分割主鏡シミュレータ II — 分割鏡動作パラメータの把握 III

岡山新技術望遠鏡グループ

平成 22 年 6 月 9 日

目 次

1 概要

分割鏡動作パラメータの把握 I では、分割主鏡の動作パラメータについてアクチュエーター制御量からギャップセンサー読み出し量への変換行列を求め、その変換行列の固有値の分布を見ることでギャップセンサーの配置の最適化が行えないかどうかを検討した。

ここでは、I で採用した簡略化されたギャップセンサー配置でなく、より現実的な配置ができるように入力データや内部の構造を改造することを検討する。

2 ギャップセンサーの配置点

I では、ギャップセンサーの配置点をギャップセンサーの検知領域の中央で代表させ、かつ代表点をセグメント間の中央点のみに限定した。ここでシミュレーター全体の設計ドキュメントなどでのように面積を持った検知平面 2 つで構成されるギャップセンサーについて、その平面間の傾きは無視でき、かつ検知領域の中央で代表させることは問題ないとする[1]。

ここでは、セグメント間中央にないギャップセンサー検出点の配置を実現できないかどうかについて検討する。検出点配置は、先ではセグメント ID、設置対象の辺 ID、辺上での位置情報を入力するものとした。理想的には完全に自由に XY 座標で配置場所を記述できるのが望ましいかもしれないが、今回のアプリケーションでは検出点配置場所はセグ

[1] 厳密には検知点間距離とセンサー内部のカウンタは非線形の関係を持つため、検知平面 2 つの間の傾きは多少なりと影響を及ぼすと考えられる。しかしながら、今回の検出対象においては傾きによって発生すると考えられる検知点間距離の変化と非線形性の非線形項の大きさを考慮すると、使用可能範囲の端でない限り影響は誤差レベル程度とも想定できる。また、以前の検討からアクチュエーターによる鏡材背面の傾き変化量はもともとの傾きに比べて上述の意味でも微小であるので、非線形項の影響は検知点間間隔に対して固定関数のバイアスとして表現でき、その影響の補正は容易であるはず。

メント間中央からそれほど離れないところに限定される。よって、配置場所の記述は、辺上 (セグメント間中央点) での位置情報と辺からの垂直距離の 2 パラメータによるものとする。

2.1 入力ファイル仕様

入力は以前と同じくテキストファイルとし、1 行 1 ギャップセンサー定義とする。追加するパラメータである、セグメント間中央点からの垂直距離は、定義対象のセグメントの内部へ入っていく方向を + とする。つまり、内周側の辺であれば R が増加する方向、外周側の辺であれば R が減少する方向である。

- セグメント ID
- 指定されたセグメント内での固定点位置が乗る辺の ID
- 位置 (x 座標)
- セグメント間中央点からセグメント側に入る距離 (垂直方向)

ここで、定義から距離は定義されている対象の位置から辺に対して垂直方向であるので、内周・外周のセグメント同士が接する辺では単に直線に垂直な方向、円弧部分では配置点の R を変化させるだけに相当する。

3 最内周の固定ギャップセンサー

最内周の固定ギャップセンサーの物理的な構造を検討する。簡単に考えられる構造としては、センサーごとに何らかのトラスに固定された対向板を用意するものであるが、トラスの変形を直接反映するために 1 点ごとに変形の自由度が 1 増えることになり、固定ギャップセンサーとは呼べない。よって、最内周部分に、円環状の非常に硬い金属ブロックをトラスに 3 点固定で配置し、そのブロック上の平面をギャップセンサーの固定側とすることを考える。

実機上では、計算誤差における桁落ちの影響を避けるため、この部分も仮想的なアクチュエーターが存在すると考え、アクチュエーターの移動量に相当するギャップセンサー配置点での変形量を制御ループまえに読み出し値から補正するなど、いくつかの操作を行う必要はあると思われるが、影響を抑えた形で制御のほうにも導入可能と考えられる。また、構造的には中央部分は 60° でほぼ対称形であるため、3 点固定は同じ半径の位置に 120° 離れた 3 固定点を配置するような設置方法をとることができる[2]。

3.1 シミュレータ上の表式

最内周の固定ギャップセンサー配置について、仮想アクチュエーターは位置を図 1 のようにとる。これに対して、ギャップセンサーは内周セグメントの内側円弧における固定位置の定義に依存するが、基本的には直交座標 x, y および極座標 r, θ で表すことができるが、いま内周セグメント側からのことを考えると、極座標のほうが取り扱いしやすいと考えられるので、以下では極座標で議論する。

ここで、極座標の角度を簡単のために、通常とは異なり Y 軸 + 方向 (セグメント ID 1 の中央点) を 0° とし、時計回りに増加する方向に θ を定義する。これにより、セグメントの内側円弧に対して求めた角度を (頭の中でも) 簡単に変換できる。ここで、仮想アクチュエーター半径を R とし、ギャップセンサーの半径を αR とする。よって、$x = \alpha R \sin\theta$, $y = \alpha R \cos\theta$ となり、セグメントに対する議論と同じく、$P_1(0,\, y_1,\, z_1)$、$P_2(-x_2,\, y_2,\, z_2)$、$P_3(x_2,\, y_2,\, z_3)$ とあらわすとすると $y_1 = R$, $x_2 = \sqrt{3}/2R$, $y_2 = R/2$ であり、任意点 x, y に対して

$$
\begin{aligned}
z &= \left(\frac{y_2 - y}{y_2 - y_1}\right) z_1 + \left(\frac{-x(y_2 - y_1) + x_2 y - x_2 y_1}{2x_2(y_2 - y_1)}\right) z_2 + \left(\frac{x(y_2 - y_1) + x_2 y - x_2 y_1}{2x_2(y_2 - y_1)}\right) z_3 \\
&= (2\alpha\cos\theta - 1)\, z_1 + \left(\sqrt{3} - \alpha\sin\theta - \sqrt{3}\alpha\cos\theta\right) z_2 + \left(\sqrt{3} + \alpha\sin\theta - \sqrt{3}\alpha\cos\theta\right) z_3
\end{aligned}
$$

[2] 3 点で固定する限り、平面の定義から固定点を動かしても同じ意味を持つ表式を作ることができるので、計算上は仮想的なアクチュエーターは理想位置に存在すると考えてよい。実際に動かすものではなく、かつその変形量を制御中に別途検出可能であるわけでもないので、仮想アクチュエーターの制御量は単なる内部制御パラメータである。

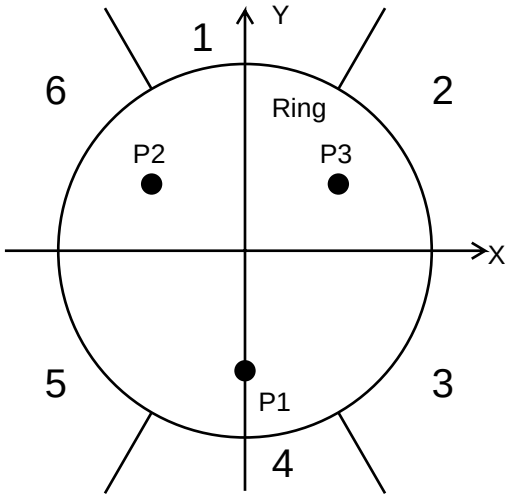

図 1: 最内周の固定ギャップセンサー配置

とあらわされる。

いま、仮想アクチュエーターの制御点とギャップセンサーが配置される半径の比 α は物理的な意味はほとんどない[3]ため、これを $\alpha = 1$ としても問題はない[4]。

3.2 シミュレーターへの実装

シミュレーターへの実装は、セグメント ID 19 を定義することとする。いま、内周セグメントの内側円弧は R 500.0 である。これを上述の R とする[5]。

4 パターン同士の比較

パターンの比較を行ってどう変化したかやどちらがより望ましいかを判断するには、パターン同士である特異ベクトルの特異値がどう変化したかをみる必要がある。この方法を検討する。

4.1 特異ベクトル同士の比較

まず、特異ベクトルがなす行列はエルミートであり、軸数分の次元を持つ空間を完全に張るようなベクトル空間を定義しているといえる。よって、このようなベクトル空間同士の比較を行うことになる。

これについて、異なる 2 パターンに含まれる特異ベクトルを一つずつとったベクトルの組についての相関は 2 ベクトルの内積をとることで判別できると考えられる。つまり、ある特異ベクトルに対して、異なるパターンのすべての特異ベクトルの中で最も内積が大きいベクトルを対応するベクトルとして定義する。

[3]仮想アクチュエーターに相当する物理的な機構はないので。
[4]ギャップセンサー配置点が異なる半径に複数あるとまた話は変わるが。
[5]注: 以前の内周円弧に対応するギャップセンサー位置の半径 R 499.0 とずれているので注意。

4.2 異なる次元のパターン同士の比較

異なる次元のパターン同士を比較する場合、つまり異なる数のアクチュエーター軸を固定したモード同士を比較する場合の扱いを考える。いま、定義から特異ベクトルのなす行列はエルミートであり、片方の特異ベクトルのなす行列をエルミート性を崩さずに拡張できれば直接比較できるといえる。

ここで、正規直交ベクトルの組 $(\mathbf{v}_1\,,\mathbf{v}_2\,,\cdots,\mathbf{v}_n)^T$ に対して、$\forall_{ij}\mathbf{v}_i\cdot\mathbf{v}_j=0$ である。いま、$\mathbf{v'}_i=(\mathbf{v}_i\,,x_i)$, $\mathbf{v'}_{n+1}=(0\,,\cdots,0\,,x_{n+1})$ と定義し、ベクトルの組を $(\mathbf{v'}_1\,,\mathbf{v'}_2\,,\cdots,\mathbf{v'}_n\,,\mathbf{v'}_{n+1})^T$ に拡張することを考える。$\forall_{ij}\mathbf{v'}_i\cdot\mathbf{v'}_j=0$ とするには $x_1=x_2=\cdots=x_n=0$ かつ $x_{n+1}=1$ であればよいといえる。

よって、何らかの理由で制御に取り入れられなかったアクチュエーター軸に対して、ベクトルのその部分の要素をすべて 0 とする拡張を行った特異ベクトルの組に対して前節のような比較を行えばよいといえる。

また、この場合、実際の計算には追加しないもののベクトル空間に付け加えられる $\mathbf{v'}_{n+1}$ に相当するベクトルと相関がいいベクトルは、アクチュエーター軸を固定することで縮退した空間の軸 (線形変換での kernel) であるといえる。

4.3 アルゴリズムの検討

実際に二つのベクトル空間の基底が、完全に一致するように張られていることは少なく、ある程度ずれた基底で張られていることが普通である。また、今回のターゲットについて 60^o 回転は相似形であるので、3 次の Zernike モードなどはそういう意味でもずれた基底に分解されている可能性が高い。このような場合にも相関判定がうまくいくようなアルゴリズムを考える。

まず、あるベクトル空間からもう一つのベクトル空間への対応関係が定義できる。これは、単にあるベクトル空間を張るベクトルの一つ一つに対して、最も相関のよいもう一つのベクトル空間を張るベクトルへの対応関係という意味である。これに対して、逆向きにも同様な対応関係を計算できる。これに対し、最も理想的な対応関係は、これら両向きの対応が一致しかつそれらの内積が両方 1 になっていることである。

いま、あるベクトルに対し、そのベクトルをターゲットとする対応関係を持つ他方のベクトル空間の基底が二つ以上ある場合について検討する。このような場合、二つのベクトル空間の中で最も相関がいい基底のペアは、両向きの対応が一致している、つまり両側から最も大きな内積を持つような基底のペアであると考えられる。よって、まずこの条件に合致するペアを相関判定済みの基底としてマークする。その後、これらの判定済みの基底を除いた部分ベクトル空間同士において同じことを繰り返す。この操作を繰り返すことで、ベクトル空間の間の線形変換で縮退した kernel に対応する基底を除くすべての基底に対する対応関係が決定できるはずである。