

## KMAP の例題 (8) – 航空機のエンジン片側 2 発故障時のシミュレーション

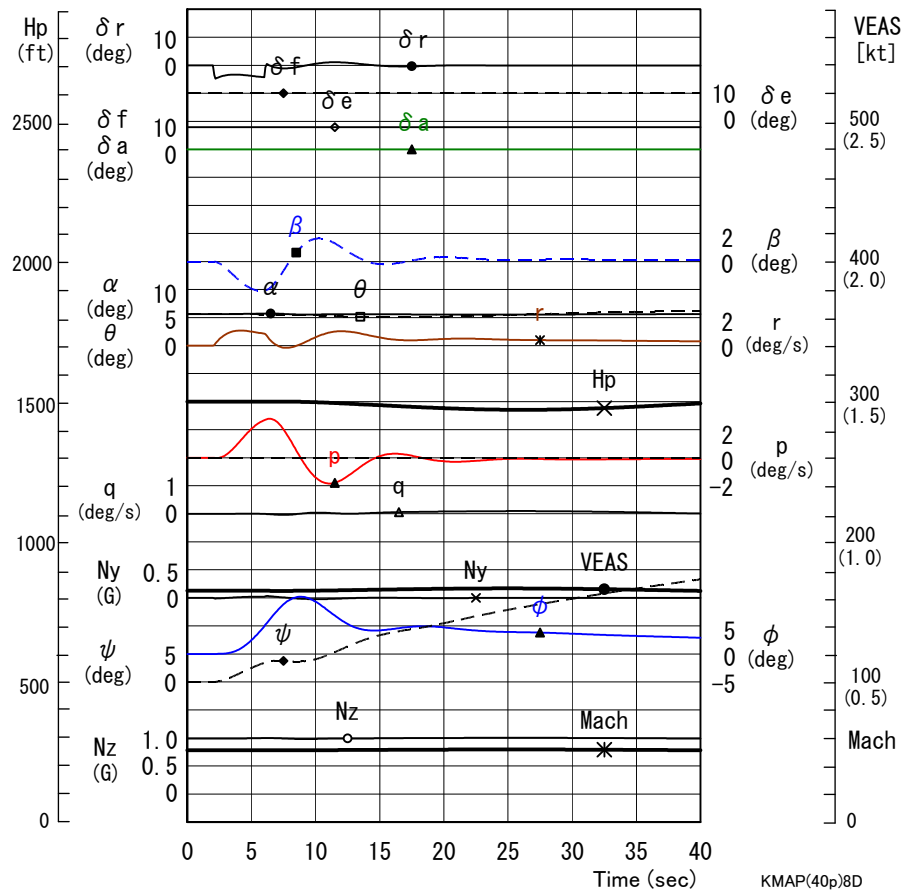
H23.9.17 片柳亮二

KMAP には、各種関数を用意されており、それらを用いて故障モードのシミュレーションを行うことができる。ここでは、安定増加装置ヨーダンパ付きの機体の片側のエンジン 2 発が故障した場合のシミュレーションを行ってみよう。これは機体のトリム能力のチェックのためである。

C:\¥KMAP フォルダ内の、“KMAP\*\*実行スタートファイル.BAT” (\*\*はバージョン番号) バッチファイルをダブルクリックすると、解析プログラム KMAP が起動する。次に、以下のようにキーイン(“ ” 部分)して、制御則修正の準備を行う。

- “1” ← 航空機などの一般解析
- “1” ← 航空機の運動・制御系解析
- “3” ← 例題ファイルをコピー利用
- “10” ← 大型旅客機のヨーダンパ有りのデータを利用
- “RK1” ← 適当なファイル名記入
- “0” ← シミュレーション
- “0” ← 修正なし
- “0” ← 修正なし

これによって、まず正常な場合のシミュレーションを行うと、次頁の結果が得られる。シミュレーション結果は、“C:\¥KMAP¥エクセル図”のフォルダを開け、“KMAP(時歴 40P)8D.x1s”のエクセル図を表示し、データ部分の適当な所にカーソルを置いて右クリックして、“データ更新”を行うと次のようなタイムヒストリーが表示できる。(ワードファイルには“拡張メタファイル”として貼り付ける)



(正常な場合のラダー操舵応答)

この計算の実行時に、画面には次のようなトリム状態が表示される。  
 これから、推力はエンジン4発で23.5(tf)発生していることがわかる。

.....CALL RYOJI.....

$\alpha = 0.56250E+01$     $\theta = 0.56250E+01$     $\phi = 0.00000E+00$     $\delta e = -0.20312E+01$   
 $\delta a = 0.00000E+00$     $\delta r = 0.00000E+00$     $\psi \text{ DOT} = 0.00000E+00$     $\text{Thrust} = 0.23516E+05$   
 KMAP Analysis has started ...

このとき、片側のエンジン2発が故障して推力が0になったと仮定すると、エンジンは片側21mと12mに装着されているので、

$23500/4 \times (21+12) = 194 \text{ (tf} \cdot \text{m)}$  のモーメントがヨー軸まわりに作用す

ることになる。また、推力も 11.8(tf)失われることになる。パイロットは故障後に何も操作しないとしてシミュレーションしてみよう。

引き続き、KMAP画面で、次のように操作していく。

“1” ←同じデータを用いて解析開始

“0” ←シミュレーション

このとき、次のように表示される。

----- (インプットデータ修正) (前半部) -----

- 1 = 滑走路位置
  - 2 = 慣性モーメント
  - 3 = 機体重量, 翼面積,  $S_{\text{plan}}$ , 平均空力翼弦
  - 4 = 重心
  - 5 = センサ位置
  - 6 = 計算時間 ( $t_{\text{max}} \geq 40$  秒 (航空機),  $\geq 4$  秒 (EIGE 解析))
  - 7 = 舵角最大最小値
  - 8 = エンジンシャフトモーメント
  - 9 = uカスト(外乱)入力
  - 10 = vカスト(外乱)入力
  - 11 = wカスト(外乱)入力
  - 12 = 外力入力
  - 13 = パラシューター
  - 14 = インプットデータのタイトル
- 何を修正しますか? (番号キー), 修正なし(完了)=0

ここで、“12” (外力入力)を選択すると、次のように表示される。

```

....<Gairyoku>.....
t1yaw(s)      = 0.2000E+02
t2yaw(s)      = 0.2500E+02
Yaw(kgf*m)    = 0.0000E+00
INPUT(99 入力元データのまま) --> t1yaw(s)      =

```

ここに値を代入すると、t1yaw(秒)～t2yaw(秒)の間、ヨー軸まわりのモーメント Yaw(単位は kgf・m)が機体に作用するようになっている。そこで、今回次のように入力してみる。

“20” ←t1yaw(s)  
 “40” ←t2yaw(s)  
 “194000” ←Yaw(kgf\*m)  
 “0” ←修正完了

このとき，次のように表示される．

----- (インプットデータ修正) (後半部) -----

- 1 = 制御則  
 (・制御ブロックにおける各ブロックの入出力関係をインプットデータに記述 )  
 2 = 初期飛行条件 (高度, 速度, G)  
 3 = パイロット操舵  
 (・U1~U5 を時間の折れ線関数として設定して利用できる )  
 4 = デバッグ時間  
 (・シミュレーション時に各状態変数を 0.1 秒毎に表示する開始時間 )  
 5 = 空力 (MACH 関数)  
 6 = 空力 (MACH, ALP 関数)  
 7 = 空力 (MACH, Hp 関数)  
 8 = 補間関数

参考 (①Z500, X50, H500, U40, R40, E80, D4 まで可能. 制御則は 900 行まで可能. )  
 (②変数 Z は, リミット関数以外は 2 回以上定義しないこと. )  
 (③舵角  $Z1 = \delta e$ ,  $Z2 = \delta a$ ,  $Z3 = \delta f$ ,  $Z4 = \delta r$ ,  $Z5 = \delta T$ . 同様にパイロット入力は U1~U5. )  
 (④x, y, z 外力 (kgf) を Z6, Z7, Z8 で追加. x, y, z モーメント (kgf·m) を Z9, Z10, Z11 で追加. )

●何を修正しますか? (番号キー), 修正なし(完了)=0

ここで，“3” (パイロット操舵) を選択すると，次のように表示される．

	時間	操作量 (トリムからの変化分)
1. NDe-----> 2 T, De	0.000 60.000	0.000 0.000
2. NDa-----> 2 T, Da	0.000 60.000	0.000 0.000
3. NDf-----> 2 T, Df	0.000 60.000	20.000 20.000
4. NDr----->10 T, Dr	0.000 2.000	0.000 0.000

	2.100	-5.000
	6.000	-5.000
	6.100	0.000
	13.000	0.000
	13.100	0.000
	17.000	0.000
	17.100	0.000
	60.000	0.000
5. N (THRUS) -> 4		
T, D (THR)	0.000	0.000
	2.000	0.000
	4.000	0.000
	200.000	0.000

何番目の操舵を修正しますか?(番号キー), 終了=98

これらは、通常はパイロット操舵入力であるが、本解析はエンジン 2 発故障を模擬するため、5 番目の推力 (THRUST) を  $t=20$  秒から半分の 117500 (kgf) を差し引く。具体的には次のように行う。

“5” ← THRUST 修正  
 “99” ← 元データのまま  
 “99” ← 元データのまま  
 “99” ← 元データのまま  
 “20” ← 時間変更  
 “99” ← 元データのまま  
 “20.1” ← 時間変更  
 “-11750” ← 時間変更  
 “99” ← 元データのまま  
 “-11750” ← 時間変更

この結果、次のように変更が表示される。

5. N (THRUS) -> 4		
T, D (THR)	0.000	0.000
	20.000	0.000
	20.100	-117500.000
	200.000	-117500.000

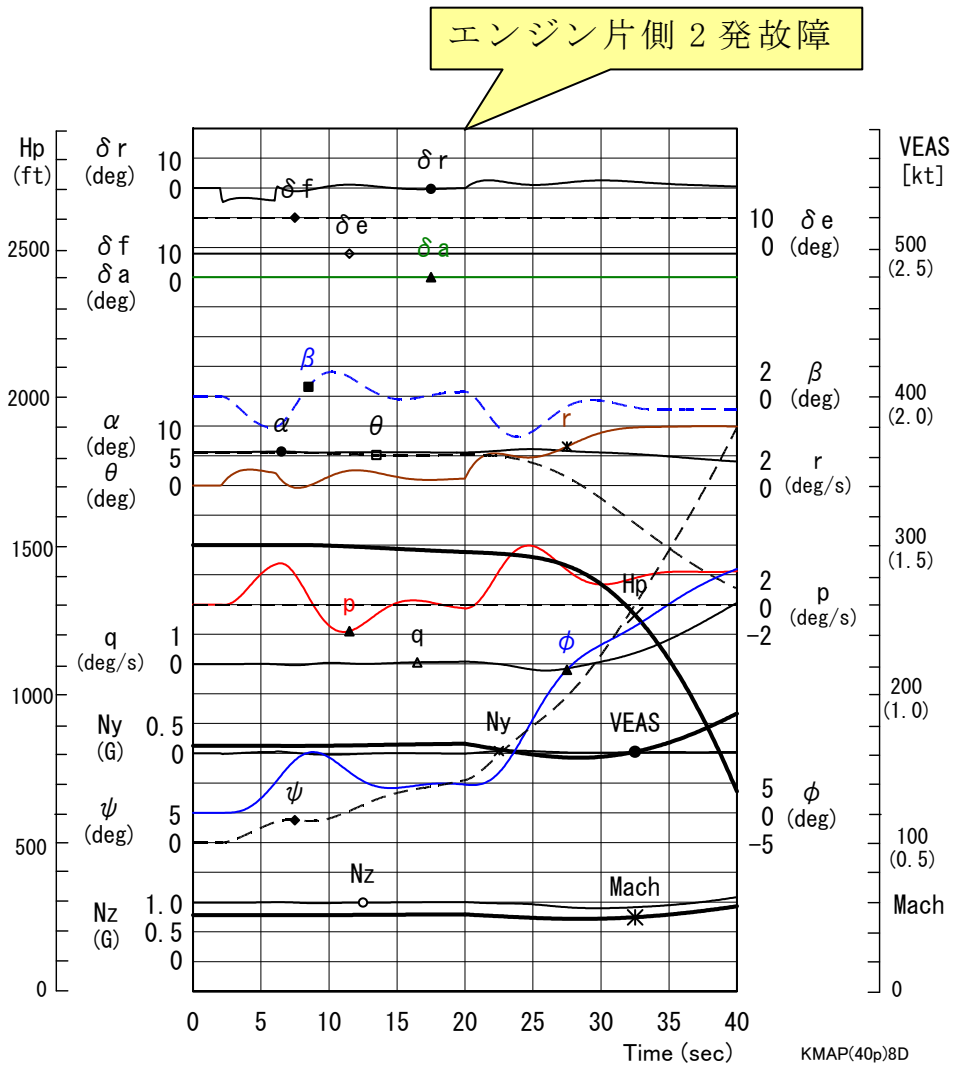
何番目の操舵を修正しますか?(番号キー), 終了=98

この後は、次のように操作すると計算が実行される。

“98” ←終了

“0” ←修正なし

シミュレーション結果のタイムヒストリーを次に示す。



( $t=20$  秒でエンジン片側 2 発故障時のシミュレーション)