

## 産業用ロボットの経路生成における干渉計算の高速化

電気通信大学 ○東 朗, 電気通信大学大学院 ◎森重功一

## 要旨

本研究は、産業用ロボットの経路生成における干渉チェック機能の高速化を目的としている。ロボットの経路生成においては、各作業点に対して干渉のない姿勢を決定する必要があり長時間を要する。本報では、干渉判定処理の見直し、モデルデータの構造やモデル間のチェック順序の最適化を行うことにより計算時間を短縮した。そして、作成した干渉チェック機能をCAMへ実装して成果を確認することが出来た。

## 1. 序論

産業用ロボットを運用する際、必ず考慮しなければならないことに干渉回避がある。干渉とは、作業中のロボットが周囲の人や物、そしてロボット自身と衝突することである。しかし、ロボットは作業に対して無数に姿勢を取り得るため、干渉の確認には長時間を要する。干渉の確認を実用的な時間内で行うことが出来れば、干渉を考慮した最適な経路を作成出来ると考えられる。

本研究では、ロボットの経路生成における干渉計算の高速化を目的としている。

## 2. 干渉チェック

本研究では、物体を三角形ポリゴンの集合モデルとして表現した STL データを利用して干渉チェックを行っている。チェックは干渉の恐れがある全てのモデルの組に対して実行され、モデル間の干渉の有無を判定している。干渉判定は、モデル間の全ての線分とポリゴン面との交差の有無を調べることで行っている。

## 2.1 モデルデータの構造最適化

本研究では、交差判定においてモデルを構成する頂点や線分の情報を利用する。しかし、STL データはモデルを構成する全ポリゴンの頂点座標と単位法線ベクトルの情報のみで構成されている。そのため、図1のように頂点と線分の情報を追加した。

初めに、STL データからモデルを構成する  $m$  個の頂点のリストを作成する。次に、頂点の組み合わせで構成される  $n$  個の線分のリストを作成した。

加えて、線分とポリゴンを含む平面の大まかな位置関係を知るために、各線分の長さ  $length$  と、各頂点を端点に持つ線分番号の情報をそれぞれリストへ追加した。

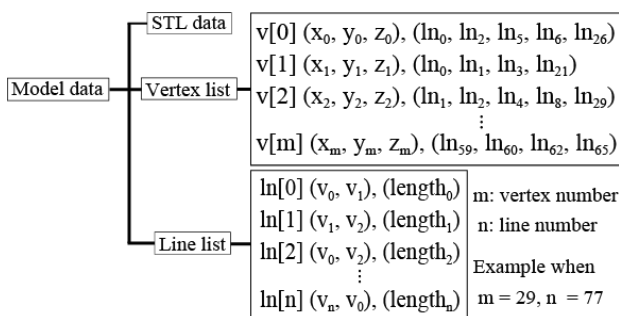


Fig.1 How to make model data form STL data.

## 3. 干渉チェックの高速化

## 3.1 交差判定の高速化

交差判定では、任意の線分とポリゴンの全ての組み合わせにおいて交差の有無を判定する。

まず初めに、線分の端点がポリゴンの点  $O$  に対してどのような

位置にあるか調べる。図2は、あるモデルを構成する任意の線分  $PP'$  が、別のモデルを構成する任意のポリゴン  $OAB$  と点  $X$  で交差する様子を示している。点  $O$  から各点  $A, B, P, P', X$  への位置ベクトルをそれぞれ、 $a, b, p, p', x$  としている。また、 $n$  はポリゴンの単位法線ベクトルを示している。

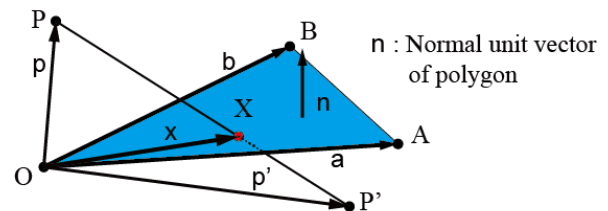


Fig. 2 Logic of Interference in polygon.

このとき、点  $P$  の位置は実数  $u, v, w$  を用いて、

$$p = ua + vb + wn \quad (1)$$

と表せる。 $w$  の大きさ  $|w|$  は、点  $P$  からポリゴン平面へ降ろした垂線の長さに相当し、余弦と内積の定義から  $w = p \cdot n$  と求められる。このとき  $|w|$  が線分  $PP'$  の長さより大きければ、線分とポリゴン平面は交差しない。

図3に示すように、図2のポリゴン  $OAB$  上にある線分  $PP'$  の他に端点  $P$  を共有する線分  $Line[0-5]$  についても、それぞれの線分の長さ  $length$  と  $|w|$  を比較することにより、ポリゴン平面と線分の交差を判定することが出来る。

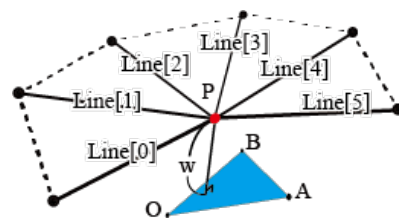


Fig. 3 Narrowing down lines for decreasing process.

$|w|$  が線分  $PP'$  の長さより小さい場合は、点  $P'$  についても同様の処理を行う。計算した  $w$  に相当するパラメータを  $w'$  とする。 $w$  と  $w'$  が、いずれも正または負であれば、線分とポリゴン平面は交差しないと判定できる。

以上に該当しない場合、線分  $PP'$  はポリゴン平面と交点を持つことになる。このとき点  $O$  に対する点  $X$  の位置は、実数  $U, V, t$  を用いて、

$$x = Ua + Vb = t(p' - p) \quad (2)$$

と表せる、(2)式を満たす実数  $U, V$  を計算して、点  $X$  の位置を求める。そのために、(1)式を満たす実数  $u, v, w$  を計算することになるが、このとき(1)式の三元連立方程式を、 $w$  が既知であることから二元連立方程式として扱うことによって、処理を高速化している。同様に、 $p'$  についても  $u, v$  に相当するパラメータを求め、結果を(2)式へ代入して  $U, V$  を求めた。

### 3.2 干渉判定順序の最適化

干渉チェックは、干渉の恐れがある全てのモデルの組に対し実行される。ロボットの姿勢ごとに繰り返し行われ、干渉を確認した姿勢を経路候補から除外する。干渉が検出され次第処理を打ち切るため、干渉は早期に検出されることが望まれる。本研究では、干渉チェックの対象とする処理の順序を以下のように決定した。

初めに、一度干渉が発生したモデルの組はその後しばらく干渉状態が続くと考え、干渉を確認した組については次回から最初にチェックを行うことで早期の検出を可能とした。

次に、ロボットは手先工具をワークへ近づけることで作業を行うため、ワークと手先周辺のモデルには干渉が起こりやすいと考え、これらのモデル同士の組を優先してチェックするものとした。さらに、交差判定の試行数が少ない組をさらに優先することとした。残りの組については、短時間でより広い領域の判定をするために、干渉面のポリゴンの平均面積が大きいものを優先した。

前回のチェックにおいて干渉が確認された組をチェックしても干渉が確認されなかった場合は、次回から規定の順序に戻りチェックを行う。

### 3.3 モデルの簡素化による交差判定の高速化

交差判定においては、モデルを単純な形状とみなすことによって処理を高速化出来る。

例として、対象モデルとして図 4(a)に示すような円錐を考える。このとき、交差がある可能性が高いと考えられる場合は、図 4(b)のようにモデルに内包される多角形を用いる。交差がない可能性が高いと考えられる場合は、モデルを図 4(c)に示すようにモデルを内包するブロックを用いる。

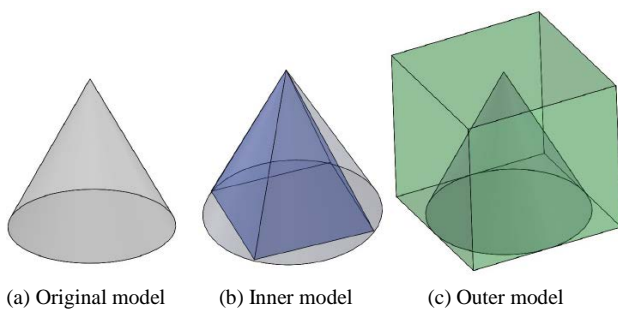


Fig. 4 Replacement of original model to simple model.

本研究では、手先工具とロボットの土台周辺は交差する可能性が低いと考えている。また、干渉チェックにおいて一度干渉が検出された組は次回のチェックにおいても交差する可能性が高いと考えている。これらの組の交差判定を行う場合のみ、通常の交差判定を図 4 のように単純化したモデルに置き換えて処理を行った。

## 4. シミュレーション実験

本研究において提案した手法により、干渉計算の高速化が出来ることを確かめるために、開発した干渉チェックプログラムを利

用して実際に経路データを生成し、シミュレーションによって干渉が回避出来ているか確認した。同時に、提案した各手法を段階的にプログラムへ反映させ、各手法によりどれだけ干渉計算を短縮出来たか確認した。

実験では、ワークとして図 5 に示すようなスポーツカーの窓ガラスを想定した薄板を用いた。

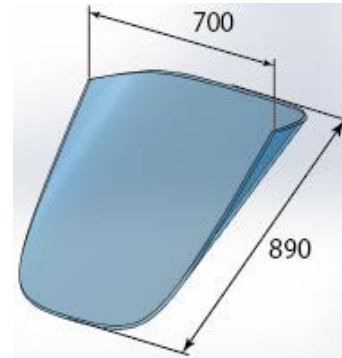


Fig. 5 Model for experiment.

ロボットには、このワークの内側の縁全周に接着剤を塗布するシーリング作業を行わせた。図 6 にシミュレーションの様子を示す。

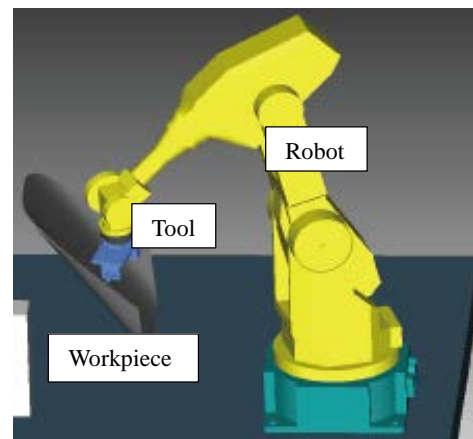


Fig. 6 Situation of simulation.

本実験で生成した経路データは、作業点 2791 点により構成されている。各作業点において 360 種類の姿勢を候補として設定し、全ての姿勢に対して干渉チェックを行った。

初めに干渉チェックプログラムを利用せず、干渉を考慮しない経路を生成した。シミュレーションにより動作を確かめたところ、干渉が確認された。

次に、開発した干渉チェックプログラムを利用して干渉を考慮した経路を生成した。シミュレーションより動作を確認した結果、干渉のない経路が生成出来ていることが確認された。

経路生成に要した計算時間は、当初、干渉計算の高速化を行わない状態で 3 時間程度であったのに対して、最短で 15 分程度まで短縮された。

## 5. 結論

本研究で作成した干渉チェックプログラムにより、産業用ロボットの経路生成における干渉計算を高速化することが出来た。また、干渉チェックの高速化によって、実用的な計算時間内で干渉を考慮した経路生成を行えることが確認出来た。