TM-1226

制約論理プログラミングによる ロボット構造設計支援システムの構築

佐藤　晋一、相場　亮

October, 1992

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

制約プログラミングによるロボット構造設計支援システムの構築

佐藤晋一、相場 亮

(財) 新世代コンピュータ技術開発機構
東京都港区三田 1－4－28 三田国際ビル 21 階

February 2, 1992

1　はじめに

本研究の目的は、制約論理プログラミング言語の応用システムとして、ハンドリングロボットの設計支援システムを構築し、制約論理プログラミング言語の有用性を示すことにある。

ハンドリングロボットの設計は、「基本構造設計」と「内部構造設計」とに大きく分けられる。基本構造設計とは、ロボットの構成、すなわちロボットの自由度、関節数、アーム長などの骨組を決定する工程であり、内部構造設計とは、関節の回転のためのモータの仕様などの設計を行う工程である。

我々の目指す設計支援は、主に基本構造設計の支援を目的としたものである。現在、ハンドリングロボットの設計支援は、ほとんど行われていないのが現状であり、構造決定後の数値的解析については、決定された各構造毎に個別の解析プログラムを作成している。制約論理プログラミング言語を用いるアプローチにおいては、このような現状における支援のさらに上流工程、すなわち基本構造の決定の支援を行うものである。この工程を支援することにより、設計工程の大幅な短縮が可能となる。制約論理プログラミング言語による設計支援システムは、任意構造のロボットに関する解析プログラムを簡単に作成できることに大きな特徴がある。

我々は、このような設計支援システムの中核となる機能である、機構学、お

よび静力学を扱うプログラムを、まず逐次環境における制約論理プログラミング言語 CAL を用いて作成し、次いでこれらをもととする設計支援システムの機能を検討し、これらを並列制約論理プログラミング言語 GDCC を用いて並列化した。

本論文の構成は次の通りである。まず第2章では、ロボット工学の基礎理論であるロボットの機構学と静力学の概要について述べる。第3章では、ロボットの設計支援の現状と問題点を示す。第4章では、設計支援システムの中核となる機構学、及び静力学のプログラムを制約プログラミングを用いて記述し、第3章に掲げた問題点を解消した点について述べる。第5章においては、それらのプログラムを用いて構築した、ロボットの構造支援システムの機能、モジュール構成、システムの実際の使用例について示す。

2 ロボットの機構学と静力学

2.1 ロボットの機構学

我々が対象とするロボットとは、主にハンドで物体をある場所から別の場所へ運び、またその姿勢を変えることを目的とするロボット(ハンドリングロボット)であり、図1のようなものである。言い換えればこのロボットの主要な機能は、ある位置姿勢にある物体を別の位置姿勢に変換することである。このためにロボットはアームの長さを変化させたり、関節機構の回転を行なう。このときの位置姿勢(ハンド空間パラメータ)と各アームの長さ、各関節の回転角度(関節空間パラメータ)との幾何学的な関係を示すものがロボットの機構学[牧野 75]である。この機構学は、個々のロボットの基本構造と密接な関係を持ち、基本構造の解析は主にこれを用いて行われる。したがってロボットの構造設計支援システムの中でも中心的な機能を果たす。

ここで物体の位置姿勢の位置とは、物体を保持するロボットのグリップの中心であり、3次元の座標 (px, py, pz) で表す。また、物体の姿勢は、その物体に固定された互いに直交する3軸の方向、すなわち、一組のトライアドによって示すことができる。ただし、トライアドにおける3番目の軸は2つの軸を決めればおのずから決まるので、これら2軸(主軸、副軸)だけによって姿勢を表現することができる。主軸、副軸は、各々 (ax, ay, az)、(bx, by, bz) という2つの

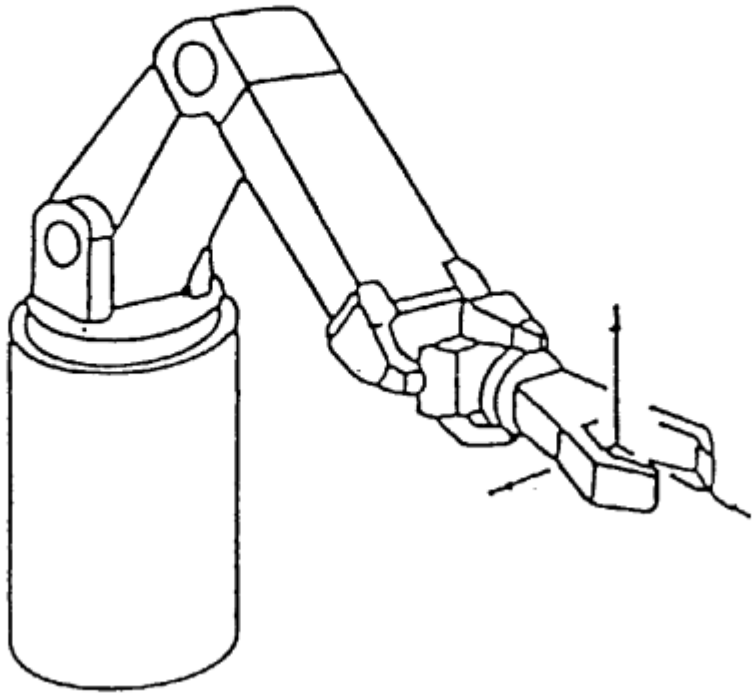

Figure 1: ハンドリングロボットの外観

単位ベクトルとして表す。これらの各成分の間には、

$$\begin{aligned} ax^2 + ay^2 + az^2 &= 1 \\ bx^2 + by^2 + bz^2 &= 1 \\ ax * bx + ay * by + az * bz &= 0 \end{aligned}$$

という関係が成り立つ。上記の位置と姿勢を一括して表現するために、つぎのような位置姿勢マトリクス $[P^{**}]$ を定義する。

$$[P^{**}] = \begin{pmatrix} px & ax & bx \\ py & ay & by \\ pz & az & bz \\ 1 & 0 & 0 \end{pmatrix}$$

ここで最下行の 1 や 0 は計算の便宜上加えたものであり、自由度には関係し

ない。このように物体の位置姿勢マトリクスには9変数が含まれるが、上記のように姿勢ベクトルの各成分の間には3つの冗長度が存在するので、実際の自由度は6である。

ところで、空間内における剛体の運動は並進運動と回転運動とからなる。いま、位置姿勢マトリクス $[P^{**}]$ で表される物体を、その始点を通る軸 $W(=(Wx,Wy,Wz))$ のまわりに θ だけ回転させ、さらにこれに変位 $dP(=(dx,dy,dz))$ なる並進運動を与えると、新しい位置姿勢マトリクス $[P'^{**}]$ は下記のように示される。

$$
\begin{aligned}
[P'^{**}] &= [M][P^{**}] \\
[M] &= \begin{pmatrix} E & dP \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} e_{11} & e_{12} & e_{13} & dx \\ e_{21} & e_{22} & e_{23} & dy \\ e_{31} & e_{32} & e_{33} & dz \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

$[M]$ は、回転変換と並進変換を一つのマトリクスの中にまとめたもので、これを変換マトリクスと呼ぶ。なお、回転変換マトリクス E 中の要素 e_{11} から e_{33} までは回転軸 W の各要素と回転角度 θ により決定される。

最初に述べたように、ハンドリングロボットのグリップの位置姿勢の変換は、各関節機構によるアームの回転、およびアーム自身の伸縮により行なわれる。これらはそれぞれ上記の回転運動と並進運動に対応するので、先に示した変換マトリクスにより表される変換である。これを拡張すると、関節数が m 個のハンドリングロボットにおけるグリップの位置姿勢マトリクス $[P^{**}]$ と各変位パラメータとの関係式が得られる。この場合上記のような変換が m 回続けて行われるので、関係式は下記のように示される。

$$
\begin{aligned}
[P^{**}] &= \begin{pmatrix} E_1 & R_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} E_2 & R_2 \\ 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} E_i & R_i \\ 0 & 1 \end{pmatrix} \cdots \\
&\quad \begin{pmatrix} E_m & R_m \\ 0 & 1 \end{pmatrix} \begin{pmatrix} G_0 & a_0 & b_0 \\ 1 & 0 & 0 \end{pmatrix} \\
&= [M_1][M_2] \ldots [M_i] \ldots [M_m][G^{**}], \qquad (1)
\end{aligned}
$$

ただしここで

E_i : 固定点から i 番目の関節の回転変換マトリクス
R_i : 固定点から i 番目のアームの回転前のベクトル (Rx_i, Ry_i, Rz_i)
G_0 : ハンドのグリップベクトルの初期値 (G_x, G_y, G_z)
a_0, b_0 : ハンドのグリップの姿勢（主軸、副軸方向）の初期値 $(ax_0, ay_0, az_0), (bx_0, by_0, bz_0)$
m : 関節数（ランク数）

である。また以下に登場するパラメーターの定義は次の通りである。

W_i : 固定点から i 番目の関節の回転軸の方向ベクトル (Wx_i, Wy_i, Wz_i)
θ_i : 固定点から i 番目の関節の回転角度

(1) 式がロボットの機構学を表現する関係式であり、特性方程式ともよばれる。

上記のパラメータの中で、$\theta_i, Wx_i, Wy_i, Wz_i, Rx_i, Ry_i, Rz_i$ は、各変換マトリクス $[M_i]$ の内容を決定するものである。これらのパラメータは、後に述べるようにロボットの基本構造を決定する重要なものであり、以下基本構造パラメータと呼ぶ。

2.2 ロボットの静力学

前節に示したロボットの機構学は、ハンドの位置姿勢と関節空間パラメータの幾何学的な関係を示している。一方、ロボットがグリップで物体を保持しているときには、そこに外力が作用しているので、これにより各関節にはある大きさのトルクが作用することになる。ロボットが関節を回転させるためには、そのトルクに逆らってモータの軸を回転させなければならないので、トルクが許容範囲内に抑えられていることが必要である。また逆に、作用する最大のトルクが決まっていれば、それに対応してモータ軸のトルク設計を行わなければならない。このときのグリップに作用する外力と各関節軸に作用するトルクとの関係を示すものがロボットの静力学である。この関係式は、以下のように示される。

$$T_i = Z_i \cdot (P_m - P_i) \times F_m \qquad (2)$$

ただし、

T_i ： 固定点から i 番目の関節に作用するトルク
Z_i ： i 番目の関節の回転軸の方向ベクトル
P_m ： 固定点からグリップまでの位置ベクトル
P_i ： 固定点から i 番目の関節までの位置ベクトル
F_m ： グリップに作用する外力のベクトル

である。

静力学の関係式は、機構学によって行われる幾何学的な解析に加え、力学的な解析を行うために不可欠なものであり、設計支援システムの中で機構学同様に中心的な機能を果たす。

3 ロボットの設計支援とその現状

3.1 ロボットの設計パラメータと評価関数

ロボットに限らず一般に設計問題とは、設計パラメータがあり、その関数として制約条件と評価関数が与えられ、制約条件の範囲内で評価関数を最大に（または最小に）するパラメータの値を求めるということである [高野 86]。まず、ロボットの設計パラメータ、設計の評価関数と拘束条件などを明らかにする。

設計パラメータはロボットの基本構造と内部構造に分けることができる。前者は、ロボットの自由度、関節数、アーム長等であり、ロボットの骨組を決定するものである。後者は、伝達機構、モーターの種類、アーム太さなどの形状等であり、基本構造の決定後に設計するものである。

一方、ロボットの評価の対象としては、作業のし易さ、モータパワー、アームの重量、ロボットの全重量等が考えられる。このうち作業のし易さというのはあいまいな表現であるが、その尺度として提案されているものの一つに「可操作度」がある [吉川 84]。可操作度は関節駆動速度とアーム先端の移動速度の比に相当するもので、これが大きいということは運動学的に先端速度を上げられることを意味する。逆にこれが小さく 0 に近いということは、アーム先端の速度に対し、関節の駆動速度を非常に大きくしなければならないということで操作性が悪くなる。可操作度が 0 になるロボットの姿勢は、「特異点」と呼ばれ、この状態では実質的にロボットの自由度が退化するので、ロボットは作業

時にこのような姿勢を極力避けるようにしなければならない。。したがって、可操作度は、ロボットの操作性を定量的に示す有効な尺度の一つである。モータパワーは、モータシャフトのトルクに比例するので、トルクの算出によりその評価が可能である。制約条件は、ロボットの仕様や作業条件により、予め決定されるものであり、関節変位の上下限、可搬重量、アームの動作範囲等である。

3.2 設計支援の現状と問題点

前節で述べたようにロボット構造には、大きく分けて、基本構造と内部構造の2つがある。通常のロボット設計の流れは、図2に示すように基本構造の設計、評価、そして内部構造の設計、評価という順に進んでいく。そして各々評価の段階で設計結果に対し、フィードバックがかけられ、場合によっては再設計が要求される。この中で基本構造の設計対象は、自由度、関節数、関節軸方向、アーム長等であり、設計者はこれにより予め与えられた作業を実行するロボットの根本的な骨組を決定する。これを誤ると後の内部構造の設計が全く無駄になってしまうので、設計上非常に重要な過程ということができる。

上記のような設計過程を支援する設計支援システムとして実際に設計現場で使われているものは、現在のところ存在せず、設計の大部分は設計者の経験と勘により進められているのが現状である。ただし、部分的にある設計パラメータを変えて一つの評価関数を最大(最小)にする研究はいくつか発表されている。これらの設計では、ロボットの自由度、関節数、関節軸方向、アーム長等、ロボットの基本構造が既に決定され、駆動モータも取り付け位置が決まっているものとしてアームの太さ、モータパワーなどの最適値を見出す問題を扱っている。このように既に基本構造が決まっていて単に寸法やモータの能力等の数値パラメータを変更するだけの設計システムを構築することは比較的簡単と考えられており、例えば代表的な産業ロボットの1つであるSCARA型組み立て用ロボットの仕様変更による諸元の変更の計算機による支援設計システムは、既存のCADシステム等に使われているいるソフトウェア技術を適用することにより、開発できるであろう。

一方、基本構造の設計を含めた総括的な設計支援システムは、従来から切望されているが、まだ部分的にも試みられた例はない。これに関しては、既存の

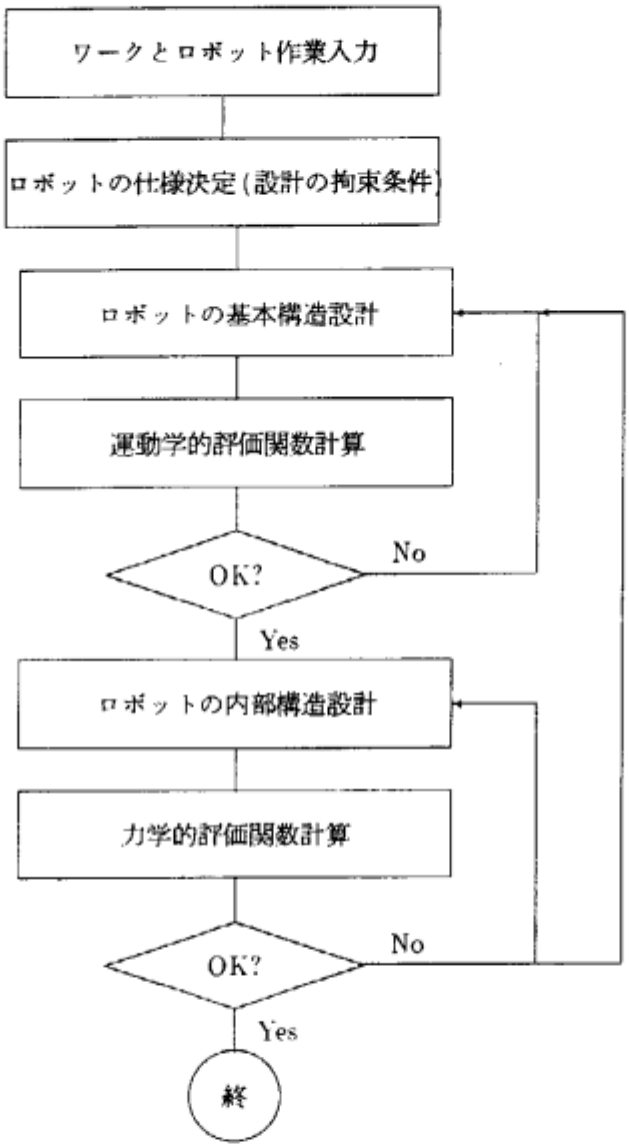

Figure 2: ロボットの一般的な設計手順

ソフトウェアの記述能力の限界により、ほとんど開発の見通しが立っていないというのが現状である。

ロボットの基本構造の設計支援を行うためには、ユーザにより設定されたロボット構造のさまざまな評価を行うための解析プログラムが必要である。しかしロボット工学の分野で従来から適用されている Fortran、Basic 等のプログラミング言語では、ロボットの基本構造が決定するまでは、プログラムを記述することは不可能である。これらのプログラミング言語を用いて基本構造の設計支援を行うものとすると、その設計過程は、図3の (a) のようになるものと考えられる。現状では、設計者自身が設定したロボット構造に対応する機構学や静力学の制約式を机上で手計算により導き、その後得られた制約式を解析する手続きをプログラムとして記述している。その場合、解析プログラムは、予め設定されたロボット構造に対応して作成しなければならない。構造設計というものは、評価と改良による試行錯誤を繰り返し行っていくものであるが、これではロボット構造の変更のたびに解析プログラムを書き直さなければならない。すなわち最も労力を要する図中の点線内の作業を何度も繰り返さなければならず、とても設計効率の向上は望めない。これが、現在まで汎用の設計支援システムの構築を妨げている最大の要因である。

本研究において我々はそのような問題点を解消し、ロボットの基本構造の設計支援を可能とする汎用システムの構築を目的とする。

4 機構学、静力学への制約プログラミングの適用

4.1 プログラミング方針

2.1 節、及び 2.2 節にて掲げた機構学、および静力学の関係式は、我々が構築する設計支援システムの中核となる基礎式であり、設計過程におけるロボット構造の解析、評価はこれらを中心として行われる。3.2 節にて述べたように従来、それらを扱う解析プログラムは、設計されたロボット構造に個々に対応して作成されていたため、構造設計に対する支援は困難なものとなっていた。本研究における我々の目的は、ロボットの基本構造の設計支援を実現することであり、このためには、ロボット構造に依存しない汎用の解析プログラムが記述できることが必要である。我々は、制約プログラミングを機構学、及び静力学に適用

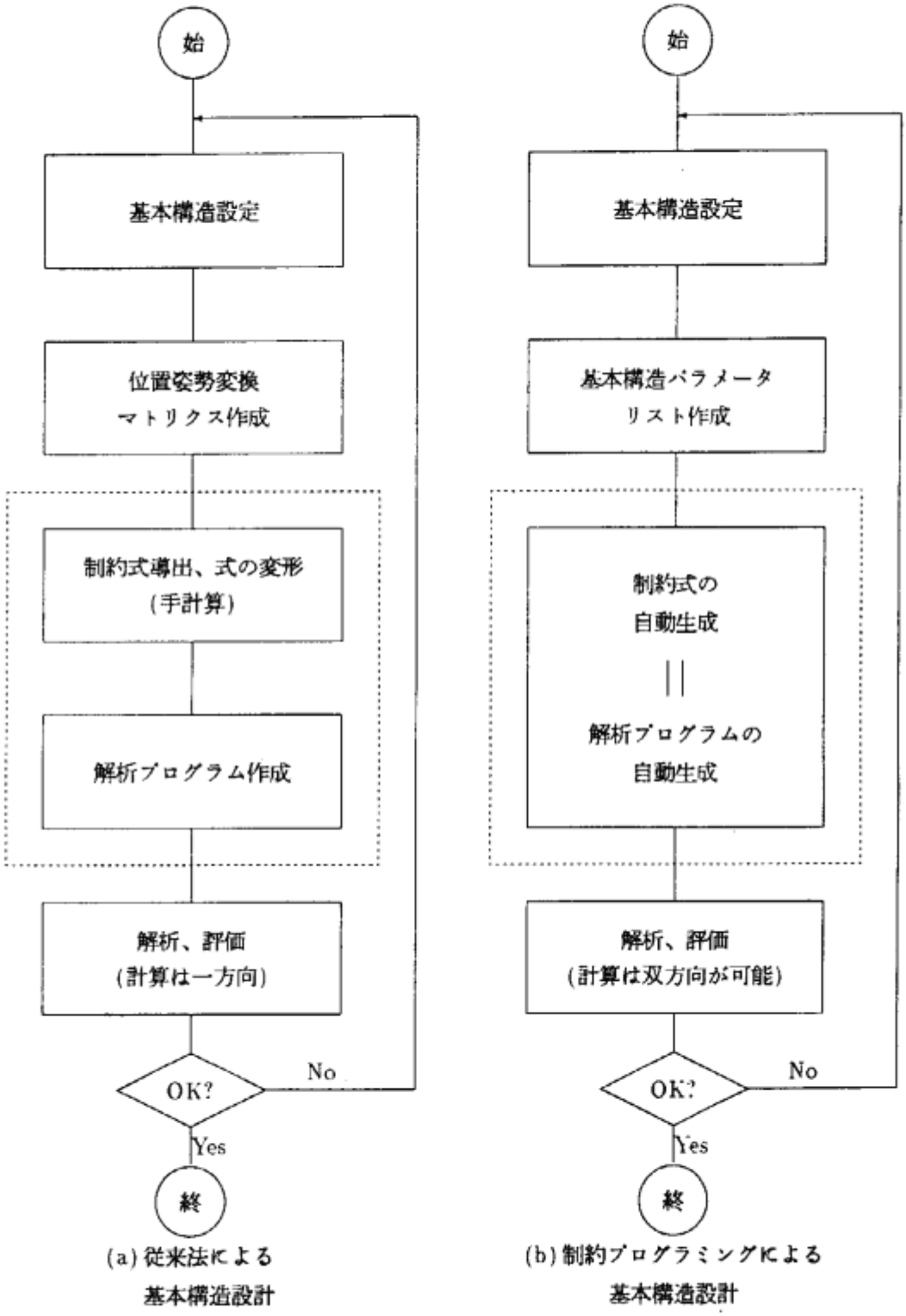

Figure 3: 従来法、及び制約プログラミングによる基本構造設計の比較

することにより、これが可能であることを見出した。具体的には、ロボットの基本構造をリストとして表現し、ユーザに対し、これを変数として入力させることを許すようなプログラムを記述することで任意の基本構造を持つロボットに対応できるようにした。これを基本的な方針として、設計支援システムの構築に先立ち、逐次環境の制約論理プログラミング実験システム CAL 上に、機構学、静力学の解析プログラムを各々記述した。以下、各々のプログラムの仕様、および特徴について述べる。

4.2 機構学プログラムの仕様

機構学の制約式である 2.1 節の (1) 式を各成分ごとに記述すると、位置姿勢パラメータに関する 9 つの関係式が得られ、これらが制約プログラミングにおけるソースプログラム上の制約式となる。プログラム中に登場するパラメータは、

1. 基本構造パラメータ
   $cos\theta_i, sin\theta_i, W_{ix}, W_{iy}, W_{iz}, R_{ix}, R_{iy}, R_{iz}$
2. ロボットハンドの位置姿勢パラメータ
   $Px, Py, Pz, Ax, Ay, Az, Bx, By, Bz$
3. グリップパラメータ
   $Gx, Gy, Gz, Ax_0, Ay_0, Az_0, Bx_0, By_0, Bz_0$

であり、グリップパラメータはロボットのグリップの位置姿勢の初期値であり、通常は具体的数値として入力する。基本構造パラメータは、(1) 式中の各変換マトリクス $[M_i]$ を決定する。このうち各アームのベクトル R_i および各関節の回転軸の方向ベクトル W_i は、市販の多くのロボットにおいてはその方向が X,Y,Z 軸のいずれかに一致しており（直交系ロボット）、3 成分のうちの 2 つは 0 である。なお、各関節の回転角度 θ_i に関しては、CAL 上では三角関数を扱うことはできないので、$cos\theta_i$,$sin\theta_i$ を各々 $cosi$,$sini$ という変数として導入し、これらの間に $cosi^2 + sini^2 = 1$ という制約条件を与えている。プログラムのトップレベルの述語 (kinematics) の引数の並びは、以下の通りである。

$$kinematics(Mlist, Gx, Gy, Gz, Ax_0, Ay_0, Az_0, Bx_0, By_0, Bz_0,$$
$$Px, Py, Pz, Ax, Ay, Az, Bx, By, Bz)$$

ここで、最初の引数 $Mlist$ は、基本構造パラメータにより構成されるリストであり、以下のようなものである。

$$[[cosm, sinm, Wx_m, Wy_m, Wz_m, Rx_m, Ry_m, Rz_m],$$
$$\cdots$$
$$\cdots$$
$$[cos2, sin2, Wx_2, Wy_2, Wz_2, Rx_2, Ry_2, Rz_2],$$
$$[cos1, sin1, Wx_1, Wy_1, Wy_3, Rx_1, Ry_1, Rz_1]]$$

上記の $Mlist$ の中にある、8つの基本構造パラメータから成る一つのリストが一組のアームと関節の構造を表現し、このリストの数がロボットの関節数に一致する。したがって、$Mlist$ の内容はロボットの全体の構造を厳密に表現するものであり、この内容を変更することにより、何関節のどのような構造を持つロボットでも扱うことが可能である。

4.3 静力学プログラムの仕様

静力学のプログラムにおけるトップレベルの述語 (statics) の引数の並びは、以下の通りである。

$statics(Mlist, Qlist, Tlist, Gx, Gy, Gz, Px, Py, Pz,$
$Fx, Fy, Fz)$

ここで

1. $Qlist$
   各関節の座標のリスト $[[qx_1, qy_1, qz_1], [qx_2, qy_2, qz_2], \ldots, [qx_m, qy_m, qz_m]]$
2. $Tlist$
   各関節に作用するトルク変数のリスト $[t_1, t_2, \ldots t_m]$
3. (Gx, Gy, Gz)
   初期のグリップのベクトル
4. (Px, Py, Pz)
   グリップの位置

5. (Fx, Fy, Fz)
   グリップに作用する外力

である。$Mlist$ は、機構学のプログラムと同様の基本構造パラメータのリストであり、この内容を変更することにより、あらゆる構造のロボットを表現することができる。機構学のプログラムでは、与えられたグリップの目標位置、姿勢と、$Mlist$ のパラメーターとの関係式を算出し、それらを出力する。これに対し、静力学のプログラムは、$Mlist$ 中のパラメータおよび (Fx, Fx, Fz) の値を参照して、$Qlist$、および $Tlist$ 中の各パラメータの値を算出する。$Qlist$,$Tlist$ の変数値は、ロボットの動きと共に変化するので、我々は、これらの値をロボットが、各アームを動かし終えるたびに計算する必要がある。ただし、これによりその時点でのロボットアームの全体の軌道を把握することができるので、障害物等のチェック等が可能になる。

4.4 プログラムの特徴

上記の2つのプログラムの最大の特徴は、その内容を全く変更することなく、これらに対する問い合わせのみの変更により、任意の構造を持つロボットを扱えることにある。これらのプログラム自体は、基本的にマトリクスの乗算を定義するだけの極めて単純なものであるが、ユーザはこれらに対し、任意のロボット構造をこれと1対1に対応する簡単な基本構造マトリクスのパラメータリスト、$Mlist$ として入力することができる。これを受けたプログラムは、そのロボット構造により決定される ロボット機構学、あるいは静力学の制約式を自動的に生成する。例えば、図4のようなベクトル図により表される3関節6自由度のロボットの機構学の制約式を求める場合のプログラムへの問い合わせは以下のようになる。

$$
\begin{aligned}
kinematics([&[cos3, sin3, 0, 0, z_3, 0, 0, 1],\\
&[cos2, sin2, x_2, 0, 0, 1, 0, 0],\\
&[cos1, sin1, 0, 0, z_1, 0, 0, 1]],\\
&5, 0, 0, 1, 0, 0, 0, 1, 0,\\
&px, py, pz, ax, ay, az, cx, cy, cz).
\end{aligned}
$$

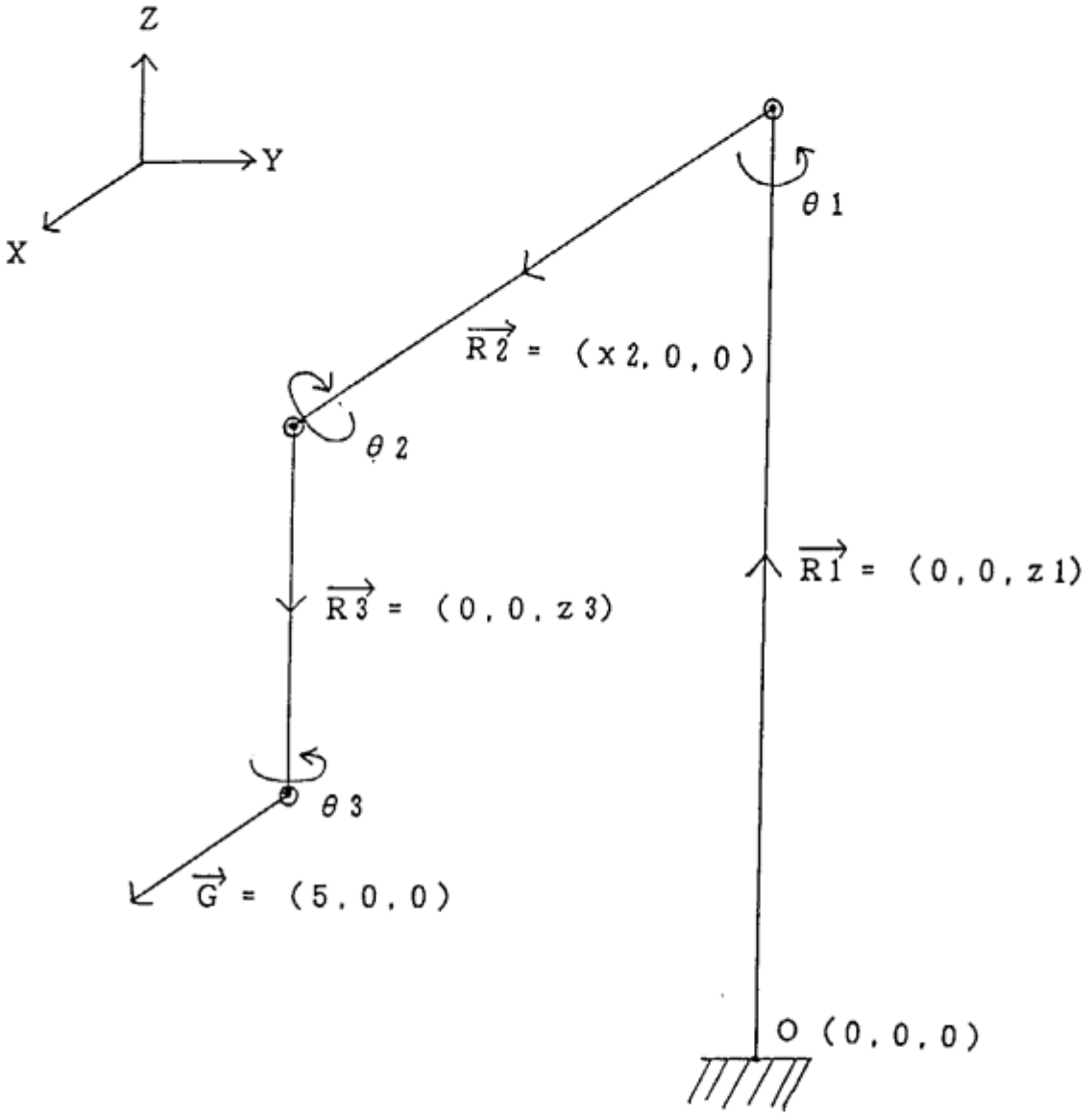

Figure 4: 6 自由度ロボットの概略図

ここで、(px, py, pz) はロボットの手先の位置、(ax, ay, az)、(cx, cy, cz) は姿勢を表すベクトルである。また、問い合わせ中の最初の変数であるリストは、4.2節、4.3節で定義した基本構造パラメータリスト Mlist に対応し、この中の $sin1$,$cos1$,$sin2$,$cos2$,$sin3$,$cos3$ は図中の各関節の回転角度 θ_1,θ_2,θ_3、また $z1$、$x2$、$z3$ は、各アームの長さに対応する。このリストは、図4のロボットの基本構造を厳密に表現している。この内容を変更すれば任意の基本構造を持つロボットの表現が可能である。これは、静力学のプログラムにおいても同様である。なお、ここではロボットのグリップの初期のベクトルを (5,0,0)、姿勢を (1,0,0)、(0,1,0) としている。この問い合わせに対するシステムの応答は、以下の通りである。

$$
\begin{aligned}
cos1^2 &= 1 - sin1^2 \\
cos2^2 &= 1 - sin2^2 \\
cos3^2 &= 1 - sin3^2 \\
px &= -5 * cos2 * sin3 * sin1 + z_3 * sin2 * sin1 \\
&\quad +5 * cos3 * cos1 + x_2 * cos1 \\
py &= 5 * cos3 * sin1 + x_2 * sin1 \\
&\quad +5 * cos1 * cos2 * sin3 - z_3 * cos1 * sin2 \\
pz &= 5 * sin3 * sin2 + z_1 + z_3 * cos2 \\
ax &= -1 * cos2 * sin3 * sin1 + cos3 * cos1 \\
ay &= cos3 * sin1 + cos1 * cos2 * sin3 \\
az &= sin3 * sin2 \\
cx &= -1 * cos1 * sin3 - cos3 * cos2 * sin1 \\
cy &= -1 * sin3 * sin1 + cos3 * cos1 * cos2 \\
cz &= cos3 * sin2
\end{aligned}
$$

このように位置姿勢パラメータと基本構造パラメータとの関係式を得ることができる。上記のような制約式が得られれば、制約プログラミングの計算の双方向性から、原理的にはあらゆるパラメータを変数とする問題が解けるので、

ユーザは設定したロボット構造について机上でさまざまな評価を簡単に行うことができる。この中で、関節空間パラメータの値からハンド空間パラメータを算出する問題は順変換問題、その逆は逆変換問題と呼ばれるが、いずれも同一プログラムで処理が可能である。例えば、図4のようなベクトル図で表されるようなロボットにたいして、グリップの位置が(40,-30,20)、姿勢が(1/3,2/3,-2/3),(-2/3,2/3,1/3)のときの各関節の回転角度、アームの長さを求めよという逆変換問題に対する問い合せ、および応答は下記のようになる。

$$
\begin{aligned}
&kinematics([[cos3, sin3, 0, 0, z_3, 0, 0, 1],\\
&\qquad [cos2, sin2, x_2, 0, 0, 1, 0, 0],\\
&\qquad [cos1, sin1, 0, 0, z_1, 0, 0, 1]],\\
&\qquad 5, 0, 0, 1, 0, 0, 0, 1, 0,\\
&\qquad 40, -30, 20, -1/3, 2/3, -2/3, 2/3, 2/3, 1/3).
\end{aligned}
$$

$$
\begin{aligned}
sin1^2 &= 4/5\\
sin2 &= 5/6 * sin1\\
sin3 &= -1 * sin1\\
cos2 &= 2/3\\
cos1 &= -1/2 * sin1\\
cos3 &= 1/2 * sin1\\
z_3 &= 26\\
x_2 &= -105/2 * sin1\\
z_1 &= 6
\end{aligned}
$$

4.5 設計支援への有効性

前述のように、我々はロボット工学の分野に制約プログラミングを取り入れることにより、ユーザの入力した任意のロボット構造に対応可能な汎用の機構学、及び静力学のプログラムを記述することに成功した。これらのプログラムを用いた場合の基本構造の設計過程を示したものが図3の(b)である。これによれば、簡単な基本構造パラメータリストの入力のみで、これと1対1に対応する基本

構造を持つロボットの機構学、及び静力学の制約式を自動生成することができる。制約式を生成することにより、制約プログラミングの計算の双方向性から、そのロボットに対するあらゆる問題の解析が可能となる。言い換えれば、生成された制約式を、指定された基本構造を持つロボット解析のプログラムそのものとみなすことができる。すなわち我々の開発したプログラムは、ユーザ指定の基本構造を持つ任意のロボットの解析プログラムを生成する汎用の変換プログラムと解釈することができる。これを利用することにより、繰り返し行われる基本構造の変更に対してもプログラムに入力するパラメータリストの変更のみで対応が可能となるので、設計効率の大幅な向上が見込まれる。図3の (a)、(b) の点線内の部分を比較すれば、われわれのアプローチにより設計者の負担が大幅に軽減されることがわかる。

5 設計支援システムの構築

5.1 構築システムの機能

既に述べたように、我々の作成したプログラムの最大の特徴は、問い合わせのみの変更により、任意のロボット基本構造に対応できることにある。したがって、図2のような設計過程において、最大限に貢献できるものは、基本構造の設計支援であり、これを念頭に置いて、構築システムの具体的な機能設計を行った。

上述のような主旨により、本システムにおける設計対象は、ロボットの基本構造である。本システムでは、まずこれに対応する基本構造パラメータリストの入力による任意のロボット構造の設定、これに対する制約式の自動生成により支援を行う。これにより、順変換、逆変換問題等のシミュレーションを行うことが可能となる。また、設計した基本構造の評価関数に関しては、既に掲げた可操作度というパラメータを算出することで支援を行う。これにより、特異点の解析の支援も可能である。また、静力学のプログラムで算出される各関節軸のトルクも、外力だけでなく基本構造に大きく依存するので、これも評価関数となる。これにより、全モータパワー、さらにはコストの見積もりが可能である。なお、3.1節において掲げた他の評価関数に関しては、現段階では定量的に表現することが困難であるので今回は対象外としたが、後に追加することも

可能である。

これらをまとめると我々の構築するロボット機構設計支援システム上の機能項目は以下の通りである。

1. 任意基本構造ロボットに対する機構学、静力学の制約式の自動生成
2. ロボット機構学による順変換、逆変換問題の求解
3. ロボットの機構学、静力学による各関節軸トルクの算出
4. ロボット機構学による可操作度の算出

なお、本システムは、ロボット構造設計を行う設計者の負担軽減、すなわち設計効率の向上が目的であり、計算機による自動設計を目指すものではない。また、本システムは、制約プログラミングをロボット工学に適用した最初のプロトタイプシステムであり、その根本的な機能に重点を置いたため、ロボットのアームの重量、及び速度、加速度等の動きを含めた力学(動力学)については無視するものとした。

5.2 システムモジュール構成

上記の機能を満たすために、開発した主なプログラムは以下の通りである。これらのプログラムは全て並列制約論理型言語 GDCC により記述されている。本システムの代数制約評価系は、ブフバーガ・アルゴリズム [Buch87] を適用しており、非線形の方程式を扱うことができる。

1. 機構学プログラム (Kinematics)
2. 静力学プログラム (Statics)
3. 行列式計算プログラム (Determinant)
4. 実根求解プログラム (find)

1, 2に関しては概ね既に述べた通りであるが、2の場合には、当初のものから、ロボットの操作性を決定する行列(ヤコビ行列と呼ばれる)を算出するための機能拡張をおこなった。3は、これに関連して開発したものであり、ヤコ

ど行列から、その行列式を算出するプログラムである。この行列式が、既に述べた可操作度と呼ばれる評価関数である。4は、CAL及びGDCC上の代数評価系に組み込まれたもので、1変数の高次方程式から全ての実根を求める機能を持つ。これは、ロボット機構学による逆変換問題の求解から必要性が生じたものである。通常ロボットの逆変換問題を解いていくと、多くの場合、最終的には1変数の2次方程式が生じるが、従来はプログラムの出力する最終的な解が複素数領域における解(グレブナ基底)[Buch 87]であったため、そのような場合には実数解を得ることはできなかった。今回、実根求解プログラムにより算出された1変数高次方程式の実数解をグレブナ基底に次々に代入することにより、実数領域における全解探索を行うことが可能となった。

上記の4つのプログラムを組み合わせて使うことにより、先に設定したシステム機能が実現され、ロボットの設計過程における基本構造設計の中核部分の支援が可能となる。この中で、基本構造設計における機構学、静力学の制約式の自動生成は、従来机上で手計算で行われていた手計算やプログラム作成の手間を完全に省くもので、これにより設計効率の大幅な向上が期待できる。また、本システムでは、機構学や静力学だけでなく、他の制約を段階的に追加していくことも可能である。例えば、予め作業条件として与えられるハンドの同一平面上の動きや外力による拘束等、物理的拘束の追加およびこれを含んだ結果(関係式)の算出も可能である。これにより、重要パラメーター同士の関係が明らかになる等、制約プログラミングの機能が最大限に発揮できる。特に専門家は、大抵の場合、具体的数値による結果よりも、できるだけ式の形で得られたものを好むので、このようなアプローチは有効である。

5.3 システム使用例

本システムを適用して、ロボットの構造設定と評価を、制約の段階的な追加により、対話的に行った例を以下に示す。

この例では、図5に示すような3関節を持つような基本構造のロボットを設定したときのモーターシャフトのトルク、および可操作度の解析をおこなう。作業内容としては、このロボットにペンを持たせて平面上に図形を書かせることを想定する。

このときの作業条件により与えられる制約としては

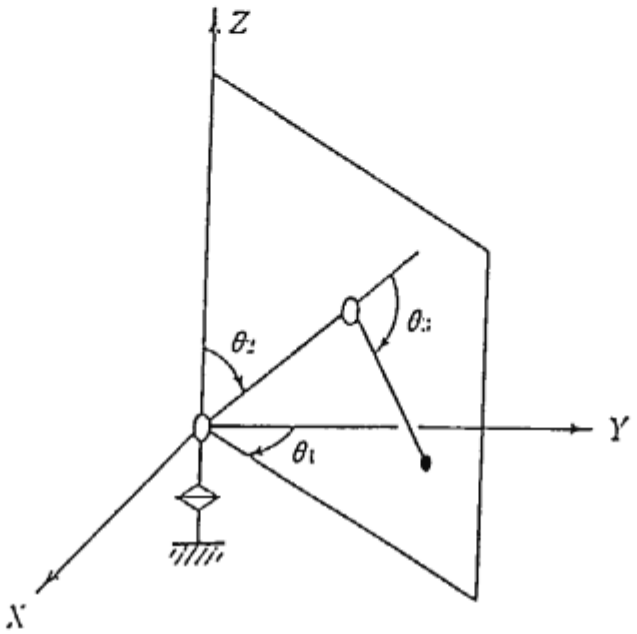

Figure 5: 3 自由度ロボットの概略図

(a) ペンがXY平面上を動くこと 及び

(b) 外力間の制約(XY方向の力は、Z軸方向の力に摩擦係数を乗じたもの)

が、追加入力される。このような条件のもとでの各関節軸に作用するトルク、及び可操作度の値の変化を解析することが目的である。以下、順を追って処理を行った例を示す。なお、一度システムに入力された制約式は、ユーザが、これをクリアするコマンドを入力しない限り、システム内に保存されている。

1. 順機構学により位置姿勢パラメータと基本構造パラメータとの関係式を求める。

$$
\begin{aligned}
&kinematics([[cos3, sin3, 0, 0, l2, 1, 0, 0],\\
&\qquad [cos2, sin2, 0, 0, 0, 1, 0, 0],\\
&\qquad [cos1, sin1, 0, 0, l1, 0, 0, 1]],\\
&\qquad 0, 0, l3, 1, 0, 0, 0, 1, 0,\\
&\qquad px, py, pz, ax, ay, az, cx, cy, cz).
\end{aligned}
$$

$$
\begin{aligned}
pz &= -sin3 * sin2 * l3 + l3 * cos3 * cos2\\
&\quad +l2 * cos2 + l1\\
py &= -sin3 * l3 * cos2 * cos1\\
&\quad -sin2 * l3 * cos3 * cos1 - sin2 * l2 * cos1\\
px &= sin3 * sin1 * l3 * cos2\\
&\quad +sin2 * sin1 * l3 * cos3 + sin2 * sin1 * l2\\
cz &= sin3 * cos2 + sin2 * cos3\\
cy &= -sin3 * sin2 * cos1 + cos3 * cos2 * cos1\\
cx &= sin3 * sin2 * sin1 - sin1 * cos3 * cos2\\
az &= 0\\
ay &= sin1\\
ax &= cos1
\end{aligned}
$$

$$
\begin{aligned}
sin3^2 &= -cos3^2 + 1 \\
sin2^2 &= -cos2^2 + 1 \\
sin1^2 &= -cos1^2 + 1
\end{aligned}
$$

問い合わせ中の最初の変数であるリストの内容が図5のロボットの基本構造を表現している。$l1, l2, l3$ は、3つのアームの長さを示す。

2. 順機構学＋静力学により、各関節軸のトルク、および可操作度を求める一般式を算出する。

$$
\begin{aligned}
&statics([[qx1, qy1, qz1], [qx2, qy2, qz2], [qx3, qy3, qz3]], \\
&\quad [cos1, sin1, 0, 0, l1, 0, 0, 1], \\
&\quad [cos2, sin2, 0, 0, 0, 1, 0, 0], \\
&\quad [cos3, sin3, 0, 0, l2, 1, 0, 0]], \\
&\quad [t1, t2, t3], 0, 0, l3, fx, fy, fz, Jacobi), \; determinant(Jacobi.det).
\end{aligned}
$$

$$
\begin{aligned}
t3 =& -sin3 * sin2 * sin1 * l3 * fx \\
&+ sin3 * sin2 * l3 * cos1 * fy + sin1 * l3 * cos3 * cos2 * fx \\
&- l3 * cos3 * cos2 * cos1 * fy - sin3 * l3 * cos2 * fz - sin2 * l3 * cos3 * fz
\end{aligned}
$$

$$
\begin{aligned}
t2 =& -sin3 * sin2 * sin1 * l3 * fx + sin3 * sin2 * l3 * cos1 * fy \\
&+ sin1 * l3 * cos3 * cos2 * fx - l3 * cos3 * cos2 * cos1 * fy \\
&- sin3 * l3 * cos2 * fz - sin2 * l3 * cos3 * fz + sin1 * l2 * cos2 * fx \\
&- l2 * cos2 * cos1 * fy - sin2 * l2 * fz
\end{aligned}
$$

$$
\begin{aligned}
t1 =& sin3 * sin1 * l3 * cos2 * fy + sin3 * l3 * cos2 * cos1 * fx \\
&+ sin2 * sin1 * l3 * cos3 * fy + sin2 * l3 * cos3 * cos1 * fx \\
&+ sin2 * sin1 * l2 * fy + sin2 * l2 * cos1 * fx
\end{aligned}
$$

$$
\begin{aligned}
det =& -sin3 * sin2 * l3^2 * l2 * cos3 + l3^2 * l2 * cos3^2 * cos2 \\
&- sin3 * sin2 * l3 * l2^2 - l3^2 * l2 * cos2
\end{aligned}
$$

$$pz = -sin3 * sin2 * l3 + l3 * cos3 * cos2 + l2 * cos2 + l1$$
$$py = -sin3 * l3 * cos2 * cos1 - sin2 * l3 * cos3 * cos1 - sin2 * l2 * cos1$$
$$px = sin3 * sin1 * l3 * cos2 + sin2 * sin1 * l3 * cos3 + sin2 * sin1 * l2$$

ここで、fx、fy、fz、は手先に作用する外力の各成分、t1、t2、t3、は各関節軸のトルク、detは可操作度の値を示す変数である。これらの式より、可操作度は、基本構造により、トルクは基本構造と外力により決定されることがわかる。なお、それ以外の変数に関する出力は省略した。以下では、着目している変数に関する出力のみを示すこにとする。

3. グリップのXY平面上の動き(pz = 0)という制約を加える。
   ?- $alg\#pz = 0.$

$$t3 = -sin3 * l3 * cos2 * fz - sin2 * l3 * cos3 * fz$$
$$-sin1 * l2 * cos2 * fx + l2 * cos2 * cos1 * fy$$

$$t2 = -sin3 * l3 * cos2 * fz - sin2 * l3 * cos3 * fz - sin2 * l2 * fz$$

$$t1 = sin3 * sin1 * l3 * cos2 * fy + sin3 * l3 * cos2 * cos1 * fx$$
$$+sin2 * sin1 * l3 * cos3 * fy + sin2 * l3 * cos3 * cos1 * fx$$
$$+sin2 * sin1 * l2 * fy + sin2 * l2 * cos1 * fx$$

$$det = -2 * l3 * l2^2 * cos3 * cos2 - l3^2 * l2 * cos2 - l2^3 * cos2$$
$$sin3 * sin2 * l3 = l3 * cos3 * cos2 + l2 * cos2$$

この段階で$t2$、及びdetの式が簡略化されている。

4. 外力による制約($fx^2 + fy^2 = mu^2 * fz^2$)を加える($mu$は摩擦係数)。
   この制約式は、ロボットが平面から受ける摩擦力に逆らってペンを動かすことを示す。これを三角関数を使って書き換えると、下記のような制約式となる。sin0.cos0によって表される角度θ_0は、XY平面に平行に働く力の方向を示す。

$?-alg\#fx = mu * fz * sin0, alg\#fy = mu * fz * cos0, alg\#cos0^2 + sin0^2 = 1.$

$$t3 = -sin3 * l3 * cos2 * fz - sin2 * l3 * cos3 * fz - sin1 * l2 * cos2 * sin0 * mu * fz + l2 * cos2 * cos1 * mu * cos0 * fz$$

$$t2 = -sin3 * l3 * cos2 * fz - sin2 * l3 * cos3 * fz - sin2 * l2 * fz$$

$$t1 = sin3 * sin1 * l3 * cos2 * mu * cos0 * fz + sin3 * l3 * cos2 * cos1 * sin0 * mu * fz + sin2 * sin1 * l3 * cos3 * mu * cos0 * fz + sin2 * l3 * cos3 * cos1 * sin0 * mu * fz + sin2 * sin1 * l2 * mu * cos0 * fz + sin2 * l2 * cos1 * sin0 * mu * fz$$

$$det = -2 * l3 * l2^2 * cos3 * cos2 - l3^2 * l2 * cos2 - l2^3 * cos2$$

$$sin3 * sin2 * l3 = l3 * cos3 * cos2 + l2 * cos2$$

5. 三角関数の加法定理による制約を加える。 $c23$、 $s23$、 $c01$、 $s01$ は各々、 $cos(\theta_2 + \theta_3), sin(\theta_2 + \theta_3), cos(\theta_0 + \theta_1), sin(\theta_0 + \theta_1)$ を示す。

$?-alg\#cos2 * cos3 - sin2 * sin3 = c23, alg\#cos2 * sin3 + cos3 * sin2 = s23, alg\#cos0 * cos1 - sin0 * sin1 = c01, alg\#cos0 * sin1 + cos1 * sin0 = s01.$

$$t3 = -l3 * fz * s23 - l3 * mu * fz * c23 * c01$$

$$t2 = -sin2 * l2 * fz - l3 * fz * s23$$

$$t1 = sin2 * l2 * mu * fz * s01 + l3 * mu * fz * s23 * s01$$

$$det = 2 * l3^2 * l2 * cos3 * c23 + l3^3 * c23 + l3 * l2^2 * c23$$

$$l2 * cos2 = -l3 * c23$$

これで、トルク、可操作度に関する全ての式が単純化されたので、これ以降は、設計者が手計算により解析を行う。

6. det、 t1、 t2、 t3 に関する式を整理する。

$$
\begin{aligned}
det &= 2 * l3^2 * l2 * cos3 * c23 + l3^3 * c23 + l3 * l2^2 * c23 \\
&= l3 * c23 * (l2^2 + l3^2 + 2 * l2 * l3 * cos3) \\
&= -l2 * cos2 * (l2^2 + l3^2 + 2 * l2 * l3 * cos3) \\
&\quad (l2 * cos2 = -l3 * c23 \text{ より})
\end{aligned}
$$

したがって、det = 0 (特異点) となるのは cos2 = c23 = 0 となるとき、すなわち

$$\theta_2 = 90(degree), \theta_3 = 0, 180(degree)$$

のときである。これは、ロボットの長さ l2、及び l3 のアームが一直線上になってXY平面に接触した状態、あるいは2つのアームが完全に折り重なった状態を示す。また、トルクに関しては、

$$
\begin{aligned}
t3 &= -l3 * fz * s23 - l3 * mu * fz * c23 * c01 \\
&= -l3 * fz * (s23 + mu * c01 * c23) \\
t2 &= -sin2 * l2 * fz - l3 * fz * s23 \\
&= -fz * (l2 * sin2 + l3 * sin23)
\end{aligned}
$$

$$
\begin{aligned}
t1 &= sin2 * l2 * mu * fz * s01 + l3 * mu * fz * s23 * s01 \\
&= mu * s01 * fz * (l2 * sin2 + l3 * sin23)
\end{aligned}
$$

となるので、t3、t2、t1 の絶対値の最大値は、それぞれ

$$
\begin{aligned}
t3max &= l3 * fz * sqrt(1 + mu^2) \\
t2max &= fz * (l2 + l3) \text{(特異点と同一の場合)} \\
t1max &= mu * fz * (l2 + l3) \text{ (同上)}
\end{aligned}
$$

となる。上記の結果から以下のようなことが言える。まず、特異点の解析結果より、特異点におけるロボットの姿勢が明らかになったので、設計者はこのような姿勢の近傍を避けるような作業を設定しなければならない。ただし、この場合は2通りの特異点の姿勢は、ロボットの作業領域の端点であるのでそれほ

ど考慮する必要はない。また、最大トルクとアームの長さ、外力との関係式が算出されたので、例えばアームの長さが決まっていれば、設計すべき各々の関節軸のモータパワーの見積もりができる。モータパワーが予め仕様として与えられている場合には、実際に各関節に作用するトルクがモータの能力を越えないように各アームの長さ等のパラメータ設計を行う必要がある。

このように基本構造の設定により、機構学、及び静力学の一般式を生成した後、作業条件による制約式を段階的に入力していくことにより、重要パラメーター同士の関係が次第に単純化され、可操作度からの特異点の解析、トルクの最大値の解析が容易になる。

6 まとめ

我々は今回、ロボット工学の世界に制約プログラミングというパラダイムを導入することにより、ロボット機構を含めた総括的なロボット設計を、部分的ではあるが、支援できる汎用的なシステムの構築を試みた。本システムは、従来のロボット工学ではほとんど試みられなかった新しいアプローチとして専門家からも比較的高い評価を受けている。今後は、実際に使ってもらうことで、機能、使い勝手等の評価を受け、システム全体にフィードバックをかけていくことが必要である。さらに、これを通じ CAL や GDCC に必要となる新たな機能を見出し、機能向上を働きかけていく予定である。

References

[信学会 85] 電子情報通信学会: ロボット工学とその応用, コロナ社, 1985.

[牧野 75] 牧野: 自動機械機構学, 日刊工業新聞社, 1975.

[遠山 89] 遠山: 機械系のためのロボティクス, 総合電子出版社, 1989

[高野 86] 高野: ロボット機構系の設計, 日本ロボット学会誌 4 巻 4 号, 1986

[吉川 84] 吉川: ロボットアームの可操作度, 日本ロボット学会誌 2 巻 1 号, 1984

[吉田 86] 吉田: ロボットシミュレーションシステムと設計, 日本ロボット学会誌 4 巻 4 号, 1986

[Buch87] B. Buchberger. : Applications of Gröbner Bases in Non-linear Computational Geometry, Lecture Notes in Computer Science 296, Trends in Computer Algebra, 1987