の種類

2節の定式化のもとでの機械の分解において必要となる協調作業には大きく分けて2種類の協調作業があると思われる。それは、

- (1) 同時に複数の部品を動かさなければならない場合。
- (2) ある部品を少しずらして、次に他の部品を少しずらして・・・、というように、同時にずらす部品は一つでも良いが、部品を一

つずつ取りはずして分解できない場合。

である。Natarajan は文献[3]において、(1)の場合の例を数多く示している。その例の一つを図1に示す。実際には、(1)の協調作業よりも(2)の協調作業が必要になる場合が多いと思われる。かんぬきのようなものが相当するが、その例を図2に示す。なお、(2)の定義において部品を同時に動かしても良いことにしてあるので、(1)の協調作業は(2)の協調作業に含まれることになる。

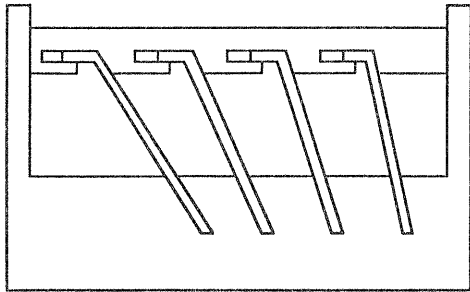


図1 部品を同時に動かす必要がある場合

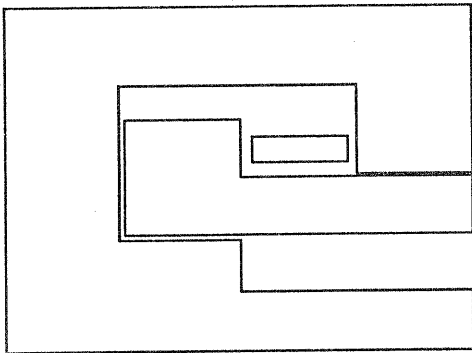


図2 部品を交互に動かす必要がある場合

#### 4. 必要となる協調作業の計算

本節では、機械の分解において前節の(2)の意味で協調して動かさなければならない部品数の下限を求めることについて考察する。より厳密な議論を行うために、分解作業を以下のステップを1からhまで繰り返して行うものとしてとらえることにする。

STEP (i): 部品、 $B_{i1}, B_{i2}, \dots, B_{iki}$   
のみを(2)の意味で同時に動かして取り外す(十分遠い位置まで移動させる)。

この時、(iを1からhまで動かした時の) $k_i$ の最大値(kとする)が、(2)の意味で同時に動かす必要のある部品の数になる。もちろん、各STEPの選び方は一意ではないので、考え得る全ての分解方法の中でkの最小値を求めることになる。初期状態Sに対するkの最小値を $k(S)$ とする。つまり、 $k(S)$ は初期状態Sから分解するのに(2)の意味で最低何個の部品を同時に動かさなければならぬかを表している。

ところで、以下の性質が成立することが簡単にわかる。

#### [性質1]

機械の分解における初期状態をS、また、 $B_1, B_2, \dots, B_n$ の中でいくつかの $B_i$ が既にはずされた状態(残りの部品は元の位置であるものとする)を $S'$ とする。この時、

$$k(S) \geq k(S')$$

が成立する。 □

この性質を利用すれば、アルゴリズム1により $k(S)$ を求めることができる。

このアルゴリズムは、とにかくはずせるものからはずしていこうという貪欲算法(greedy algorithm)であるが、性質1のためにこれで正しく動作することがわかる。また、このアルゴリズムの時間計算量は(\*)の部分の時間計算量を $C(k, n)$ とすると(但し、nは入力全体のサイズ)、

$$O\left(\sum_{k=1}^m 2^m C(k, n)\right) = O(m 2^m C(m, n))$$

となるが、文献[1]のアルゴリズムを用いると、 $C(k, n)$ がkに関する一重指数時間、nに関する多項式時間となるので、アルゴリズム全体としては、mに関する一重指数時間、nに関する多

項式時間となる。但し、文献[1]のアルゴリズムは複雑で実用的でないので、このアルゴリズムも実用的ではない。

[アルゴリズム 1]

Procedure Find-k(S)

```

begin
  for k=1 to m do
    begin
      B := { B1, ..., Bm };
    L1: flag := FALSE;
      for all subset B' ⊂ B such that
        | B' | ≤ k do
          begin
            B - B' を障害物とみなし、B' だけを
            動かすことにより B' の各部品を取り外
            すことができるかどうかを調べる;
              - (*)
            if 取り外すことができる then
              begin
                flag := TRUE;
                break
              end
            end;
          if flag = TRUE then
            begin
              B := B - B';
              if B = φ then
                begin
                  output k;
                  halt
                end
              else
                goto L1
              end
            end
          end
        end
      end
    end
  end
end

```

5. Configuration Space

ロボットのプランニングにおいては Configuration Space という考え方が良く用いられる[2]。今、自由度 d のロボット・システム B があったとする。この時、B の状態 (配置) は d 次元空間上の一点により表わすことができる。この空間が Configuration Space と呼ばれるものであるが、この空間上の各点は更にその点が自分自身もしくは障害物と干渉する点であるかないかにより 2 種類にわけることができる。ここで、干渉する点の集合を IP、干渉しない点の集合を FP とする。この時、初期配置を表わす点から目標配置を表わす点を結ぶ曲線で FP に含まれるものが干渉しないで初期配置から目標配置へ至るロボットの動作に対応することになる (図 3)。

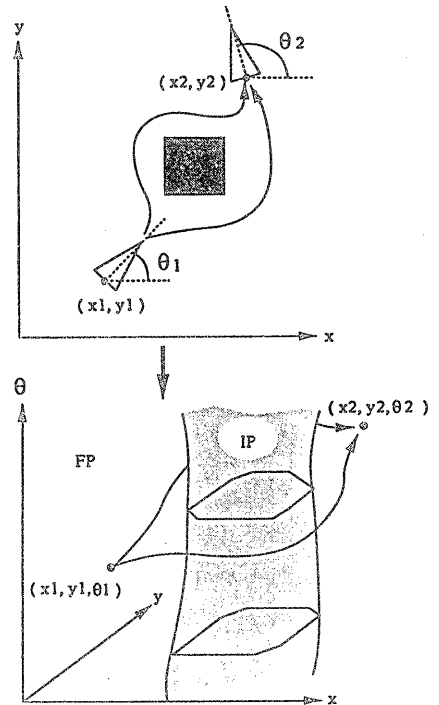


図 3 Configuration Space

さて、機械の分解を考える際には、部品全体をロボットとして考えれば良いことを 2 節で示したが、この Configuration Space 上で考えれば、3 節の (1)(2) それぞれの協調動作がどのようなもの

であるかがわかりやすくなる。簡単のために図4および図5を用いて説明する。図4および図5は1自由度の部品(AとBとする)が二つあった場合の Configuration Space を示している。ここで、初期配置を表わす点から無限遠(もしくは目標配置)とを結ぶ曲線でFPに含まれるものが存在す

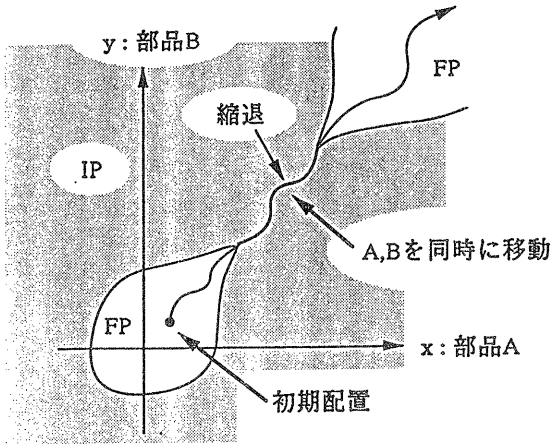


図4 (1)に対応するConfiguration Space

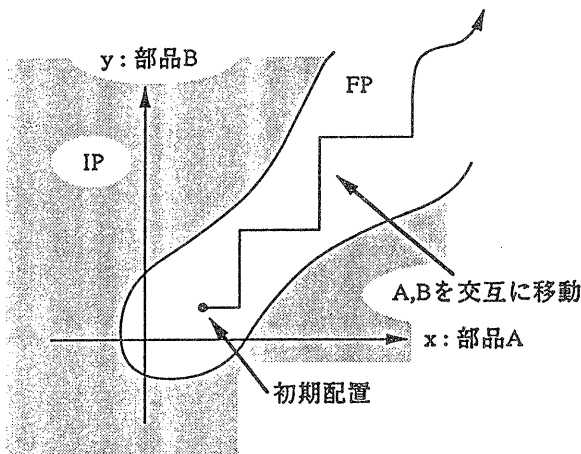


図5 (2)に対応するConfiguration Space

れば機械の分解が行えるわけであるが、そのような曲線は、図4においては必ず縮退した部分を通らなければならない。これは、AとBを同時に動

かさなければならないことを意味する。このことは(1)の協調動作に対応する。また、図5においては図に示す曲線に対応する動作を行えば、A、Bを同時に動かす必要がないことがわかるが、どうい曲線を考えてもAもしくはBだけを単独で動かすことにより分解することはできないことも図5よりわかる。このように考えると、必要となる協調作業の検出というのは Configuration Space 上でのある性質を持った領域の探索問題としてとらえることができる。しかしながら、現在のところそのためのアルゴリズムを得るには至っていない。

## 6. おわりに

本稿では、機械の組立・分解において必要となる協調作業について考察した。特に、最低何個の部品を協調して動かさなければならないか、を求める方法について考察した。しかしながら、そのための実用的なアルゴリズムはまだ得ていず、今後の課題である。また、本稿で示した考え方が機械の組立・分解といった作業以外の協調作業の検出にも利用できるかどうか調べることも今後の課題である。

## 謝辞

日頃討論頂く機械技術研究所、次世代構造化ソフトウェア・プロジェクト・グループの皆さんに感謝の意を表します。

## 参考文献

- [1] J. F. Canny: "The Complexity of Robot Motion Planning", The MIT Press, 1988.
- [2] T. Lozano-Perez: "Spatial Planning: A Configuration Space Approach", IEEE Trans. on Computers, C32, pp.108-120, 1983.
- [3] B. K. Natarajan: "On Planning Assemblies", Proc. 4th ACM Symp. Comp. Geo., pp.299-308, 1988.
- [4] J. T. Schwartz, M. Sharir and J. Hopcroft eds.: "Planning, Geometry and Complexity of Robot Motion", Ablex Publishing, 1987.