



制御関連進捗状況



分割主鏡セグメント間位相合わせ制御

- ・制御系シミュレーターの検討

主鏡制御系のハードウェア部分を代替するようなシミュレーターモジュールを利用して、制御系システム構築上の問題点を探る

- ・ギャップセンサー配置の検証ツール

安定した主鏡制御を可能にするギャップセンサーの配置場所をモデルから検証するためのツール



ギャップセンサー配置の検証ツール

ギャップセンサーデータから主鏡制御パラメータへの変換をモデル化する。

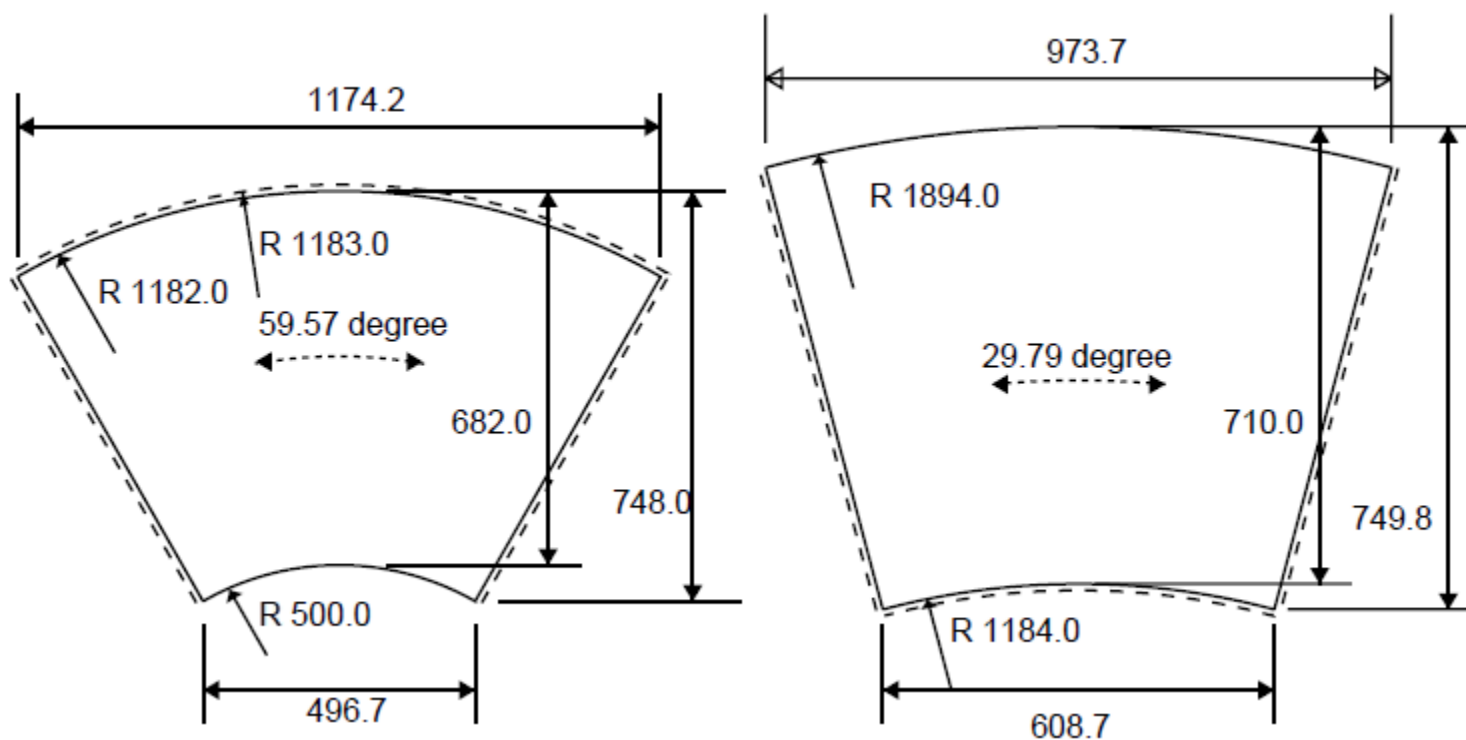
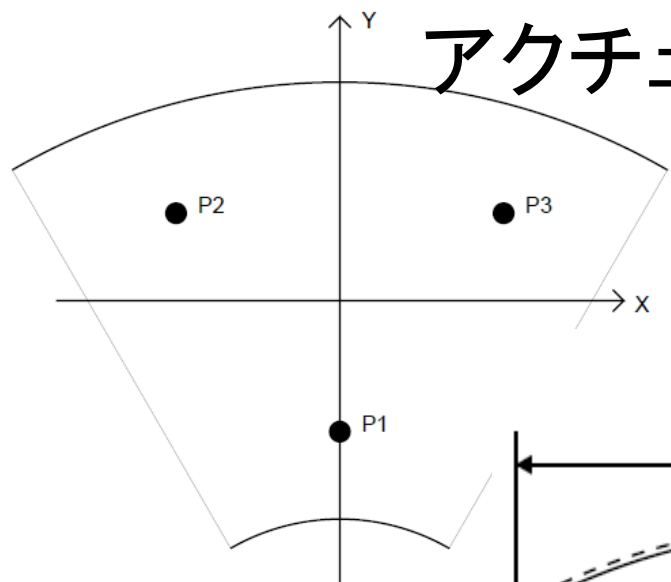
あるギャップセンサーの配置に対するモデルが安定かどうかを”数学的に”検証する。

ギャップセンサー以外の制約条件についても導入可能なものについては考慮して検証する。



ギャップセンサー配置の検証ツール

アクチュエーターとギャップセンサーの配置



まず、セグメントにおける変換係数に関して、簡単化のために、セグメント ID を i とし

$$S_1^i(x_A, y_A) = \frac{y_2 - y_A}{y_2 - y_1}$$

$$S_2^i(x_A, y_A) = \frac{-x_A(y_2 - y_1) + x_2 y_A - x_2 y_1}{2x_2(y_2 - y_1)}$$

$$S_3^i(x_A, y_A) = \frac{x_A(y_2 - y_1) + x_2 y_A - x_2 y_1}{2x_2(y_2 - y_1)}$$

と定義する。いま、ギャップセンサーの取り付け位置を $G_A^i(x_A^i, y_A^i, 0)$ とすると、この点での鏡材の移動量は

$$(S_1^i(x_A, y_A), S_2^i(x_A, y_A), S_3^i(x_A, y_A)) \cdot (z_1^i, z_2^i, z_3^i)^t$$

と表すことができる。つまり、一つのセグメントにある全ての取り付け位置 $(G_A^i, G_B^i, \dots, G_N^i)$ に対して

$$\begin{pmatrix} z_A^i \\ z_B^i \\ \vdots \\ z_N^i \end{pmatrix} = \begin{pmatrix} S_1^i(x_A, y_A), & S_2^i(x_A, y_A), & S_3^i(x_A, y_A) \\ S_1^i(x_B, y_B), & S_2^i(x_B, y_B), & S_3^i(x_B, y_B) \\ \vdots & & \\ S_1^i(x_N, y_N), & S_2^i(x_N, y_N), & S_3^i(x_N, y_N) \end{pmatrix} \cdot \begin{pmatrix} z_1^i \\ z_2^i \\ z_3^i \end{pmatrix}$$

$$(z_A^1, z_B^1, \dots, z_N^1, \dots, z_A^{18}, z_B^{18}, \dots, z_N^{18})^t =$$

$$\begin{pmatrix} S_1^1(x_A, y_A), & S_2^1(x_A, y_A), & S_3^1(x_A, y_A), & \dots, & 0, & 0, & 0 \\ S_1^1(x_B, y_B), & S_2^1(x_B, y_B), & S_3^1(x_B, y_B), & \dots, & 0, & 0, & 0 \\ \vdots, & \vdots, & \vdots, & \ddots, & \vdots, & \vdots, & \vdots \\ S_1^1(x_N, y_N), & S_2^1(x_N, y_N), & S_3^1(x_N, y_N), & \dots, & 0, & 0, & 0 \\ \vdots, & \vdots, & \vdots, & \ddots, & \vdots, & \vdots, & \vdots \\ 0, & 0, & 0, & \dots, & S_1^{18}(x_A, y_A), & S_2^{18}(x_A, y_A), & S_3^{18}(x_A, y_A) \\ 0, & 0, & 0, & \dots, & S_1^{18}(x_B, y_B), & S_2^{18}(x_B, y_B), & S_3^{18}(x_B, y_B) \\ \vdots, & \vdots, & \vdots, & \ddots, & \vdots, & \vdots, & \vdots \\ 0, & 0, & 0, & \dots, & S_1^{18}(x_N, y_N), & S_2^{18}(x_N, y_N), & S_3^{18}(x_N, y_N) \end{pmatrix} \cdot \begin{pmatrix} z_1^1 \\ z_2^1 \\ z_3^1 \\ \vdots \\ \vdots \\ z_1^{18} \\ z_2^{18} \\ z_3^{18} \end{pmatrix}$$



ギャップセンサー配置の検証ツール

つまり、基本的にはモデルは行列形式で表現できる。

いわゆる線形システム論？ノルムを使うと H^∞ とかに近い？

なので、この行列のさまざまなふるまいを検証する。



ギャップセンサー配置の検証ツール

m 行 n 列 ($m > n$) の要素を持つ行列 A に対し、その共役転置行列 (随伴行列) A^* をかけた A^*A の n 個の固有値の平方根が、行列 A の特異値と呼ばれる。いま、 A^*A は随伴行列を掛けているため半正定値エルミート行列であるので、行列 A の特異値は実数かつ非負である。逆に、 AA^* の m 個の特異値は大きなほうから n 個は A^*A のものと一致し、残りはすべて 0 となる。

この特異値に対して、 A^*A の n 個の n 次元の固有ベクトルは右特異ベクトル、 AA^* の n 個の m 次元の固有ベクトルは左特異ベクトルと呼ばれる。また、これらの特異ベクトルはそれぞれ互いに直交するように選択することと、一般性を失うことなく単位ベクトルとして選択することができる。いま、固有値 $\sigma_1^2, \dots, \sigma_n^2$ に対応する右特異ベクトルを v_1, \dots, v_n 、左特異ベクトルを u_1, \dots, u_n とする。これらについて、 $Av_i = \sigma_i u_i$ 、 $A^*u_i = \sigma_i v_i$ が成立する。

いま、すべての i について

$$Av_i^T = \sigma_i u_i^T$$

であるので

$$A(v_1^T, v_2^T \dots, v_n^T) = (\text{diag}(\sigma_i)) (u_1^T, u_2^T \dots, u_n^T)$$

である。ここで

$$V = (v_1^T, v_2^T \dots, v_n^T), U = (u_1^T, u_2^T \dots, u_n^T)$$

とすると、 V は nn 行列、 U は mn 行列となり、上述のように u_i, v_i が互いに直交する単位ベクトルとなるように選択しておくと、行列 V, U は正規直交行列である。よって、上記の式は

$$A = U \cdot (\text{diag}(\sigma_i)) \cdot V^T$$

とかける。また、行列 V は正規直交行列であるので、座標系の回転を示すともいうことができる。



ギャップセンサー配置の検証ツール

ここで、ベクトル \mathbf{v} と \mathbf{u} は単位ベクトルであるので、 $\alpha_i \sigma_i \mathbf{u}_i$ の大きさは、特異ベクトル \mathbf{v}_i に対応する特異値 σ_i に依存することになる。つまり、特異値自体が小さな特異ベクトルに対応する動作が全体に及ぼす影響は小さい。そして、行列 \mathbf{A} に対し、 $0 < k < n$ である整数 k について同じ固有ベクトルを持つが階数が k となる行列 \mathbf{A}_k を定義する。いま、この二つ行列の差 $\mathbf{X} = \mathbf{A} - \mathbf{A}_k$ の Frobenius norm は

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i,j} X_{ij}^2}$$

と定義され、Frobenius norm を最小化する \mathbf{A}_k について考える。上述の特異ベクトルと特異値の関係からも直感的にわかる Eckart and Young による定理

$$\min_{Z|\text{rank}(Z)=k} \|\mathbf{A} - Z\|_F = \|\mathbf{A} - \mathbf{A}_k\|_F = \sqrt{\sum_{i=k+1}^n \sigma_i^2}$$

から、最も大きな特異値から k 個を残し、残りすべてを 0 とした行列が \mathbf{A}_k であることがわかる。

以上から、行列 \mathbf{A} の特異値と特異ベクトルについて、それぞれの特異ベクトルはある種の行列が示す動作モードをあらわすことと、行列 \mathbf{A} の動作において小さな特異値 σ_i とそれに対応する特異ベクトルの影響は小さく、かつ行列 \mathbf{A} をある階数 k までで近似する際には、大きいほうから $k+1$ 個め以上の特異値を 0 とする近似が最適であることがわかった。



ギャップセンサー配置の検証ツール

ということで

SVDで分解して特異値と特異ベクトルをしらべる！

縮退した特異値に対応する特異ベクトル

=

擾乱が発生した時に弱いモード

- ・できるだけそろった特異値が出るような配置
- ・縮退した特異ベクトルのモードを制約で消せないか

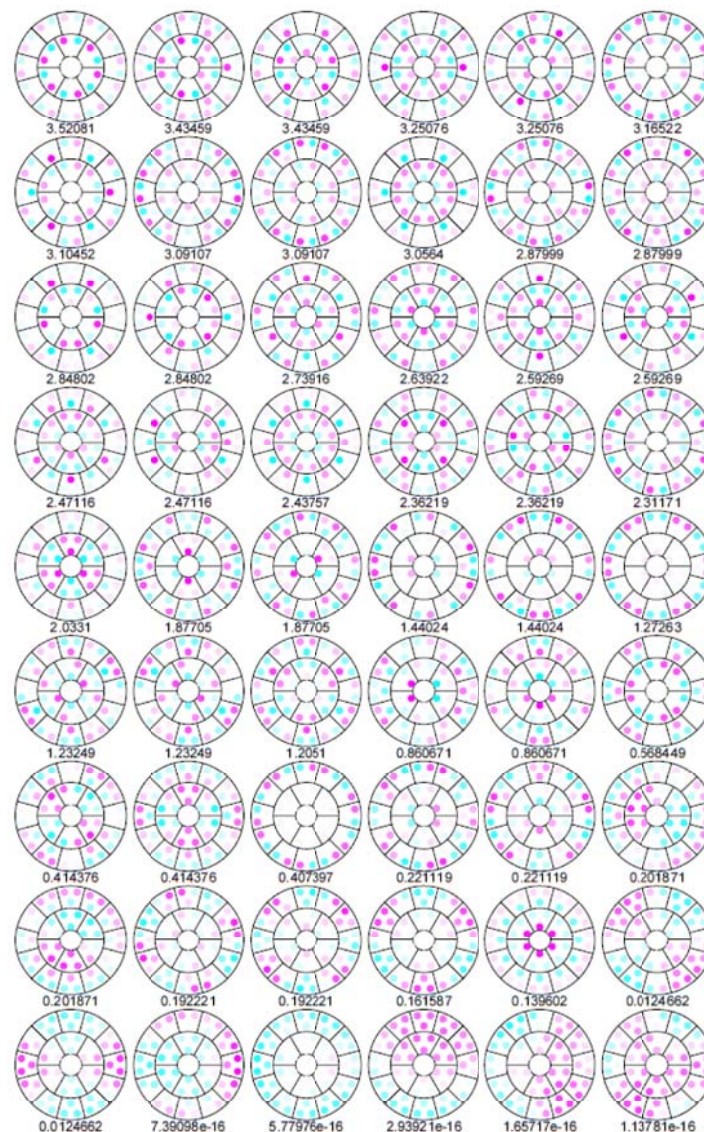


ギャップセンサー配置の検証ツール

わかったこと

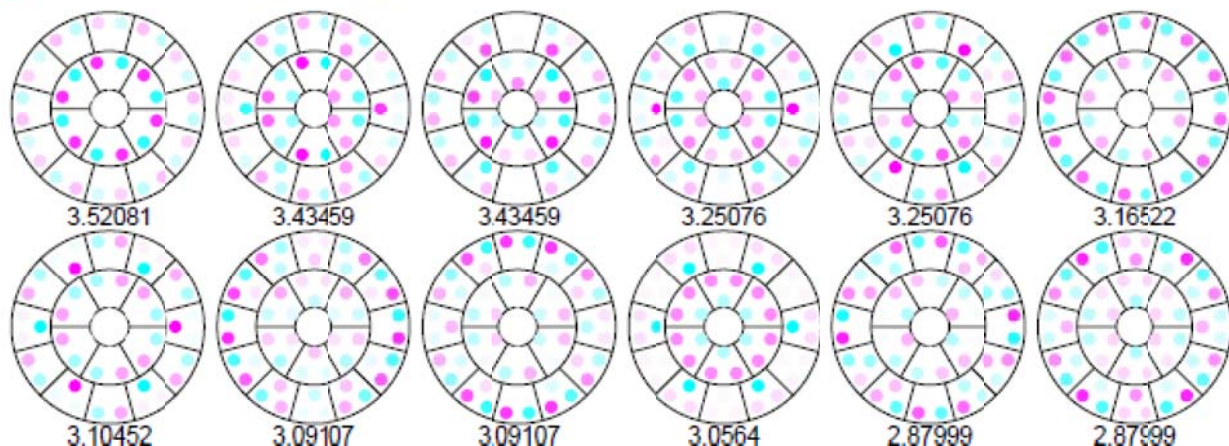
- ・ツールの動作は大丈夫そう
- ・想像に近いベクトルが縮退してる
- ・縮退を完全に消すのは一手間

model	mode 1	mode 2	mode 3	mode 4	備考
Model A I	7 (5)	4 (2)	2	3 (1)	最もシンプル
Model A II	7 (5)	4 (2)	2	1	辺間を1/4,3/4に
Model A III	7 (5)	5 (2)	3 (2)	3 (1)	辺間を1/4,1/2,3/4で分散
Model A IV	6 (4)	3 (1)	1	2 (0)	内・外周間に1つずつ追加
Model B I	4 (2)	3 (2)	1	0	
Model B II	4 (2)	3 (1)	1	0	
Model B III	4 (2)	3 (1)	1	0	
Model B IV	3 (1)	2 (0)	0	0	



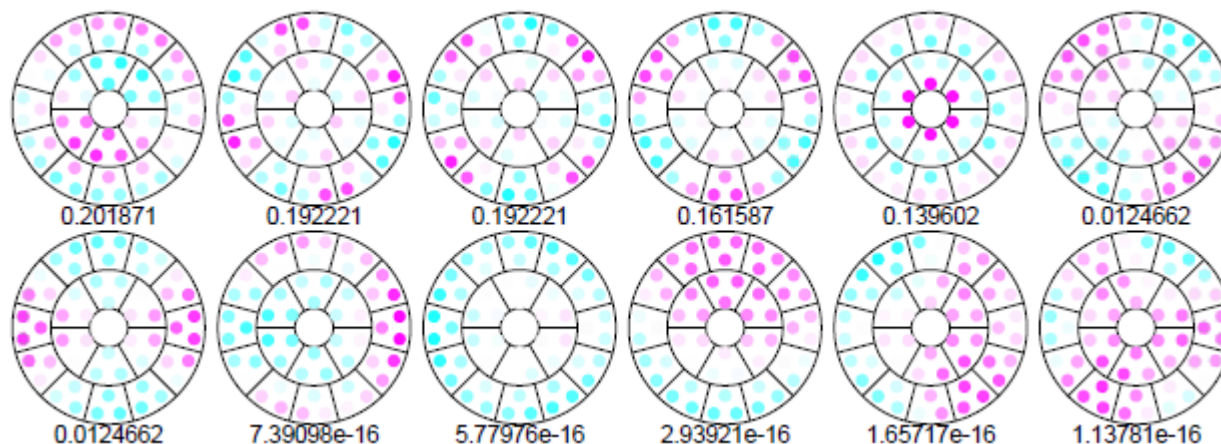


ギャップセンサー配置の検証ツール



↑ 特異値が大きなベクトルの例

特異値が小さいか縮退しやすいベクトルの例 ↓





ギャップセンサー配置の検証ツール

- ・検証ツールは完成しました 遊んでください！
- ・全体がゆるく変形するモードは解消しにくい
- ・SHCamなどの情報を反映させるとどうなるか