

計測点群の補間に基づいた タイヤ空気圧モニタリングシステムの検討

†庄司こずえ †今野晃市 †蛸島昭之 †梶田屋秀樹
†岩手大学工学部 †アルプス電気株式会社

Study on tire pressure monitoring system based on interpolating of measured point set

†Kozue SHOJI †Kouichi KONNO †Akiyuki TAKOSHIMA †Hideki MASUDAYA
†Iwate University †Alps Electric co.,ltd.

1 はじめに

組み込み技術の進歩により、様々なセンシング技術とソフトウェア技術を用いた組み込みシステムが研究・開発されている。特に、自動車、家電などに実装されている様々な機能は、組み込みソフトにより実現されていることが知られている。

近年、タイヤ空気圧低下に起因するタイヤバーストによる事故を防止し、自動車の走行安全性を確保するため、タイヤ内に空気圧センサーを設置し、タイヤ内の空気圧情報を自動車運転者へ常時通知する空気圧モニタリングシステム (TPMS: Tire Pressure Monitoring System) の装着が必要とされている [1]。

従来の TPMS は、バッテリーにより駆動されるものが主流であり、バッテリーの寿命やそれに起因する通信頻度、使用温度範囲、重量などの問題がある [2]。そこで、次世代の TPMS ではバッテリーを搭載しないものが主流となると考えられる。バッテリーレスの TPMS では、上記の問題を一気に解決することが可能であり、将来への発展が期待できる。

次世代 TPMS のひとつに、水晶振動子方式を用いたバッテリーレスな装置がある。このような装置では、共振周波数と圧力の関係を明確にしておき、共振周波数を測定することで、圧力を得ることができる。しかし、水晶の温度変化を考慮しないと、精度の良い圧力値を得ることは困難であることから、水晶の温度を関係式に組み込む必要がある。

例えば、圧力周波数、温度周波数、圧力値を複数計測して、3次元ベクトルを構成する。計測した3次元ベクトルを連続関数で適切に補間することができれば、任意の圧力周波数と温度周波数から、精度よく圧力を獲得することが可能である。しかし、計測した3次元ベクトルは、格子状に整列されているわけではないので、整列されていない3次元ベクトルデータの補間技術が必要である。

本論文では、圧力周波数、温度周波数、圧力値を補間した連続関数を生成し、任意の圧力周波数と温度周波数から圧力値を得るシステムを提案する。本手法は、3次元ベクトルデータの補間には、パラメトリック曲面の一種である双3次 B-spline 曲面 [3] を用いる。また、車載 PC のメモリ容量の制限を考慮した、曲面の u, v パラメータを算出するための差分法によるインバージョンアルゴリズムを提案する。

本手法を、車載 PC および、センサ (トランスポンダ) に組み込み、その効果を確認した。

2 タイヤ空気圧モニタリングシステム

2.1 タイヤ空気圧モニタリングシステムとは

タイヤ空気圧モニタリングシステム (TPMS: Tire Pressure Monitoring System) は、タイヤの空気圧を計測して低下の警告を出すシステムである。TPMS はアメリカにおいては、TREAD act[1] にて義務化されており、欧州に

においても環境問題から、義務化がなされようとしている。TPMSのシステムには、タイヤの空気圧を直接計測する直接式と、ABSの輪速計測を利用して輪速の差からタイヤ空気圧を推定する間接式とがある。現在は直接式が主流となっている。以降では、特に断らない限りTPMSは直接式とする。

直接方式のTPMSは、タイヤ内に空気圧センサを内蔵し、車体側に受信機を備えているシステムである。このシステムでは、4本のタイヤの空気圧を計測した結果を、受信機で受信して、空気圧の低下をドライバーに警告する。現在は、タイヤに内蔵するセンサに電池を使った、電池内蔵式が実用化されている。しかし、このシステムには、電池の寿命や使用温度範囲、重量、電池性能を所定の期間保持できるように設定される通信頻度や、廃棄の際の環境負荷の問題がある。そこで、電池の装着が不要である電池レスのシステムが必要とされている。

2.2 電池レス TPMS システム

電池レスシステムについては、いくつかのシステムが考案されている。本研究で使用するシステムは、IQMにて開発されたTPMSシステムを用いている[4]。システムの構成を図1に示す。このシステムは、センサからの信号を受信するECU(Electric Control Unit)と、それぞれのタイヤに組み込まれるセンサ(トランスポンダ)で構成されている。センサは、計測データをECUへ送信するため、無線回路とアンテナを内蔵している。トランスポンダは共振器と容量式圧力センサから構成されており、センサの容量変化によって共振周波数が変化する。本システムは、トランスポンダの周波数変化をECUにて検知することで、圧力を算出することが可能である。

また、精度よく圧力値を算出するために、トランスポンダには温度センサも搭載し、温度特性に応じて圧力値の補正をかける構成になっている。まず温度センサから得られる温度値と、圧力センサと共振器における共振周波数の相関をとり、温度に対する共振周波数の標準値を算出しておく。そして、圧力センサから得られる返信信号の周波数

と、標準値の差分を分析することによって、温度変化を考慮した圧力値を精度良く得ることができる。

前述の手法は物理的には分かりやすい。一般に多項式展開を用いて計算モデルを構築することができる。しかし、多項式の次数が高くなると、補間結果が振動するといった問題がある。そこで本論文では、前述の問題を解決する手法について提案する。

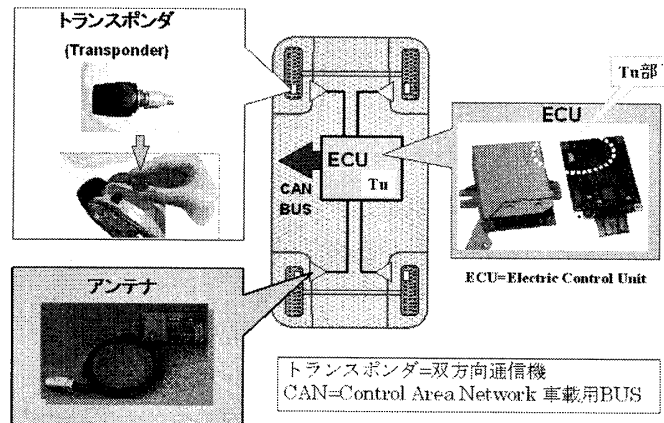


図1: 電池レス TPMS システム

3 提案手法

3.1 概要

2.2節で述べたシステムは、電源レスなシステムである。水晶振動子を利用して、温度周波数と圧力周波数を測定することが可能である。これを利用することで、2.1節で述べた、電池寿命、通信頻度、重量、地球環境の問題を一気に解決することができる。

2.2節のシステムでは、個々のトランスポンダの構成部品にはばらつきがあるため、温度周波数と圧力周波数はトランスポンダごとに特性が異なる。そのため、圧力を計算するための補間係数はトランスポンダが保存しており、ECUは各トランスポンダ固有の補間係数を取得して、決められた補間関数で圧力値を計算する。タイヤに取り付けられたトランスポンダは、1秒に10回以上の頻度で温度周波数と圧力周波数を計測し、その結果を無線

通信を利用して ECU へ転送する。ECU では温度周波数と圧力周波数を入力として、トランスポンダから得られた B-spline 補間係数を補関数に適用して圧力値を計算する。

個々のトランスポンダに保存される B-spline 補間係数は、あらかじめ計測しておいた温度周波数、圧力周波数、圧力値の 3 次元の離散データを利用して、B-spline 補間したときの補関数の係数である。トランスポンダに搭載されているメモリは約 2Kbit である。そのうち、B-spline 補間係数の保存に割り当てられるメモリは約 500bit となるため、B-spline 補間係数のデータ圧縮が必要となる。

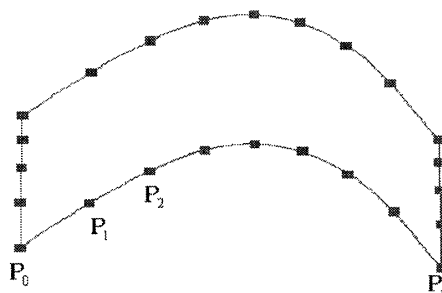


図 2: 境界曲線を表す点群と B-spline 曲線

3.2 B-spline 補間法

測定した温度周波数、圧力周波数、圧力値の 3 つのデータを 3 次元座標点群とみなし、点群を補間する 3 次 B-spline 補関数を生成する手法について説明する。以下では、B-spline 補関数は、3 次元空間の B-spline 曲面とみなして処理を行う。空間におかれている 3 次元座標点群を滑らかに近似する曲面の制御点を発生し、その x, y, z 座標を B-spline 補関数の補間係数とみなす。

点群を近似する B-spline 曲面を生成する手順は次のようになる。最初に境界曲線を表す点群から、曲面の境界となる B-spline 曲線を求め、次に B-spline 曲面を生成する。以下の手順で B-spline 曲線を近似する [5]。

1. 曲面の境界曲線を表す点列を抽出する。図 2 に境界曲線を表す点列を示す。図 2 の青いマーカーが境界曲線を表す点列を示している。この図で、 $\mathbf{P}_i (i = 0, \dots, m)$ が 1 本の境界曲線を生成するとき利用する点列である。また点列から生成された B-spline 曲線も表示している。
2. 点列を近似する B-spline 曲線を生成するためには、その点が曲線のパラメータと対応している必要がある。 $m + 1$ 個の点列 $\mathbf{P}_k (k = 0, \dots, m)$ があるとき、各点における曲線のパラメータ $u_k (k = 1, \dots, m - 1)$ を、式 (1) のように決定する。

$$u_k = u_{k-1} + \frac{|\mathbf{P}_k - \mathbf{P}_{k-1}|}{d} \quad (1)$$

ただし、

$$d = \sum_{k=1}^m |\mathbf{P}_k - \mathbf{P}_{k-1}|, \\ u_0 = 0, u_m = 1,$$

とする。

3. 最小二乗法により曲線の制御点を求める。次数 3、制御点数 $n + 1$ 個の B-spline 曲線は、次のように表わされる。

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,3}(u) \mathbf{P}_i \quad (2)$$

ここで、 $N_{i,3}(u)$ は B-spline 基底関数 [3] である。

4. 手順 3 で得られた B-spline 曲線と、元になった点列の距離を計算し、距離がしきい値以上となったパラメータに対応する区間に、その区間の平均値を新たなノットとして挿入する。例えば、ノットベクトルの初期値は、 $[0, 0, 0, 0, 1, 1, 1, 1]$ とし、区間 $(0, 1)$ の間に、その平均値 $1/2$ を挿入する。その後、手順 3 に戻り、曲線を生成しなおす。

次に、4 本の境界曲線と境界曲線で囲まれる領域内の点群で、B-spline 曲面を生成する。4 本の境界曲線と領域内の点群を図 3 に示す。以下の手順で、点群を B-spline 曲面で近似する。

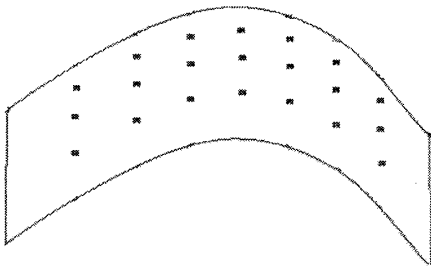


図3: 境界を表す B-spline 曲線と内部点群

1. 境界点列を近似する4本の B-spline 曲線を生成し、近似される B-spline 曲面の境界曲線とする。
2. 境界曲線を生成した点列以外の内部の各点が、曲面上でとる u, v パラメータ値を推定する。具体的には、相対する2本の B-spline 曲線からルールド曲面を生成し、その曲面に内部のサンプリング点を射影して u, v パラメータを得る。
3. 最小二乗法により曲面の制御点を求める。近似する B-spline 曲面を $S(u, v)$ 、曲面の制御点列を $P_{ij} (0 \leq i \leq n, 0 \leq j \leq m)$ とすると、曲面 $S(u, v)$ は、式(3)のように表すことができる [6].

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,3}(u) N_{j,3}(v) P_{i,j} \quad (3)$$

ここで $N_{i,3}(u), N_{j,3}(v)$ は B-spline 基底関数である。

3.3 圧力値の計算方法

圧力周波数を x 、温度周波数を y とし、圧力値 z を算出するために、本手法では x, y 値と B-spline 曲面から、インバージョン法を用いて、 (x, y) に対応するパラメータ (u, v) を算出する。その後、その (u, v) を用いて z 値を計算する。

一般に3次元座標値に対応する曲面の u, v パラメータを得るには、非線形の偏微分方程式を、ニ

ュートン法を用いて計算するインバージョン法を利用する。しかし、曲面上の点の位置の計算と、偏微分ベクトルの計算の両方のコードを実装することは、メモリ量の増加をまねく。ECUでは、2Kbyte程度のメモリ使用量を想定しているの、従来のインバージョン法をそのまま実装することは難しい。本システムでは、できるだけ少ないメモリ使用量で目的とする値が得られるように、曲面の座標値の計算のみを実装し、必要な値を取り出すシステムを提案する。

3.4 提案するインバージョン手法

曲面上の座標値を得る機能だけを利用して、 u, v パラメータを近似する手法について述べる。手順は以下ようになる。

1. 初期入力データから生成した曲面 $S(u, v)$ について、 u, v 各方向に4等分したグリッド線を引き、その交点上に点群を発生する。

図4に示すように、発生した点を $Q_{ij} (i = 0, \dots, 4, j = 0, \dots, 4)$ としたとき、各点における u, v パラメータは、 (u_i, v_j) となる。 $u_i = i/4, v_j = j/4$ であることから、パラメータの間隔は0.25である。

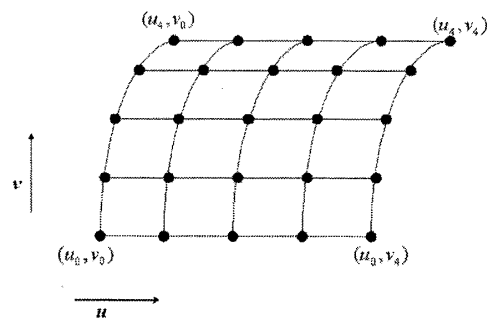


図4: 曲面上に発生させた点群

2. 任意の点 P と最も近い面上の点 Q_{ij} を求める。初期状態としては、点 P の x, y は与えられた値を利用し、 z は0に設定する。求めた面上の点 Q_{ij} の z 値を、点 $P(x, y, z)$ の z 値に代入する。

3. 図5に示すように、 Q_{ij} の近傍の4点 $Q_{i-1,j}$, $Q_{i+1,j}$, $Q_{i,j-1}$, $Q_{i,j+1}$ を取り出し、 Q_{ij} を中心に4つの三角形面を構成する。各三角形面の法線ベクトルの平均ベクトルを求め、点 Q_{ij} の接平面を表す法線ベクトル n_{ij} とする。

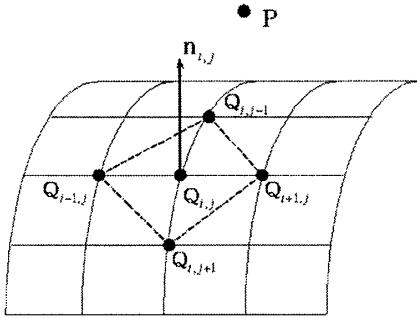


図5: 近点 Q_{ij} での接平面の法線ベクトル

4. Q_{ij} の接平面に、点 P を射影する。射影した点を P' とする。
5. 射影した点 P' が、 Q_{ij} と近傍の4点で分けられる領域AからDのどこに位置するのかわかる。 Q_{ij} と近傍の4点で分けられる領域を図6に示す。AからDのどの領域に P' が位置するかは依存して、 Q_{ij} に隣接する点と Q_{ij} との差分ベクトルにより、 u, v 各方向の方向ベクトル U, V を決定する。例えばAの領域では、式(4)、式(5)のように U, V ベクトルが決定される。

$$U = Q_{i+1,j} - Q_{ij} \quad (4)$$

$$V = Q_{i,j+1} - Q_{ij} \quad (5)$$

6. 式(6)から、パラメータの差分値を求める。

$$w = \alpha U + \beta V \quad (6)$$

ここで、 $w = P' - Q_{ij}$ とする。式(6)を表した図を図6に示す。 α は区間 Q_{ij} と $Q_{i+1,j}$ の長さを1としたときの w の u 方向成分の割合を表し、 β は、区間 Q_{ij} と $Q_{i,j+1}$ の長さを1としたときの w の v 方向成分の割合を表す。

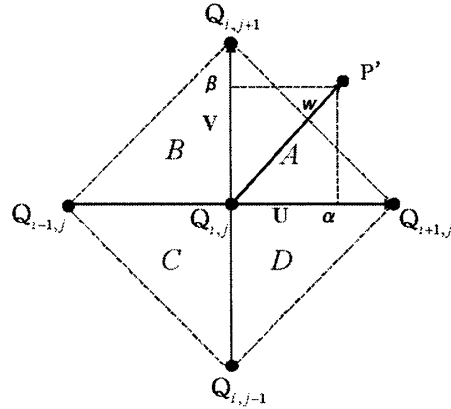


図6: Q_{ij} と近点で分けられる領域と座標値評価関数によるパラメータ

よって、 P' を表すパラメータ値 u'_i, v'_i は、式(7)から式(10)より求められる。

$$\Delta u = \frac{\alpha}{|U|} \quad (7)$$

$$\Delta v = \frac{\beta}{|V|} \quad (8)$$

$$u'_i = u_i + \Delta u \quad (9)$$

$$v'_i = v_i + \Delta v \quad (10)$$

ここで、 u_i, v_i は、面上の点 Q_{ij} における u, v パラメータとする。

7. 次に u'_i, v'_i の近傍4点のパラメータ値を次式のように設定する。

$$u'_{i-1} = u'_i - 0.25 \times \frac{1}{2^j} \quad (11)$$

$$u'_{i+1} = u'_i + 0.25 \times \frac{1}{2^j} \quad (12)$$

$$v'_{i-1} = v'_i - 0.25 \times \frac{1}{2^j} \quad (13)$$

$$v'_{i+1} = v'_i + 0.25 \times \frac{1}{2^j} \quad (14)$$

ここでは j は繰り返し回数とし、繰り返し回数が増加するごとに4近傍の点との区間が狭められる。

P' を新たに P と仮定し、手順3から手順7を繰り返す。ベクトル w の大きさが十分小さくなったところで、処理を終了する。本システムでは、一定間隔で結果を返す必要があるため、反復回数を4回とした。

4 車載PC上での実装

4.1 マイコン上での計算の高速化

本システムでは、曲面補間係数の計算はPC上で行うが、車両のタイヤ空気圧を推定するのは車載ECU上のマイコンが行う。そのため、測定のリアルタイム性を確保するためには、マイコン上で高速に圧力値を推定するための工夫が必要である。

マイコン上での計算高速化のために、浮動小数点演算はすべて整数演算で行えるように、データ型を整数型で表現するように変更する。また、ほとんどの16bitマイコンでは、32bitの乗除算はハードウェアではなく関数ライブラリで処理されるために、処理時間が数msかかることがある。そのため、整数演算でも32bitの乗除算は、できるだけ使用しないようなアルゴリズムを開発する。

4.2 基底関数テーブルの圧縮

B-spline補間により圧力を求めるには、式(3)に示すB-spline曲面 $S(u, v)$ をECU上のマイコンで計算する必要がある。しかし、基底関数 $N(u)$ の結果を逐一計算していたのではシステムのリアルタイム性が損なわれる。そこで、本システムではあらかじめ基底関数 $N(u)$ をPC上で計算しておき、その結果を格納したテーブルをマイコン上のROMに格納する。B-spline基底関数は連続関数であるが、マイコンのROM上に格納するためにはこれを離散化する必要がある。本システムではB-spline基底関数を最大値512で正規化し、 $\Delta = 1$ として離散化する。ノットベクトルを u 方向 $(0, 0, 0, 0, 256, 512, 512, 512, 512)$ 、 v 方向 $(0, 0, 0, 0, 128, 256, 384, 512, 512, 512, 512)$ として離散化した結果をプロットすると図7、図8のようになる。

図から分かるように、 u 方向では5本、 v 方向では7本の基底関数データが必要である。本システムではC言語で実装を行っているので、これら12本の曲線データを格納するには、一点につき16bit (unsigned short int型) とすると $16 \times 513 \times 12 = 98,496$ bit (約10KB) のROM容量が必要となり、マイコンROM (32KByte) の30%以上を

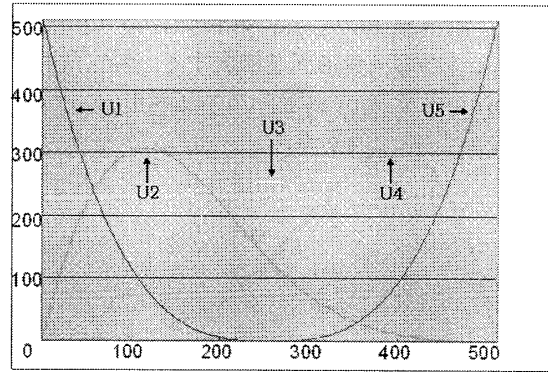


図7: u 方向のB-Spline基底関数

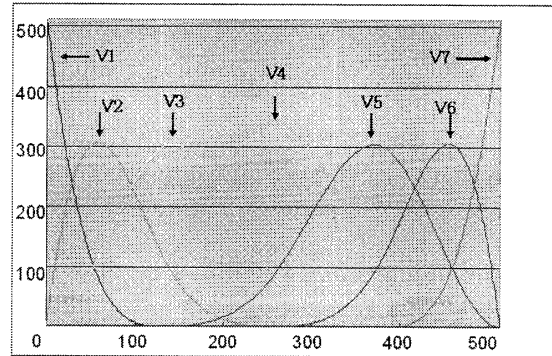


図8: v 方向のB-Spline基底関数

消費してしまう。このため、何らかの方法でテーブルサイズを圧縮する必要がある。

ここで、基底関数のグラフにもう一度注目すると、曲線U1とU5、U2とU4は相似形であることが分かる。そのため、U1が必要なときは配列の先頭から、U5が必要なときは配列の末尾から配列データを参照するようにインデックスを操作すれば、同一のデータで曲線U1とU5の両方を表せるため、必要なROMサイズを1/2に圧縮することができる。次に、曲線U3に着目すると、頂点を軸に左右対称となっていることが分かる。従って、軸から片側半分だけのデータを持っていれば曲線全体を表すことができ、必要なROMサイズは1/2となる。また、曲線V1とV7、V2とV6、V3とV5、V4についても同様のことが言える。さらに、表1に示すように、曲線U2とV2の配列データを比較すると、V2の配列はU2の配列のインデックスが偶数のものだけを抜き出してきたものと等しいため曲線U2の配列データだけで曲線

表 1: U2 と V2 の関係

インデックス	0	1	2	3	4	5	6
U2	0	6	12	18	23	29	35
V2	0	12	23	35	46	57	67

V2も表すことができる。また、同様のことが曲線 U4 と V6 にも言える。

以上のデータ圧縮を行った結果、必要な ROM 容量は 32,832 bit (4KB) となり、元のデータサイズから 57% まで圧縮することができた。ただし、配列インデックス操作などによって増加するプログラム ROM サイズについては、ここでは考慮していないが、プログラムの増加量はわずかである。

5 結果

5.1 補間曲面の評価

測定した温度周波数、圧力周波数、圧力値の 3 つのデータを 3 次元座標点群とみなし、離散データの補間をおこなう。

圧力周波数を x 、温度周波数を y とし、 x, y 値と B-spline 曲面から、3.4 節で述べたインバージョン法を用いて (x, y) を想定するパラメータ (u, v) を算出し、圧力値 z を求める。

B-spline 曲面の生成に使用される B-spline 補間係数は、整数に正規化した後で圧縮して、タイヤに取り付けられたトランスポンダに保存する。圧縮データをマイコンへ転送し、マイコン上で解凍して B-spline 補間係数を得る。解凍された補間係数を用いて圧力値を計算する。

実験では、正規化による丸め誤差の影響を調べるため、整数値で表された B-spline 補間係数を用いて PC 上で圧力値を求め、評価用の圧力データと比較し誤差評価を行う。実験環境は、Celeron(R) 2.2GHz、メモリ 768MB の PC である。

図 9 に示すように、初期入力点として、 u 方向に 5 点、 v 方向に 9 点の合計 45 点の点群データから、補間曲面を生成する。

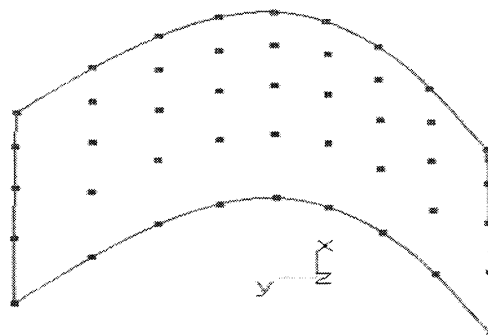


図 9: 初期入力点

B-spline 曲線の生成に使用するノットは、最も良好に曲線を近似できることと、生成される補間係数の総数を考慮して、 u 方向 $(0, 0, 0, 0, 0.5, 1, 1, 1, 1)$ 、 v 方向 $(0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1)$ とする。このノットで生成される B-spline 曲面補間係数は u 方向に 5 点 v 方向に 7 点の計 35 点である。生成された B-spline 曲面補間係数を図 10 に示す。トランスポンダへは、B-spline 補間係数を整数に正規化して保存する。

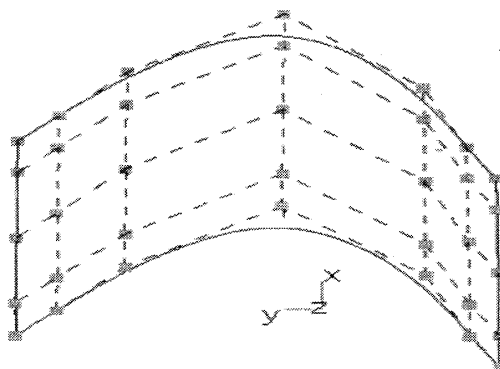


図 10: B-spline 曲面補間係数

補間曲面を評価するために別途計測しておいた 561 点の点群を用いて、得られた形状を評価する。図 11 に示すように、評価点群は x 方向に 17 点、 y 方向に 33 点、計 561 点配置されている。この 561 点の圧力値である z 値と、点群補間手法で求めた点の z 値の誤差を評価する。結果を図 11 に示す。

表 2: z 値の誤差

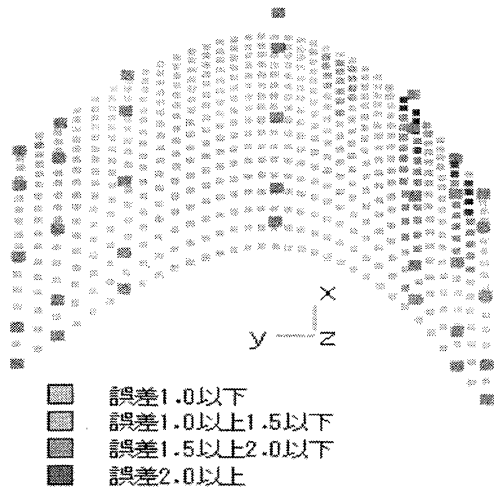


図 11: 点群補間を行った結果

図 11 では、誤差が 1.0 未満の点を水色で、誤差が 1.0 以上 1.5 未満の点を緑で、誤差 1.5 以上 2.0 未満の点を紫で、誤差 2.0 以上の点を青で示す。また、PC とマイコンでそれぞれ実行したときの z 値の誤差を、表 2 に示す。PC での実行結果は、評価点 561 点に対して、74.5% を占める 418 点が誤差 1.0 未満となった。また、全体の 88.8% を占める点が、誤差 1.5 未満となった。誤差が 2.0 以上となったのは、12 点のみで、誤差の最大値は 2.8 であった。マイコンでの計算では、4 章の車載 PC 上への実装で述べた、演算での制約があるが、算出された z 値の誤差範囲は、PC で計算した結果とほぼ同様な結果を得ることができ、マイコンへ実装したときの有効性が確認できた。

実験の結果、ほとんど全ての点の誤差値が 2 以下で補間できた。点群の z 値の最小値は 100、最大値は 500 であるので、 z 値における点群の範囲は 400 である。誤差値が 2 というのは z 値の範囲に対して相対的に 0.5% となり、非常に小さい値であると言える。

6 まとめと今後の課題

本論文では、離散データである圧力周波数、温度周波数、圧力値を B-spline 曲面補間した連続関数を生成し、任意の圧力周波数と温度周波数から

誤差範囲	点数		全体に占める割合	
	PC	マイコン	PC	マイコン
0 以上 1.0 未満	418	457	74.5 %	81.5 %
1.0 以上 1.5 未満	80	73	14.3 %	13.0 %
1.5 以上 2.0 未満	51	18	9.1 %	3.2 %
2.0 以上	12	13	2.1 %	2.3 %

圧力値を得る、タイヤ空気圧モニタリングシステムを提案した。また、本システムは車載マイコンへ搭載するため、限られたメモリ使用量を想定し、計算コストが低くなるアルゴリズムについて述べた。本アルゴリズムで推定した圧力値は、相対誤差 0.5% 以下となり、十分に小さい誤差で値を得ることができた。また、車載マイコン上での計算を高速にするために、すべての計算を整数型で実装した。

本システムでは、タイヤに取り付けられたトランスポンダに、B-spline 補間係数を保持しているが、トランスポンダに搭載されているメモリのうち、B-spline 補間係数に割り当てられるメモリは約 500bit である。そのため、B-spline 補間係数のデータ圧縮が今後の課題となる。

参考文献

- [1] TREAD 法 FMVSS138, タイヤ空気圧監視システム標準化規格.
- [2] Alps's Environmental Report 2004, <http://www.mlit.go.jp/seisakutokatsu/freight/kannkyou/24alps.pdf>
- [3] Piegl, L. and Tiller, W., The NURBS Book, Springer-Verlag, 1995.
- [4] 特許公開 2008-201369, タイヤ情報検出装置.
- [5] 徳山喜政, 今野晃市: “N 角形領域を覆うような B-spline 曲面の生成方法”, 情報処理学会論文誌, 第 43 巻, 第 10 号, pp. 3209-3218, (2002).
- [6] Farin, G., Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, Academic Press, 1996.