計測自動制御学会東北支部 第179回研究集会 (1998.12.14)
資料番号 179-5

Bスプラインによる移動ロボットの動的障害物回避軌道の設計

Dynamic Path Planning Based on B-spline Curve Considering Obstacle Avoidance for Mobile Robots

○横田 尚大, 王 志東, 高橋 隆行, 中野 栄二

○Takahiro Yokota, Zhi-Dong Wang, Takayuki Takahashi, Eiji Nakano

東北大学

Tohoku University

キーワード： 自律移動ロボット (autonomous vehicle robot), 軌道生成 (trajectory generation), 障害物回避 (obstacle avoidance), スプライン曲線 (spline curve)

連絡先： 〒980-8579 仙台市青葉区荒巻字青葉01 東北大学大学院 情報科学研究科 中野研究室
横田 尚大, Tel.: (022)217-7025, Fax.: (022)217-7023,
E-mail: takahiro@robotics.is.tohoku.ac.jp

1. はじめに

移動ロボットが実環境において走行することを想定すると，ロボットの持つ地図情報にはない，椅子などの静的な障害物，あるいは人などの動的な障害物が存在することが多々ある．このような環境に移動ロボットが適応するためには，軌道の空間的および時間的な局所変更性が要求される．このため，ロボットの軌道計画を，環境モデルをもとにして，目的地へ至る経路を決定する大域的な計画と，計画された経路から実際に移動機構を制御するための軌道データを作成する局所的な軌道生成とに分けて考える．ここで，経路とはロボットが空間内で通過する道筋を表し，軌道とはロボットの位置，速度，加速度の時間的な履歴を意味する．つまり，経路設計においては時間的な拘束は含まれない．

本研究の目的は，空間的および時間的に局所変更性を有する軌道生成法を提案し，実環境におけるリアルタイム障害物回避軌道の設計法を提案することである．ただし，ロボットはvision等により，ある範囲内でなら障害物全体の大まかな位置と形を認識できるものとし，その範囲は障害物回避に十分な距離とする．想定する状況は，ロボットは地図情報をもっているが，地図情報にはない静的な障害物が 存在する2次元平面を走行するものとする．

2. 経路および軌道計画

まず，障害物が存在しない環境下における経路および軌道の生成法を示す．

2.1 経路・軌道に要求される曲線の性質

経路および軌道に要求される曲線の性質として，(1)局所変更が可能，(2)曲線長に関する微分可能性が任意に指定できる，(3)演算が高速にで

きることなどが挙げられる．ユニフォームなBスプライン曲線はこのような要求を満たすため，この曲線を用いて経路および軌道を生成する．

2.2 ユニフォームなBスプライン曲線

ユニフォームなBスプライン曲線は，与えられる制御点を $\{Q_0, Q_1, \cdots, Q_n\}$，曲線パラメータを t とすると，

$$P(t) = \sum_{j=0}^{n} Q_j B_{j,k}(t) \quad (t \in [0, n-k+1])$$

として定義できる．ここで基底関数 $B_{j,k}(t)$ は k 次の多項式であり次式で定義される．

$k=1$の場合

$$B_{j,1}(t) = \begin{cases} 1 & (t_j \le t < t_{j+1}) \\ 0 & (\text{上記以外}) \end{cases}$$

$k>1$の場合

$$B_{j,k}(t) = \frac{t-t_j}{t_{j+k}-t_j} B_{j,k-1}(t) + \frac{t_{j+k+1}-t}{t_{j+k+1}-t_{j+1}} B_{j+1,k-1}(t)$$

なお，t_j はノットであり，ノットベクトル T は，

$$T = [t_0, t_1, \cdots, t_{n+k+1}] = [-k, -(k-1), \cdots, n+1]$$

である．パラメータ t の変化によって位置ベクトル P は曲線を描くが，t が隣接するノット間を変化するときに生成される曲線の部分は曲線セグメント，ノットに対応する曲線上の点はノット点と呼ばれる．t が $0 \le t \le n-k+1$ の範囲で変化するとき，P は連続する $n-k+1$ 個の曲線セグメントから構成される．この曲線は $k-1$ 次の導関数まで連続となり，連続する $k+1$ 個の制御点のみにより定義され，連続する $k+1$ 個の制御点の凸包領域に存在する．このため，制御点を変更した場合，その影響は曲線全体には及ばず，局所的に曲線を変更することができる．以下では曲率までの連続を考慮し，3次のスプライン曲線を対象とする．このとき，各曲線セグメントは4個の連続する制御点 $\{Q_{j-1}, Q_j, Q_{j+1}, Q_{j+2}\}$ により決定される．図1に3次のBスプライン曲線における制御点 Q_j とノット点 P_j の関係を示す．

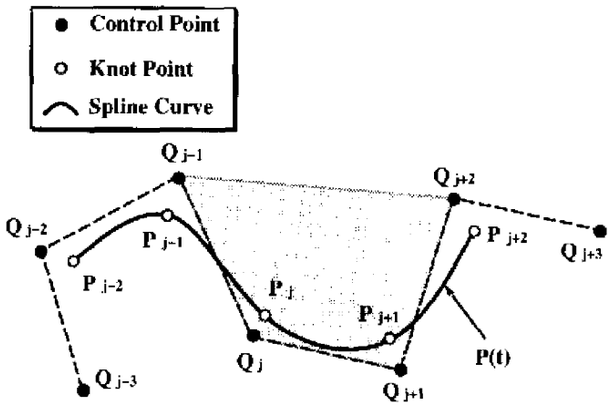

図1 Relation between Q_j, P_j and B-spline curve

2.3 経路設計

経路設計では，Globalな制御点でロボットが通過する道筋を大域的に計画する．ここで，時間的な拘束は含まないことを注意しておく．ロボットの経路は車体の位置および姿勢の空間的変化であるので，それを $P(s) = [x(s), y(s), \theta(s)]^T$ のように表すと，$P(s)$ は $n+1$ 個の制御点 Q_j により，

$$P(s) = \sum_{j=0}^{n} Q_j B_{j,k}(s) \quad (s \in [0, n-k+1])$$

のように表せる．ここで，s は曲線パラメータである．

2.4 Localな制御点の算出法

大域的に計画された経路から実際に移動機構を制御するためのLocalな制御点を算出する方法について述べる．次の節で述べる単位運動により，ロボットが経路上を単位時間で移動する距離毎にLocalな制御点を打っていく必要がある．これには，時間に対する速度グラフをもとに，解析的にLocalな制御点を算出する方法が知られている[1]．しかし，この方法では演算量が多くなってしまうのでリアルタイム制御には適さない．そこで，次のようにしてLocalな制御点を求める．すなわち，Localな制御点をGlobalな制御点間の曲線パラメー

タ s を $[d/D]$ 分割して求められる曲線上の点とするものである．ここで，d はGlobalな制御点間のユークリッド距離，D は単位時間でロボットが移動する距離である．図 2 にGlobalな制御点とLocalな制御点の関係を示す．

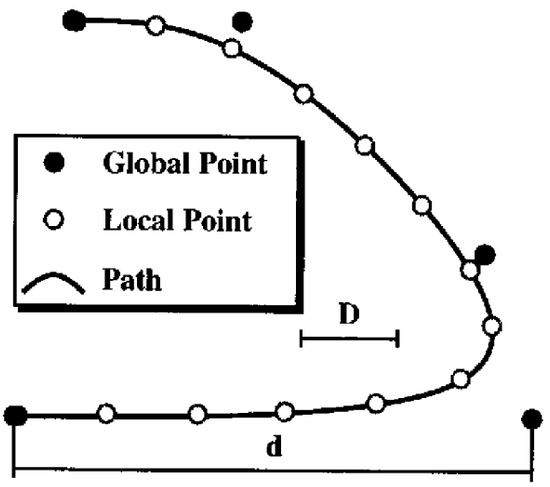

図 2　Relation between Global and Local Points

2.5　単位運動による軌道生成

高山ら[2]によって提案された単位運動による移動ロボットの軌道生成の概要について述べる．この手法において，ロボットの位置および姿勢の軌道は，正規化された一様なB-スプライン関数を重み付けし，それを時間軸についてずらしながら重ね合わせることによって生成される．軌道はロボットの位置および姿勢の時間的変化であるので，それを $p(t) = [x(t), y(t), \theta(t)]^T$ のように表すと，$p(t)$ は以下のように生成される．

$$p(t) = \sum_{j=i-k}^{i} q_{j+k+1} B_k(\omega(t - t_j)) \quad (t \in [t_i, t_{i+1}])$$

ただし，$q_i = [x_i, y_i, \theta_i]$ は，Localな制御点である．$B_k(t)$ は，基底関数 $B_{j,k}(t)$ が曲線パラメータ t に対して水平方向に平行移動した関係にあることに着目し操作された式で，以下のように定義される．

$$B_k(t) = \begin{cases} N_{k-j,k}(t-j) \\ \qquad (j \leqq t < j+1, j = 0, 1, \cdots, k) \\ 0 \qquad (t < 0, k+1 \leqq t) \end{cases}$$

ここで，関数 $N_{0,k}(t), \cdots, N_{k,k}(t)$ は以下のように再帰的に定義される関数である．

$$\begin{cases} N_{0,0}(t) = 1 \\ N_{0,k}(t) = \dfrac{1-t}{k} N_{0,k-1}(t) \\ N_{j,k}(t) = \dfrac{k-j+t}{k} N_{j-1,k-1}(t) \\ \qquad\qquad + \dfrac{1+j-t}{k} N_{j,k-1}(t) \\ \qquad\qquad\qquad (j = 1, 2, \cdots, k-1) \\ N_{k,k}(t) = \dfrac{t}{k} N_{k-1,k-1}(t) \end{cases}$$

ω は $B_k(t)$ における変数 t を縮尺するための正定数スカラーであり，ω の逆数を単位時間 T_{spc} と定義すると T_{spc} はノット点間の移動時間を表すパラメータとなる．

また，境界条件として，

$$\begin{cases} q_1 & = \cdots & = q_{k-1} & = q_k \\ q_{N-k+1} & = \cdots & = q_{N-1} & = q_N \end{cases}$$

のように，移動開始時および停止時に経路点を k 個重ねることにより，ロボットは滑らかに運動を開始し，滑らかに運動を停止することができる．

単位運動による軌道生成の具体例を示す．図 3 はデカルト座標系において設計された経路を表している．丸印はLocalな制御点を表しており，始点と終点においては多重頂点になっている．図 4 は生成された軌道(x 方向)を示しており，軌道は時間軸に対して，単位運動を重ね合わせて生成されていることがわかる．

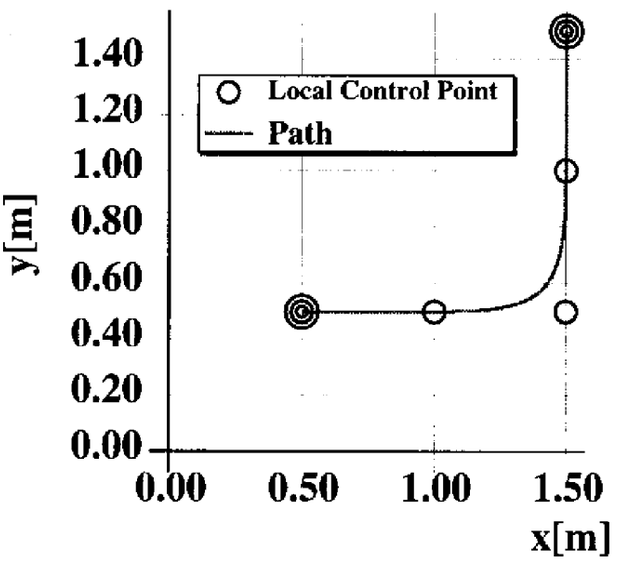

図 3　Path generated by Unit Motions

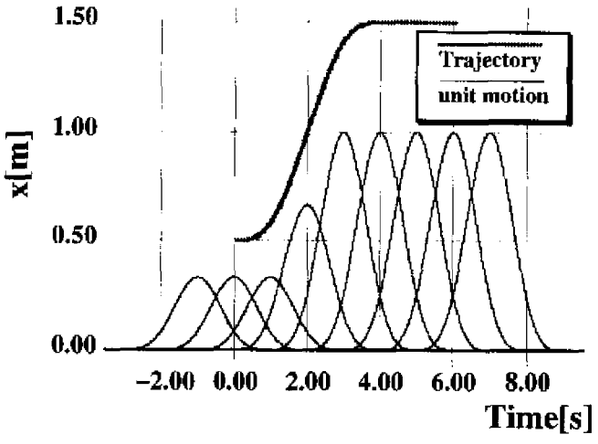

図 4　Trajectory generated by Unit Motions

3.　障害物回避経路の設計

ロボットが走行中，静的な障害物を検知した場合の障害物回避経路の設計法を述べる．ロボットはあらかじめ設計した経路が障害物に衝突していないか判断し，もし衝突しているなら動的に障害物との衝突を回避する経路を再設計する必要がある．以下，障害物の表現法，衝突判定法，衝突回避法について順次説明していく．

3.1　障害物の表現法

ロボットを点とみなすことができれば，障害物回避経路の設計は容易になる．そこで，障害物に対し以下のような操作を行い，最終的に得られたものを障害物とみなし，障害物の頂点の座標を求めておく．操作の過程を図 5 に示す．

1．障害物を凸包で表現する．
2．ロボットを円筒近似し，1で得られた障害物にその円筒の半径分だけ障害物を拡大する．
3．2で得られた障害物を凸多角形で近似する．

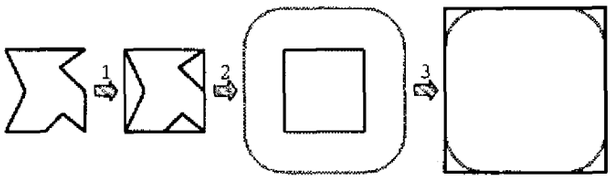

図 5　Expression of Obstacle

3.2　衝突物との衝突判定

頂点の座標がわかっている障害物に対し，設計した経路が障害物に衝突していないか判定する必要がある．衝突判定を高速かつ任意の精度で行う方法を述べる．この方法は，Bスプライン曲線セグメントをBezier曲線の制御点で表し，Bezier曲線を細分割していくことにより実現される．Bezier曲線は制御点に対し凸包性をもち任意に細分割できる曲線である．

3.2.1　BezierによるBスプライン曲線の表現法

3次のBスプライン曲線の制御点 $\{Q_{j-1}, Q_j, Q_{j+1}, Q_{j+2}\}$ により定義されているBスプライン曲線セグメントを，Bezier曲線の制御点 $\{Q'_0, Q'_1, Q'_2, Q'_3\}$ で表現すると，次式が成り立つ．

$$\begin{cases} Q'_0 &= \frac{1}{6}Q_{j-1} + \frac{2}{3}Q_j + \frac{1}{6}Q_{j+1} \\ Q'_1 &= \frac{2}{3}Q_j + \frac{1}{3}Q_{j+1} \\ Q'_2 &= \frac{1}{3}Q_j + \frac{2}{3}Q_{j+1} \\ Q'_3 &= \frac{1}{6}Q_j + \frac{2}{3}Q_{j+1} + \frac{1}{6}Q_{j+2} \end{cases}$$

幾何的関係を図 6 に示す．3辺 $\overline{Q_{j-1}Q_j}$，$\overline{Q_jQ_{j+1}}$，

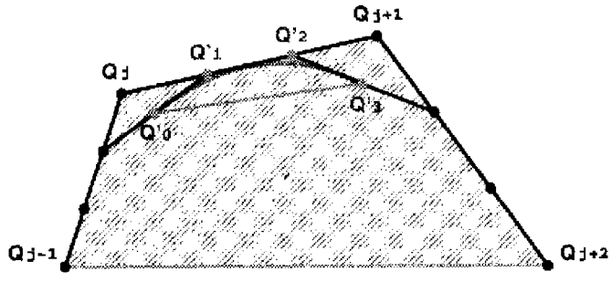

図 6　Relation between B-spline's Control Points and Bezier's Control Points

$\overline{Q_{j+1}Q_{j+2}}$ をそれぞれ 3 等分すると，辺 $\overline{Q_jQ_{j+1}}$ 上で Q_j に近い等分点が Q'_1，また Q_{j+1} に近い方の等分点が Q'_2 となる．辺 $\overline{Q_{j-1}Q_j}$ 上で Q_j に近い等分点と Q'_1 とを直線で結び，その線分の中点は曲線セグメントの始点で Q'_0 となり，また辺 $\overline{Q_{j+1}Q_{j+2}}$ 上で Q_{j+1} に近い方の等分点と Q'_2 を直線で結び，その線分の中点は曲線セグメントの終点で Q'_3 となる．このように，Bスプライン曲線

セグメントをBezier曲線の制御点で表現すると，曲線の存在する領域をより限定することができる．また，Bezier曲線を細分割することにより，さらにその領域を小さくすることができる．

3.2.2 衝突判定アルゴリズム

衝突判定は以下のアルゴリズムにより行う．

1．軌道データをもとに移動しているロボットが障害物を検知する．
2．Localな制御点を用いてBスプラインの凸包性を利用した衝突判定をする（図 7(a)）．衝突していないなら 7 へ進む．
3．Bezierの凸包性を利用した衝突判定をする（図 7(b)）．衝突していないなら 7 へ進む．
4．凸包領域の面積が閾値以下なら 6 へ進む．
5．Bezierを細分割し凸包性を利用した衝突判定をする（図 7(c)）．衝突しているなら 4 へ戻る．衝突していないなら 7 へ進む．
6．衝突していると判断し，障害物と衝突した座標を記憶しておく．
7．衝突していないと判断する．

3.3 衝突回避アルゴリズム

経路と障害物が衝突している場合，障害物を回避するような経路を再設計する必要がある．衝突回避には，図 8 に示すように，衝突領域に対し，制御点のシフトやあらかじめ用意している形状(円・楕円)のあてはめなどが有効であると考えられる．しかし，これらの方法は，過剰回避が問題となると思われる．そこで，Bスプラインの逆変換を用いた障害物の最小回避経路の設計法について述べる．

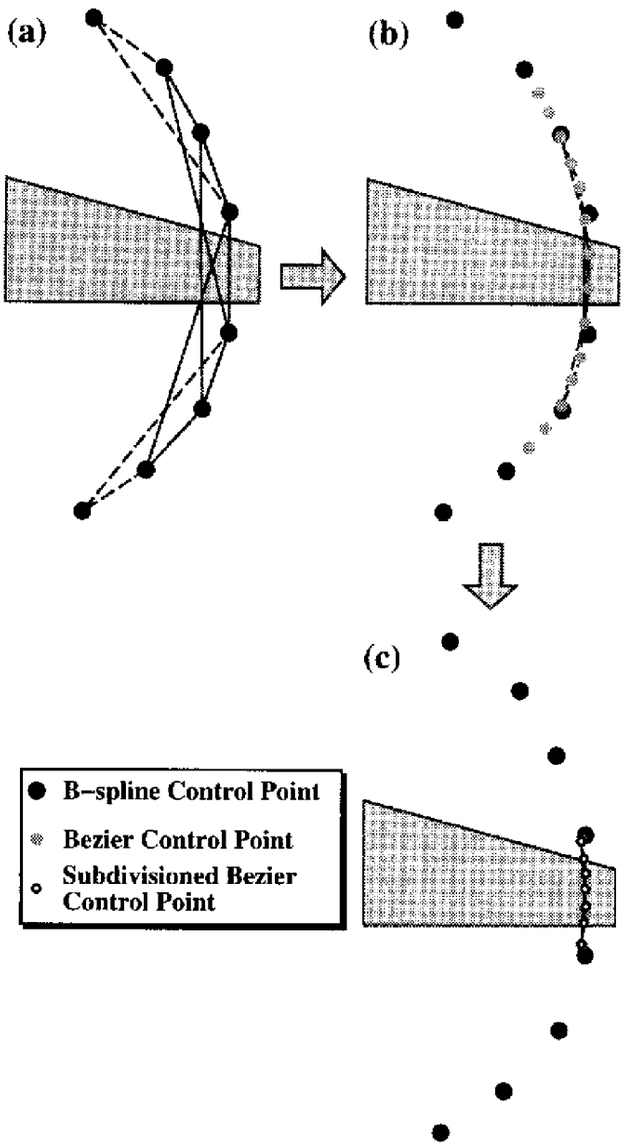

図 7　Crash Judgement Algorithm

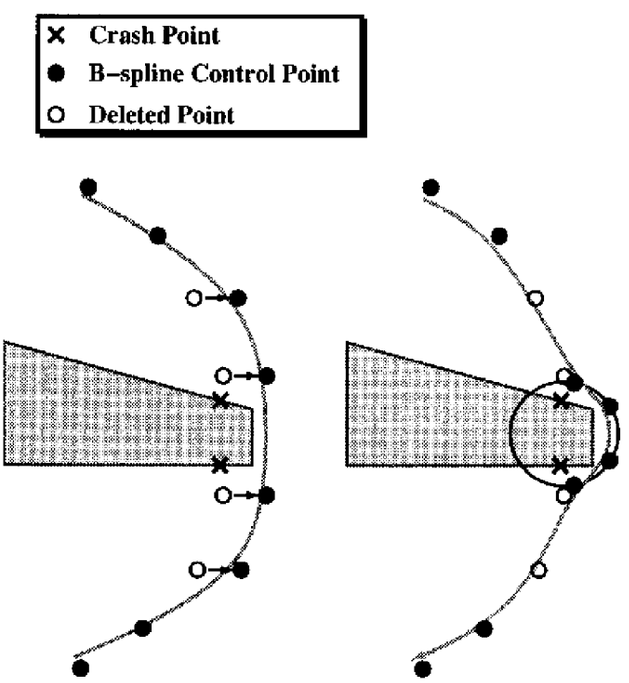

図 8　Examples of Obstacle Avoidance

3.3.1 Bスプライン曲線の逆変換

3次のBスプライン曲線のノット点 $P_i(i=1,2,\cdots,n-1)$ が与えられて，制御点 $Q_j(j=1,2,\cdots,n)$ を求める逆変換について述べる．曲線パラメータが整数のとき，次の連立方程式が成り立つ．

$$\frac{1}{6}Q_{i-1}+\frac{2}{3}Q_i+\frac{1}{6}Q_{i+1}=P_i \quad (i=1,2,\cdots,n-1)$$

この連立方程式は未知数に対し条件数が二つ少ないので，通常，両端の制御点を多重頂点（曲線の両端の曲率を0とする）とすることで，未知数と条件数を一致させることにより，解くことができる．ただし，この両端の制御点に関する条件設定は任意に決定できる．

3.3.2 衝突回避アルゴリズム

衝突回避アルゴリズムは障害物との衝突に寄与した制御点の両端点を結ぶ線分が障害物と交わる場合と交わらない場合で分類される．それぞれについて，順次説明していく．

(I)交わる場合のアルゴリズム(Case 1)

1．障害物との衝突に寄与した制御点 Q_a と障害物の頂点を用い，障害物に対し凸包を求める（図 9(a)）．

2．1.で求めた凸包上の点をノット点とし逆変換する．ただし，条件設定として，逆変換される両端の制御点を障害物との衝突に寄与しなかった両端の制御点 Q_b と一致させる（図 9(b)）．

3．2.で求めた制御点と衝突に寄与しなかった制御点を用い，経路設計をする（図 9(c)）．

(II)交わらない場合のアルゴリズム(Case2)

1．衝突に寄与した制御点 Q_a の両端点を結ぶ線分と障害物の頂点との距離が最短である頂点を求める（図 10(a)）．

2．1.で求めた頂点と衝突に寄与した制御点の両端点と衝突に寄与しなかった制御点を用い，経路設計をする（図 10(b)）．

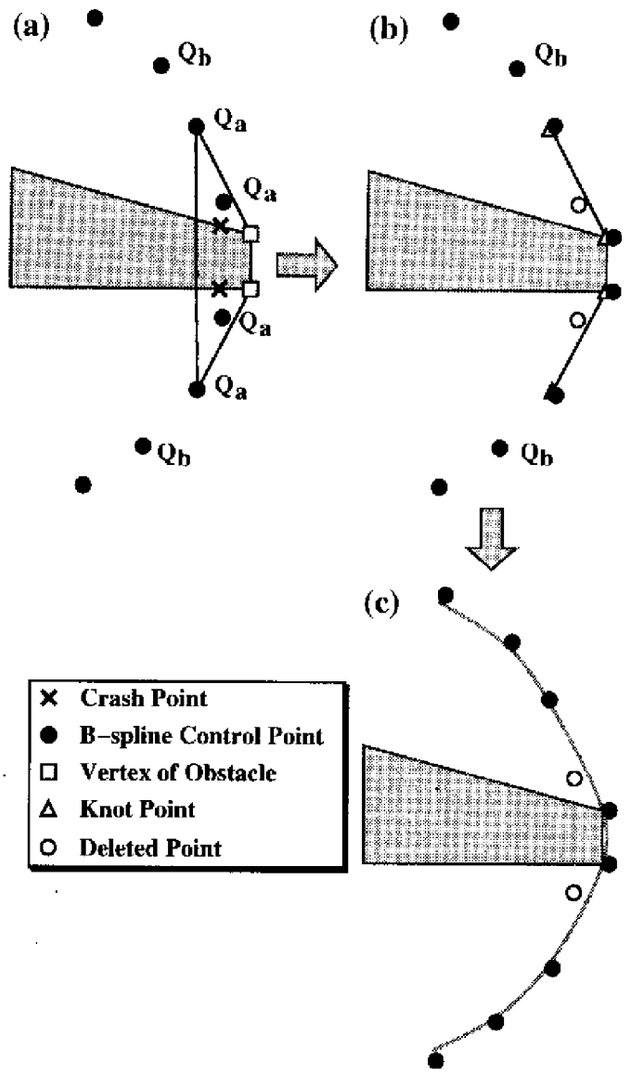

図 9　Obstacle Avoidance (Case1)

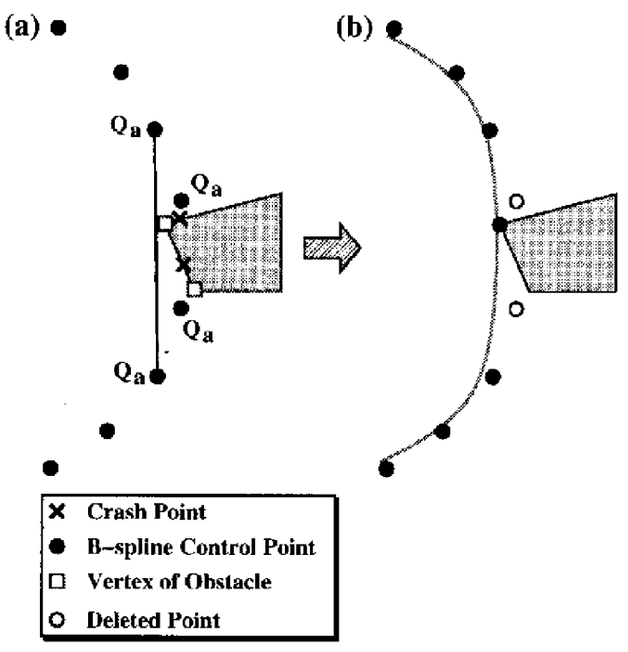

図 10　Obstacle Avoidance (Case2)

4. おわりに

自律移動ロボットの空間的および時間的に局所変更性を有する軌道生成法を提案し，障害物回避経路の設計法を示した．今後は，本研究で提案した手法の有効性を実機を用いて検証するとともに，単位運動においてより有効な制御点の選定法を検討する予定である．

参考文献

1) B.Guenter and R.Parent： Computing the Arc Length of Parametric Curves, IEEE Computer Graphics & Applications, 72/78 (1990)

2) 高山，狩野： 運動の単位に基づくロボットマニピュレータの運動制御，計測自動制御学会第3回ロボット工学部会研究会資料，1/10 (1987)

3) 森 善一： 非ホロノミックな全方向移動ロボットの機構およびその制御に関する研究，東北大学大学院情報科学研究科博士学位論文，(1998)

4) 山口富士夫： コンピュータディスプレイによる形状処理工学[II]，69/177，日刊工業新聞社(1992)

5) 小森谷，谷江： スプライン曲線による車輪型移動ロボットの軌道制御，日本ロボット学会誌，8-2 133/143 (1990)