

行政院國家科學委員會專題研究計畫 成果報告

智慧型導引技術

計畫類別：個別型計畫

計畫編號：NSC94-2623-7-009-005-

執行期間：94年01月01日至94年12月31日

執行單位：國立交通大學電機與控制工程學系(所)

計畫主持人：梁耀文

共同主持人：廖德誠

報告類型：完整報告

處理方式：本計畫可公開查詢

中華民國 95 年 3 月 17 日

國防科技學術合作研究計畫成果報告

*
* 智 慧 型 導 引 技 術 *
*
*

計畫編號： NSC 94-2623-7-009-005

執行期間： 94 年 1 月 1 日至 94 年 12 月 31 日

計畫主持人： 梁耀文

共同主持人： 廖德誠

執行單位： 國立交通大學電機與控制工程學系

中華民國 94 年 12 月 31 日

摘要

近年來，由於中共積極發展彈道飛彈，使得我方受到的威脅日益增加。另一方面，由於國軍先進飛彈的獲得幾乎都受制於歐美強國，取得不易。而自行研發攔截飛彈將花費龐大並且曠日費時，研發期間又較易招致國際阻擾。因此，如果能在中科院既有良好的飛彈製造基礎上，增加智慧型的導引系統將能有效的反制解放軍逐年增加部署彈道飛彈的威脅。因此，本計畫主要目的在於研發新一代的非容錯控制技術使系統具有智慧型的容錯控制能力及可靠性，同時能有效的提升彈道預估的精準程度，達到快速且有效的攔截目的。本計畫過去一年內已完成了下列研究項目：

- (1) 推導一般基因演算法則應用於來襲飛彈之彈道軌跡估測。
- (2) 建構適應性基因演算法則以應用於具有高度移動能力之來襲飛彈的彈道軌跡估測。
- (3) 經由估測來襲飛彈之彈道來決定我方攔截飛彈之攔截軌跡。
- (4) 推導彈體座標與地面座標之變換關係以執行攔截軌跡追蹤之任務。
- (5) 利用可變結構控制理論來設計達成攔截軌跡追蹤目的之被動式容錯控制法則。
- (6) 發展控制系統致動器線上即時錯誤偵測與診斷之機制。
- (7) 發展結合錯誤偵測結果之攔截軌跡追蹤主動式容錯控制法則。
- (8) 比較一般(非容錯)設計與兩容錯設計之閉迴路控制系統在致動器正常及異常下之性能表現。
- (9) 針對具有模式誤差或外界干擾之系統進行容錯控制之穩健性分析。
- (10) 進行電腦模擬來驗證所設計之容錯控制的有效性。

目錄

摘要	I
目錄	II
圖目錄	IV
1 前言	1
2 遺傳演算法在飛彈導引上之應用	3
2.1 遺傳演算流程	3
2.1.1 遺傳演算法的參數設定	4
2.1.2 遺傳演算法的演算步驟	6
2.2 導引律設計	8
2.2.1 飛彈導引律的動態方程式	8
2.2.2 目標逃逸運動	9
2.2.3 以基因演算法為基礎的補償設計	10
2.3 數值模擬結果	11
2.3.1 目標具螺旋逃逸模式	12
2.3.2 目標具方形-螺旋逃逸模式	15
3 HAVE DASH II 飛彈系統之動態數學模式	20
3.1 座標軸系統	20
3.1.1 慣性座標系	20
3.1.2 彈體座標系	20
3.1.3 座標系轉換	21
3.2 飛彈模型	22

4	控制器之設計	29
4.1	非容錯控制律之設計	29
4.2	容錯控制律之設計	31
4.2.1	問題描述	31
4.2.2	被動式容錯控制律之設計	32
4.2.3	主動式容錯控制律之設計	35
5	錯誤偵測與診斷觀察器之設計	37
5.1	問題描述	37
5.2	觀察器之設計	40
6	模擬結果與討論	43
6.1	飛彈參數設定	43
6.2	觀察器之設計	44
6.3	控制器設計	46
6.4	模擬結果與討論	50
7	結論	89
	參考書目	91
	附錄	94

圖目錄

圖 2.1	遺傳演算法之演算流程圖	4
圖 2.2	三種不同交配的方法	6
圖 2.3	自然界與基因演算的突變機制	7
圖 2.4	飛彈攔截過程的幾何圖形	8
圖 2.5	四種逃逸模式	9
圖 2.6	導引律的方塊圖	11
圖 2.7	目標具螺旋逃逸和 GA 學習結果 ($a_t = 0.1\text{km}/\text{sec}^2$)	12
圖 2.8	不同參數的時間響應 ($a_t = 0.1\text{km}/\text{sec}^2$)	13
圖 2.9	RPTN 和具 GA 補償的 RTPN 之飛彈加速度比較 ($a_t = 0.1\text{km}/\text{sec}^2$)	13
圖 2.10	目標具螺旋逃逸和 GA 學習結果 ($a_t = 0.25\text{km}/\text{sec}^2$)	14
圖 2.11	不同參數的時間響應 ($a_t = 0.25\text{km}/\text{sec}^2$)	14
圖 2.12	RPTN 和具 GA 補償的 RTPN 之飛彈加速度比較 ($a_t = 0.25\text{km}/\text{sec}^2$)	15
圖 2.13	目標具方形-螺旋逃逸和 GA 學習結果 ($a_t = 0.1\text{km}/\text{sec}^2$)	16
圖 2.14	不同參數的時間響應 ($a_t = 0.1\text{km}/\text{sec}^2$)	16
圖 2.15	RPTN 和具 GA 補償的 RTPN 之飛彈加速度比較 ($a_t = 0.1\text{km}/\text{sec}^2$)	17
圖 2.16	目標具方形-螺旋逃逸和 GA 學習結果 ($a_t = 0.25\text{km}/\text{sec}^2$)	18
圖 2.17	不同參數的時間響應 ($a_t = 0.25\text{km}/\text{sec}^2$)	18
圖 2.18	RPTN 和具 GA 補償的 RTPN 之飛彈加速度比較 ($a_t = 0.25\text{km}/\text{sec}^2$)	19
圖 3.1	慣性座標與彈體座標轉換	21
圖 3.2	HAVE DASH II 飛彈系統外觀圖	22
圖 3.3	HAVE DASH II 飛彈系統結構圖	22
圖 5.1	偵測與診斷架構圖	38
圖 6.1	非容錯控制設計，在所有致動器正常之下其追蹤性能之表現	53

圖 6.2	非容錯控制設計，在所有致動器正常之下其致動器之狀態	53
圖 6.3	非容錯控制設計，在所有致動器正常之下其飛彈之追蹤誤差	54
圖 6.4	非容錯控制設計，在所有致動器正常之下其飛彈之推力	54
圖 6.5	非容錯控制設計， δ_1 於 $t=0.3$ 秒卡死其追蹤性能之表現	55
圖 6.6	非容錯控制設計， δ_1 於 $t=0.3$ 秒卡死其致動器之狀態	55
圖 6.7	非容錯控制設計， δ_1 於 $t=0.3$ 秒卡死其飛彈之追蹤誤差	56
圖 6.8	非容錯控制設計， δ_1 於 $t=0.3$ 秒卡死其飛彈之推力	56
圖 6.9	非容錯控制設計， δ_2 於 $t=0.3$ 秒卡死其追蹤性能之表現	57
圖 6.10	非容錯控制設計， δ_2 於 $t=0.3$ 秒卡死其致動器之狀態	57
圖 6.11	非容錯控制設計， δ_2 於 $t=0.3$ 秒卡死其飛彈之追蹤誤差	58
圖 6.12	非容錯控制設計， δ_2 於 $t=0.3$ 秒卡死其飛彈之推力	58
圖 6.13	非容錯控制設計， δ_3 於 $t=0.3$ 秒卡死其追蹤性能之表現	59
圖 6.14	非容錯控制設計， δ_3 於 $t=0.3$ 秒卡死其致動器之狀態	59
圖 6.15	非容錯控制設計， δ_3 於 $t=0.3$ 秒卡死其飛彈之追蹤誤差	60
圖 6.16	非容錯控制設計， δ_3 於 $t=0.3$ 秒卡死其飛彈之推力	60
圖 6.17	非容錯控制設計， δ_4 於 $t=0.3$ 秒卡死其追蹤性能之表現	61
圖 6.18	非容錯控制設計， δ_4 於 $t=0.3$ 秒卡死其致動器之狀態	61
圖 6.19	非容錯控制設計， δ_4 於 $t=0.3$ 秒卡死其飛彈之追蹤誤差	62
圖 6.20	非容錯控制設計， δ_4 於 $t=0.3$ 秒卡死其飛彈之推力	62
圖 6.21	所有致動器皆正常之下，針對 δ_1 故障所設計之被動式控制器與非容錯 控制器其追蹤性能的表現	63
圖 6.22	所有致動器皆正常之下，針對 δ_1 故障所設計之被動式控制器與非容錯 控制器其致動器之狀態	63
圖 6.23	所有致動器皆正常之下，針對 δ_1 故障所設計之被動式控制器與非容錯 控制器其飛彈之追蹤誤差	64

圖6.24 所有致動器皆正常之下，針對 δ_1 故障所設計之被動式控制器與非容錯 控制器其飛彈之推力	64
圖6.25 所有致動器皆正常之下，針對 δ_2 故障所設計之被動式控制器與非容錯 控制器其追蹤性能的表現	65
圖6.26 所有致動器皆正常之下，針對 δ_2 故障所設計之被動式控制器與非容錯 控制器其致動器之狀態	65
圖6.27 所有致動器皆正常之下，針對 δ_2 故障所設計之被動式控制器與非容錯 控制器其飛彈之追蹤誤差	66
圖6.28 所有致動器皆正常之下，針對 δ_2 故障所設計之被動式控制器與非容錯 控制器其飛彈之推力	66
圖6.29 所有致動器皆正常之下，針對 δ_3 故障所設計之被動式控制器與非容錯 控制器其追蹤性能的表現	67
圖6.30 所有致動器皆正常之下，針對 δ_3 故障所設計之被動式控制器與非容錯 控制器其致動器之狀態	67
圖6.31 所有致動器皆正常之下，針對 δ_3 故障所設計之被動式控制器與非容錯 控制器其飛彈之追蹤誤差	68
圖6.32 所有致動器皆正常之下，針對 δ_3 故障所設計之被動式控制器與非容錯 控制器其飛彈之推力	68
圖6.33 所有致動器皆正常之下，針對 δ_4 故障所設計之被動式控制器與非容錯 控制器其追蹤性能的表現	69
圖6.34 所有致動器皆正常之下，針對 δ_4 故障所設計之被動式控制器與非容錯 控制器其致動器之狀態	69
圖6.35 所有致動器皆正常之下，針對 δ_4 故障所設計之被動式控制器與非容錯 控制器其飛彈之追蹤誤差	70

圖6.36 所有致動器皆正常之下，針對 δ_4 故障所設計之被動式控制器與非容錯 控制器其飛彈之推力	70
圖6.37 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_2 發生故障之追蹤性能 表現	71
圖6.38 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_2 發生故障之致動器狀態	71
圖6.39 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_2 發生故障之飛彈追蹤 誤差	72
圖6.40 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_2 發生故障之飛彈推力 .	72
圖6.41 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_3 發生故障之追蹤性能 表現	73
圖6.42 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_3 發生故障之致動器狀態	73
圖6.43 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_3 發生故障之飛彈追蹤 誤差	74
圖6.44 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_3 發生故障之飛彈推力 .	74
圖6.45 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_4 發生故障之追蹤性能 表現	75
圖6.46 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_4 發生故障之致動器狀態	75
圖6.47 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_4 發生故障之飛彈追蹤 誤差	76
圖6.48 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_4 發生故障之飛彈推力 .	76
圖6.49 針對 δ_1 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之追蹤性能	77
圖6.50 針對 δ_1 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之致動器 變化狀態	77
圖6.51 針對 δ_1 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之飛彈追 蹤誤差	78

圖6.52 針對 δ_1 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之飛彈推力曲線	78
圖6.53 針對 δ_1 於0.3秒時卡死，錯誤偵測與診斷機制所提供之residual signal與alarm signal	79
圖6.54 針對 δ_1 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計，致動器在0~2秒之運作狀態	79
圖6.55 針對 δ_2 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之追蹤性能	80
圖6.56 針對 δ_2 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之致動器變化狀態	80
圖6.57 針對 δ_2 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之飛彈追蹤誤差	81
圖6.58 針對 δ_2 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之飛彈推力曲線	81
圖6.59 針對 δ_2 於0.3秒時卡死，錯誤偵測與診斷機制所提供之residual signal與alarm signal	82
圖6.60 針對 δ_2 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計，致動器在0~2秒之運作狀態	82
圖6.61 針對 δ_3 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之追蹤性能	83
圖6.62 針對 δ_3 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之致動器變化狀態	83
圖6.63 針對 δ_3 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之飛彈追蹤誤差	84
圖6.64 針對 δ_3 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之飛彈推力曲線	84

圖6.65 針對 δ_3 於0.3秒時卡死，錯誤偵測與診斷機制所提供之 residual signal 與 alarm signal	85
圖6.66 針對 δ_3 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計，致動器 在0~2秒之運作狀態	85
圖6.67 針對 δ_4 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之追蹤性能	86
圖6.68 針對 δ_4 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之致動器 變化狀態	86
圖6.69 針對 δ_4 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之飛彈追 蹤誤差	87
圖6.70 針對 δ_4 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計之飛彈推力	87
圖6.71 針對 δ_4 於0.3秒時卡死，錯誤偵測與診斷機制所提供之 residual signal 與 alarm signal	88
圖6.72 針對 δ_4 於0.3秒時卡死，非容錯式、被動式與主動式容錯設計，致動器 在0~2秒之運作狀態	88

第一章

前言

民國八十年發生波灣戰爭，使得對岸中共的國防與軍事理論、戰略、戰備等事務之思維產生革命性的改變。於波灣戰爭開戰前夕，北京的軍事專家一致認為，伊拉克擁有非常強大的武力，美國將再一次掉入泥淖，陷入另一個越戰。但出乎意料，由於美國使用最新的武器與戰術，速戰速決，使伊拉克在短期內兵敗撤軍。而在民國八十八年的科索夫之戰，以美國為主的北約部隊更在未動用地面部隊的情況下，藉由智慧型的精密打擊武器，以空中力量屈服南斯拉夫。這都說明了智慧型科技優勢在戰爭中所具有的潛力。

由於1972年美蘇反彈道飛彈條約(ABM Treaty)及其1976年協定(Protocol)的限制，美國早期(1970-1983)在攔截彈道飛彈方面的研究相當有限，只有零星的實驗性計畫如：Homing Overlay Experiment (HOE)，及Exoatmospheric Reentry Vehicle Interceptor System (ERIS)等。1983年開始的戰略防衛計畫(Strategic Defense Initiative, SDI)曾投入相當大的經費，但由於爭議性相當高，在1988年以後幾乎完全終止。許多研究及構想均仍停留在可行性分析的階段，並沒有完成接近可服役的武器系統。在1990年的波斯灣戰爭，伊拉克對以色列及沙烏地阿拉伯的軍事設施及都會區發射多枚飛毛腿(Scud)飛彈，美國在以色列及沙烏地阿拉伯佈署愛國者二型飛彈(Patriot Advanced Capability Level-2, PAC-2)攔截。根據戰後的分析愛國者的攔截高度大約在10公里上下，其平均命中率則約在45%左右。由於攔截高度不夠，攔截命中之後仍有造成地面相當傷害的報導。除了積極研發愛國者三型(PAC-3)外，美國

目前研發已接近可部署的反飛彈系統包括標準型飛彈(Standard Missile)為船艦防衛系統(1997年1月24日試射成功攔截射程400公里的藍斯飛彈(Lance)，攔截高度約12公里)，及改良型鷹式三型(Hawk III)飛彈。

以目前國軍先進精密武器的採購獲得，幾乎皆受制於歐美強國，例如，美國之所以會同意出售先進中程空對空飛彈給台灣，是因為台灣自製的天劍二型(TC-II)中程空對空飛彈，擁有與先進中程空對空飛彈幾乎同等級的性能，美國認為有錢可賺，順勢傾銷同等級武器給台灣，進一步壓縮台灣好不容易建立的本土軍火工業。另外美國在出售給予台灣的F-16戰機，卻不附帶出售可以連結天劍飛彈與F-16戰機的空電系統軟體，用意在於強迫台灣購買其中程空對空飛彈，更誇張的是台灣買的飛彈還要存放在美國境內，美國的心態與法國在出售幻象機，順便傾銷雲母飛彈是一樣的，也形成台灣同時擁有世界上三種最先進中程空對空飛彈的世界紀錄。同樣的美國眼見台灣雄風家族技術成熟，更索性傾銷魚叉飛彈，除了作生意賺錢外，連帶削弱台灣自主國防工業生機，愛國者防禦飛彈也是類似的情形。所以面對台海的威脅，我們更應戮力於發展智慧型的防禦武器，提昇技術層次，達到國防自主的目標。

本計畫利用基因演算法提出來襲彈道飛彈之軌跡估測，經由估測之彈道軌跡引導我方反飛彈鑽入敵方彈道中以提高攔截的命中率。以及達到即時、有效的攔截目標。此外，本計畫也引進了容錯控制之技術使攔截技術更具有智慧型的容錯控制能力及高可靠性。

本報告之結構如下：第二章介紹遺傳演算法在飛彈導引之應用。第三章針對HAVE DASH II飛彈系統做完整的數學模型介紹。第四章利用可變結構控制方法推導主動式(active)及被動式(passive)容錯控制律。第五章介紹錯誤偵測診斷與觀察器的設計。經由觀察器所產生的殘量可以得知發生故障的位置與嚴重程度。第六章節將針對HAVE DASH II飛彈系統作尾翼故障之偵測診斷與容錯控制之模擬分析與比較。最後，在第七章提出結論。

第二章

遺傳演算法在飛彈導引上之應用

遺傳演算法是由 John Holland 教授於 1975 年首度提出，主要是藉由達爾文進化論中的適者生存的觀點，模擬自然界中的物競天擇與物種遺傳之演化方式，對某些特定問題進行最佳解的搜尋。經過了許多年的研究與發展，遺傳演算法被證明為最有效率的最佳化搜尋方法，且可廣泛被應用在許多實際的問題上。由於遺傳演算法能在空間中更有效率的搜尋最佳解，雖然遺傳演算法從 1975 年提出至今已有 30 年的時間，但因為這個特點，使得遺傳演算法歷久不衰，且更有許多專家學者競相投入這個領域，繼續對遺傳演算法做更深入的探索。在此報告中，我們將使用遺傳演算法來判別和補償在飛彈導引律 (missile guidance) 中的目標逃逸 (target's maneuver)。

2.1 遺傳演算流程

遺傳演算法是一種透過基因遺傳之最佳化搜尋法則，其基本精神在於模仿自然界中物競天擇及適者生存的生物法則。亦即選擇物種中具有較佳特性的母代，並隨機交換彼此的基因，以期能產生比母代更優秀的子代，經由如此不斷的擇優與交配，優良的基因將被保留，而劣質基因將會被慢慢淘汰，如此可產生具有最佳適應性之個體。遺傳演算法依據上述過程有三個主要程序，分別為擇優 (Selection)、交配 (Crossover) 及突變 (Mutation)，其相關演算流程如圖 2.1 所示。

在遺傳演算法中，每一個個體皆稱為染色體 (Chromosome)，且每一染色體的基因值皆由亂數產生，而所謂的群體 (Population) 是指每一世代的染色體所形成的集

合。在每一世代中，每一個染色體會互相競爭，如果某一染色體比較能適合環境而生存下來，則其具有較高的適存值(Fitness value)，有較高適存值的染色體將可複製出較多的子代，接著再從這些子代中選擇較優的染色體來進行交配(Crossover)，期望可產生適存度更高的下一代。然而，為了避免錯過某些有用的資訊，在遺傳演算法中加入突變(Mutation)的機制來產生有用的資訊，通常突變的機率(Mutation Rate)是很低的。經由如此不斷的演化下去，將產生具有非常高適存值的染色體，此染色體便是我們所需要的最佳解。

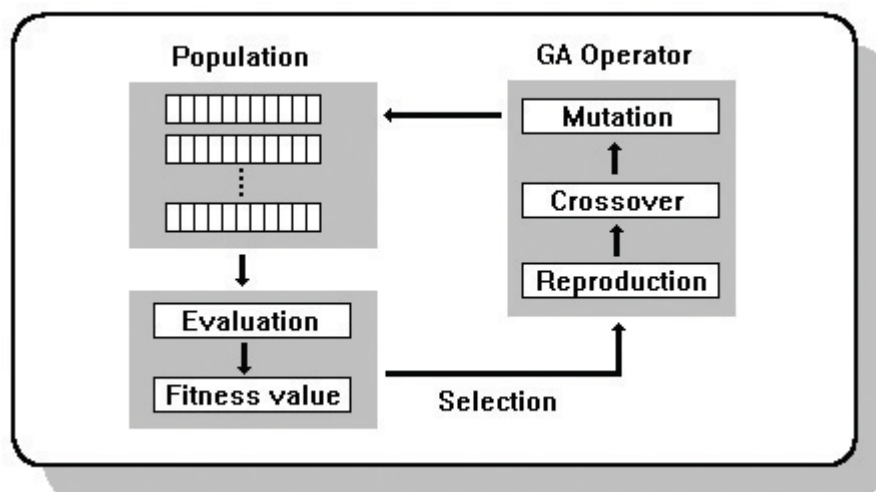


圖 2.1 遺傳演算法之演算流程圖

2.1.1 遺傳演算法的參數設定

在採用遺傳演算法來解最佳化問題時，首先需先決定編碼和解碼的方式，再針對所要解決的問題來定義適應函數和初始族群。由於參數之間的關聯性非常的大，所以不同的參數設定將會產生不同的搜尋結果。以下將介紹遺傳演算法的參數設定準則。

a. 初始族群

所謂的族群就是指每一世代中共同生存的個體。族群最主要的目的是為了提供具有不同表現的個體，由於個體間的具有差異性，在天擇與基因交換的機制影響下，才能產生更好的下一代族群。因此，在遺傳演算法的開始，將以隨機的方式產

生S個染色體，此S個染色體即所謂的初始族群。染色體的個數依問題的複雜度而有所不同，一般而言，越複雜的問題所需的染色體個數也會越多，然而，一旦染色體數目過多，將會影響演算法的收斂速度。

b. 染色體

在自然界中，遺傳性狀的最基本單位被稱為基因 (gene)，而相關連的 gene 則以一連串的字串 (string) 方式存在於染色體中。因此，藉由染色體便可決定該個體之遺傳特徵。在遺傳演算法中，每一個染色體代表參數搜尋中之一組解，為所有基因組合而成之字串，其型態可表示成其中，P 代表一個染色體，為 n 位元所組成的二進位碼，m 代表所有基因個數。

c. 編碼與解碼方式

無論用什麼方法來做 gene 編碼，在執行 reproduction、交配或突變等運算後，所出現的基因表示法必須是有效的表示方式。亦即，好的基因編碼方法不但能夠表示所有合法的基因組合，且不會因為執行遺傳演算法的基本運算而演化出不合法的個體。本報告採用二進位編碼方法，將所欲搜尋之參數轉換成 n 位元的二進制碼；當 n 值越大時，相對的編碼長度也會越長，雖然參數搜尋之精確度會越高，但將佔據較多的計算時間和記憶體空間。

d. 適應函數

根據達爾文的進化論觀點，對環境適應程度越高的個體將有更高的生存機率，且更有機會繁衍下一代。在自然界中，生物的演化是天擇的結果，而天擇所依據的即為該個體的適應函數。在遺傳演算法中，同樣需要適應函數的機制，在此，適應函數的定義變成越符合所要求的個體或解，其適應值將越高。適應函數在遺傳演算法中將扮演引導演化方向的角色。由於適應函數的存在，將使得演化過程不是漫無目的，而是朝著我們所要求的方向前進。而適應函數是評估族群中染色體適應能力之性能指標，依個別問題的不同而有所不同。適應函數值愈高，表示此染色體適合能力愈佳，比較有可能將本身的優良基因遺傳給下一代。

2.1.2 遺傳演算法的演算步驟

a. 複製

複製機制是依據適應函數值的大小來決定子代中是否保留或淘汰某基因，適應函數愈高則染色體愈可能完全複製到子代中，子代之適應函數將因而更提高。

b. 交配

在自然界中，生物常常藉由交配而達到基因交換的目的。交配後的子代將混合親代的特徵，交配的目的是希望能夠製造出具備親代各項優點的新個體。然而，子代也可能同時遺傳親代的某些缺點，亦即，交配並不一定能保證造出更好的子代，但是透過天擇的結果，較差的子代將會慢慢被淘汰。在遺傳演算法中，仍然保留了自然界中的交配機制，以達成混合母代優點的目的。交配的過程是先隨機選取交配池中的任兩個染色體，再彼此交換基因而組成新的染色體，藉由母代優秀基因將產生更優秀之子代。然而，必須強調的一點是，並非所有在交配池中的染色體均可進行交配程序，而是由預先設定之交配機率所控制。在實作上，交配的方法可分為圖 2.2 中的三種不同法則。

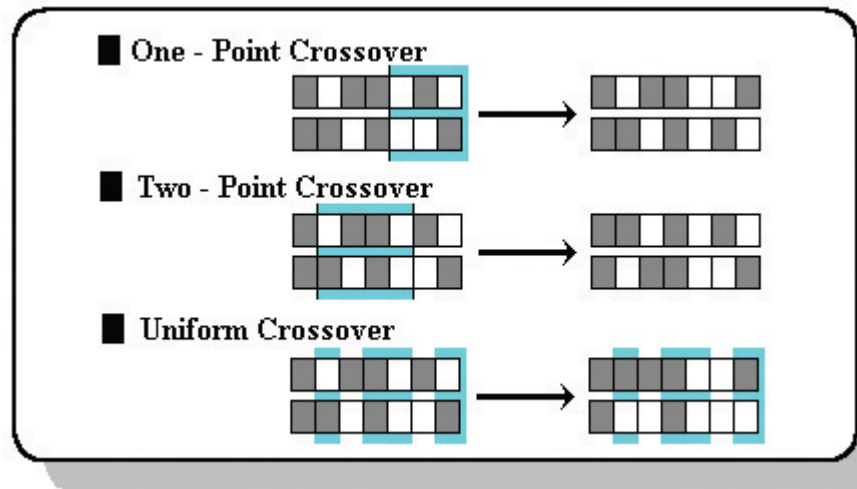


圖 2.2 三種不同交配的方法

c. 突變

如果只單單依靠複製與交配機制，並不能使演化程序產生一個具有新特性的個

體。因為複製與交配機制所產生的個體僅繼承了親代的混合特性，很難發展出具有新特徵的個體，如此不但使得演化的速度過於緩慢，同時使得演化的結果侷限在某些特定的點上。在自然界中，生物藉著突變機制來產生新的物種。在遺傳演算法中，仍然藉著自然界中的這種機制來增加族群的多樣性，使得演化能盡可能的朝多個新方向前進，避免侷限在少數的個體上。突變機制是隨機在交配池中選取染色體，再隨機地選取突變點，可針對單一位元或字串進行突變，突變的方式為取其補數。圖 2.3 為自然界與基因演算法的突變機制示意圖。

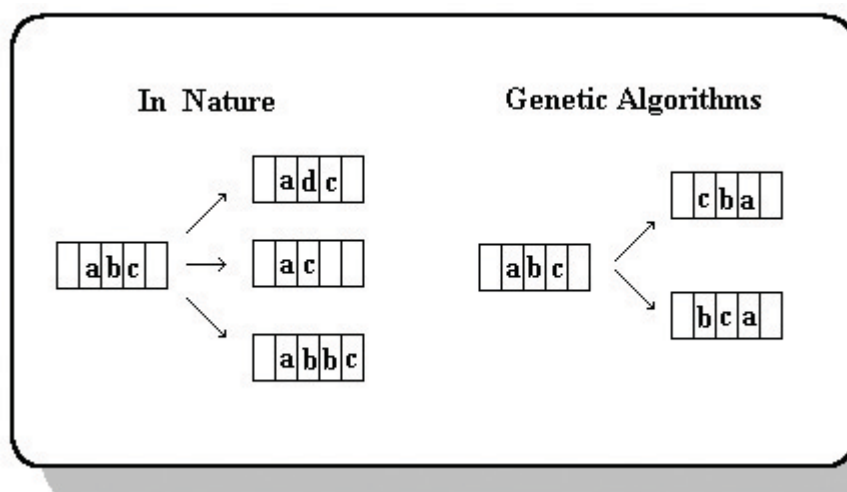


圖 2.3 自然界與基因演算法的突變機制

d. 選擇

在經過複製、交配和突變的演化過程後，所產生的新染色體將取代舊染色體繼續演化下一代。選擇方式是根據每個染色體的適應值高低來進行，在排定每個染色體的適應能力順序後，適應值高的染色體將有較高的機會被選擇來進行交配以產生新的下一代，其選擇的方式是隨機的，所以，適應值低的染色體也有機會被選擇來進行交配，因為適應值低的染色體也許帶有某些有用的資訊。不過一般在選擇過程中，還是會保留舊染色體中適應值較高者，以取代新染色體中適應值較低者，以確保新世代的染色體適應值不會降低。

2.2 導引律設計

在本節中，我們將針對導引飛彈攔截中的目標逃逸問題採用判別法則。首先，我們將回顧導引飛彈的動態方程式，接著討論目標飛彈可能有的逃逸方式，而針對導引飛彈攔截問題，我們也會提出以基因演算法(GA)為主的補償設計。

2.2.1 飛彈導引律的動態方程式

一般而言，飛彈與目標物間的相對運動可被描述成原點固定在飛彈上的球座標系統(spherical coordinate system)，如圖2.4所示，其中， e_r 、 e_θ 和 e_ϕ 分別表在球座標系統中的三個單位軸向量。為了簡化導引律設計，我們只考慮系統的動力學，且假

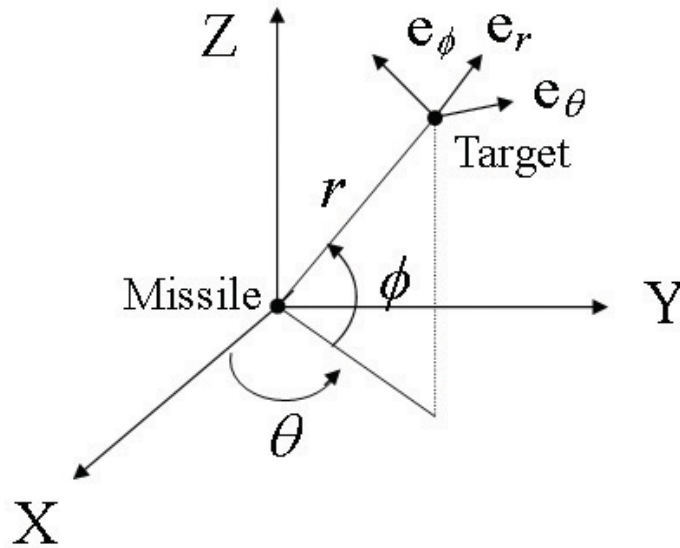


圖 2.4 飛彈攔截過程的幾何圖形

設飛彈和目標皆為質點運動。因此，此系統的運動方程式可被整理如下：

$$\ddot{r} - r\dot{\phi}^2 - r\dot{\theta}^2 \cos^2 \phi = a_{Tr} - a_{Mr} \quad (2.1)$$

$$r\ddot{\theta} \cos \phi + 2\dot{r}\dot{\theta} \cos \phi - 2r\dot{\phi}\dot{\theta} \sin \phi = a_{T\theta} - a_{M\theta} \quad (2.2)$$

$$r\ddot{\phi} + 2\dot{r}\dot{\phi} + r\dot{\theta}^2 \cos \phi \sin \phi = a_{T\phi} - a_{M\phi} \quad (2.3)$$

其中， r 表飛彈和目標間的相對距離， θ 和 ϕ 分別表方位角和傾斜角。而 a_{Mr} 、 $a_{M\theta}$ 和 $a_{M\phi}$ 分別表飛彈在球座標中三個指令加速度的分量，且 a_{Tr} 、 $a_{T\theta}$ 和 $a_{T\phi}$ 表相對的目標

加速度分量。此研究的主要目標是設計導引律以達成攔截任務，亦即，在目標逃逸的過程中達到 $r \rightarrow 0$ 的狀態。

2.2.2 目標逃逸運動

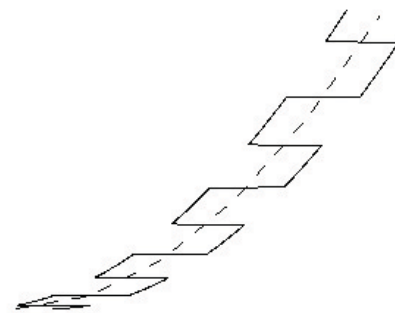
眾所周知，目標逃逸的主要函數被設計成可躲避導引飛彈的攔截，但此過程一般而言是非常難以預測的。然而，在終端階段中的目標逃逸不能太過複雜，此主要是為了要符合目標的性能需求。典型的目標逃逸模式如圖 2.5 所示，雖然逃逸運動的類型可以是非常多樣化的，但在此研究中，我們將焦點放在兩個已知的目標逃逸類型來驗證所提出方法的可行性，即為螺旋方程式 (2.4) 和方波方程式 (2.5)，分別定義如下：

$$a_T = a_{t1} \cos(\omega_{t1}t + \beta_{t1})e_\theta + a_{t2} \sin(\omega_{t2}t + \beta_{t2})e_\phi \quad (2.4)$$

$$a_T = a_{t1} \cdot \text{sign}(\cos(\omega_{t1}t + \beta_{t1}))e_\theta + a_{t2} \cdot \text{sign}(\sin(\omega_{t2}t + \beta_{t2}))e_\phi \quad (2.5)$$



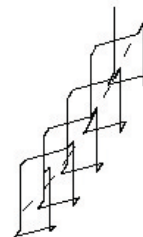
(a) the maneuvering type is sin-function



(b) the maneuvering type is square-function



(c) the maneuvering acceleration is spiral



(d) the maneuvering type is square-spiral

圖 2.5 四種逃逸模式

2.2.3 以基因演算法為基礎的補償設計

以下，我們將採用第2.1節所介紹的基因遺傳演算法來判別式(2.4)和(2.5)中的目標逃逸運動參數。圖2.6為一混和的設計架構，其內迴路(inner-loop)為realistic true proportional navigation (RTPN) 導引律[7]，主要是在控制非逃逸動態方程式；而外迴路為基因演算法，主要目的在判別和消除目標逃逸特性。因此，系統(2.1)-(2.3)可被改寫如下：

$$\ddot{r} - r\dot{\phi}^2 - r\dot{\theta}^2 \cos^2 \phi = a_{Tr} - (a_{Mrpn} + a_{Mrga}) \quad (2.6)$$

$$r\ddot{\theta} \cos \phi + 2\dot{r}\dot{\theta} \cos \phi - 2r\dot{\phi}\dot{\theta} \sin \phi = a_{T\theta} - (a_{M\theta pn} + a_{M\theta ga}) \quad (2.7)$$

$$r\ddot{\phi} + 2\dot{r}\dot{\phi} + r\dot{\theta}^2 \cos \phi \sin \phi = a_{T\phi} - (a_{M\phi pn} + a_{M\phi ga}) \quad (2.8)$$

其中，參數 a_{Mrpn} 、 $a_{M\theta pn}$ 和 $a_{M\phi pn}$ 為已知的RTPN導引律，而 a_{Mrga} 、 $a_{M\theta ga}$ 和 $a_{M\phi ga}$ 為針對目標逃逸的混合補償器。

在此研究中，我們採用RTPN導引律[7]來做補償器設計，其形式如下所示：

$$a_M = -\lambda \dot{r} (\dot{\theta} \cos \phi e_\theta + \dot{\phi} e_\phi) \quad (2.9)$$

在參考文獻[7]中，如果式(2.10)成立，則飛彈可成功攔截到目標物：

$$\frac{|\dot{r}|}{h_0/r_0} > \frac{1}{\sqrt{\lambda-1}} \quad (2.10)$$

其中， $h_0 = r_0^2 \sqrt{\dot{\phi}_0^2 + \dot{\theta}_0^2 \cos^2 \phi_0}$ ，而下標0表相關參數的初始值。在此研究中，我們將式(2.9)做如下的變形

$$a_M = -\lambda \dot{r} (\dot{\theta} \cos \phi e_\theta + \dot{\phi} e_\phi + a_{ga1} \cos(\omega_{ga1}t + \beta_{ga1})e_\theta + a_{ga2} \sin(\omega_{ga2}t + \beta_{ga2})e_\phi) \quad (2.11)$$

從式(2.11)可清楚的看到，額外增加的項是利用sine函數來做逃逸補償近似用。逃逸補償參數 $(a_{ga}, \omega_{ga}, \beta_{ga})$ 將被所提出的線上基因學習法則所決定。根據PN控制法則的設計經驗，如果 $\dot{\theta}$ 和 $\dot{\phi}$ 都趨近零的話，目標物將可被成功的攔截。因此，適當定義的適應函數，如式(2.12)，可被選擇在基因演算法的尋找過程中扮演導引的角色。

$$F(k) = 2 - \exp(-|\dot{\theta}(k)| \sigma_1) - \exp(-|\dot{\phi}(k)| \sigma_2) \quad (2.12)$$

其中 σ_1 和 σ_2 為常數

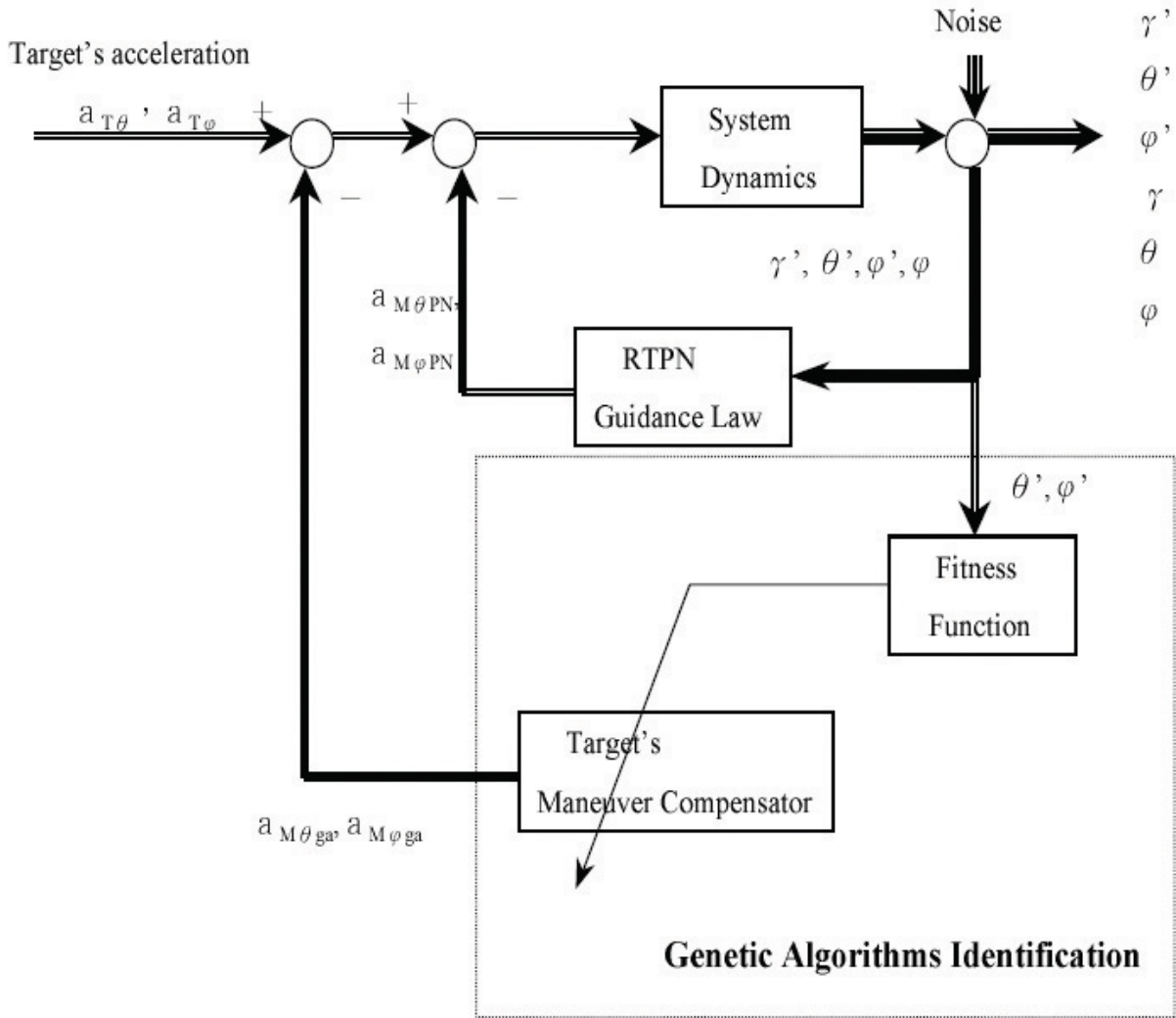


圖 2.6 導引律的方塊圖

2.3 數值模擬結果

在此節中，我們提出一個範例來證明 GA 演算法的效用，並利用 Matlab 的 GA Toolbox 來做數值模擬。考慮目標物有逃逸和非逃逸兩種模式，對具有逃逸模式的狀況而言，目標加速度被假設成位於垂直瞄準線 (line of sight, LOS) 的方向，且具有式 (2.7) 或式 (2.8) 的形式。為了完成式 (2.11) 的要求，我們選擇 $\lambda = 3$ ，且初始狀態為 $r_0 = 8\text{km}$ 、 $\theta_0 = -0.2\text{rad}$ 、 $\psi_0 = 0.1\text{rad}$ 、 $\dot{r}_0 = -3\text{ km/sec}$ 、 $\dot{\theta}_0 = -0.01\text{ rad/sec}$ 且 $\dot{\psi}_0 = 0.05\text{ rad/sec}$ ；另外，我們也假設加速度 a_t 和 a_m 被限制在 0.5 km/sec^2 之內。考慮兩種不

同類型的目標逃逸模式，分別為螺旋和方形-螺旋模式，其目標加速度的參數為 $a_{t1} = a_{t2} = 0.1 \text{ km/sec}^2$ 、 $\omega_{t1} = \omega_{t2} = 6 \text{ rad/sec}$ 且 $\beta_{t1} = \beta_{t2} = \pi/6 \text{ rad}$ 。假設 a_{ga} 的學習範圍在 -0.6 至 0.6 km/sec^2 之間，則我們可得到下列的數值模擬結果。

2.3.1 目標具螺旋逃逸模式

假設目標具有振幅 0.1 km/sec^2 的螺旋逃逸模式，圖 2.7 為目標加速度(x)和 GA 學習結果(o)的示意圖，可發現 GA 的學習效果非常好，在 1 秒之後，GA 不管在 θ 或 ψ 方向，皆可很接近目標加速度。而圖 2.8 顯示攔截過程中動態參數的時間響應圖，其中虛線表示 RTPN 控制在目標具有螺旋逃逸模式時的軌跡；而實線為具 GA 補償的 RTPN 控制在目標具有螺旋逃逸模式時的軌跡，可發現 $\dot{\theta}$ 和 $\dot{\psi}$ 都趨近於零，且在 1 秒左右就達到 GA 演算法中，適應函數的要求。圖 2.9 為 RTPN (虛線表示) 和具 GA 補償的 RTPN (實線表示) 之飛彈加速度比較，可發現具 GA 補償的 RTPN 在初始階段需要較多的加速度，且達到 0.5 km/sec^2 的限制範圍。在此小節中，所有的圖示說明與此部分的模擬結果相同。

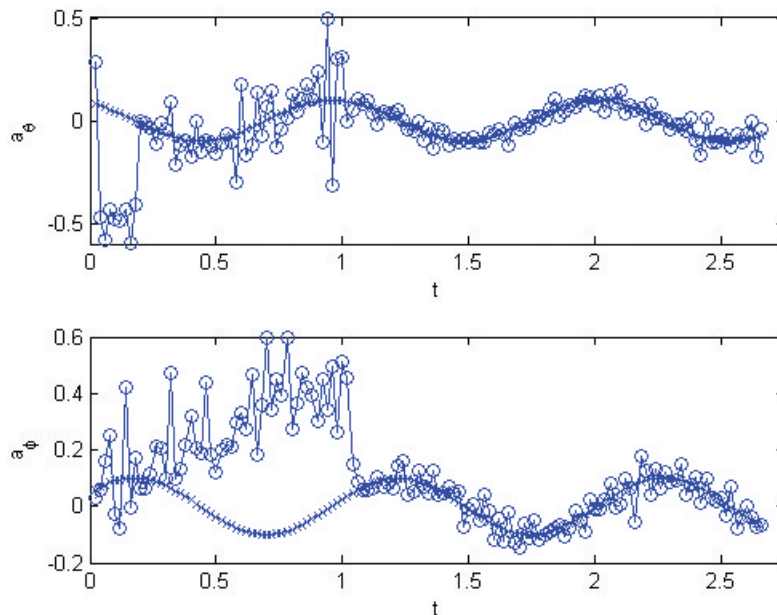


圖 2.7 目標具螺旋逃逸和 GA 學習結果 ($a_t = 0.1 \text{ km/sec}^2$)

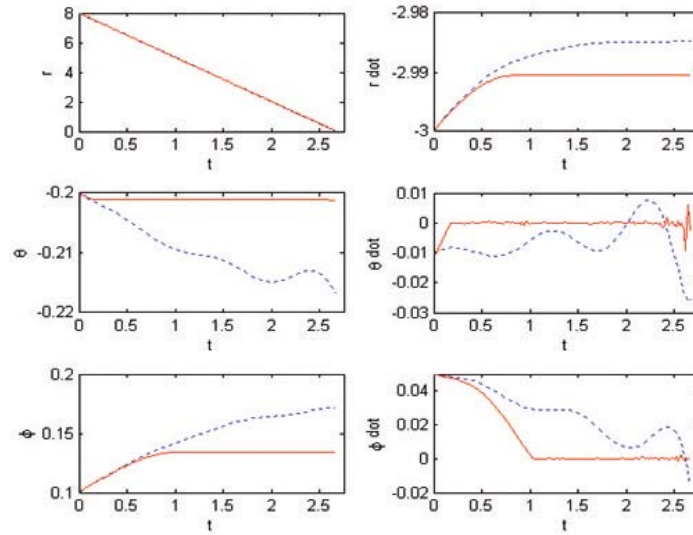


圖 2.8 不同參數的時間響應 ($a_t = 0.1\text{km}/\text{sec}^2$)

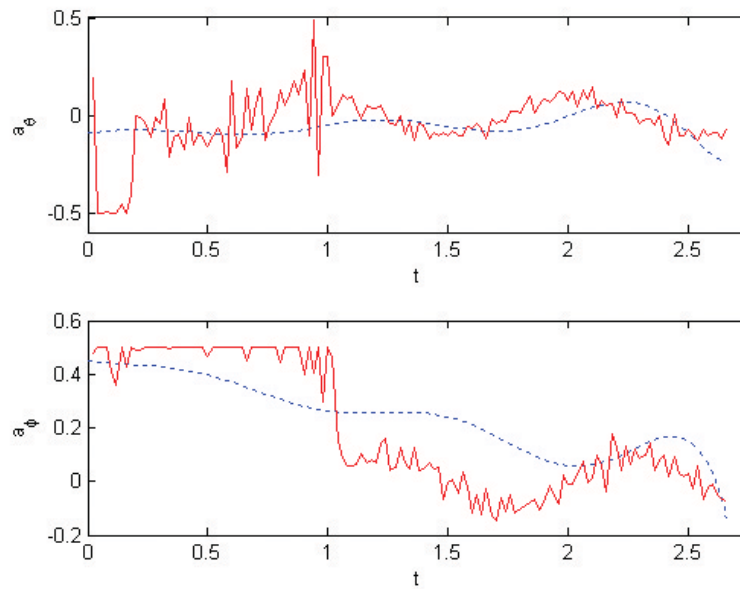


圖 2.9 RTPN 和具 GA 補償的 RTPN 之飛彈加速度比較 ($a_t = 0.1\text{km}/\text{sec}^2$)

為了驗證 GA 演算法的學習效果，我們將目標螺旋逃逸加速度的振幅增加至 $0.25\text{km}/\text{sec}^2$ ，則圖 2.10 為目標加速度(x)和 GA 學習結果(o)的示意圖，可發現 GA 還是可以很接近目標加速度。而圖 2.11 為攔截過程中的動態參數時間響應圖，可發現雖然目標逃逸加速度增加了，但 $\dot{\theta}$ 和 $\dot{\psi}$ 還是可以趨近於零。圖 2.12 為 RTPN (虛線表示) 和具 GA 補償的 RTPN (實線表示) 之飛彈加速度比較，可發現目標的逃逸加速度

增加時，飛彈攔截時所需的加速也會增加。而純RTPN在導引終端也需要較多的能量且會趨向不穩定。

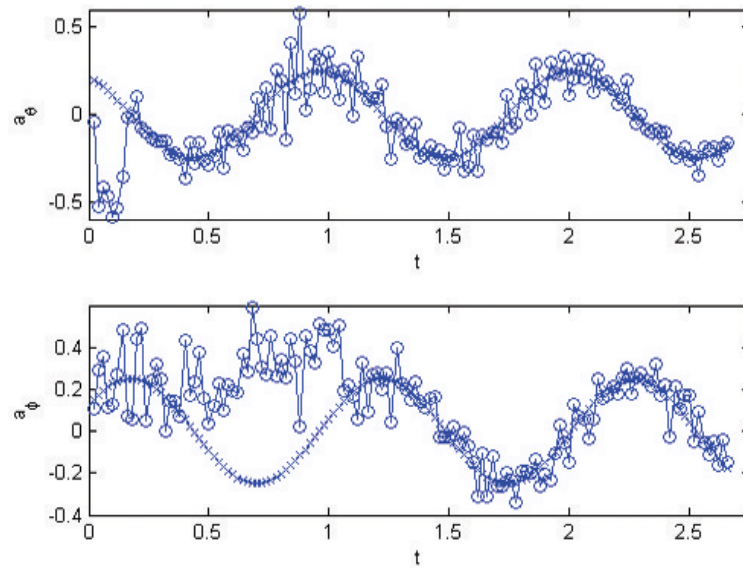


圖 2.10 目標具螺旋逃逸和 GA 學習結果 ($a_t = 0.25\text{km}/\text{sec}^2$)

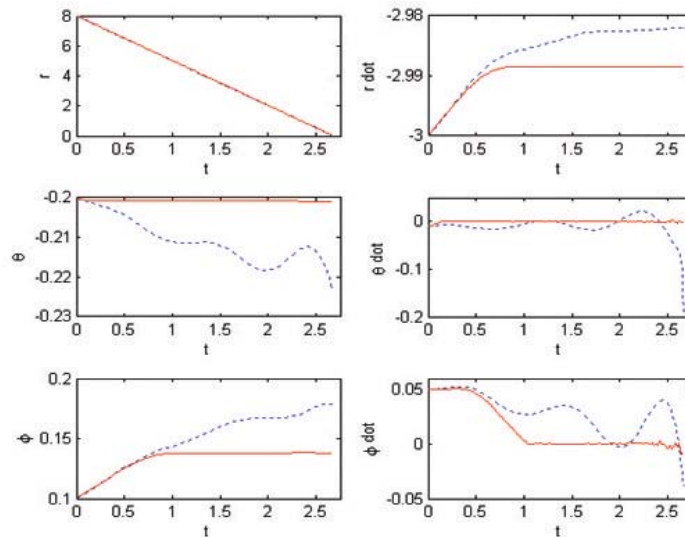


圖 2.11 不同參數的時間響應 ($a_t = 0.25\text{km}/\text{sec}^2$)

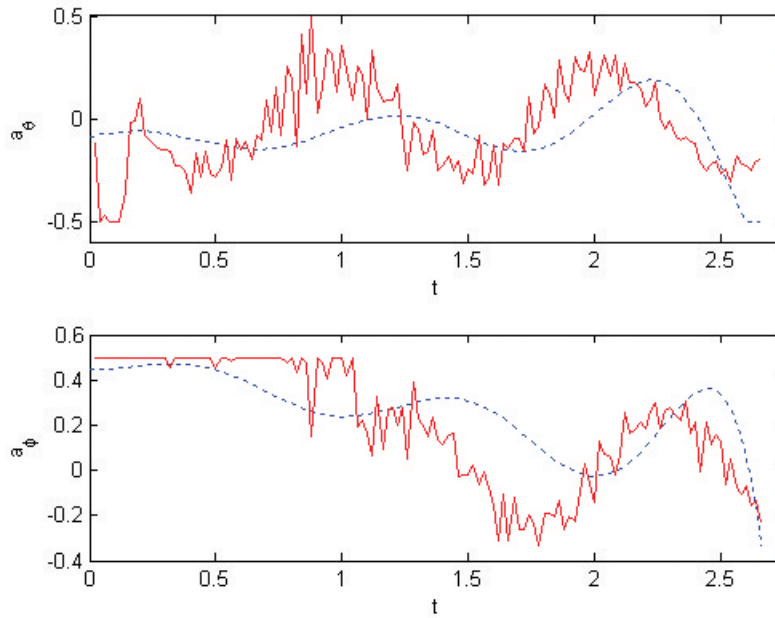


圖 2.12 RTPN 和具 GA 補償的 RTPN 之飛彈加速度比較 ($a_t = 0.25\text{km}/\text{sec}^2$)

2.3.2 目標具方形-螺旋逃逸模式

在 2.3.1 節中，我們是使用式 (2.11) 來做飛彈的加速度控制，其中，GA 是用來學習 sine 函數的參數，以做為逃逸補償近似之用。由 2.3.1 節中可觀察出，GA 在目標物具有螺旋逃逸模式時擁有非常好的學習效果，在此小節中，我們一樣使用式 (2.11) 來做飛彈的加速度控制，然而，目標的逃逸模式改為方形-螺旋的形式。首先，考慮目標加速度的振幅 $a_{t1} = a_{t2} = \text{km}/\text{sec}^2$ 的情況。圖 2.13 為目標加速度 (x) 和 GA 學習結果 (o) 的示意圖，可發現目標的逃逸形式雖然與 GA 學習的形式不同，但 GA 還是可以學習的很好。而圖 2.14 為攔截過程中的動態參數時間響應圖，可發現目標物確實是以方形-螺旋的方式逃逸，而具 GA 補償的 RTPN 控制還是可在短時間內達到 θ 和 ψ 趨近於零的狀況。圖 2.15 為 RTPN (虛線表示) 和具 GA 補償的 RTPN (實線表示) 之飛彈加速度比較，可發現具 GA 補償的 RTPN 控制為了達到較佳的效果，所需要的加速度就比純 RTPN 控制來的高。

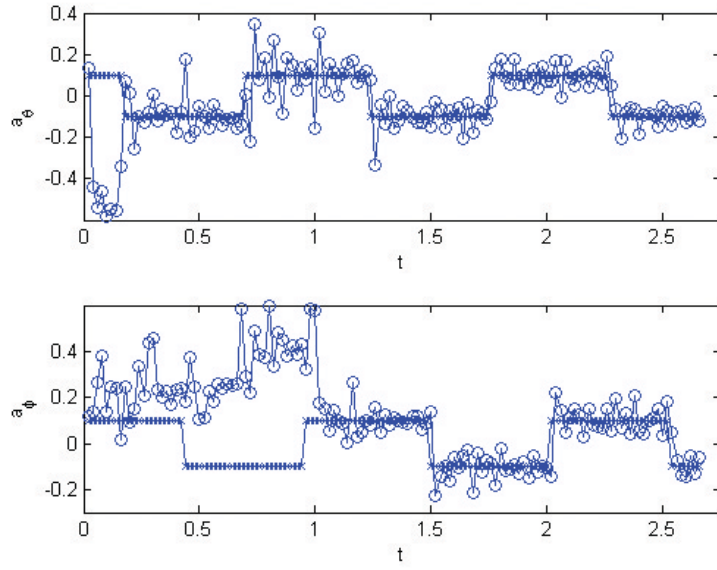


圖 2.13 目標具方形-螺旋逃逸和 GA 學習結果 ($a_t = 0.1\text{km}/\text{sec}^2$)

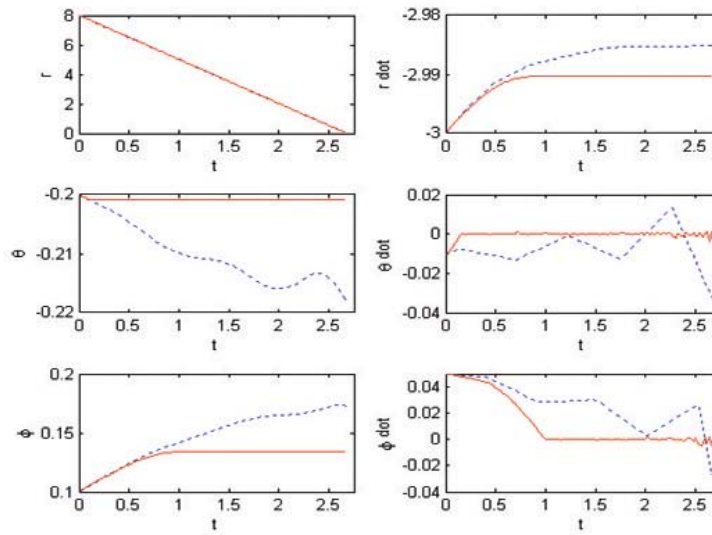


圖 2.14 不同參數的時間響應 ($a_t = 0.1\text{km}/\text{sec}^2$)

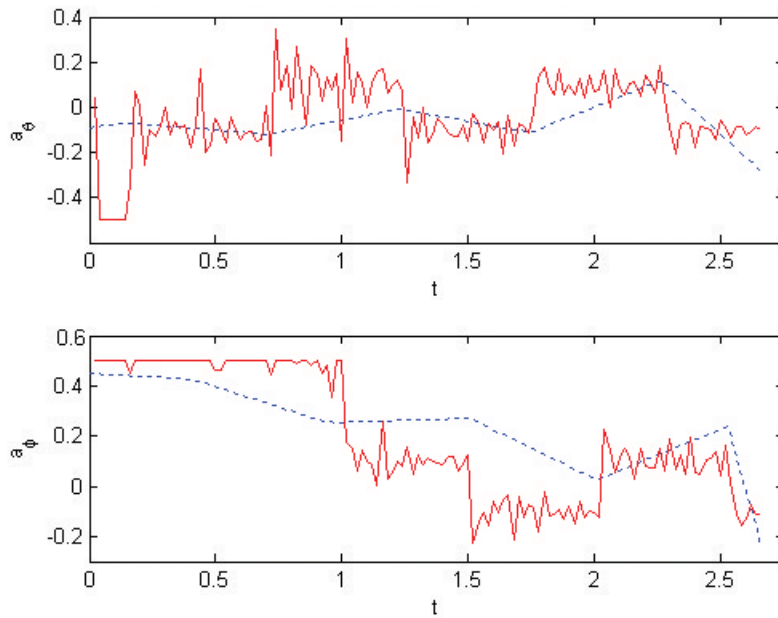


圖 2.15 RTPN 和具 GA 補償的 RTPN 之飛彈加速度比較 ($a_t = 0.1\text{km}/\text{sec}^2$)

接著，我們將目標物方形-螺旋逃逸模式的加速度振幅增加至 $0.25\text{km}/\text{sec}^2$ 。圖 2.16 為目標加速度 (x) 和 GA 學習結果 (o) 的示意圖，可發現雖然 GA 學習的模式與目標的逃逸模式不同，但 GA 還是有很好的學習效果。圖 2.17 為攔截過程中的動態參數時間響應圖，圖 2.18 為 RTPN (虛線表示) 和具 GA 補償的 RTPN (實線表示) 之飛彈加速度比較，由以上的模擬圖可觀察出，RTPN 配合 GA 的設計可以達到較好的性能。此種狀況下，兩種控制法則皆需較多的能量，但在導引終端，純 RTPN 會趨向不穩定。

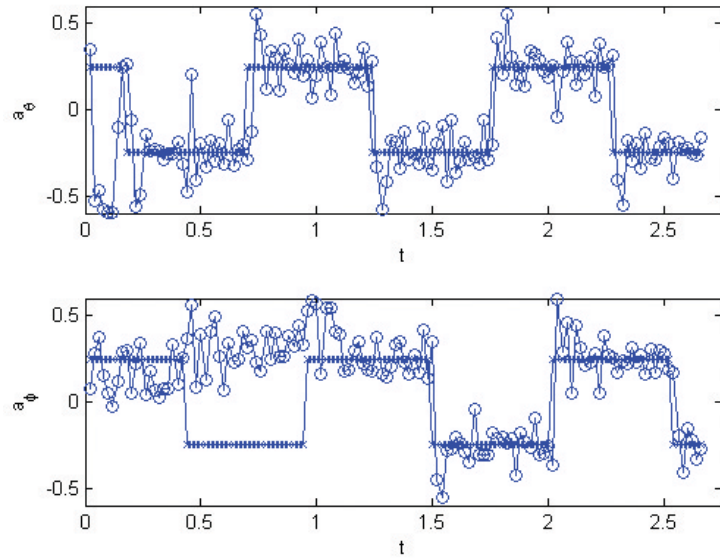


圖 2.16 目標具方形-螺旋逃逸和 GA 學習結果 ($a_t = 0.25\text{km}/\text{sec}^2$)

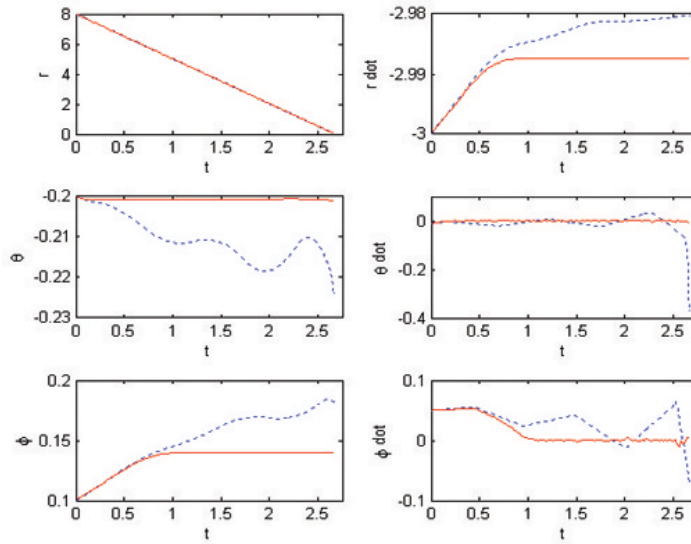


圖 2.17 不同參數的時間響應 ($a_t = 0.25\text{km}/\text{sec}^2$)

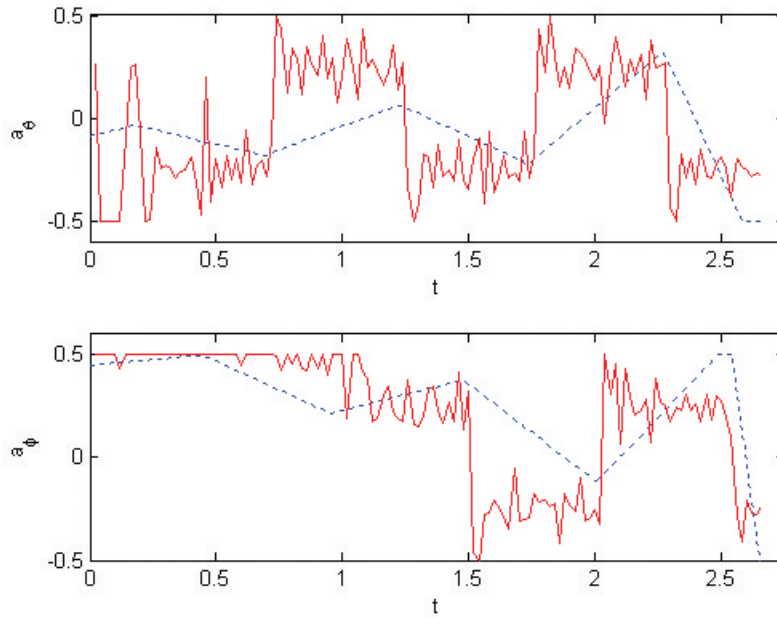


圖 2.18 RTPN 和具 GA 補償的 RTPN 之飛彈加速度比較 ($a_t = 0.25\text{km}/\text{sec}^2$)

此節的研究，主要是提出線上學習的基因演算法用於目標逃逸之判別，並可用在 ATBM 的終端導引設計上。由以上的數值模擬結果可驗證簡單的 GA 學習模型可用在各種逃逸模式的判別上，且很快就達到合理的精確度。由模擬結果也可觀察出，當目標逃逸運動可被準確預測時，飛彈攔截就達到比較好的效果。

第三章

HAVE DASH II 飛彈系統之動態數學模式

由於飛彈在運行期間，其位移及滾轉隨時間改變，為了有效瞭解並掌控其飛行動態，本章將探討描述其飛行運動之數學動態模式。一般而言，用來描述飛彈飛行的座標系統包含有慣性座標系統與彈體座標系統。在此章中我們將於第一小節介紹這兩種座標系統，並推導兩種座標系統的轉換關係。第二小節除了介紹HAVE DASH II數學模式之外，也從第一小節中的座標轉換結果，說明飛彈運行軌跡與參考軌跡的相關性。

3.1 座標軸系統

3.1.1 慣性座標系 (X, Y, Z)

此座標原點在地球上，視為固定不動的座標，不隨時間而改變，可以表示飛彈與目標的位置，通常 X 軸指向北方， Y 軸指向東方，而 Z 軸則指向下，三軸互相垂直符合右手定則。

3.1.2 彈體座標系 (X_b, Y_b, Z_b)

彈體座標系用來表示飛彈本身的姿態及動態，其原點在飛彈質心， X_b 軸沿著飛彈中心指向前方， Y_b 軸則垂直 X_b 軸，在飛彈左右的平面上，指向飛彈右邊， Z_b 軸則垂直 $X_b - Y_b$ 平面，指向飛彈下方，三軸互相垂直且符合右手定則。

3.1.3 座標轉換

接下來，我們推導慣性座標與彈體座標轉換，由慣性座標經過三個角度的旋轉來得到彈體座標，這種有順序的角度旋轉稱為尤拉角(Eular Angle)，首先將彈體座標和慣性座標原點重合起來：第一、將慣性座標 (X, Y, Z) 固定 Z 軸向右轉動 X - Y 軸 Ψ 角，得到 (x_1, y_1, z_1) 座標，其轉換公式為

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} =: M_1 \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \quad (3.1)$$

第二、再固定 y_1 軸，向上轉動 $x_1 - z_1$ 軸 Θ 角，得到 (x_2, y_2, z_2) 座標，轉換公式為

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} \cos(\Theta) & 0 & \sin(\Theta) \\ 0 & 1 & 0 \\ -\sin(\Theta) & 0 & \cos(\Theta) \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} =: M_2 \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \quad (3.2)$$

第三、再固定 x_2 軸向順時針轉動 $y_2 - z_2$ 軸 Φ 角，便可以得到彈體座標 (X_b, Y_b, Z_b) ，轉換公式為

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & -\sin(\Phi) \\ 0 & \sin(\Phi) & \cos(\Phi) \end{pmatrix} \begin{pmatrix} X_b \\ Y_b \\ Z_b \end{pmatrix} =: M_3 \begin{pmatrix} X_b \\ Y_b \\ Z_b \end{pmatrix} \quad (3.3)$$

於是結合以上三式，可得到慣性座標與彈體座標轉換式

$$\begin{aligned} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} &= M_1 M_2 M_3 \begin{pmatrix} X_b \\ Y_b \\ Z_b \end{pmatrix} \\ &= \begin{pmatrix} C_\Theta C_\Psi & S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi & C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi \\ C_\Theta S_\Psi & S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi & C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi \\ -S_\Theta & C_\Theta S_\Phi & C_\Theta C_\Phi \end{pmatrix} \begin{pmatrix} X_b \\ Y_b \\ Z_b \end{pmatrix} \end{aligned} \quad (3.4)$$

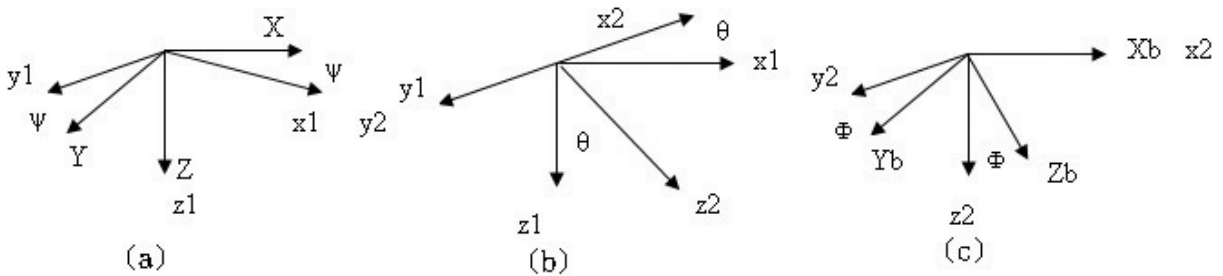


圖 3.1 慣性座標與彈體座標轉換

3.2 飛彈模型

本計畫將採用HAVE DASH II 飛彈系統來進行容錯控制之分析與模擬，HAVE DASH II 飛彈之外觀與構造圖分別如圖 3.2 及圖 3.3 所示 [16]：

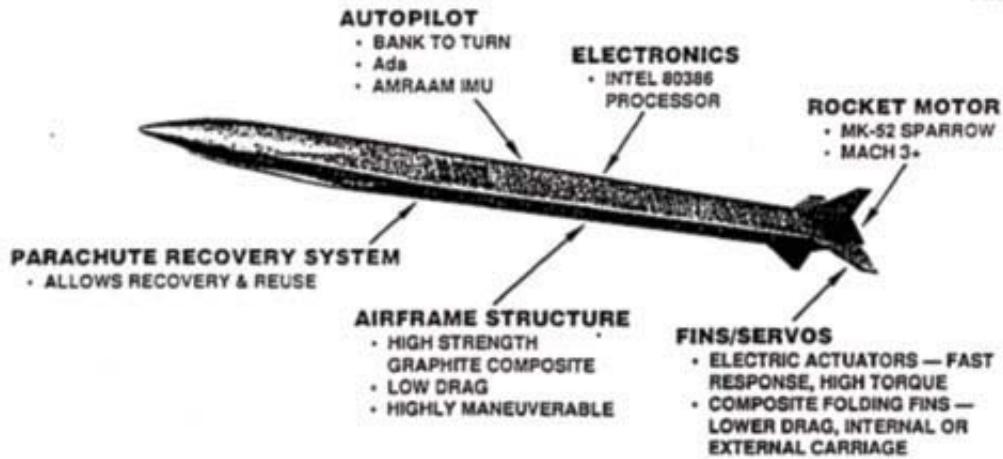


圖 3.2 HAVE DASH II 外觀圖

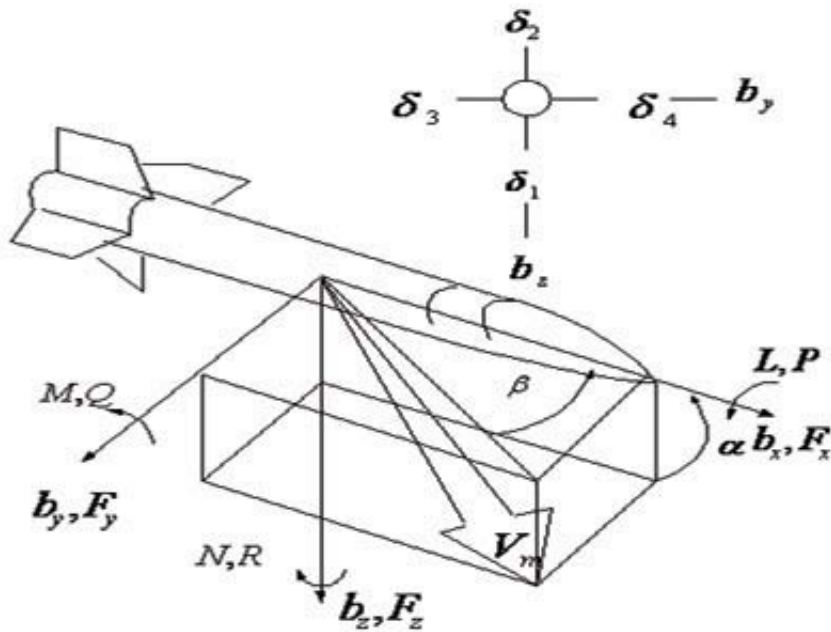


圖 3.3 HAVE DASH II 構造圖

接下來，我們將針對HAVE DASH II 飛彈系統描述其動態控制模型，首先，飛彈彈體座標之運動方程式可表示如下[10]:

$$\dot{P} = -\frac{I_{zz} - I_{yy}}{I_{xx}}QR + \frac{L}{I_{xx}} \quad (3.5)$$

$$\dot{Q} = -\frac{I_{xx} - I_{zz}}{I_{yy}}PR + \frac{M}{I_{yy}} \quad (3.6)$$

$$\dot{R} = -\frac{I_{yy} - I_{xx}}{I_{zz}}PQ + \frac{N}{I_{zz}} \quad (3.7)$$

$$\dot{\Psi} = \sin \Phi \sec \Theta Q + \cos \Phi \sec \Theta R \quad (3.8)$$

$$\dot{\Theta} = \cos \Phi Q - \sin \Phi R \quad (3.9)$$

$$\dot{\Phi} = P + \sin \Phi \tan \Theta Q + \cos \Phi \tan \Theta R \quad (3.10)$$

$$\dot{V}_m = \frac{1}{m} \{ \cos \alpha \cos \beta (F_x + g_x + T_x) + \sin \beta (F_y + g_y) \} \quad (3.11)$$

$$\begin{aligned} \dot{\alpha} = & Q - (P \cos \alpha + R \sin \alpha) \tan \beta \\ & + \frac{1}{mV_m \cos \beta} \{ -\sin \alpha (F_x + g_x + T_x) + \cos \alpha (F_z + g_z) \} \end{aligned} \quad (3.12)$$

$$\begin{aligned} \dot{\beta} = & P \sin \alpha - R \cos \alpha + \frac{1}{mV_m} \{ -\cos \alpha \sin \beta (F_x + g_x + T_x) \} \\ & + \frac{1}{mV_m} \{ \cos \beta (F_y + g_y) - \sin \alpha \sin \beta (F_z + g_z) \} \end{aligned} \quad (3.13)$$

其中受重力場影響部份為

$$g_x = -mg \sin \Theta \quad (3.14)$$

$$g_y = mg \cos \Theta \sin \Phi \quad (3.15)$$

$$g_z = mg \cos \Theta \cos \Phi \quad (3.16)$$

在上述表示式中，P、Q、R分別代表飛彈彈體座標 (X_b, Y_b, Z_b) 軸三個方向的角速度

V_m 、 α 和 β 分別代表彈體座標的飛彈速度、攻角及側滑角

Θ 、 Φ 、 Ψ 分別代表飛彈之傾斜角、轉動角、偏離角

I_{xx} 、 I_{yy} 、 I_{zz} 分別代表彈體座標 (X_b, Y_b, Z_b) 軸三個方向的轉動慣量

m 表飛彈質量， g 表重力常數， T_x 為飛彈推力

F_x, F_y, F_z 分別代表彈體座標 (X_b, Y_b, Z_b) 軸的氣動力

L, M, N 則為彈體座標的轉動力矩、傾斜力矩和偏離力矩

它們分別滿足下列的式子：

$$F_x = k_F \rho V_m^2 C_x \quad (3.17)$$

$$F_y = k_F \rho V_m^2 C_y \quad (3.18)$$

$$F_z = k_F \rho V_m^2 C_z \quad (3.19)$$

$$L = k_M \rho V_m^2 C_l \quad (3.20)$$

$$M = k_M \rho V_m^2 C_m \quad (3.21)$$

$$N = k_M \rho V_m^2 C_n \quad (3.22)$$

其中 ρ 為空氣密度； k_F 和 k_M 是由飛彈幾何尺寸所決定的常數； C_x, C_y, C_z, C_l, C_m 和 C_n 為氣動力參數並表示成氣動力表的形式。氣動力參數 $C_i, i = x, y, z, l, m, n$ 的函數形式表示如下：

$$C_i = C_{io}(\alpha, \beta, M_m) + C_{i\delta_e}(\alpha, \delta_e, M_m) + C_{i\delta_a}(\alpha, \delta_a, M_m) + C_{i\delta_r}(\alpha, \delta_r, M_m) \quad (3.23)$$

其中 $M_m = V_m/c$ 為馬赫數 (c 為聲速)。 δ_e, δ_a 和 δ_r 為相對的有效傾斜、滾動、偏離尾翼控制偏角。它們隨著致動器和尾翼調節器所下的偏角 $\delta_{ec}, \delta_{ac}, \delta_{rc}$ 命令而改變。

HAVE DASH II 飛彈系統有四個分布不對稱的尾翼。根據下列式子控制偏角產生四個尾翼偏角 δ_{ic} 如下：

$$\delta_{1c} = \delta_{ec} - \delta_{ac} + \delta_{rc} \quad (3.24)$$

$$\delta_{2c} = -\delta_{ec} - \delta_{ac} + \delta_{rc} \quad (3.25)$$

$$\delta_{3c} = -\delta_{ec} - \delta_{ac} - \delta_{rc} \quad (3.26)$$

$$\delta_{4c} = \delta_{ec} - \delta_{ac} - \delta_{rc} \quad (3.27)$$

根據上述式子，可以得到結合實際尾翼偏角所產生的等效傾斜角、滾動角及偏離角

如下：

$$\delta_e = \frac{1}{4}(\delta_1 - \delta_2 - \delta_3 + \delta_4) \quad (3.28)$$

$$\delta_a = -\frac{1}{4}(\delta_1 + \delta_2 + \delta_3 + \delta_4) \quad (3.29)$$

$$\delta_r = \frac{1}{4}(\delta_1 + \delta_2 - \delta_3 - \delta_4) \quad (3.30)$$

更進一步，我們將氣動力係數根據下列函數來做近似：

$$\begin{aligned} \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} &= \begin{pmatrix} C_{x0} \\ C_{y0} \\ C_{z0} \end{pmatrix} + \begin{pmatrix} C_{xe} & C_{xa} & C_{xr} \\ C_{ye} & C_{ya} & C_{yr} \\ C_{ze} & C_{za} & C_{zr} \end{pmatrix} \begin{pmatrix} \delta_e \\ \delta_a \\ \delta_r \end{pmatrix} \\ &=: C_{F0} + C_{Fu}u \end{aligned} \quad (3.31)$$

$$\begin{aligned} \begin{pmatrix} C_l \\ C_m \\ C_n \end{pmatrix} &= \begin{pmatrix} C_{l0} \\ C_{m0} \\ C_{n0} \end{pmatrix} + \begin{pmatrix} C_{le} & C_{la} & C_{lr} \\ C_{me} & C_{ma} & C_{mr} \\ C_{ne} & C_{na} & C_{nr} \end{pmatrix} \begin{pmatrix} \delta_e \\ \delta_a \\ \delta_r \end{pmatrix} \\ &=: C_{M0} + C_{Mu}u \end{aligned} \quad (3.32)$$

矩陣 $C_{F0}, C_{Fu}, C_{M0}, C_{Mu}$ 為 α, β, M_m 的函數。根據最小平方誤差準則，利用三次的曲線近似方法可得：

$$C_{F0} = \begin{pmatrix} -0.57 + 0.0083\alpha \\ -0.21\beta \\ C_{z0}(\beta, M_m) \end{pmatrix} \quad (3.33)$$

$$C_{Fu} = \begin{pmatrix} 0.004 & 0 & 0 \\ 0 & 0 & 0.08 \\ -0.09 & 0 & 0 \end{pmatrix} \quad (3.34)$$

$$C_{M0} = \begin{pmatrix} 0.116\beta \\ C_{m0}(\alpha, M_m) \\ 0.08\beta \end{pmatrix} \quad (3.35)$$

$$C_{Mu} = \begin{pmatrix} 0 & -0.127 & 0 \\ -0.675 & 0 & 0 \\ 0 & 0 & -0.584 \end{pmatrix} \quad (3.36)$$

其中 $C_{z0}(\beta, M_m)$ 和 $C_{m0}(\alpha, M_m)$ 如下所示：

$$C_{z0}(\alpha, M_m) = C_{z1}(\alpha) + C_{z2}(\alpha)M_m \quad (3.37)$$

$$C_{m0}(\alpha, M_m) = C_{m1}(\alpha) + C_{m2}(\alpha)M_m \quad (3.38)$$

而

$$C_{z1}(\alpha) = -0.0015\alpha^3 + 0.0125\alpha^2 - 0.5052\alpha + 0.0429 \quad (3.39)$$

$$C_{z2}(\alpha) = 0.0006\alpha^3 - 0.0138\alpha^2 + 0.123\alpha - 0.0191 \quad (3.40)$$

$$C_{m1}(\alpha) = -0.0055\alpha^3 + 0.2131\alpha^2 - 2.7419\alpha - 0.0381 \quad (3.41)$$

$$C_{m2}(\alpha) = 0.0014\alpha^3 - 0.0623\alpha^2 + 0.8715\alpha - 0.4041 \quad (3.42)$$

且符合 $0 \leq \alpha \leq 25deg$ ， $2.0 \leq M_m \leq 3.0$ 這兩項條件。

以上為完整的飛彈數學模型，接下來我們利用3.1節的座標轉換關係，來推導飛彈在慣性座標的飛行情形，假設 (U, V, W) 分別代表彈體座標 (X_b, Y_b, Z_b) 的速度分量，而與狀態 V_m, α, β 的關係式如下：

$$V_m = \sqrt{U^2 + V^2 + W^2} \quad (3.43)$$

$$\alpha = \tan^{-1} \frac{W}{U} \quad (3.44)$$

$$\beta = \sin^{-1} \frac{V}{V_m} \quad (3.45)$$

由上述可求得

$$U = V_m \cos \alpha \cos \beta \quad (3.46)$$

$$V = V_m \sin \beta \quad (3.47)$$

$$W = V_m \sin \alpha \cos \beta \quad (3.48)$$

最後我們加上座標轉換將得到飛彈在慣性座標 (X, Y, Z) 的速度分量

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = V_m \begin{pmatrix} C_\alpha C_\beta C_\Theta C_\Psi & S_\beta (S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi) & C_\alpha C_\beta (C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi) \\ C_\alpha C_\beta C_\Theta S_\Psi & S_\beta (S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi) & C_\alpha C_\beta (C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi) \\ -C_\alpha C_\beta S_\Theta & S_\beta (C_\Theta S_\Phi) & C_\alpha C_\beta (C_\Theta C_\Phi) \end{pmatrix} \quad (3.49)$$

利用 (3.49) 將可以得知飛彈在慣性座標上飛行的位置與速度，因此為了達到軌跡追蹤的目的，假設給定一空間軌跡 $r_d(t) = (r_1, r_2, r_3)^T$ ，其中 (r_1, r_2, r_3) 分別代表慣性座標系中的 (X, Y, Z) 位置的分量，為使設計較簡單我們將作以下假設：

- 忽略致動器的動態，我們假設 $\delta_e = \delta_{ec}, \delta_a = \delta_{ac}, \delta_r = \delta_{rc}$
- 忽略氣動力和尾翼控制偏角的耦合現象，即假設 $C_{Fu} = 0$
- 假設 α 很小，接近一常數，且 $\beta \approx 0$

由以上假設，若我們欲控制 $\Phi = 0$ ，而參考軌跡 $r_d(t)$ 與 $V_m(t), \Psi(t), \Theta(t)$ 有以下之關係式

$$\Theta(t) = \alpha - \sin^{-1}\left(\frac{\dot{r}_3}{V_m}\right) + h_1(z - r_3) \quad (3.50)$$

$$\Psi(t) = \tan^{-1}\left(\frac{\dot{r}_2}{\dot{r}_1}\right) + h_2(y - r_2) \quad (3.51)$$

$$V_m(t) = \sqrt{\dot{r}_1^2 + \dot{r}_2^2 + \dot{r}_3^2} + h_3\sqrt{(x - r_1)^2 + (y - r_2)^2 + (z - r_3)^2} \quad (3.52)$$

其中 (x, y, z) 代表飛彈在慣性座標的空間位置， h_1, h_2, h_3 依據飛彈位置與參考軌跡 $r_d(t)$ 來做適當補償。

綜合上述，飛彈系統的動態方程式可由下式來表示：

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \quad (3.53)$$

$$y(t) = h(x(t)) \quad (3.54)$$

其中

$$x = (\Psi \ \Theta \ \Phi \ \alpha \ \beta \ P \ Q \ R)^T \quad (3.55)$$

$$u = (\delta_1 \ \delta_2 \ \delta_3 \ \delta_4 \ T_x)^T \quad (3.56)$$

$$y = (V_m \ \Psi \ \Theta \ \Phi)^T \quad (3.57)$$

$$f(x) = \begin{pmatrix} Q \sin \Phi \sec \Theta + R \cos \Phi \sec \Theta \\ Q \cos \Phi - R \sin \Phi \\ P + Q \sin \Phi \tan \Theta + R \cos \Phi \tan \Theta \\ \frac{1}{m}\{C_\alpha C_\beta (F_x + g_x) + S_\beta (F_y + g_y) + S_\alpha C_\beta (F_z + g_z)\} \\ Q - (PC_\alpha + RS_\alpha \tan \beta + \frac{1}{mV_m C_\beta}\{-S_\alpha (F_x + g_x) + C_\alpha (F_z + g_z)\}) \\ PS_\alpha - RC_\alpha + \frac{1}{V_m}\{-C_\alpha S_\beta (F_x + g_x) + C_\beta (F_y + g_y) - S_\alpha S_\beta (F_z + g_z)\} \\ -\frac{I_{zz}-I_{yy}}{I_{xx}}QR + \frac{L}{I_{xx}} \\ -\frac{I_{xx}-I_{zz}}{I_{yy}}PR + \frac{M}{I_{yy}} \\ -\frac{I_{yy}-I_{xx}}{I_{zz}}PQ + \frac{N}{I_{zz}} \end{pmatrix} \quad (3.58)$$

$$g(x) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{C_\alpha C_\beta}{m} \\ 0 & 0 & 0 & 0 & \frac{-S_\alpha}{mV_m C_\beta} \\ 0 & 0 & 0 & 0 & \frac{-C_\alpha S_\beta}{mV_m} \\ C_{l1} & C_{l2} & C_{l3} & C_{l4} & 0 \\ C_{m1} & C_{m2} & C_{m3} & C_{m4} & 0 \\ C_{n1} & C_{n2} & C_{n3} & C_{n4} & 0 \end{pmatrix} \quad (3.59)$$

因此，控制的目標在於設計適當的控制律來達成 $y \rightarrow y_d$ ，其中 y_d 為預設估測來襲的飛彈軌跡。

第四章

控制器之設計

由於可變結構控制器在設計上的容易實現以及其高抗雜訊的能力，在本計畫中我們將採用可變結構控制技術來進行控制器之設計工作。本章將介紹三種控制器型態之設計，第一部份探討使用傳統非容錯的控制法則來設計控制器，第二部份將探討被動式容錯控制器的設計，其設計理念是在設計控制器時就已將系統可能的不正常情況考慮進來，使設計的控制器在正常及異常的狀態下不需切換控制法則仍能保持系統穩定的運作及維持一定的性能。而第三部份則介紹主動式容錯控制器的設計，此種設計的有效性需仰賴一個高可靠度的錯誤偵測與診斷系統，經由診斷系統所提供的系統資訊來切換控制法則，使系統能獲得較好的性能。

4.1 非容錯控制律之設計

考慮如下非線性系統：

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \quad (4.1)$$

$$y(t) = h(x(t)) \quad (4.2)$$

其中 $x(t) \in \mathcal{R}^n$ 為系統的狀態變數， $u(t) \in \mathcal{R}^m$ 為系統的輸入訊號， $y(t) \in \mathcal{R}^p$ 為系統的輸出， $f: \mathcal{R}^n \rightarrow \mathcal{R}^n$ ， $g: \mathcal{R}^{n \times m} \rightarrow \mathcal{R}^m$ ， $h: \mathcal{R}^n \rightarrow \mathcal{R}^m$ 是平滑(smooth)的函數，且 $m \geq p$ 。根據(4.1)(4.2)，我們將介紹針對系統(4.2)所設計的非容錯可變結構控制器。

定義 1 如果存在正整數 ρ_1, \dots, ρ_p ，使得系統 (4.2) 具有如下的輸出-輸入關係：

$$\begin{pmatrix} y_1^{(\rho_1)}(t) \\ y_2^{(\rho_2)}(t) \\ \vdots \\ y_p^{(\rho_p)}(t) \end{pmatrix} = B(x(t)) + A(x(t)) \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{pmatrix} \quad (4.3)$$

其中 $B: \mathcal{R}^n \rightarrow \mathcal{R}^p$ 而 $A: \mathcal{R}^n \rightarrow \mathcal{R}^{p \times m}$ 是平滑函數且 $A(x)$ 為滿秩 (*full rank*) 的矩陣。則系統 (4.2) 稱為輸出-輸入回授可線性化 (*input-output feedback linearizable*) 的系統。而 ρ_1, \dots, ρ_p 稱為系統 (4.2) 的相對階數 (*relative degree*)。

針對第三章所述之 HAVE DASH II 飛彈系統，經過直接計算我們可獲得其相對階數為 (1,2,2,2)，其輸出-輸入關係式如下：

$$\begin{aligned} \begin{pmatrix} \dot{V}_m \\ \ddot{\Psi} \\ \ddot{\Theta} \\ \ddot{\Phi} \end{pmatrix} &= \begin{pmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{pmatrix} + \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ T_x \end{pmatrix} \\ &= B(x) + A(x)u \end{aligned} \quad (4.4)$$

接下來我們選取順滑平面 $s_i = 0, i = 1, \dots, p$ ，其中

$$s_i(t) = e_i^{(\rho_i-1)} + k_{i(\rho_i-1)}e_i^{(\rho_i-2)} + \dots + k_{i2}e_i^{(1)} + k_{i1}e_i \quad (4.5)$$

且 $e_i = y_i - y_{id}$ ， y_{id} 是參考軌跡， $k_{i(\rho_i-1)}, \dots, k_{i1}$ 是選取的常數使得如下多項式

$$\lambda^{(\rho_i-1)} + k_{i(\rho_i-1)}\lambda_i^{(\rho_i-2)} + \dots + k_{i1} \quad (4.6)$$

為 Hurwitz 多項式。在這樣的選取下，如果系統持續留在順滑平面上 (即 $s = 0$ 且 $\dot{s} = 0$) 時，(4.5) 式保證追蹤誤差 $e(t) \rightarrow 0$ ，也就是軌跡追蹤的任務可以達成。其次，為了使系統狀態在有限時間到達順滑平面，我們要求閉迴路系統要滿足如下的順滑條件

$$\frac{1}{2} \frac{ds_i^2}{dt} \leq -\eta_i |s_i|, (\eta > 0) \quad (4.7)$$

其中 $\eta_i, i = 1, \dots, p$ 為正數。為了達到 (4.7) 式，我們定義

$$\dot{s} = \begin{pmatrix} \dot{s}_1 \\ \vdots \\ \dot{s}_p \end{pmatrix}, Y^{(\rho)} = \begin{pmatrix} y_1^{(\rho_1)} \\ \vdots \\ y_p^{(\rho_p)} \end{pmatrix}, \text{sgn}(s) = \begin{pmatrix} \text{sgn}(s_1) \\ \vdots \\ \text{sgn}(s_p) \end{pmatrix} \quad (4.8)$$

由 (4.7) 式的順滑條件我們可以得到如下之控制律

$$u = A^{-1}(-(\dot{s} - Y^{(\rho)}) - B - \Lambda \text{sgn}(s)) \quad (4.9)$$

其中 $\Lambda = \text{diag}[\lambda_1 \dots \lambda_p]$ 且 λ_i 是大於 η_i 的正數。附帶一提的是

$$(\dot{s}_i - y_i^{(\rho_i)}) = -y_{id}^{\rho_i} + k_{i(\rho_i-1)} e_i^{(\rho_i-1)} + \dots + k_{i2} e_i^{(2)} + k_{i1} e_i^{(1)} \quad (4.10)$$

此外，符號函數 (sign function) 的不連續性將造成閉迴路系統切跳的現象，在實際的應用上經常以飽和函數 $\text{sat}(\frac{s_i}{\varepsilon_i})$ 取代，飽和函數的數學表示式如下：

$$\text{sat}(x) = \begin{cases} x, & \text{if } |x| \leq 1 \\ \text{sgn}(x), & \text{if } |x| \geq 1 \end{cases} \quad (4.11)$$

針對飛彈系統，非容錯控制器可利用 (4.9) 來實現，而順滑平面的選取可由 (4.5) 式來得到：

$$s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} = \begin{pmatrix} e_1 \\ \dot{e}_2 - k_{21} e_2 \\ \dot{e}_3 - k_{31} e_3 \\ e_4 \end{pmatrix} \quad (4.12)$$

其中 $e_1 = V_m - V_{mc}, e_2 = \Psi - \Psi_c, e_3 = \Theta - \Theta_c$ 且 $e_4 = \Phi = 0$ ，而 V_{mc}, Ψ_c 及 Θ_c 為欲追蹤的飛彈速度、偏離角與傾斜角之軌跡，且可得 $\dot{s} - Y^{(\rho)}$ 為

$$(\dot{s} - Y^{(\rho)}) = \begin{pmatrix} \dot{s}_1 - \dot{V}_m \\ \dot{s}_2 - \dot{\Psi} \\ \dot{s}_3 - \dot{\Theta} \\ \dot{s}_4 - \dot{\Phi} \end{pmatrix} = \begin{pmatrix} -\dot{V}_{mc} \\ k_{21}(\dot{\Psi} - \dot{\Psi}_c) - \ddot{\Psi}_c \\ k_{31}(\dot{\Theta} - \dot{\Theta}_c) - \ddot{\Theta}_c \\ k_{41}\dot{\Phi} \end{pmatrix} \quad (4.13)$$

4.2 容錯控制律之設計

4.2.1 問題描述

考慮非線性系統如下：

$$\begin{aligned}\dot{x} &= f_0(x(t)) + G_0(x(t))u(t) \\ y &= h_0(x(t))\end{aligned}\tag{4.14}$$

其中 $x \in \mathcal{R}^n$, $u \in \mathcal{R}^m$, $y \in \mathcal{R}^p$ 分別表示系統狀態變數、控制輸入與系統輸出，假設 $f_0(\cdot)$ 、 $G_0(\cdot)$ 、 $h_0(\cdot)$ 為平滑向量場，除此之外，假設控制系統有以下特性：

【假設 1】：系統(4.14)有相對階數 (k_1, \dots, k_p)

在假設 1 中表示系統式(4.14)具有以下輸出-輸入的數學關係式：

$$\begin{pmatrix} y_1^{(k_1)} \\ y_2^{(k_2)} \\ \vdots \\ y_p^{(k_p)} \end{pmatrix} = f(x) + G(x)u + d\tag{4.15}$$

其中 $f: \mathcal{R}^n \rightarrow \mathcal{R}^p$, $u \in \mathcal{R}^m$, $d \in \mathcal{R}^p$, $G: \mathcal{R}^n \rightarrow \mathcal{R}^{p \times m}$ 且 $G(x)$ 為滿秩 (full rank), d 用來表現可能出現的不確定模式，量測雜訊跟外部干擾，此外，在假設 1 中 $m > p$ 表示控制系統在輸出追蹤任務中將有多餘的控制元件，在不失一般性的情況下，我們假設 $(g_1(x) \cdots g_p(x))$ 是一個非奇異的矩陣，且容許 u_{p+1}, \dots, u_m 是可以發生錯誤的，在此定義 $g_i(x)$ 為矩陣 $G(x)$ 的第 i 個行。

此研究的主要目的是綜合一種控制律使得即使當系統致動器 u_{p+1}, \dots, u_m 發生錯誤時，輸出追蹤性能仍然可以達成。也就是說當 $t \rightarrow \infty$ 時對所有 $i = 1, \dots, p$ 而言仍可達成 $y_i(t) \rightarrow y_{id}(t)$ ，其中 y_{id} 是指參考輸出軌跡。

4.2.2 被動式容錯控制律之設計

假設致動器 u_{p+1}, \dots, u_m 發生錯誤，我們將 (4.15) 改寫為

$$\begin{pmatrix} y_1^{(k_1)} \\ y_2^{(k_2)} \\ \vdots \\ y_p^{(k_p)} \end{pmatrix} = f(x) + G(x)u + d + \sum_{i=p+1}^m g_i(x)(u_j^* - u_j) \quad (4.16)$$

其中 u_j^* 為第 j 個控制輸入的實際值，且 u_j 為欲達到的控制輸入，定義輸出追蹤誤差為

$$e_i(t) = y_i(t) - y_{id}(t) \quad (4.17)$$

以及順滑平面為

$$s_i(t) = e_i^{(k_i-1)}(t) + a_{i(k_i-1)}e_i^{(k_i-2)}(t) + \dots + a_{i2}e_i^{(1)}(t) + a_{i1}e_i(t) \quad (4.18)$$

其中 $i = 1, \dots, p$, $a_{ij}, i = 1, \dots, p, j = 1, \dots, k_i - 1$ 為所選擇的常數，使得

$$\lambda^{(k_i-1)} + a_{i(k_i-1)}\lambda_i^{(k_i-2)} + \dots + a_{i2}\lambda + a_{i1} \quad (4.19)$$

為 Hurwitz 多項式，如果系統狀態維持在順滑平面上，則輸出追蹤性能將可以被達成，此外，對於所有 i, \dots, p 而言，利用 (4.16)(4.17)(4.18) 式可以推導得到下列式子

$$\begin{aligned} \dot{s}_i(t) = & f_i(x) + \sum_{j=1}^p g_{ij}(x)u_j + \sum_{j=p+1}^m g_{ij}(x)u_j^* + d_i - y_{id}^{(k_i)} \\ & + a_{i(k_i-1)}e_i^{(k_i-1)}(t) + \dots + a_{i2}\ddot{e}_i(t) + a_{i1}\dot{e}_i(t) \end{aligned} \quad (4.20)$$

$g_{ij}(x)$ 為矩陣 $G(x)$ 的 (i, j) 項。而完整的控制器為以下型式

$$u_i = u_i^{eq} + u_i^{re}, \quad i = 1, \dots, n \quad (4.21)$$

u_i^{re} 使系統狀態達到順滑平面， u_i^{eq} 能使得順滑平面成為一不變的曲面 (invariant manifold)，為了滿足要求我們假設

$$\begin{pmatrix} u_1^{eq} \\ \vdots \\ u_p^{eq} \end{pmatrix} = - (g_1 \quad \dots \quad g_p)^{-1} \begin{pmatrix} f_1(x) - y_{1d}^{(k_1)} + a_{1(k_1-1)}e_1^{(k_1-1)}(t) + \dots + a_{11}\dot{e}_1(t) \\ \vdots \\ f_n(x) - y_{nd}^{(k_n)} + a_{n(k_n-1)}e_n^{(k_n-1)}(t) + \dots + a_{n1}\dot{e}_n(t) \end{pmatrix} \quad (4.22)$$

由 (4.20)(4.21)(4.22) 式，可以得到

$$\dot{s}_i(t) = \sum_{j=1}^p g_{ij}(x)u_j^{re} + \sum_{j=p+1}^m g_{ij}(x)u_j^* + d_i \quad (4.23)$$

為了保證滿足追蹤條件，增加下一個假設：

【假設 2】：存在一個非負的函數 $\rho_i(x, t), i = 1, \dots, p$ ，使得

$$\left| \sum_{j=p+1}^m g_{ij}(x)u_j^* + d_i \right| \leq \rho_i(x, t) \quad (4.24)$$

由 (4.23) 可以得到

$$s_i(t)\dot{s}_i(t) = s_i(t)\left(\sum_{j=1}^p g_{ij}(x)u_j^{re} + \sum_{j=p+1}^m g_{ij}(x)u_j^* + d_i\right) \quad (4.25)$$

我們選擇

$$\begin{pmatrix} u_1^{re} \\ \vdots \\ u_p^{re} \end{pmatrix} = -\begin{pmatrix} g_1(x) & \cdots & g_p(x) \end{pmatrix}^{-1} \begin{pmatrix} (\rho_1 + \eta_1) \text{sgn}(s_1) \\ \vdots \\ (\rho_p + \eta_p) \text{sgn}(s_p) \end{pmatrix} \quad (4.26)$$

其中 $\eta_i, i = 1, \dots, p$ 為所選擇的正數，可獲得

$$s_i(t)\dot{s}_i(t) \leq -\eta_i |s_i|, \quad (\eta > 0) \quad (4.27)$$

上式能保證系統狀態能在有限時間內到達所選擇的順滑平面。

上述為考慮控制輸入如果發生故障的情形，接下來我們考慮操作在正常情況時，

在相同的 $e_i, s_i, u_i, i = 1, \dots, p$ 設定下，我們得到

$$\dot{s}_i(t) = \sum_{j=1}^p g_{ij}(x)u_j^{re} + \sum_{j=p+1}^m g_{ij}(x)u_j + d_i \quad (4.28)$$

以及

$$s_i(t)\dot{s}_i(t) = s_i(t)\left(\sum_{j=1}^p g_{ij}(x)u_j^{re} + \sum_{j=p+1}^m g_{ij}(x)u_j + d_i\right) \quad (4.29)$$

我們可選取

$$(u_{p+1}^{eq}, \dots, u_m^{eq})^T \in N(g_{p+1}(x) \cdots g_m(x)) \quad (4.30)$$

與

$$u_i^{re} = -\eta \cdot \text{sgn}(s^T g_i(x)) \text{ for } j = p+1, \dots, m \quad (4.31)$$

(4.29) 及 (2.31) 式將會保持順滑平面為一不變的曲面 (invariant manifold) 以及增加系統狀態收斂至順滑表面的速度，在此定義 $N(\cdot)$ 為矩陣零空間 (null space) 對於 $i = p+1, \dots, m$ 而言，而 $u_i^{eq} = 0$ 為滿足 (4.30) 式的一明顯解。

4.2.3 主動式容錯控制律之設計

上一節所討論的是被動式的容錯控制律，雖然也可以達到追蹤性能，但被動式控制是一種較保守的方法，它往往會過度評估錯誤程度，然而過度預估將可能導致設計出控制力大於系統的最大物理轉矩限制，並且導致不良的性能，本節將探討主動式控制率的設計方法。我們假設當錯誤發生時主動式控制會被成功的偵測與診斷如下：

$$u_j^* = u_j^0 + \Delta u_j, \quad j = p+1, \dots, m \quad (4.32)$$

其中 u_j^0 與 Δu_j 分為定義為估測控制值與估算誤差值，因此在 (4.20) 式中的 $\dot{s}_i(t)$ 與 (4.22) 式中的 u_i^{eq} 被修正為

$$\begin{aligned} \dot{s}_i(t) = f_i(x) + \sum_{j=1}^p g_{ij}(x)u_j + \sum_{j=p+1}^m g_{ij}(x)[u_j^0 + \Delta u_j] + d_i - y_{id}^{(k_i)} \\ + a_{i(k_i-1)}e_i^{(k_i-1)}(t) + \dots + a_{i2}\ddot{e}_i(t) + a_{i1}\dot{e}_i(t) \end{aligned} \quad (4.33)$$

$$\begin{pmatrix} u_1^{eq} \\ \vdots \\ u_p^{eq} \end{pmatrix} = - (g_1 \quad \dots \quad g_p)^{-1} \quad (4.34)$$

$$\begin{pmatrix} f_1(x) + \sum_{j=p+1}^m g_{1j}(x)u_j^0 - y_{1d}^{(k_1)} + a_{1(k_1-1)}e_1^{(k_1-1)}(t) + \dots + a_{11}\dot{e}_1(t) \\ \vdots \\ f_n(x) + \sum_{j=p+1}^m g_{nj}(x)u_j^0 - y_{nd}^{(k_n)} + a_{n(k_n-1)}e_n^{(k_n-1)}(t) + \dots + a_{n1}\dot{e}_n(t) \end{pmatrix}$$

在(4.34)式中的 u_i^{eq} 多了一項包含診斷的訊息 $\sum_{j=p+1}^m g_{ij}(x)u_j^0$ ，且可以得到以下的結果：

$$s_i(t)\dot{s}_i(t) = s_i(t)\left(\sum_{j=1}^p g_{ij}(x)u_j^{re} + \sum_{j=p+1}^m g_{ij}(x)\Delta u_j + d_i\right) \quad (4.35)$$

在主動式的控制律中，我們使用 $\sigma_i(x, t)$ 去表示對系統模型不確定性與干擾的估測上界如下：

$$\left| \sum_{j=1}^p g_{ij}(x)u_j^{re} + \sum_{j=p+1}^m g_{ij}(x)\Delta u_j + d_i \right| \leq \sigma_i(x, t) \quad (4.36)$$

由(4.36)式中可以看出如果 $|\Delta u_j|$ 夠小，估測上界 $\sigma_i(x, t)$ 將會比在(4.24)式中得到的值來得更小，這也是說，如果診斷越精確則估測上界越小，最後將(4.36)所得到的 $\sigma_i(x, t)$ 代入(4.26)式中可求得 u_i^{re} 。

第五章

錯誤偵測與診斷觀察器之設計

在上一章中，我們提到了主動式容錯控制律之設計，其需要一有效的錯誤偵測與診斷的機制，用來提供系統的資訊，而使得主動式容錯控制器能夠適時地轉換控制律，以期系統能得到較佳的性能。故本章將介紹錯誤偵測診斷與觀察器的設計。透過變數變換所設計的觀察器，能將其產生的殘值訊號(residual signal)作解耦(decouple)，經由解耦的訊號可以得知致動器發生故障的位置與嚴重性。

5.1 問題描述

在此計畫中我們考慮如下之非線性系統

$$\dot{x} = f(x) + g(x)u, \quad f(0) = 0 \quad (5.1)$$

其中 $x \in \mathcal{R}^n$ 代表狀態變數(State variable)， $u \in \mathcal{R}^m$ 代表控制輸入(Control input)， $g(x) = (g_1(x), \dots, g_m(x)) \in \mathcal{R}^{n \times m}$ ，且 $f(x)$ 及 $g(x)$ 都是平滑向量場(Smooth vector field)。本計畫之錯誤偵測與診斷的主要目標為設計一觀察器使得當致動器發生故障時能夠即時的偵測出來，且能成功的診斷出哪一個致動器損壞，以提供主動式容錯控制切換控制律的依據。

接下來，我們將透過設計觀察器來進行錯誤偵測與診斷的工作，並由觀察器所產生的residual來判斷系統故障的產生及故障的來源。整個錯誤偵測與診斷系統描述如圖5.1所示， $\delta_1, \delta_2, \delta_3, \delta_4$ 分別為四個致動器，另一輸入飛彈推力(Tx)為不可故障之輸入，在此我們不討論(Tx)故障的情形，只考慮致動器故障的情況，而Plant為

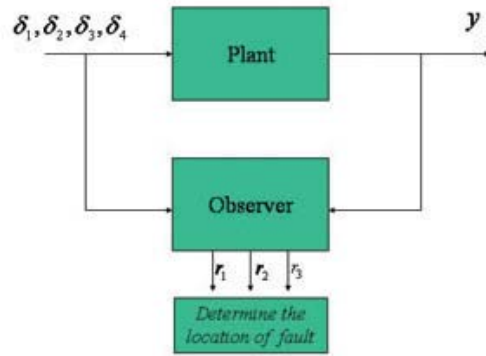


圖 5.1 偵測與診斷架構圖

所要被觀測的系統，在此為飛彈系統。藉由輸出 y 以及四個致動器的輸入來設計觀察器，我們要求此觀察器所產生的 residual signals，可以偵測出致動器的故障並診斷出故障的來源。

有了設計的概念後，我們接下來要做的是將原系統作變數變換，由第三章飛彈系統模型的描述，可以得知飛彈系統為九個狀態 $x = (\Psi, \Theta, \Phi, V_m, \alpha, \beta, P, Q, R)$ 所組成的系統。假設其中 α, β, P, Q, R 五個狀態是可以量測得到的訊號，將飛彈的模型化為 (3.54) 式的型式，可如下：

$$\begin{aligned}
 \dot{x}_1 &= f_1(x_1 \cdots x_9) \\
 &\vdots \\
 \dot{x}_4 &= f_4(x_1 \cdots x_9) + T_x \\
 \dot{x}_5 &= f_5(x_1 \cdots x_9) + T_x \\
 \dot{x}_6 &= f_6(x_1 \cdots x_9) + T_x \\
 \begin{pmatrix} \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{pmatrix} &= \begin{pmatrix} f_7(x_5 \cdots x_9) \\ f_8(x_5 \cdots x_9) \\ f_9(x_5 \cdots x_9) \end{pmatrix} + \begin{pmatrix} g_{11} \cdots g_{14} \\ g_{21} \cdots g_{24} \\ g_{31} \cdots g_{34} \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{pmatrix} \quad (5.2)
 \end{aligned}$$

將原來的模型化為 (3.54) 式的型式後，可以清楚的看到只有四個致動器輸入只出現

在 $\dot{x}_7, \dot{x}_8, \dot{x}_9$ 三個狀態方程式中，而在 $\dot{x}_4, \dot{x}_5, \dot{x}_6$ 有 T_x 的控制輸入，但 T_x 為不可損壞的輸入裝置，在此不考慮 T_x 損壞的情形。為了達到我們所希望的解耦現象，我們針對原系統作變數變換

$$Z = \bar{M}X \quad (5.3)$$

其中

$$\bar{M} = \begin{pmatrix} I_{6 \times 6} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_{3 \times 3} \end{pmatrix} \quad (5.4)$$

$$M_{3 \times 3} = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix}^{-1} \quad (5.5)$$

新的狀態方程式可改寫成

$$\begin{aligned} \dot{Z} = \bar{M}\dot{X} &= \bar{M}f(x) + \bar{M}g(x)u \\ &= \bar{M}f(\bar{M}^{-1}Z + \bar{M}g(x)) \\ &= f_{new}(z) + \bar{M}g(x) \end{aligned} \quad (5.6)$$

其中 $g(x)$ ， $f_{new}(z)$ 與 $\bar{M}g(x)$ 分別為

$$g(x) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ g_{11} & g_{12} & g_{13} & g_{14} \\ g_{21} & g_{22} & g_{23} & g_{24} \\ g_{31} & g_{32} & g_{33} & g_{34} \end{pmatrix} \quad (5.7)$$

$$f_{new}(z) = \begin{pmatrix} f_1(z) \\ f_2(z) \\ f_3(z) \\ f_4(z) \\ f_5(z) \\ f_6(z) \\ f_{7new}(z) \\ f_{8new}(z) \\ f_{9new}(z) \end{pmatrix} \quad (5.8)$$

$$\bar{M}g(x) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & l_2 \\ 0 & 0 & 1 & l_3 \end{pmatrix} \quad (5.9)$$

我們可將新的狀態方程式整理成

$$\begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \\ \dot{z}_5 \\ \dot{z}_6 \\ \dot{z}_7 \\ \dot{z}_8 \\ \dot{z}_9 \end{pmatrix} = \begin{pmatrix} f_{1new}(z) \\ f_{2new}(z) \\ f_{3new}(z) \\ f_{4new}(z) \\ f_{5new}(z) \\ f_{6new}(z) \\ f_{7new}(z) + \delta_1 + l_1\delta_4 \\ f_{8new}(z) + \delta_2 + l_2\delta_4 \\ f_{9new}(z) + \delta_3 + l_3\delta_4 \end{pmatrix} \quad (5.10)$$

將原來的飛彈系統模型經過變數變換後，得到的新的狀態方程式(5.10)式。在(5.10)中， \dot{z}_7 受到 δ_1 與 δ_4 兩致動器的影響， \dot{z}_8 受到 δ_2 與 δ_4 兩致動器的影響， \dot{z}_9 受到 δ_3 與 δ_4 兩致動器的影響。由於任何兩個尾翼故障時，軌跡追蹤無法達成，因此本計畫只考慮單一尾翼故障的狀況。

5.2 觀察器的設計

針對座標轉換後的系統(5.10)我們設計觀察器及殘量信號(residual signal)如下：

$$\dot{o}_1 = f_{7new}(z) + \delta_1 + l_1\delta_4 + k_1(z_7 - o_1) \quad (5.11)$$

$$\dot{o}_2 = f_{8new}(z) + \delta_2 + l_2\delta_4 + k_2(z_8 - o_2) \quad (5.12)$$

$$\dot{o}_3 = f_{9new}(z) + \delta_3 + l_3\delta_4 + k_3(z_9 - o_3) \quad (5.13)$$

$$r_1 = z_7 - o_1 \quad (5.14)$$

$$r_2 = z_8 - o_2 \quad (5.15)$$

$$r_3 = z_9 - o_3 \quad (5.16)$$

在 (5.11)(5.12)(5.13) 中， $k_i > 0, i = 1, 2, 3$ 。

在此設計下，任何一個單一致動器故障都可經由所產生的 residual 信號偵測及診斷出來。實際上，當第一個致動器 δ_1 故障時，residual 信號將會出現 $|r_1| \neq 0$ 但 $|r_2| = |r_3| = 0$ ；第二個致動器 δ_2 故障時，residual 信號將會出現 $|r_2| \neq 0$ 但 $|r_1| = |r_3| = 0$ ；第三個致動器 δ_3 故障時，residual 信號將會出現 $|r_3| \neq 0$ 但 $|r_1| = |r_2| = 0$ ；第四個致動器 δ_4 故障時， $|r_1| \neq 0$ 、 $|r_2| \neq 0$ 且 $|r_3| \neq 0$ 。如此一來，錯誤訊號就可以被解耦並且成功的被偵測診斷出來。詳細情況說明如下：

考慮第一個致動器 δ_1 出現卡死的情形時，假設實際卡死的角度為 δ_1^* ，我們將錯誤訊號以如下之 m_1 來表示

$$m_1 = \delta_1^* - \delta_1 \quad (5.17)$$

在此情況下 \dot{z}_7 的狀態方程式變成

$$\dot{z}_7 = f_{7new} + \delta_1 + l_1 \delta_4 + m_1 \quad (5.18)$$

由 (5.10)~(5.16) 式可以得到誤差訊號

$$\begin{aligned} e_1 &\equiv z_7 - o_1 \\ \Rightarrow \dot{e}_1 &= -k_1 e_1 + m_1 \end{aligned} \quad (5.19)$$

根據前面所提到 $k_i > 0$ ，所以只要適當的選取 $k_i > 0$ ，則經過一段短暫時間後 e_1 會趨近至 $\frac{-m_1}{k_1}$ 。從 (5.14)(5.15)(5.16) 及 (5.19) 式可以看出 residual signal r_1 只受到錯誤訊號 m_1 的影響，並且

$$m_1 \neq 0 \Leftrightarrow r_1 \neq 0 \quad (5.20)$$

同樣地，我們可以定義

$$m_i = \delta_i^* - \delta_i \quad i = 1, 2, 3, 4 \quad (5.21)$$

$$e_2 = z_8 - o_2 \quad (5.22)$$

$$e_3 = z_9 - o_3 \quad (5.23)$$

則當第*i*個致動器， $i = 1, 2, 3$ ，發生卡死情況時，

$$\dot{e}_i = -k_i e_i + m_i \quad (5.24)$$

而當第4個致動器發生卡死時，

$$\dot{e}_i = -k_i e_i + m_i \quad i = 1, 2, 3 \quad (5.25)$$

由(5.24)及(5.25)可以清楚的看到當 $\delta_i, i = 1, 2, 3$ ，發生卡死現象時只有 $m_i \neq 0$ 及 $r_i \neq 0$ 其餘 $r_j = 0, j \neq i$ 。但若 δ_4 發生故障時表示 $m_4 \neq 0$ ，由(5.25)知對所有 $i = 1, 2, 3, e_i \neq 0$ ，也就是說所有residual信號 r_i 都會反映出 δ_4 的故障。因此，透過監測residual信號可有效的偵測並診斷單一尾翼發生故障的狀況。由於把握飛彈系統的動態特性，利用變數變換的方式將殘值訊號作解耦分析，可以清楚的偵測出錯誤的發生及錯誤的來源，此方法簡單可免去許多複雜偵錯的步驟或演算法。

第六章

模擬結果與討論

本章節將針對HAVE DASH II 飛彈系統之非線性動態模式進行飛彈尾翼錯誤偵測與診斷及容錯控制器設計之模擬，並觀察飛彈追蹤的性能表現。本次模擬是利用MATLAB及SIMNON兩種套裝軟體來進行系統模擬。模擬包含下列幾種結果：

- 致動器發生故障時之偵測結果(包含 δ_1 、 δ_2 、 δ_3 及 δ_4 中任何一組產生故障之情況)。
- 用傳統非容錯控制律設計方式模擬全部致動器皆正常及致動器 δ_1 、 δ_2 、 δ_3 及 δ_4 中任何一組產生故障之情況。
- 將致動器 δ_1 、 δ_2 、 δ_3 及 δ_4 中任何一組產生故障之情況所設計對應之被動式容錯控制律用在模擬全部致動器皆正常的情況。
- 針對 δ_1 、 δ_2 、 δ_3 及 δ_4 任何一組尾翼發生故障時，由主動式容錯控制器的模擬情況

最後，我們將比較正常情況下設計之主動式控制器和各尾翼卡死之情況時設計之主動式和被動式控制器的性能表現。

6.1 飛彈參數設定

在此次模擬中，我們採用的飛彈參數設定如下：

(1) 飛彈之轉動慣量：

$$I_{xx} = 1.1913(\text{slug} - ft^2),$$

$$I_{yy} = 100.51(\text{slug} - ft^2),$$

$$I_{zz} = 100.57(\text{slug} - ft^2)$$

(2) 重力加速度： $g = 32.174(ft/sec^2)$

(3) 飛彈質量： $m = 9.89(\text{slug})$

(4) $k_F = 0.1534(ft^2), k_M = 0.0959(ft^3)$ (由飛彈幾何模型決定)

(5) 聲速： $c = 968(ft/sec)$

(6) 大氣密度： $\rho = 5.124 \times 10^{-4}(\text{slug}/ft^3)$

6.2 觀察器設計

為了方便觀察器的設計，首先，將第三章所介紹的飛彈系統模型化成底下(6.1)式的型式：

$$\dot{x} = \begin{pmatrix} \dot{\Psi} \\ \dot{\Theta} \\ \dot{\Phi} \\ \dot{V}_m \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{P} \\ \dot{Q} \\ \dot{R} \end{pmatrix} = \begin{pmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \\ f_5(x) \\ f_6(x) \\ f_7(x) \\ f_8(x) \\ f_9(x) \end{pmatrix} + g(x) \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{pmatrix} \quad (6.1)$$

其中 $(f_1(x), \dots, f_9(x))^T$ 及 $g(x)$ 分別如 (3.58) 與 (3.59)。將實際的參數 $k_M, \rho, I_{xx}, I_{yy}, I_{zz}$

帶入 $g(x)$ 可得

$$g(x) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.31 \times 10^{-6} V_m^2 & 1.31 \times 10^{-6} V_m^2 & 1.31 \times 10^{-6} V_m^2 & 1.31 \times 10^{-6} V_m^2 \\ -8.25 \times 10^{-8} V_m^2 & 8.25 \times 10^{-8} V_m^2 & 8.25 \times 10^{-8} V_m^2 & -8.25 \times 10^{-8} V_m^2 \\ -7.13 \times 10^{-8} V_m^2 & -7.13 \times 10^{-8} V_m^2 & 7.13 \times 10^{-8} V_m^2 & -7.13 \times 10^{-8} V_m^2 \end{pmatrix} \quad (6.2)$$

接下來我們選擇一變數變換矩陣 \bar{M}

$$\bar{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.82 \times 10^5 / V_m^2 & -0.06 \times 10^8 / V_m^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.06 \times 10^8 / V_m^2 & -0.07 \times 10^8 / V_m^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.82 \times 10^5 / V_m^2 & 0 & 0.07 \times 10^8 / V_m^2 \end{pmatrix} \quad (6.3)$$

由 (6.3) 可得到新的狀態變數 Z

$$\dot{Z} = f_{new}(z) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \delta_1 + \delta_4 \\ \delta_2 - \delta_4 \\ \delta_3 + \delta_4 \end{pmatrix} = \begin{pmatrix} f_{1new}(z) \\ f_{2new}(z) \\ f_{3new}(z) \\ f_{4new}(z) \\ f_{5new}(z) \\ f_{6new}(z) \\ f_{7new}(z) + \delta_1 + \delta_4 \\ f_{8new}(z) + \delta_2 - \delta_4 \\ f_{9new}(z) + \delta_3 + \delta_4 \end{pmatrix} \quad (6.4)$$

其中 $f_{new}(z)$ 為

$$f_{new}(z) = \bar{M}f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \\ f_5(x) \\ f_6(x) \\ 3.82 \times 10^5 / V_m^2 f_7(x) - 0.06 \times 10^8 / V_m^2 f_8(x) \\ 0.06 \times 10^8 / V_m^2 f_8(x) - 0.07 \times 10^8 / V_m^2 f_9(x) \\ 3.82 \times 10^5 / V_m^2 f_7(x) + 0.07 \times 10^8 / V_m^2 f_9(x) \end{pmatrix} \quad (6.5)$$

因此，經過座標轉換後可以得到新的系統方程式

$$\begin{aligned}
\dot{z}_1 &= f_1(x) \\
\dot{z}_2 &= f_2(x) \\
\dot{z}_3 &= f_3(x) \\
\dot{z}_4 &= f_4(x) \\
\dot{z}_5 &= f_5(x) \\
\dot{z}_6 &= f_6(x) \\
\dot{z}_7 &= 3.82 \times 10^5 / V_m^2 f_7(x) - 0.06 \times 10^8 / V_m^2 f_8(x) + \delta_1 + \delta_4 \\
\dot{z}_8 &= 0.06 \times 10^8 / V_m^2 f_8(x) - 0.07 \times 10^8 / V_m^2 f_9(x) + \delta_2 - \delta_4 \\
\dot{z}_9 &= 3.82 \times 10^5 / V_m^2 f_7(x) + 0.07 \times 10^8 / V_m^2 f_9(x) + \delta_3 + \delta_4
\end{aligned} \tag{6.6}$$

由(6.6)式，可以設計觀察器如下：

$$\begin{aligned}
o_1 &= 3.82 \times 10^5 / V_m^2 f_7(x) - 0.06 \times 10^8 / V_m^2 f_8(x) + \delta_1 + \delta_4 + k_1(z_7 - o_1) \\
o_2 &= 0.06 \times 10^8 / V_m^2 f_8(x) - 0.07 \times 10^8 / V_m^2 f_9(x) + \delta_2 - \delta_4 + k_2(z_8 - o_2) \\
o_3 &= 3.82 \times 10^5 / V_m^2 f_7(x) + 0.07 \times 10^8 / V_m^2 f_9(x) + \delta_3 + \delta_4 + k_3(z_9 - o_3)
\end{aligned} \tag{6.7}$$

選擇 $k_1 = k_2 = k_3 = 10$ ，並且令 residual signals 為

$$\begin{aligned}
r_1 &= z_7 - o_1 \\
r_2 &= z_8 - o_2 \\
r_3 &= z_9 - o_3
\end{aligned} \tag{6.8}$$

就完成了觀察器的設計。

6.3 控制器設計

根據可變結構控制法則，我們先使用傳統非容錯控制律來設計，隨後再進一步討論被動式容錯控制器與主動式容錯控制器結合觀察器的設計方法，在

(3.55)~(3.59) 我們已把 HAVE DASH II 飛彈系統化做了標準型式 (3.54)，經過計算可以得到飛彈系統的相對階數為 (1,2,2,2)，利用此特性可將飛彈系統描述為下列輸出-輸入之動態數學模式：

$$\begin{aligned} \begin{pmatrix} \dot{V}_m \\ \ddot{\Psi} \\ \ddot{\Theta} \\ \ddot{\Phi} \end{pmatrix} &= \begin{pmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{pmatrix} + \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \end{pmatrix} \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ T_x \end{pmatrix} \\ &= B(x) + A(x)u \end{aligned} \quad (6.9)$$

其中 $A(x)$ 可表示如下：

$$A_{11} = A_{12} = A_{13} = A_{14} = 0, A_{15} = C_\alpha C_\beta / m$$

$$A_{25} = -h_2 C_\alpha C_\beta (C_\alpha C_\beta S_\Psi C_\Theta + S_\alpha C_\beta (S_\Psi S_\Theta C_\Phi - C_\Psi S_\Phi)) / m$$

$$A_{35} = -h_1 C_\alpha C_\beta (-C_\alpha C_\beta S_\Theta + S_\alpha C_\beta C_\Theta C_\Phi) / m$$

$$\begin{aligned} &-h_3 \frac{\dot{r}_3 C_\alpha C_\beta (C_\alpha C_\beta C_\Psi C_\Theta + S_\alpha C_\beta (C_\Psi S_\Theta C_\Phi + S_\Psi S_\Phi)) (x - r_1)}{m V_m^2 \sqrt{1 - (\dot{r}_3 / V_m)^2} \sqrt{(x - r_1)^2 + (y - r_2)^2 + (z - r_3)^2}} \\ &-h_3 \frac{\dot{r}_3 C_\alpha C_\beta (C_\alpha C_\beta S_\Psi C_\Theta + S_\alpha C_\beta (S_\Psi S_\Theta C_\Phi - C_\Psi S_\Phi)) (y - r_2)}{m V_m^2 \sqrt{1 - (\dot{r}_3 / V_m)^2} \sqrt{(x - r_1)^2 + (y - r_2)^2 + (z - r_3)^2}} \\ &-h_3 \frac{\dot{r}_3 C_\alpha C_\beta (-C_\alpha C_\beta S_\Theta + S_\alpha C_\beta C_\Theta C_\Phi) (z - r_3)}{m V_m^2 \sqrt{1 - (\dot{r}_3 / V_m)^2} \sqrt{(x - r_1)^2 + (y - r_2)^2 + (z - r_3)^2}} \end{aligned}$$

$$\begin{pmatrix} A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{pmatrix} = \frac{k_M \rho V_m^2}{4} \begin{pmatrix} 0 & \frac{S_\Phi}{C_\Theta I_{yy}} & \frac{C_\Phi}{C_\Theta I_{zz}} \\ 0 & \frac{C_\Phi}{I_{yy}} & -\frac{S_\Phi}{I_{zz}} \\ \frac{1}{I_{xx}} & \frac{S_\Phi S_\Theta}{C_\Theta I_{yy}} & \frac{C_\Phi S_\Theta}{C_\Theta I_{zz}} \end{pmatrix} C_M u W$$

其中 $C_M u$ 與 W 分別為

$$C_M u = \begin{pmatrix} 0 & -0.127 & 0 \\ -0.675 & 0 & 0 \\ 0 & 0 & -0.584 \end{pmatrix} \quad W = \begin{pmatrix} 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 \end{pmatrix}$$

在確定上述飛彈系統之輸入-輸出數學動態模式後，接下來要進行控制器的推導，首先選取順滑表面為下列的形式：

$$s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} = \begin{pmatrix} e_1 \\ \dot{e}_2 - k_{21} e_2 \\ \dot{e}_3 - k_{31} e_3 \\ e_4 \end{pmatrix}$$

定義 $e_1 = V_m - V_{mc}$ 、 $e_2 = \Psi - \Psi_c$ 、 $e_3 = \Theta - \Theta_c$ 且 $e_4 = \Phi$ ，其中 V_{mc} 、 Ψ_c 及 Θ_c 分別為欲追蹤的飛彈速度、偏離角與傾斜角之軌跡。根據上述順滑曲面的選取，再利用 (4.9) 式即可得到非容錯的控制器。

接下來，我們考慮容錯控制器的設計，在被動式容錯控制器中曾經提到，需先把致動器故障的情形考慮在控制器的設計當中，我們以 δ_4 故障的情況與例來設計被動式容錯控制器，其 $\delta_1, \delta_2, \delta_3$ 也類似此設計。同時，根據錯誤偵測與診斷結果設計符合 $\delta_1, \delta_2, \delta_3$ 及 δ_4 任一致動器故障時之主動式容錯控制器，其結果設計如下：

(1) 被動式容錯控制器 (針對 δ_4 故障設計)

$$\begin{aligned} \begin{pmatrix} \delta_1^{eq} \\ \delta_2^{eq} \\ \delta_3^{eq} \\ T_x^{eq} \end{pmatrix} &= (g_1 \quad g_2 \quad g_3 \quad g_5)^{-1} \begin{pmatrix} -B_1 + \dot{V}_{mc} - e_1 \\ -B_2 + \ddot{\Psi} - \dot{e}_2 - a_{21}e_2 \\ -B_3 + \ddot{\Theta} - \dot{e}_3 - a_{31}e_3 \\ -B_4 - a_{41}\dot{e}_4 \end{pmatrix} \\ \delta_4^{eq} &= 0 \\ \begin{pmatrix} \delta_1^{re} \\ \delta_2^{re} \\ \delta_3^{re} \\ T_x^{re} \end{pmatrix} &= (g_1 \quad g_2 \quad g_3 \quad g_5)^{-1} \begin{pmatrix} -(\rho_1 + \eta_1) \cdot \text{sgn}(s_1) \\ -(\rho_2 + \eta_2) \cdot \text{sgn}(s_2) \\ -(\rho_3 + \eta_3) \cdot \text{sgn}(s_3) \\ -(\rho_5 + \eta_5) \cdot \text{sgn}(s_5) \end{pmatrix} \\ \delta_4^{re} &= -\eta_4 \cdot \text{sgn}(s^T g_4) \end{aligned}$$

(2) 主動式容錯控制器

a. 針對 δ_1 故障之狀況設計

$$\begin{aligned} \begin{pmatrix} \delta_2^{eq} \\ \delta_3^{eq} \\ \delta_4^{eq} \\ T_x^{eq} \end{pmatrix} &= (g_2 \quad g_3 \quad g_4 \quad g_5)^{-1} \begin{pmatrix} -B_1 + \dot{V}_{mc} - e_1 - g_{21}\delta_1^0 \\ -B_2 + \ddot{\Psi}_c - \dot{e}_2 - a_{21}e_2 - g_{31}\delta_1^0 \\ -B_3 + \ddot{\Theta}_c - \dot{e}_3 - a_{31}e_3 - g_{41}\delta_1^0 \\ -B_4 - a_{41}\dot{e}_4 - g_{51}\delta_1^0 \end{pmatrix} \\ \begin{pmatrix} \delta_2^{re} \\ \delta_3^{re} \\ \delta_4^{re} \\ T_x^{re} \end{pmatrix} &= (g_2 \quad g_3 \quad g_4 \quad g_5)^{-1} \begin{pmatrix} -(\sigma_2 + \eta_2) \cdot \text{sgn}(s_2) \\ -(\sigma_3 + \eta_3) \cdot \text{sgn}(s_3) \\ -(\sigma_3 + \eta_4) \cdot \text{sgn}(s_4) \\ -(\sigma_5 + \eta_5) \cdot \text{sgn}(s_5) \end{pmatrix} \end{aligned}$$

其中 δ_1^0 為 δ_1 偵測器偵測到卡死的角度

b. 針對 δ_2 故障之狀況設計

$$\begin{pmatrix} \delta_1^{eq} \\ \delta_3^{eq} \\ \delta_4^{eq} \\ T_x^{eq} \end{pmatrix} = (g_1 \quad g_3 \quad g_4 \quad g_5)^{-1} \begin{pmatrix} -B_1 + \dot{V}_{mc} - e_1 - g_{12}\delta_2^0 \\ -B_2 + \ddot{\Psi}_c - \dot{e}_2 - a_{21}e_2 - g_{32}\delta_2^0 \\ -B_3 + \ddot{\Theta}_c - \dot{e}_3 - a_{31}e_3 - g_{42}\delta_2^0 \\ -B_4 - a_{41}\dot{e}_4 - g_{52}\delta_2^0 \end{pmatrix}$$

$$\begin{pmatrix} \delta_1^{re} \\ \delta_3^{re} \\ \delta_4^{eq} \\ T_x^{re} \end{pmatrix} = (g_1 \quad g_3 \quad g_4 \quad g_5)^{-1} \begin{pmatrix} -(\sigma_1 + \eta_1) \cdot \text{sgn}(s_1) \\ -(\sigma_3 + \eta_3) \cdot \text{sgn}(s_3) \\ -(\sigma_4 + \eta_4) \cdot \text{sgn}(s_4) \\ -(\sigma_5 + \eta_5) \cdot \text{sgn}(s_5) \end{pmatrix}$$

其中 δ_2^0 為 δ_2 偵測器偵測到卡死的角度

c. 針對 δ_3 故障之狀況設計

$$\begin{pmatrix} \delta_1^{eq} \\ \delta_2^{eq} \\ \delta_4^{eq} \\ T_x^{eq} \end{pmatrix} = (g_1 \quad g_2 \quad g_4 \quad g_5)^{-1} \begin{pmatrix} -B_1 + \dot{V}_{mc} - e_1 - g_{13}\delta_3^0 \\ -B_2 + \ddot{\Psi}_c - \dot{e}_2 - a_{21}e_2 - g_{23}\delta_3^0 \\ -B_3 + \ddot{\Theta}_c - \dot{e}_3 - a_{31}e_3 - g_{43}\delta_3^0 \\ -B_4 - a_{41}\dot{e}_4 - g_{53}\delta_3^0 \end{pmatrix}$$

$$\begin{pmatrix} \delta_1^{re} \\ \delta_2^{re} \\ \delta_4^{eq} \\ T_x^{re} \end{pmatrix} = (g_1 \quad g_2 \quad g_4 \quad g_5)^{-1} \begin{pmatrix} -(\sigma_1 + \eta_1) \cdot \text{sgn}(s_1) \\ -(\sigma_2 + \eta_2) \cdot \text{sgn}(s_2) \\ -(\sigma_4 + \eta_4) \cdot \text{sgn}(s_4) \\ -(\sigma_5 + \eta_5) \cdot \text{sgn}(s_5) \end{pmatrix}$$

其中 δ_3^0 為 δ_3 偵測器偵測到卡死的角度

d. 針對 δ_4 故障之狀況設計

$$\begin{pmatrix} \delta_1^{eq} \\ \delta_2^{eq} \\ \delta_3^{eq} \\ T_x^{eq} \end{pmatrix} = (g_1 \quad g_2 \quad g_3 \quad g_5)^{-1} \begin{pmatrix} -B_1 + \dot{V}_{mc} - e_1 - g_{14}\delta_4^0 \\ -B_2 + \ddot{\Psi}_c - \dot{e}_2 - a_{21}e_2 - g_{24}\delta_4^0 \\ -B_3 + \ddot{\Theta}_c - \dot{e}_3 - a_{31}e_3 - g_{34}\delta_4^0 \\ -B_4 - a_{41}\dot{e}_4 - g_{54}\delta_4^0 \end{pmatrix}$$

$$\begin{pmatrix} \delta_1^{re} \\ \delta_2^{re} \\ \delta_3^{eq} \\ T_x^{re} \end{pmatrix} = (g_1 \quad g_2 \quad g_3 \quad g_5)^{-1} \begin{pmatrix} -(\sigma_1 + \eta_1) \cdot \text{sgn}(s_1) \\ -(\sigma_2 + \eta_2) \cdot \text{sgn}(s_2) \\ -(\sigma_3 + \eta_3) \cdot \text{sgn}(s_3) \\ -(\sigma_5 + \eta_5) \cdot \text{sgn}(s_5) \end{pmatrix}$$

其中 δ_4^0 為 δ_4 偵測器偵測到卡死的角度

在設計完容錯控制器後，以下將對控制器參數及系統狀態之初始值進行設定：

- 在被動式與主動式容錯控制器的模擬中，我們假設致動器 δ_1 、 δ_2 、 δ_3 及 δ_4 於 $t=0.3$ 秒卡死。
- 選定之順滑表面中參數 $a_{21} = 50$, $a_{31} = 40$, $a_{41} = 18$
- 為了避免利用 sign 函數所產生的切換現象 (chattering)，因此將 sign 函數轉成 saturation 函數的形式，即 $\text{sgn}(s_i)$ 用 $\text{sat}(s_i/\varepsilon_i)$ 取代，其中邊界層厚度分別選取為 $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 1.0$
- 系統狀態初始值除飛彈速 $V_m(0) = 2662(ft/sec)$ 和攻角 $\alpha(0) = 10(deg)$ 、偏離角 $\Psi(0) = 11.45(deg)$ 及傾斜角 $\Theta(0) = 10(deg)$ 外，其餘皆設為零

- 飛彈尾翼擺動角之物理限制為 $-30^\circ \leq \delta_i \leq 30^\circ$ $i = 1, 2, 3, 4$
- 設定飛彈欲追蹤之軌跡 $r_d(t)$ ，其在慣性座標的分量為：

$$r_1(t) = 2620.5t + 8000$$

$$r_2(t) = 462t + 2000$$

$$r_3(t) = \begin{cases} -5(t-7)^2 + 10245 & \text{if } 0 \leq t \leq 10 \\ 5(t-13)^2 + 10155 & \text{if } 10 \leq t \leq 20 \end{cases}$$

而飛彈之初始位置為(8500,2200,9800)

- 在(3.20)式位置補償常數(h_1, h_2, h_3)的選取則為

$$h_1 = 0.0005$$

$$h_2 = -0.0005$$

$$h_3 = \begin{cases} -0.14 & \text{if } \dot{r}_d \cdot (r_d - x) < 0 \\ 0.14 & \text{if } \dot{r}_d \cdot (r_d - x) > 0 \end{cases}$$

6.4 模擬結果與討論

飛彈系統反飛彈導引設計模擬結果列於圖6.1-6.60。模擬圖形主要包括針對致動器完全正常或只有一致動器故障時運用非容錯控制、被動式容錯控制及主動式容錯控制配合錯誤診斷機制所模擬的結果，模擬結果詳述如下：

- (a) 圖6.1-圖6.4展示針對全部致動器皆正常時設計非容錯控制器的模擬圖，由這些模擬圖可以明顯地看出所有尾翼皆正常操作的情況下，導引飛彈在三度空間的軌跡皆能在不耗費很大能量下追蹤到參考軌跡。
- (b) 圖6.5-圖6.20顯示利用非容錯控制器針對 δ_1 、 δ_2 、 δ_3 和 δ_4 其中任一致動器故障時之模擬結果，在這些模擬中，我們假設對應之尾翼都在0.3秒時產生故障(卡死)。由模擬圖形可看出系統狀態呈現發散現象。這些模擬圖形說明了使用非容錯控制器設計時，飛彈系統是不允許有任何致動器故障，這也更突顯了容錯控制的重要性。

- (c) 接下來為利用被動式容錯控制針對 δ_1 故障設計，但致動器全部皆正常的情況下，如圖6.21-圖6.24，其暫態響應時間較非容錯控制器短，而致動器 δ_1 未發生故障，並未影響軌跡追蹤的性能。
- (d) 接續以上的結果，在圖6.25-圖6.36我們模擬針對 δ_2 、 δ_3 和 δ_4 故障的被動式容錯控制，在所有致動器皆正常的情況下，與非容錯控制器的比較下，其追蹤性能並未有太大的改變，在此說明了被動式容錯控制器的可行性。
- (e) 圖6.49-圖6.72為針對有一組致動器發生故障時採用被動式容錯控制器、主動式容錯控制器與非容錯控制器之模擬結果，在模擬中我們假設故障發生時間在 $t=0.3$ 秒，由圖形可以看出無論是哪一個致動器故障，被動式容錯控制器與主動式容錯控制器皆能達到軌跡追蹤的目的。
- (f) 由圖6.21-6.24及圖6.49-6.52為針對 δ_1 故障所設計的被動式容錯控制器，從模擬結果可以看出無論是 δ_1 正常或故障時，系統皆能達到良好的性能，完成軌跡追蹤的任務，並不因為致動器的故障而發生系統發散的現象，這也證實了我們所設計的被動式控制器確實是可行的。
- (g) 圖6.37-圖6.48提供被動式容錯控制器對 δ_1 發生故障所設計，但 δ_i 發生故障($i \neq 1$)之模擬圖，我們可以明顯看出，飛彈系統無法達成軌跡追蹤的目的，在此說明被動式容錯控制器雖可以針對較有可能發生錯誤的元件來設計，但若故障不在預期之內時，軌跡追蹤的任務無法保證達成。
- (h) 圖6.49-圖6.54中，正常情況下當致動器 δ_1 在 $t=0.3$ 秒卡死時，我們設residual signal超過0.005時，視為錯誤發生，所設計的觀察器偵測到residual signal 1有訊號產生，而residual signal 2、residual signal 3皆未超過0.005，這表示觀察器能正確地偵測到致動器 δ_1 錯誤，且在0.65秒警報器的訊號可以傳回系統，將即時切換控制律，達到系統穩定的效果。圖6.59、圖6.65及圖6.71分別為 δ_2 、 δ_3

和 δ_4 故障時，觀察器皆能成功地偵測到哪一個致動器錯誤，達到切換控制律的功能。

- (i) 最後，在圖 6.49-圖 6.72 我們比較被動式、主動式控制器之性能，就能量的觀點上，被動式所消耗的能量為最大，在飛彈的推力被動式以約為主動式的兩倍進行，而在致動器的表現也可以看出，被動式也較大於主動式，在圖 6.54、圖 6.60、圖 6.66、及圖 6.72 我們放大在 0 ~ 2 秒，觀察致動器的變化，可以明顯看出被動式致動器的擺動角度皆較主動式大。另外因為被動式容錯控制器為針對某一特定元件故障所設計，而當非預期的元件發生故障時，將使得軌跡追蹤任務失敗，而主動式容錯控制器結合錯誤偵測與診斷機制，可適當地轉換控制律，達到系統的要求。因此，結合上述結果可知主動式控制器的性能比被動式控制器佳，這也和我們所提出的理論相符。

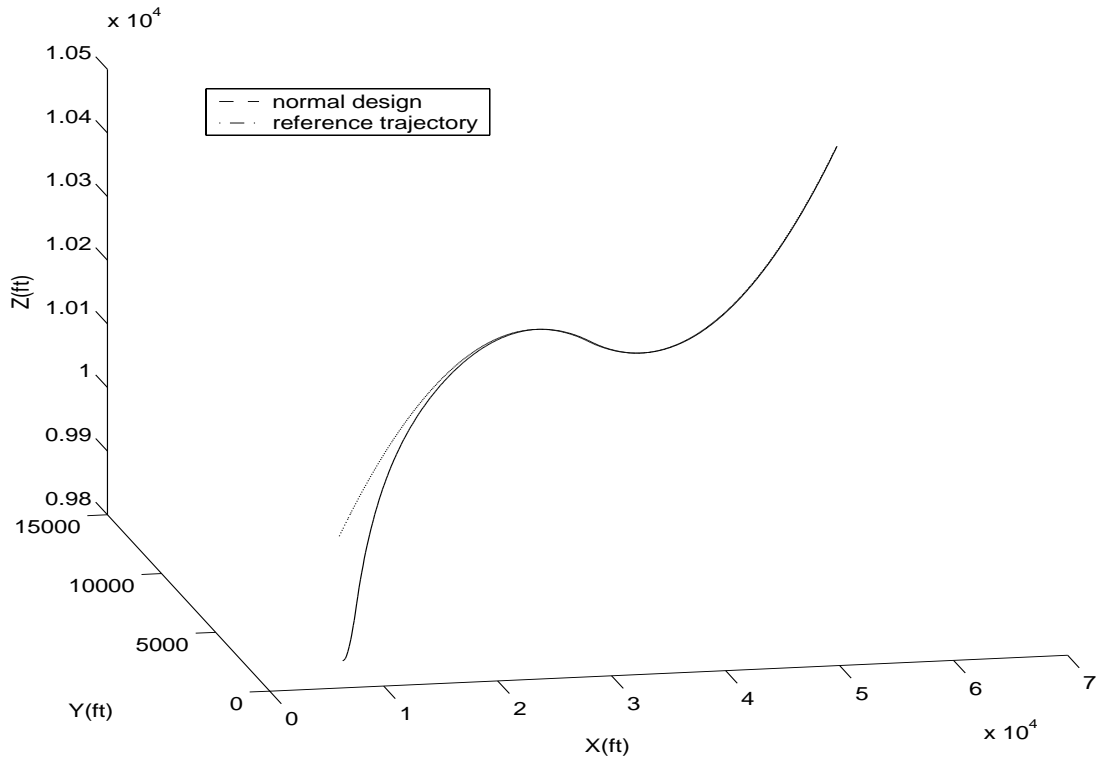


圖 6.1 非容錯控制設計，在所有致動器正常之下其追蹤性能之表現

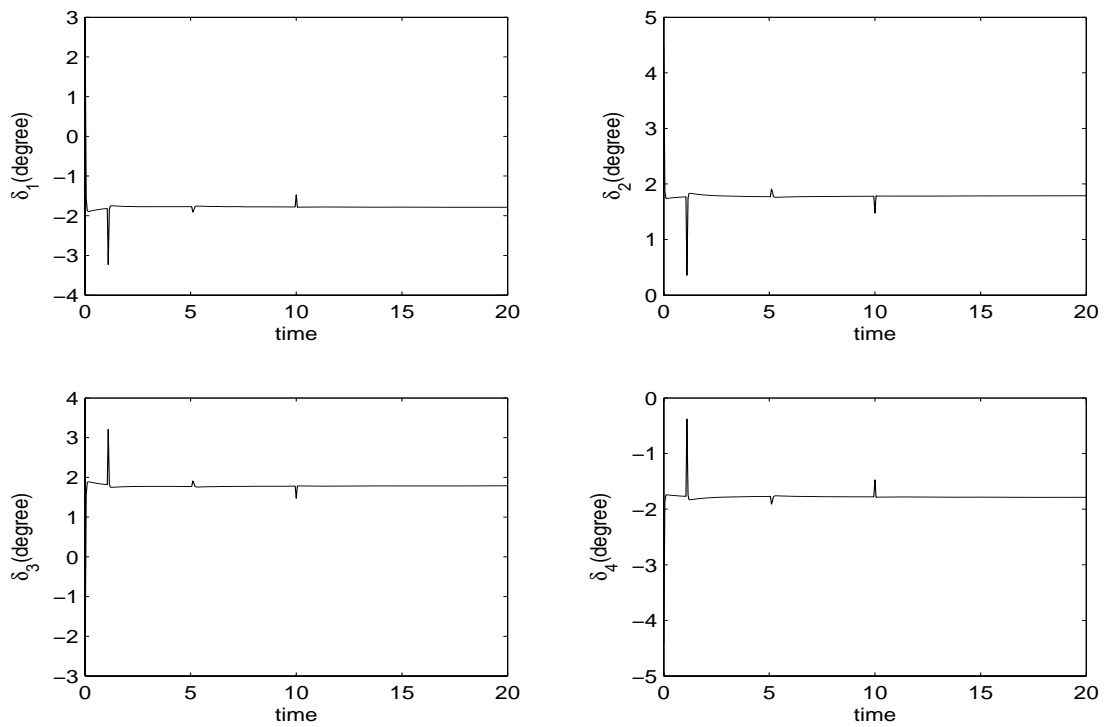


圖 6.2 非容錯控制設計，在所有致動器正常之下其四個致動器之狀態

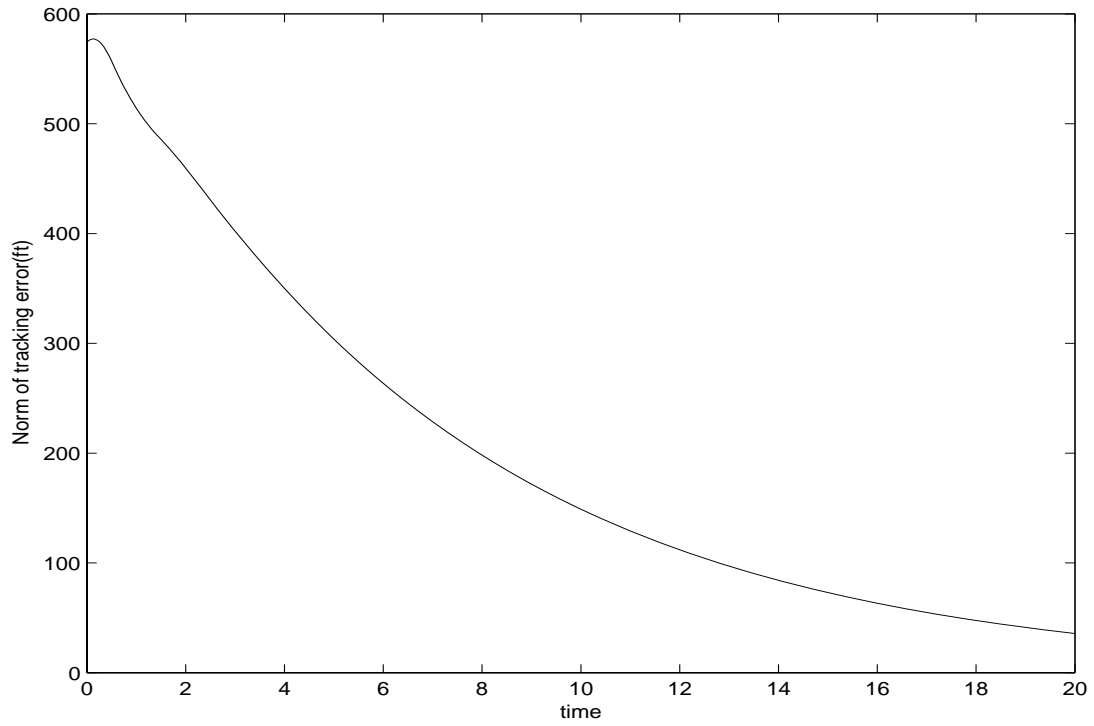


圖 6.3 非容錯控制設計，在所有致動器正常之下其飛彈之追蹤誤差

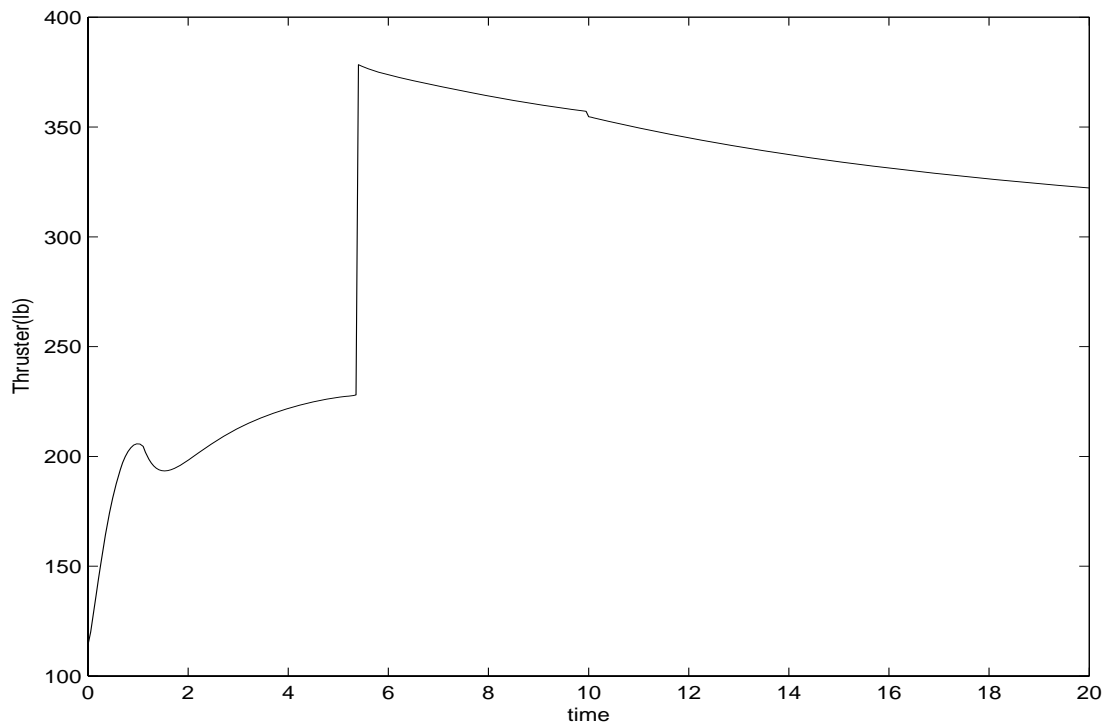


圖 6.4 非容錯控制設計，在所有致動器正常之下其飛彈之推力

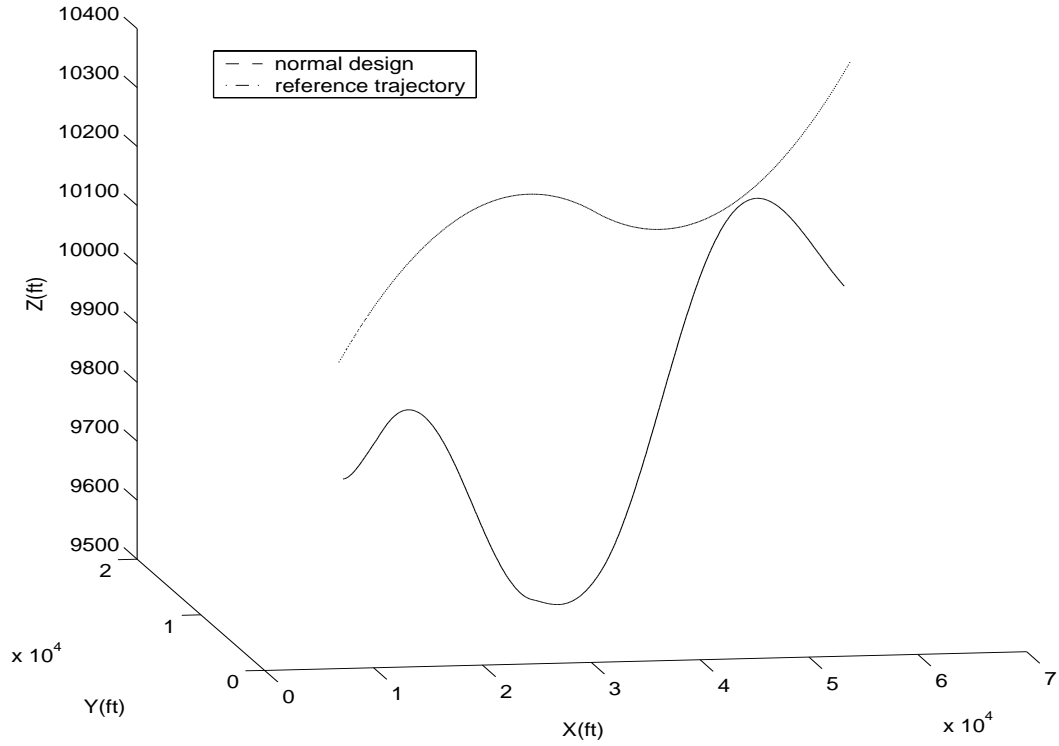


圖 6.5 非容錯控制設計， δ_1 於 $t=0.3$ 秒卡死其追蹤性能之表現

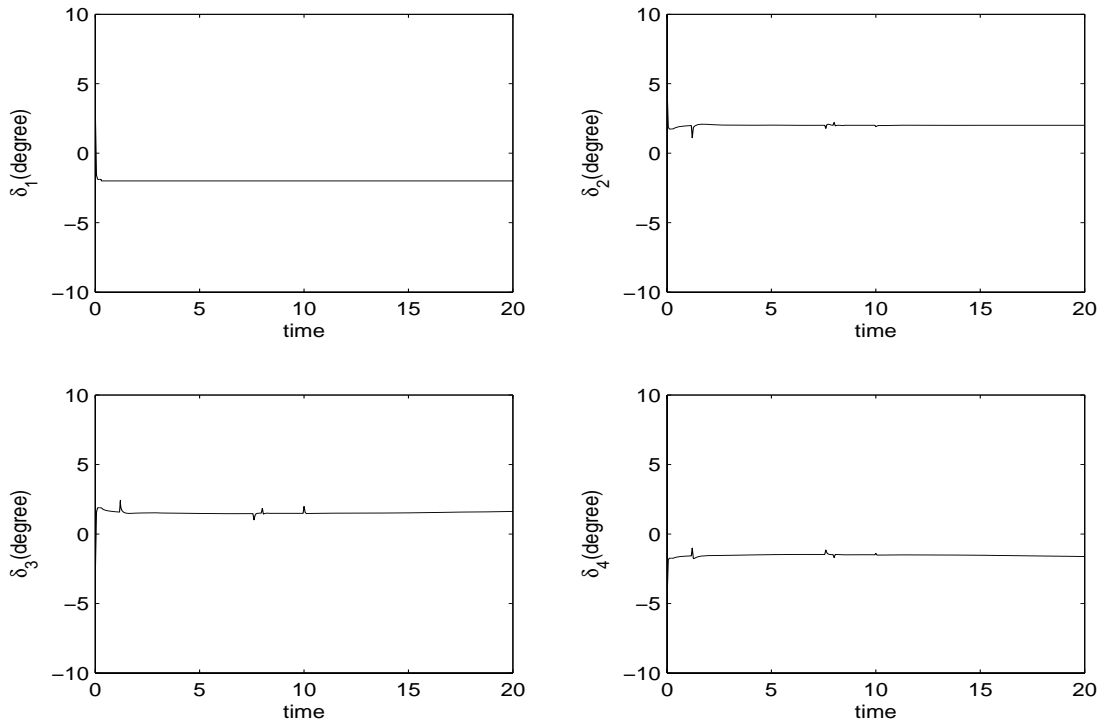


圖 6.6 非容錯控制設計， δ_1 於 $t=0.3$ 秒卡死其致動器之狀態

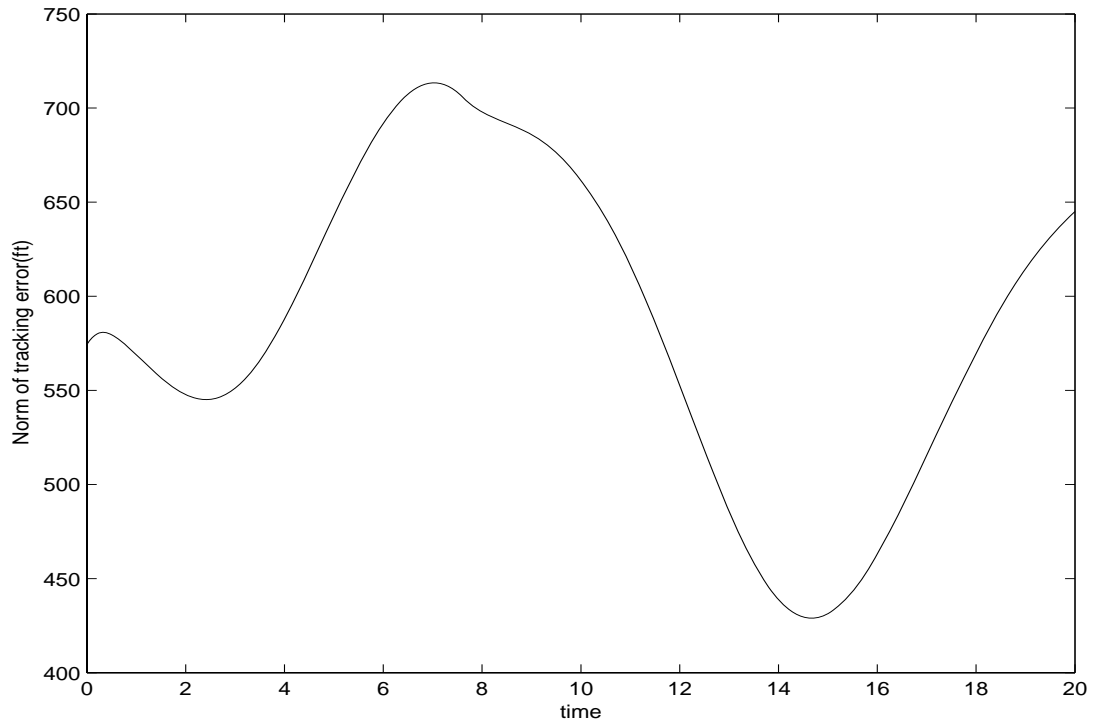


圖 6.7 非容錯控制設計， δ_1 於 $t=0.3$ 秒卡死其飛彈之追蹤誤差

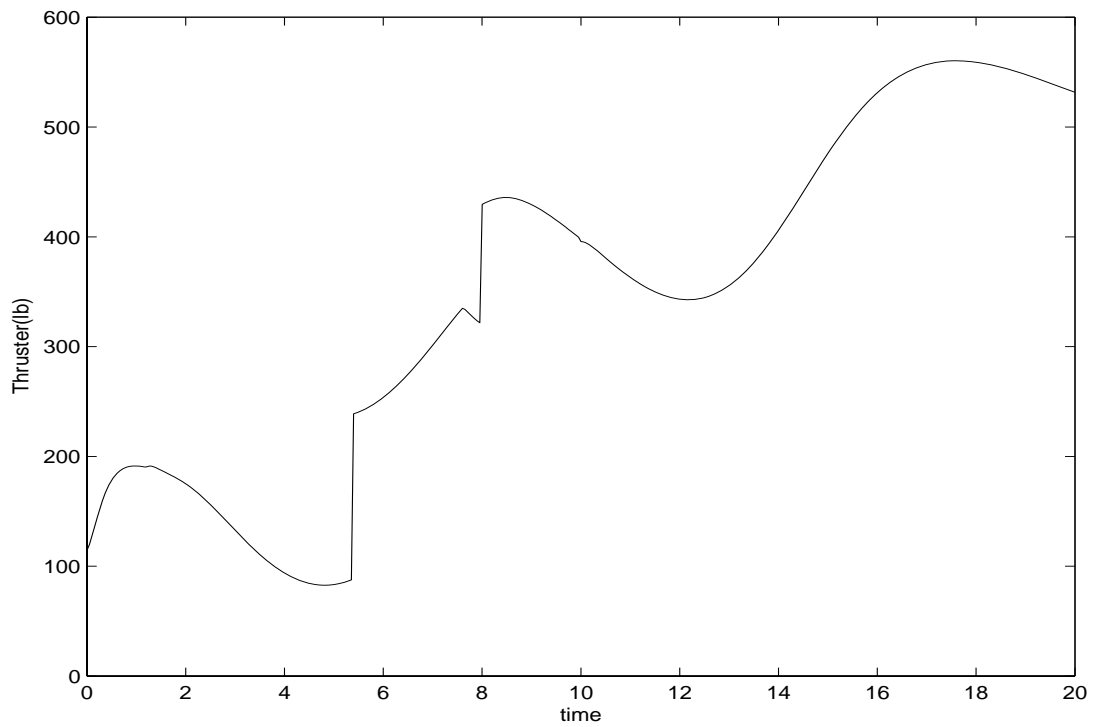


圖 6.8 非容錯控制設計， δ_1 於 $t=0.3$ 秒卡死其飛彈之推力

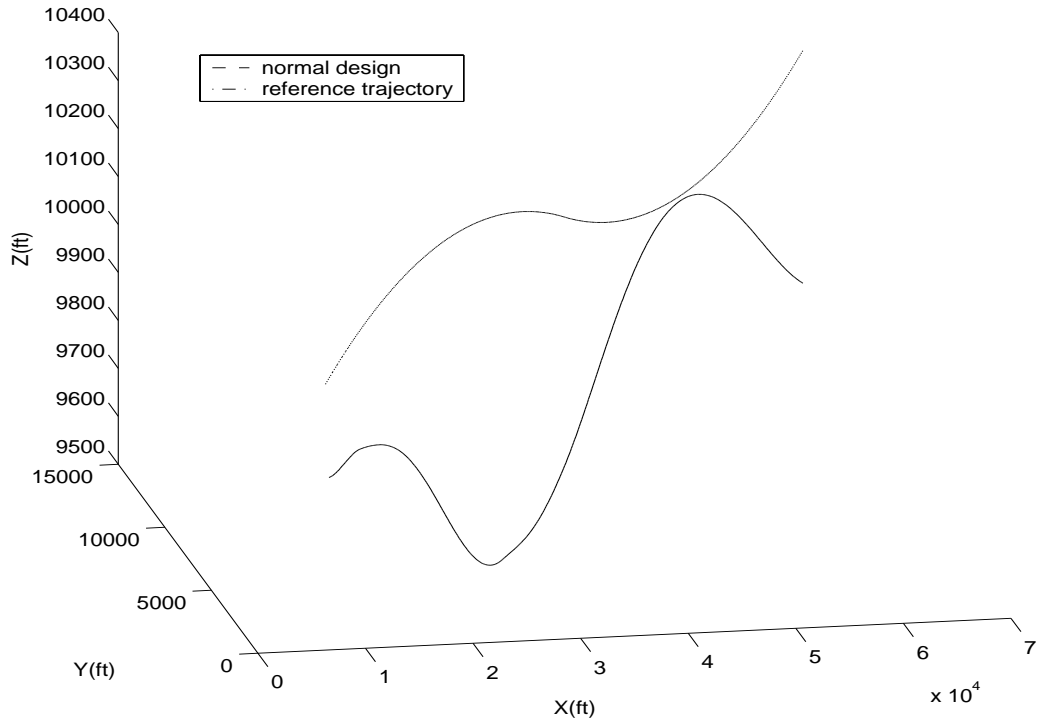


圖 6.9 非容錯控制設計， δ_2 於 $t=0.3$ 秒卡死其追蹤性能之表現

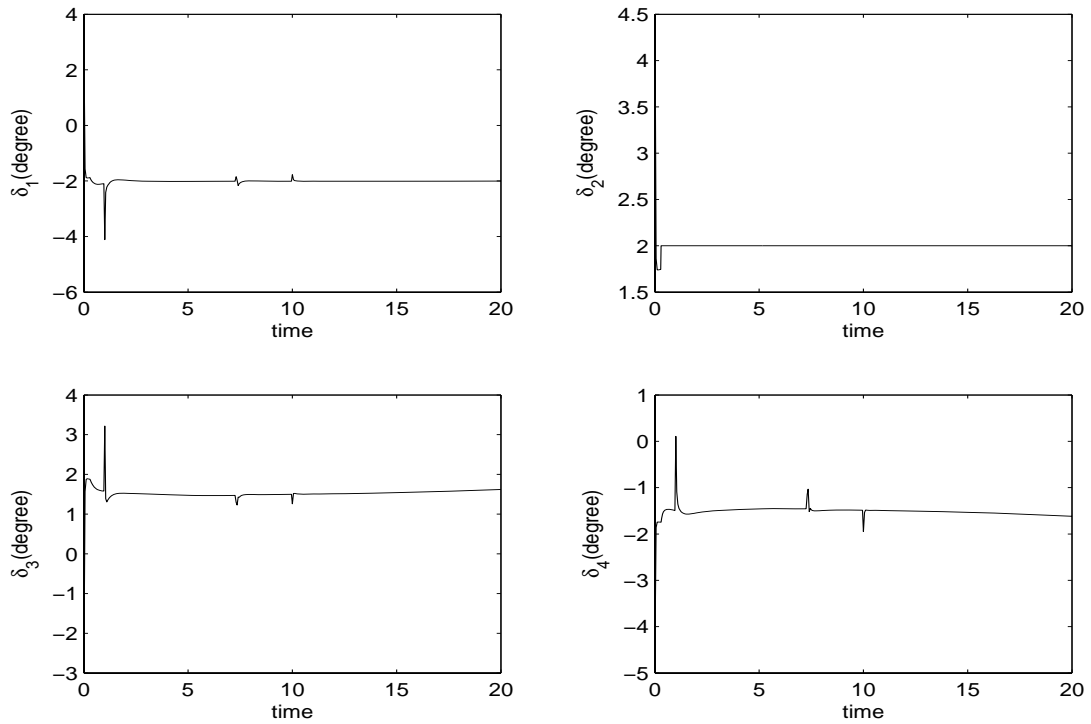


圖 6.10 非容錯控制設計， δ_2 於 $t=0.3$ 秒卡死其致動器之狀態

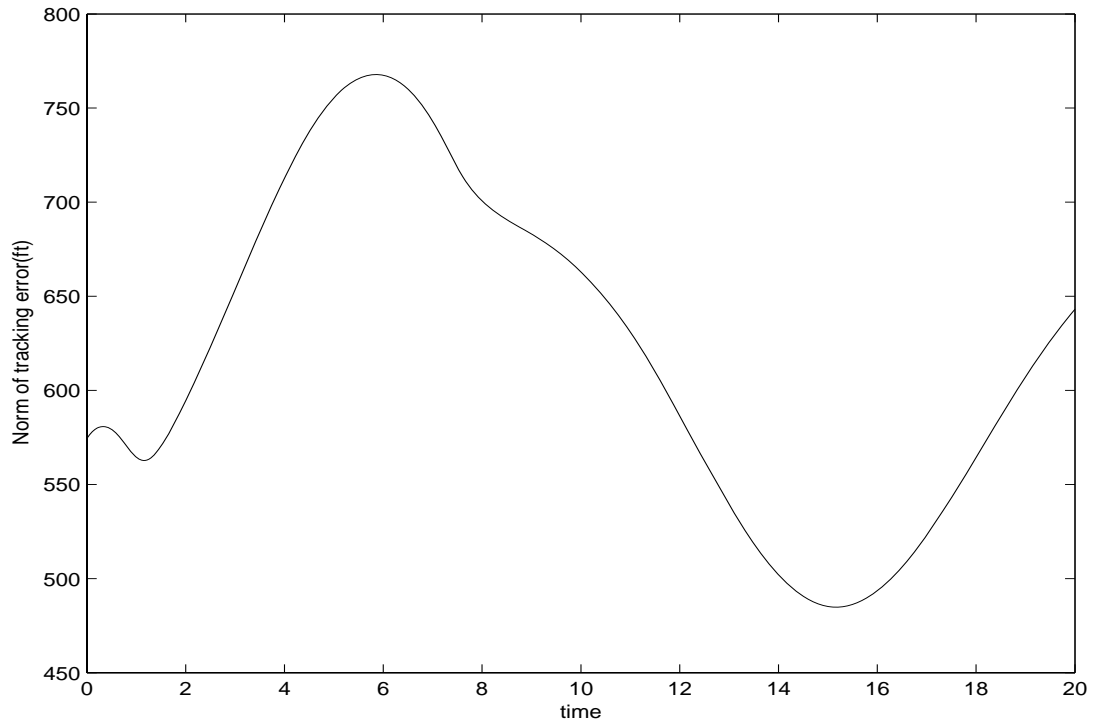


圖 6.11 非容錯控制設計， δ_2 於 $t=0.3$ 秒卡死其飛彈之追蹤誤差

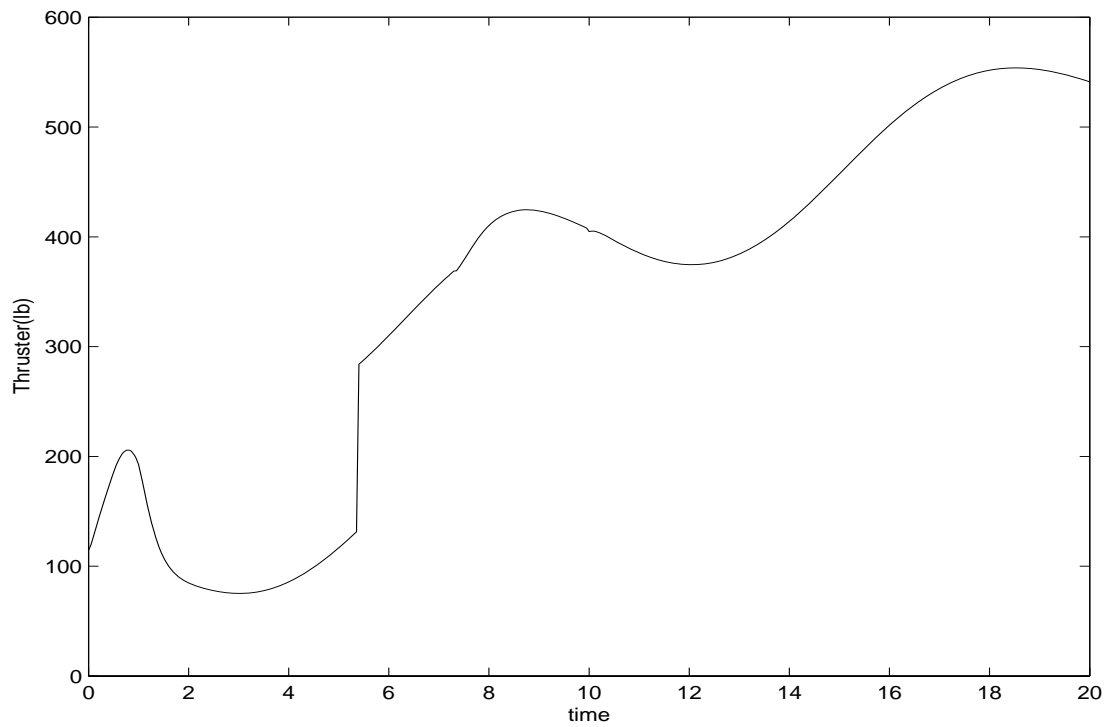


圖 6.12 非容錯控制設計， δ_2 於 $t=0.3$ 秒卡死其飛彈之推力

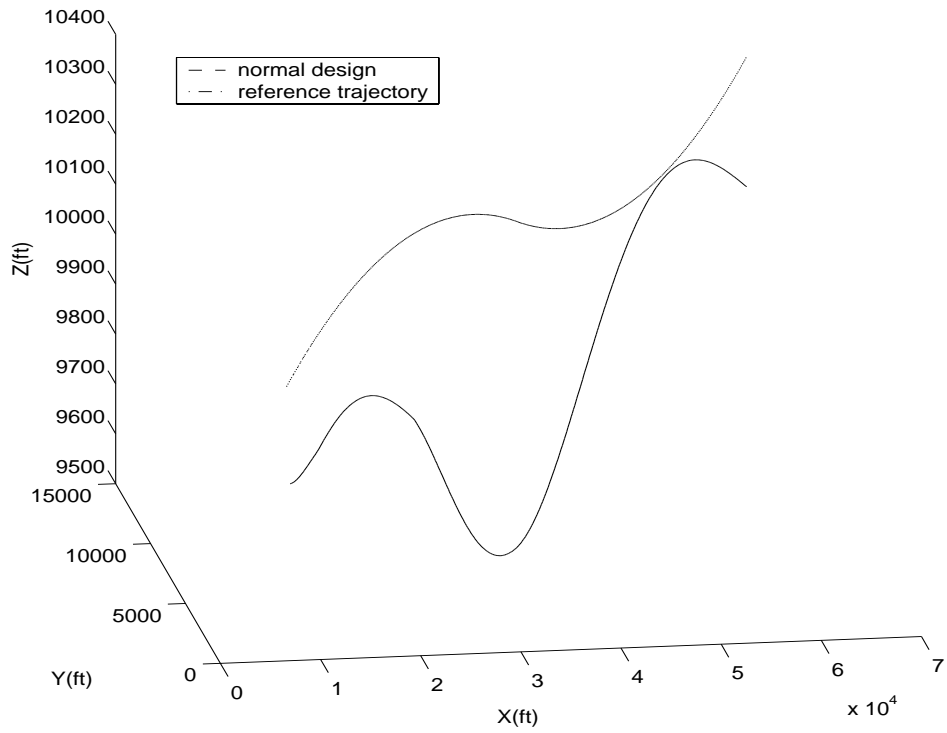


圖 6.13 非容錯控制設計， δ_3 於 $t=0.3$ 秒卡死其追蹤性能之表現

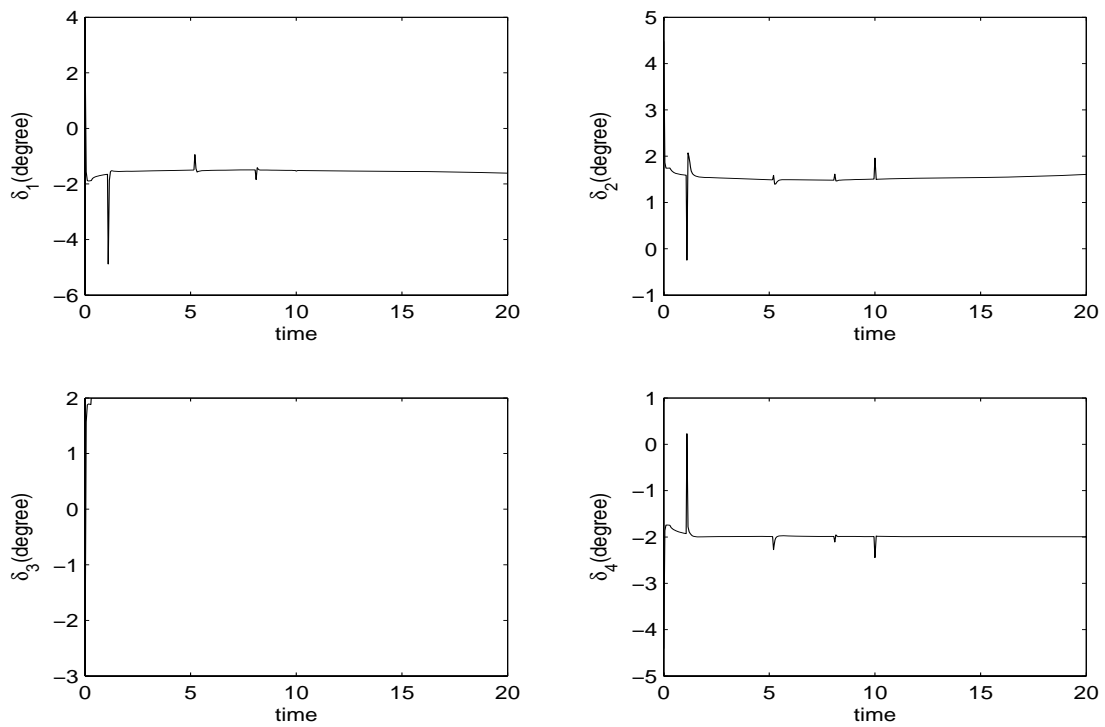


圖 6.14 非容錯控制設計， δ_3 於 $t=0.3$ 秒卡死其致動器之狀態

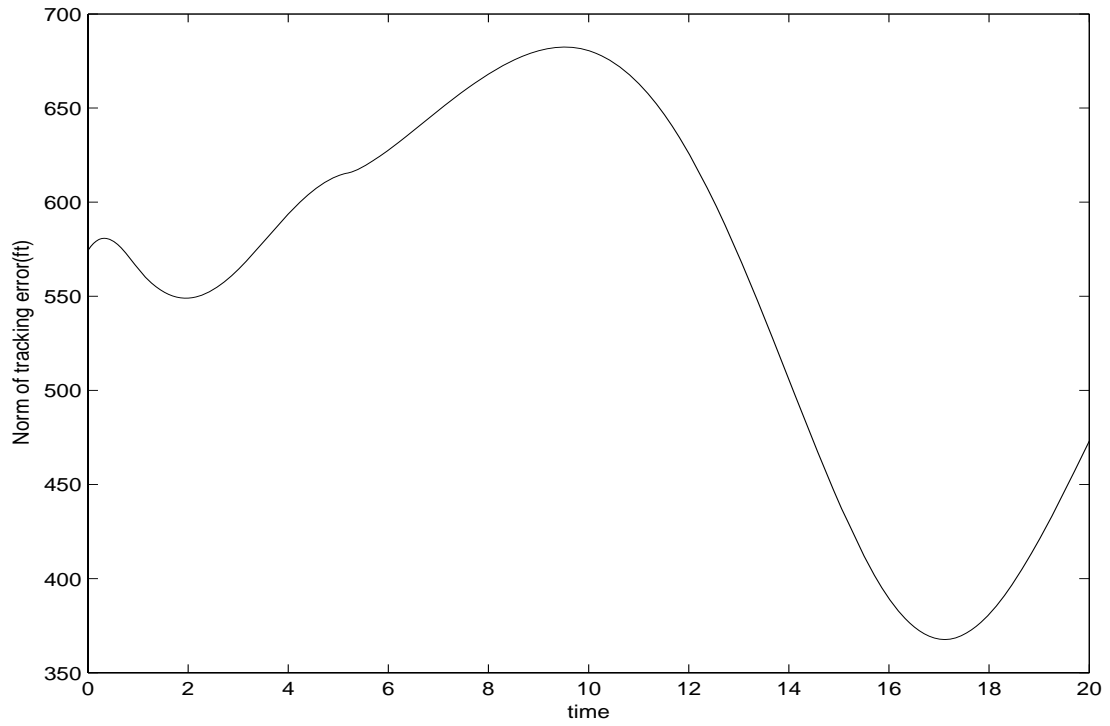


圖 6.15 非容錯控制設計， δ_3 於 $t=0.3$ 秒卡死其飛彈之追蹤誤差

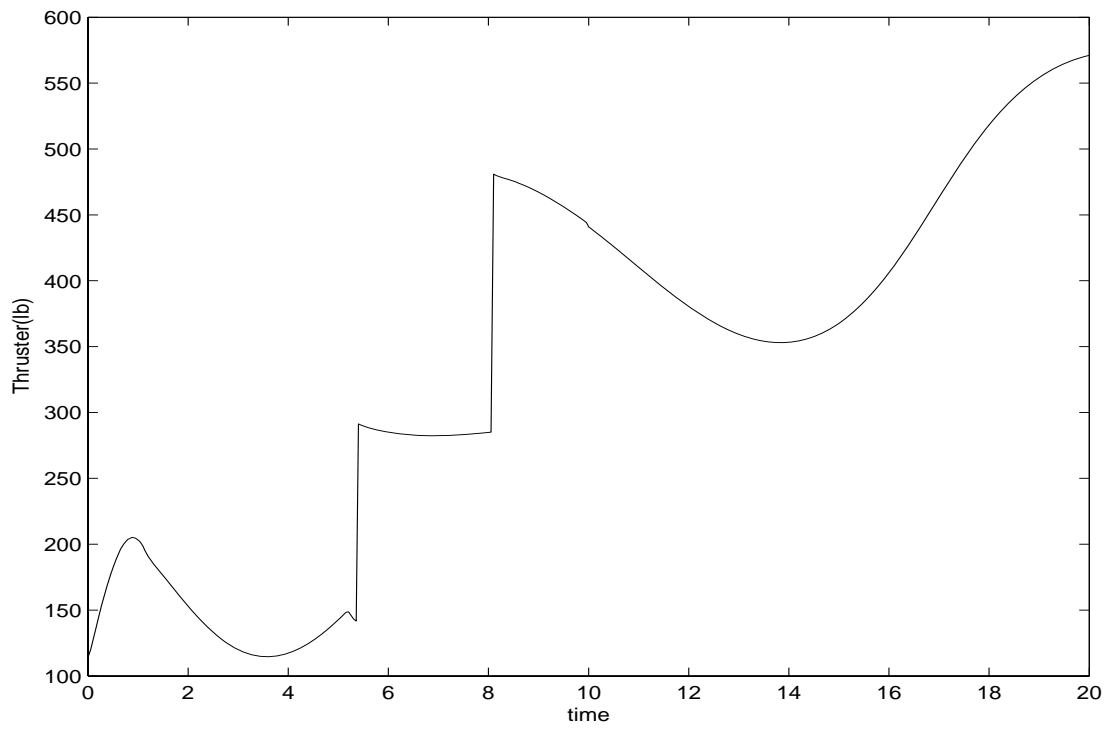


圖 6.16 非容錯控制設計， δ_3 於 $t=0.3$ 秒卡死其飛彈之推力

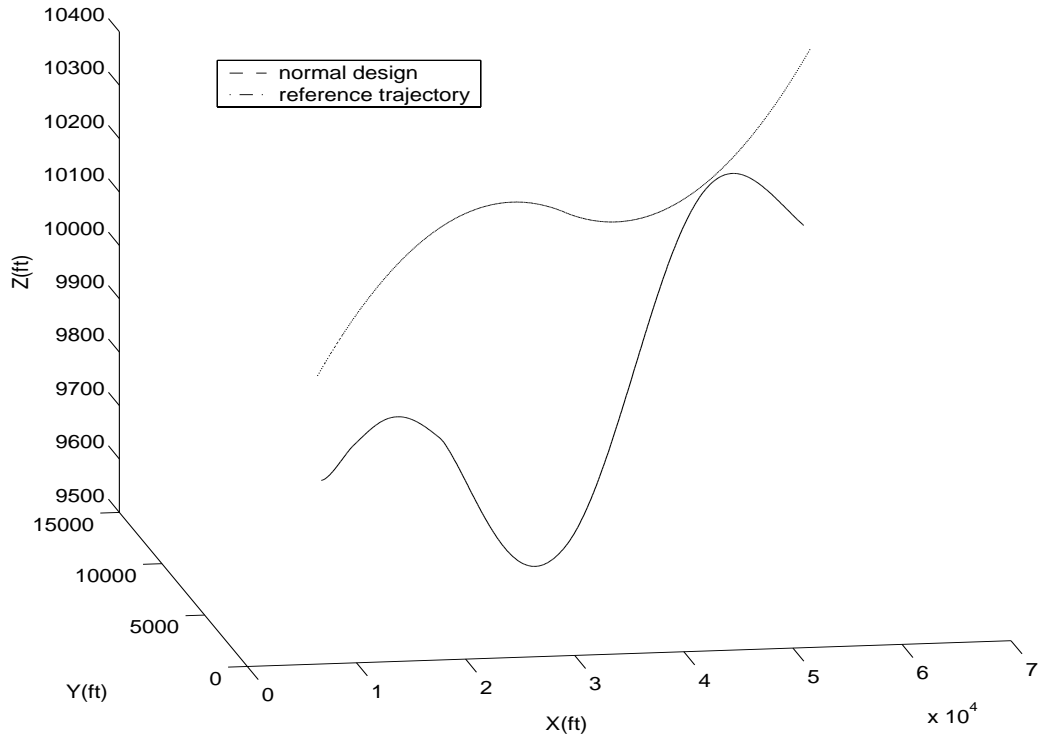


圖 6.17 非容錯控制設計， δ_4 於 $t=0.3$ 秒卡死其追蹤性能之表現

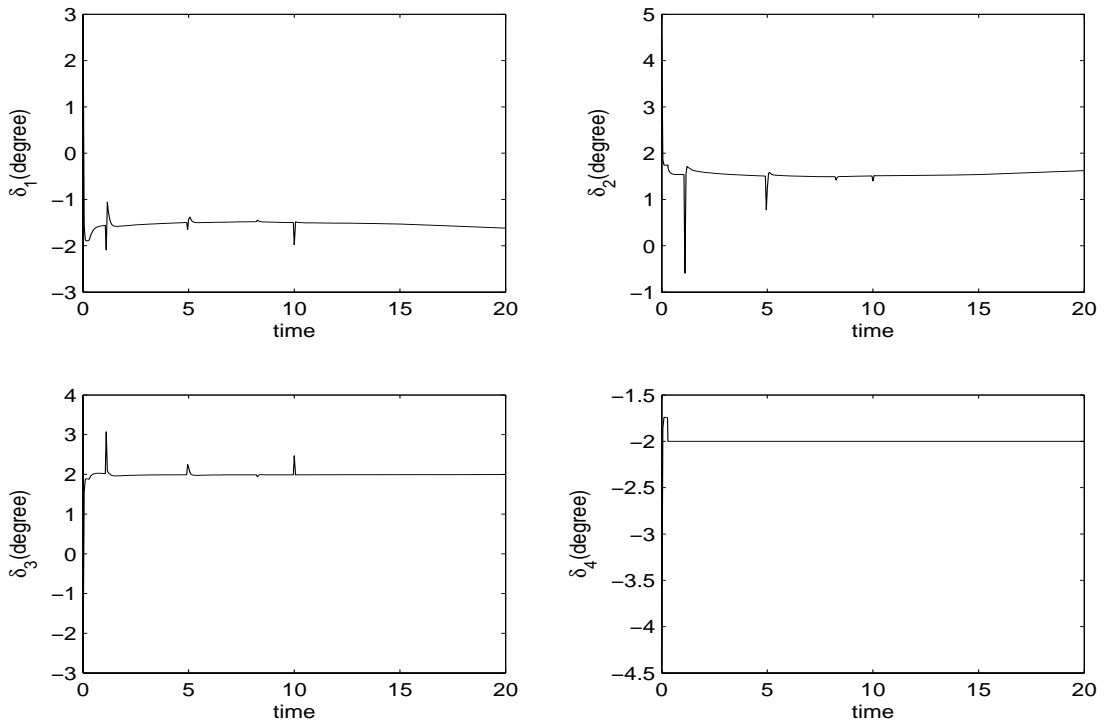


圖 6.18 非容錯控制設計， δ_4 於 $t=0.3$ 秒卡死其致動器之狀態

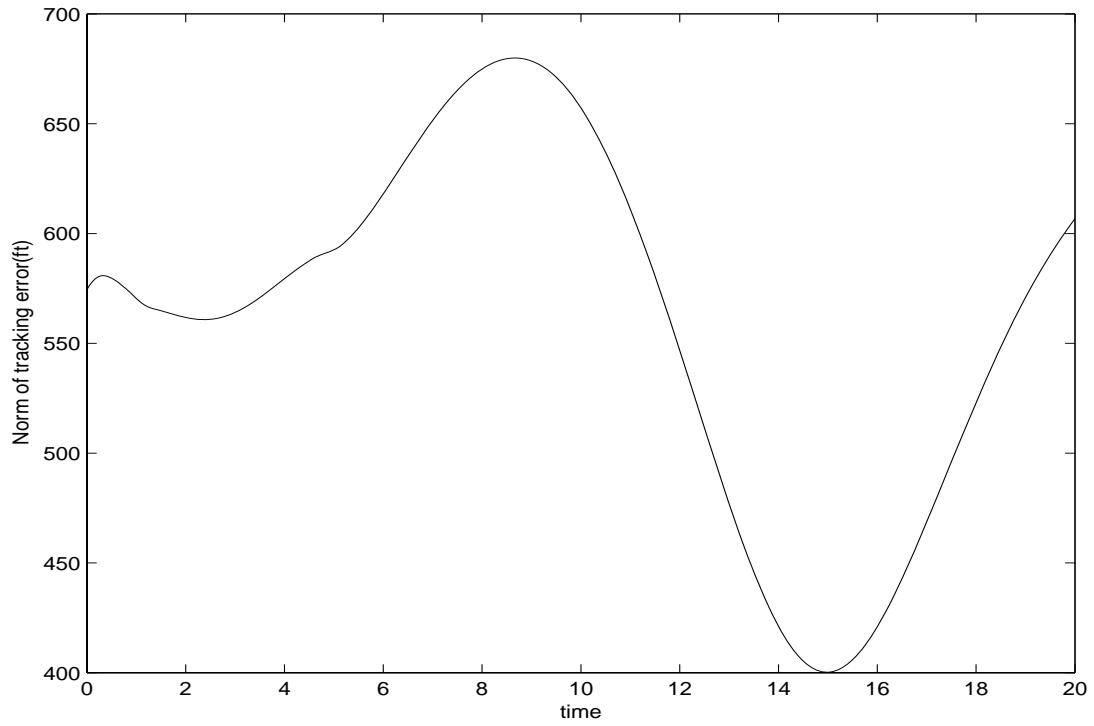


圖 6.19 非容錯控制設計， δ_4 於 $t=0.3$ 秒卡死其飛彈之追蹤誤差

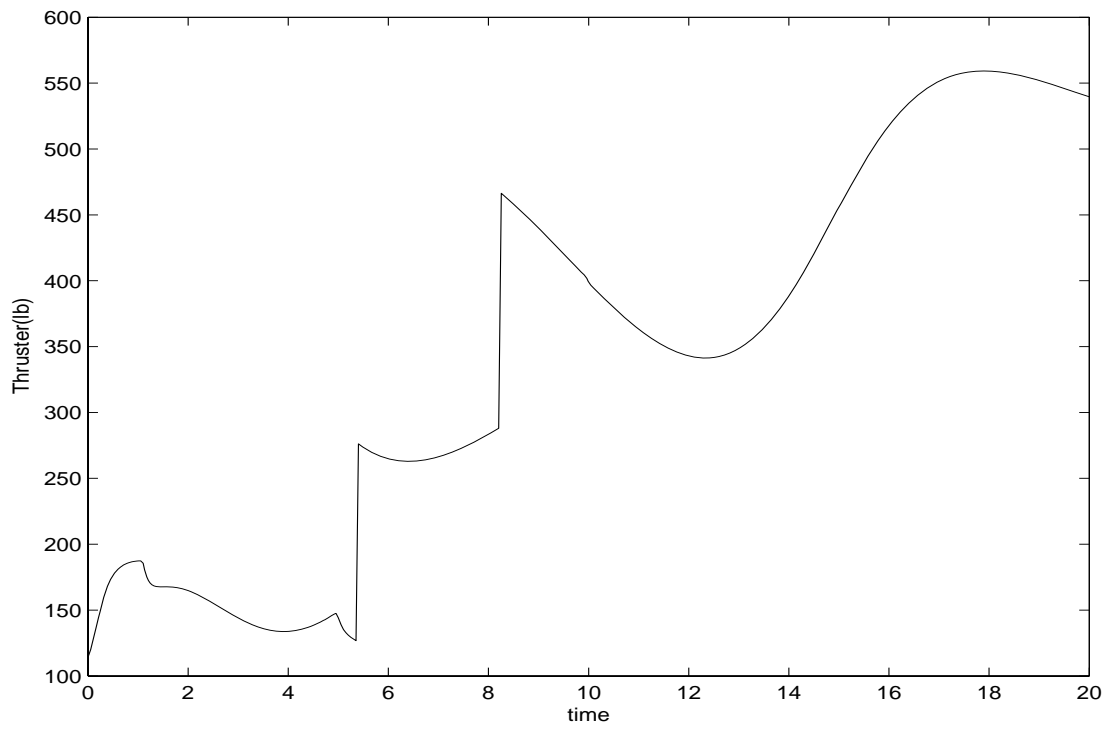


圖 6.20 非容錯控制設計， δ_4 於 $t=0.3$ 秒卡死其飛彈之推力

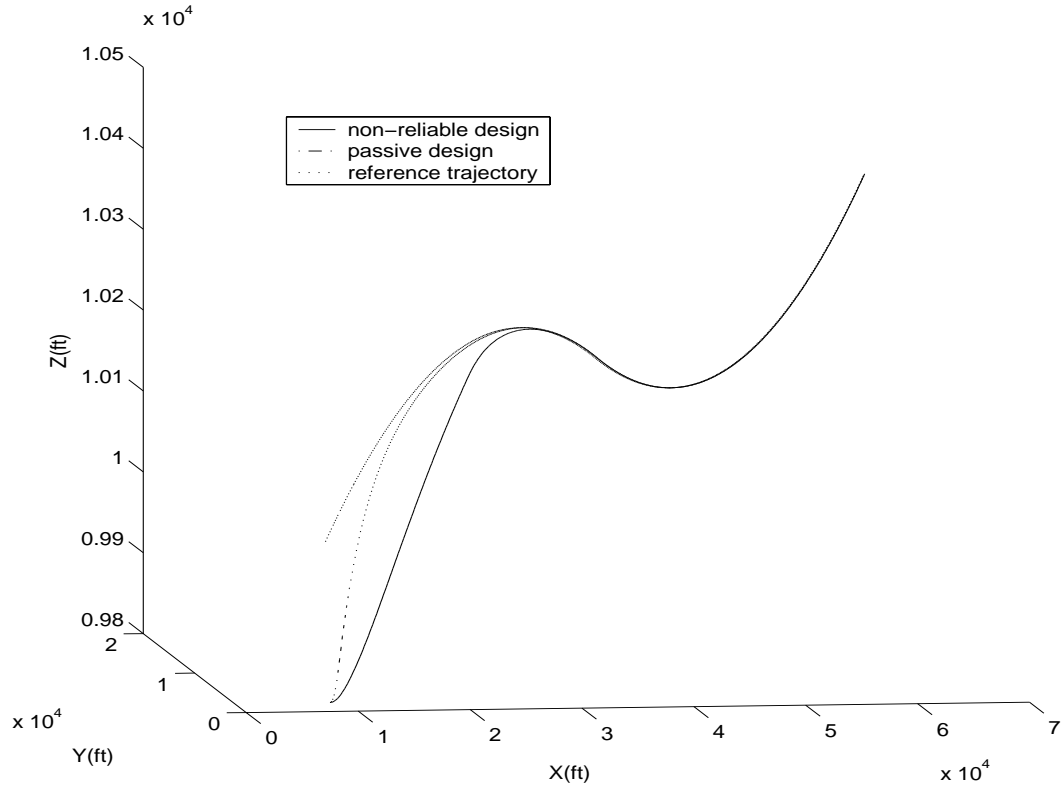


圖 6.21 所有致動器皆正常之下，針對 δ_1 故障所設計之被動式控制器與非容錯控制器其追蹤性能的表現

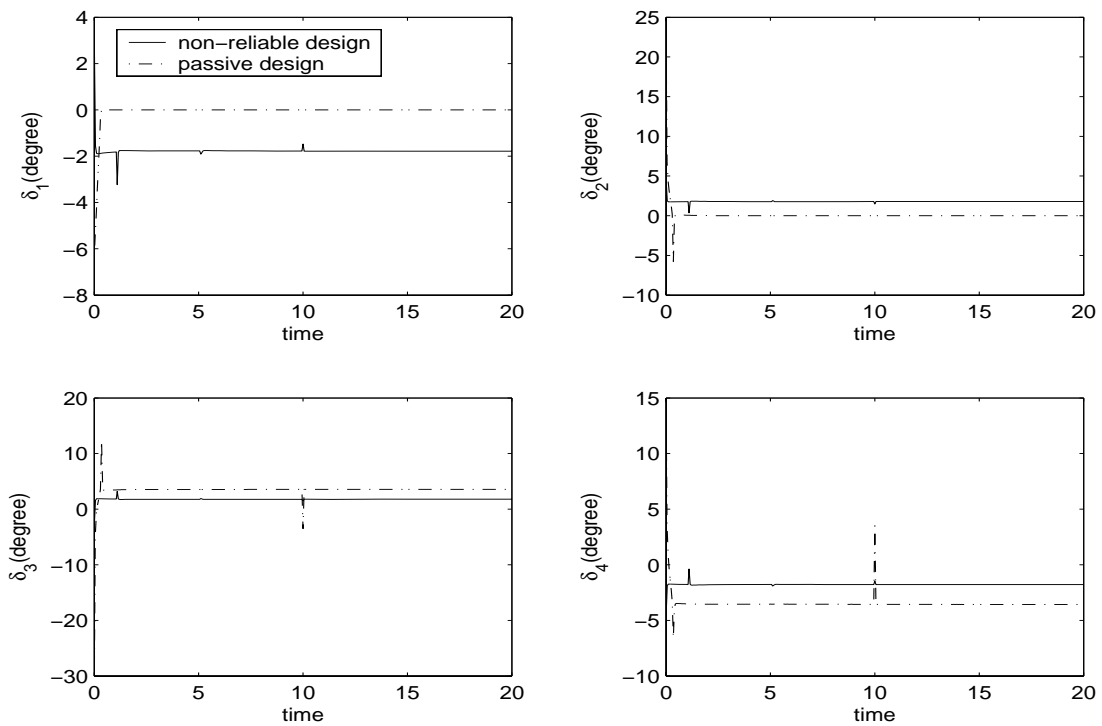


圖 6.22 所有致動器皆正常之下，針對 δ_1 故障所設計之被動式控制器與非容錯控制器其致動器的狀態

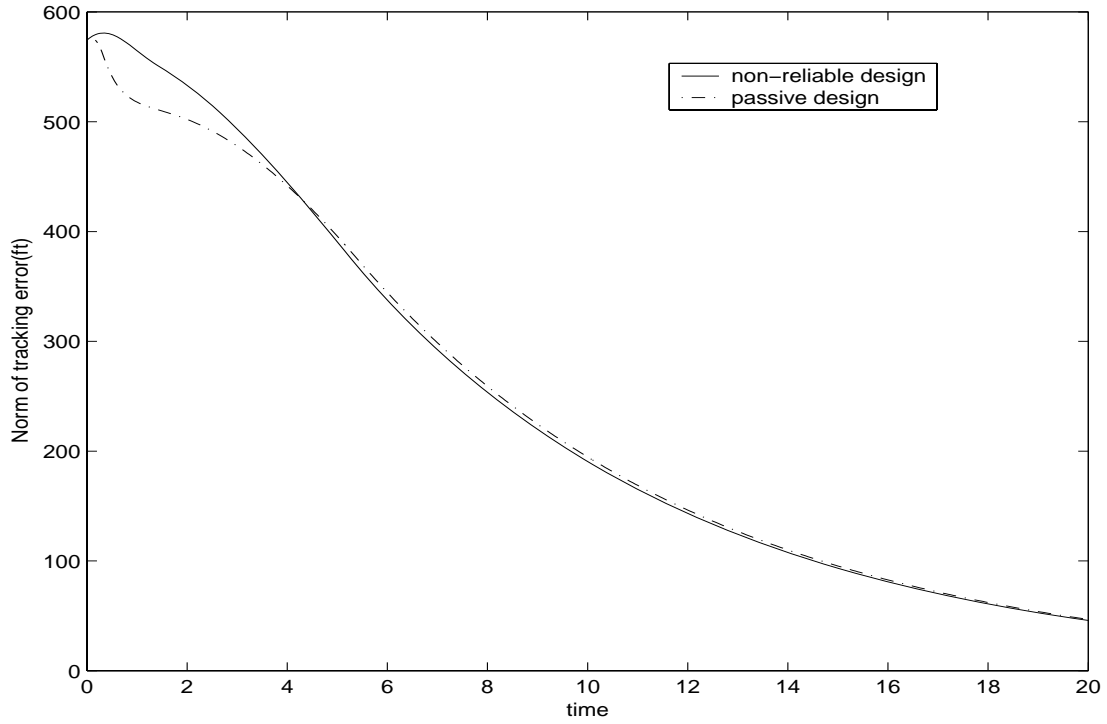


圖 6.23 所有致動器皆正常之下，針對 δ_1 故障所設計之被動式控制器與非容錯控制器其飛彈之追蹤誤差

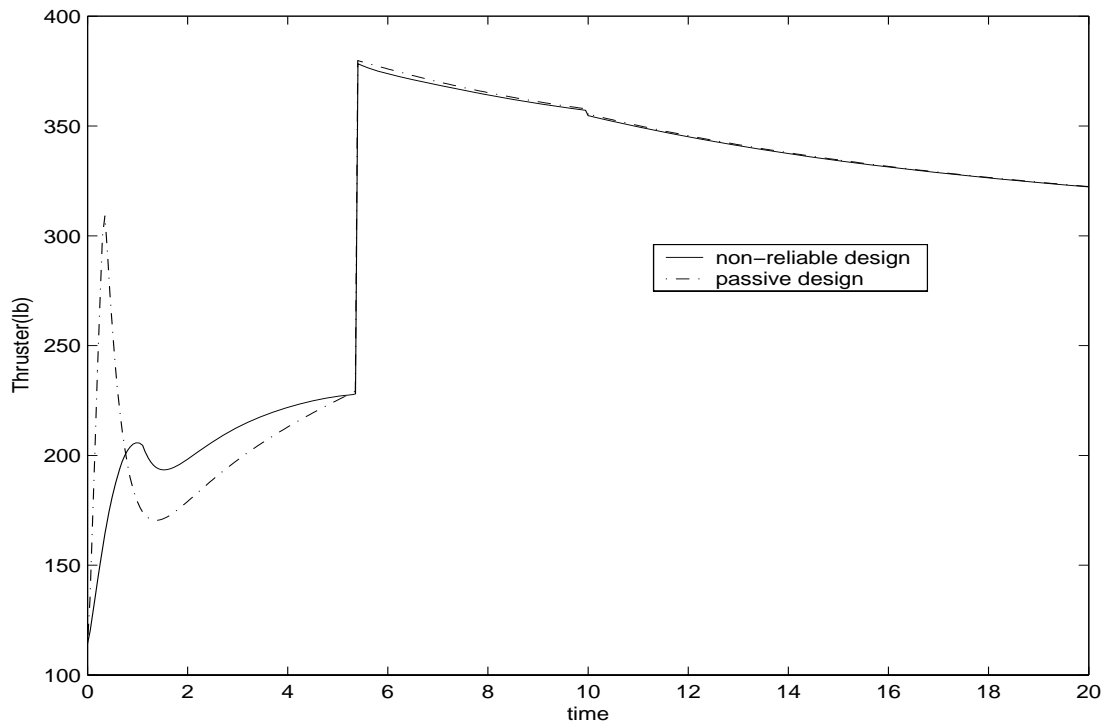


圖 6.24 所有致動器皆正常之下，針對 δ_1 故障所設計之被動式控制器與非容錯控制器其飛彈之推力

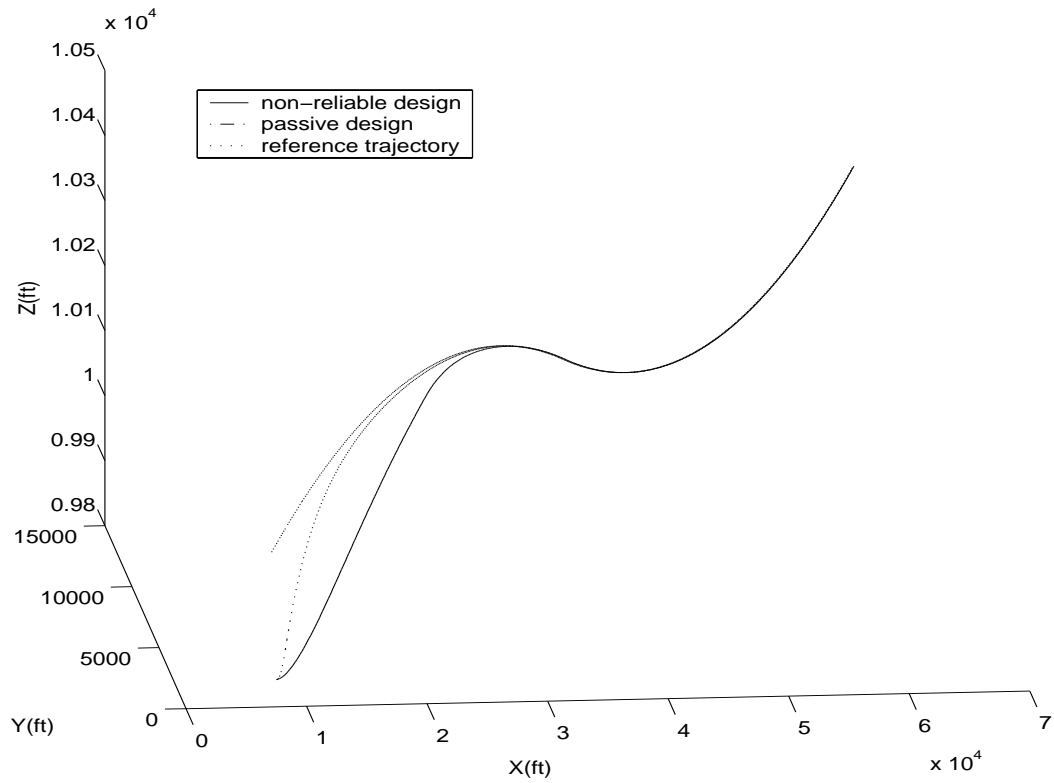


圖 6.25 所有致動器皆正常之下，針對 δ_2 故障所設計之被動式控制器與非容錯控制器其追蹤性能之表現

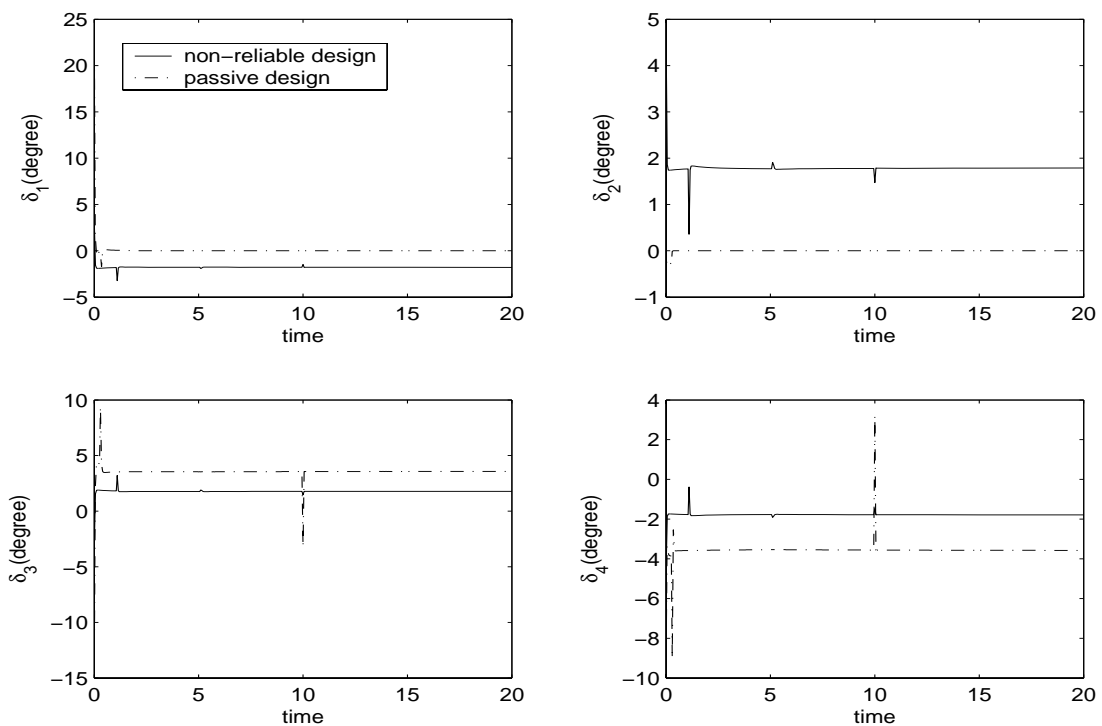


圖 6.26 所有致動器皆正常之下，針對 δ_2 故障所設計之被動式控制器與非容錯控制器其致動器之狀態

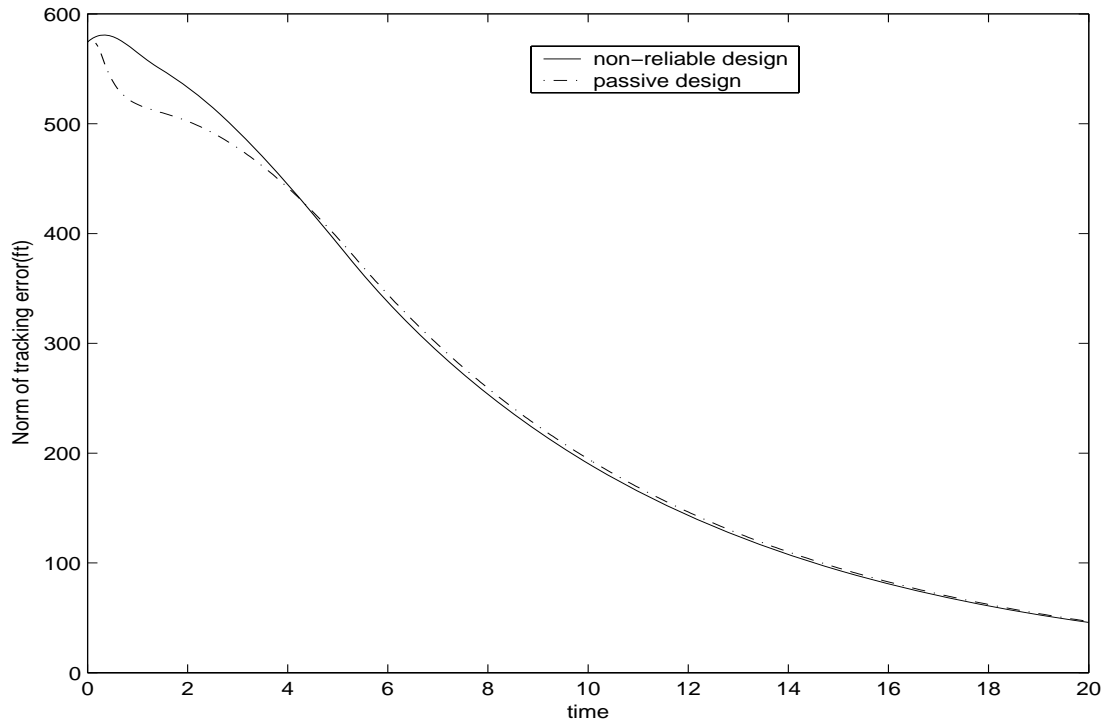


圖 6.27 所有致動器皆正常之下，針對 δ_2 故障所設計之被動式控制器與非容錯控制器其飛彈追蹤之誤差

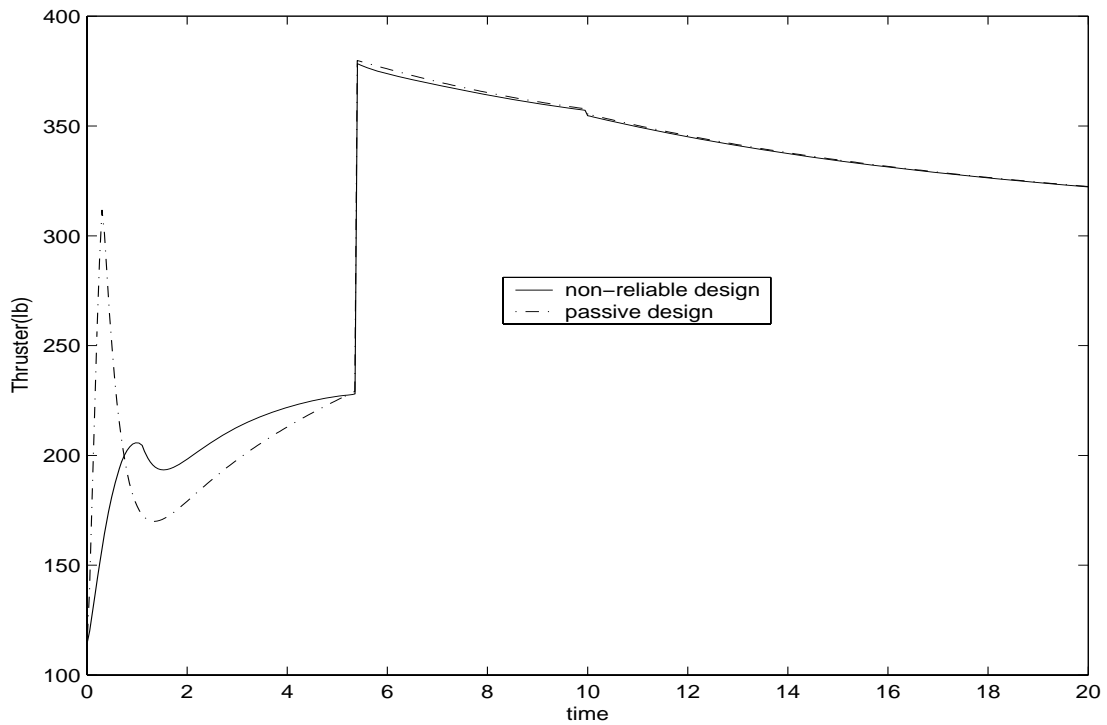


圖 6.28 所有致動器皆正常之下，針對 δ_2 故障所設計之被動式控制器與非容錯控制器其飛彈之推力

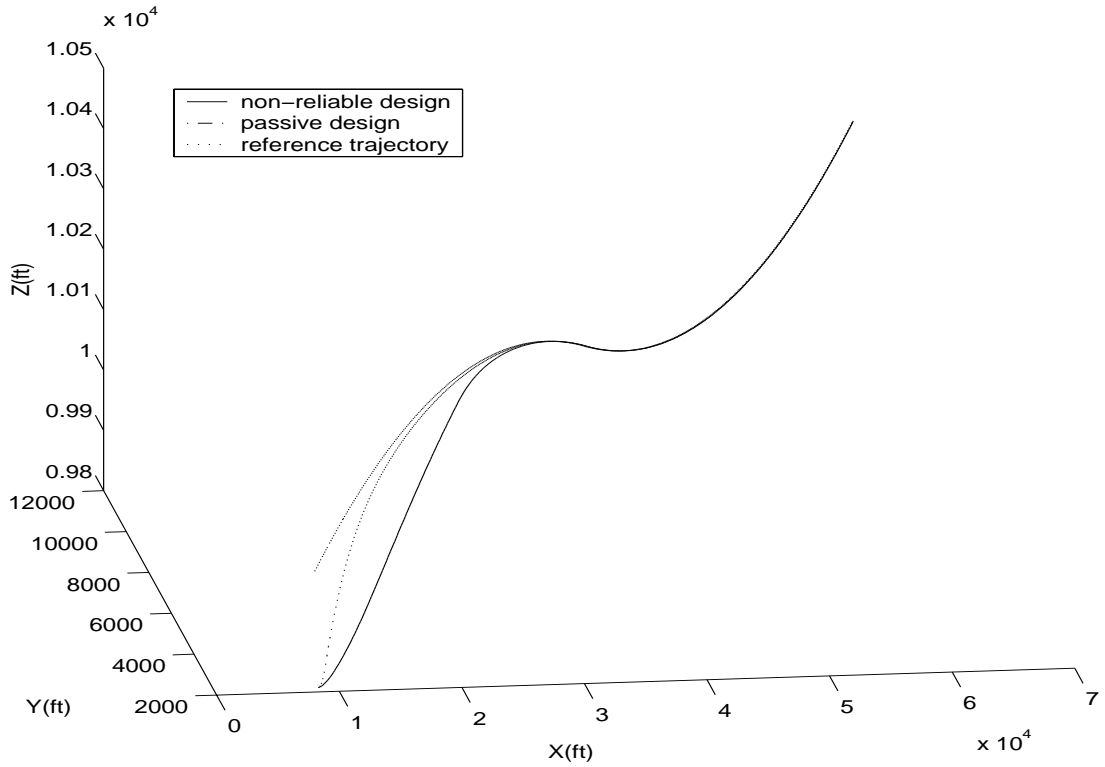


圖 6.29 所有致動器皆正常之下，針對 δ_3 故障所設計之被動式控制器與非容錯控制器其追蹤性能之表現

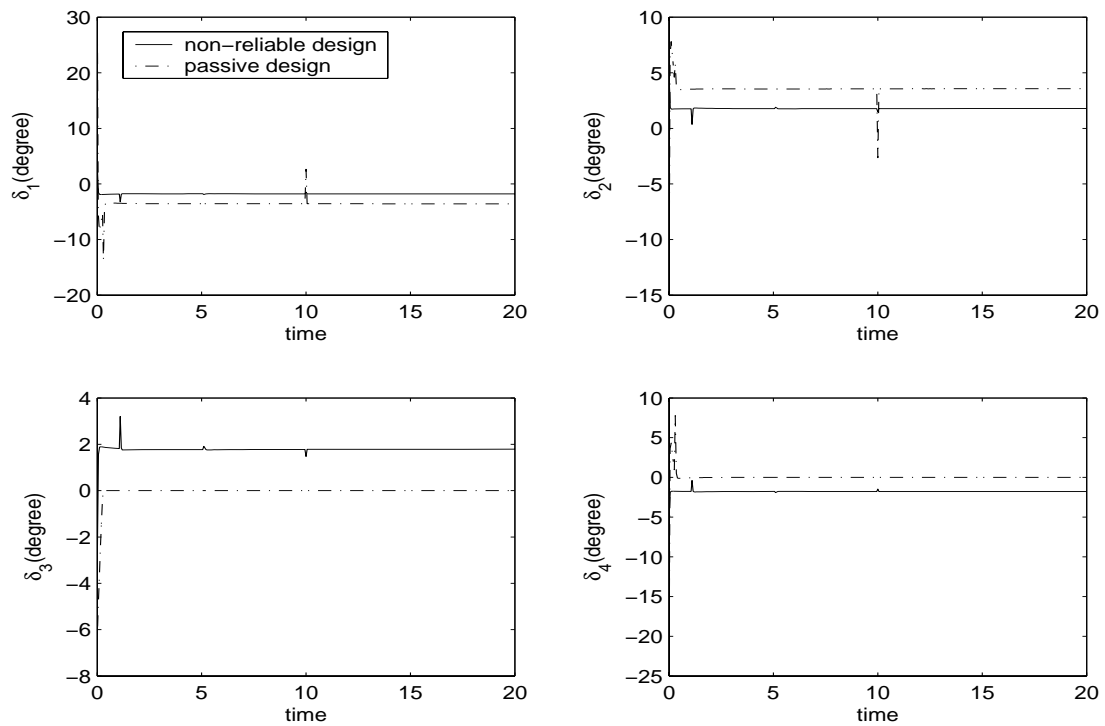


圖 6.30 所有致動器皆正常之下，針對 δ_3 故障所設計之被動式控制器與非容錯控制器其所有致動器之狀態

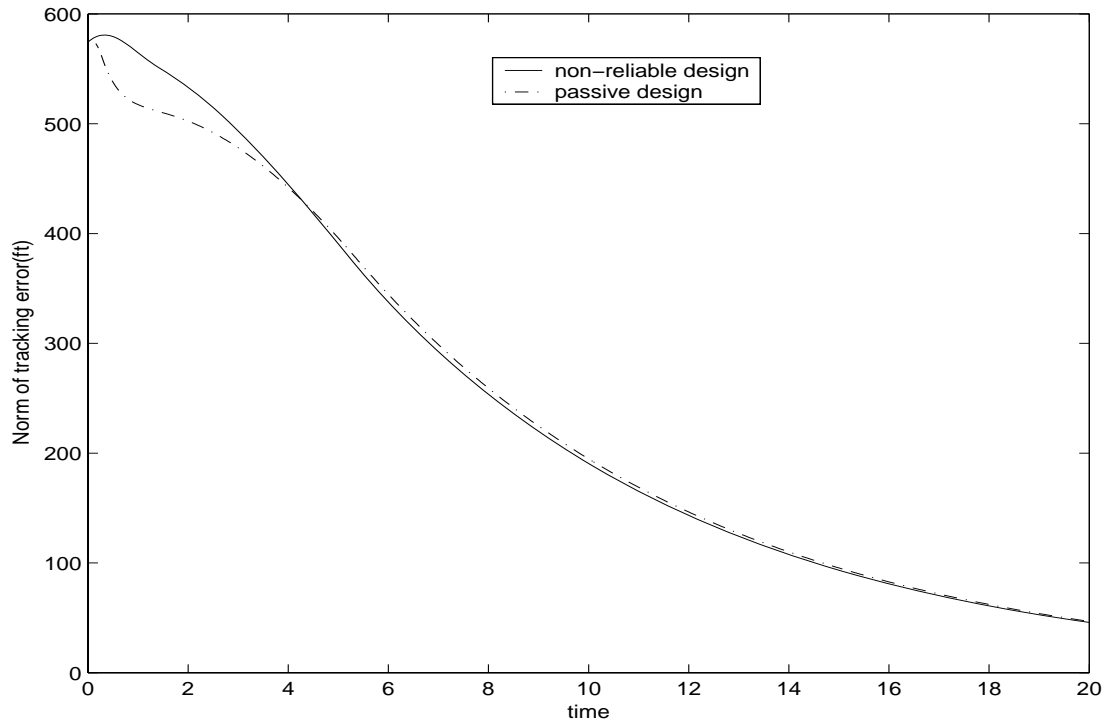


圖 6.31 所有致動器皆正常之下，針對 δ_3 故障所設計之被動式控制器與非容錯控制器其飛彈之追蹤誤差

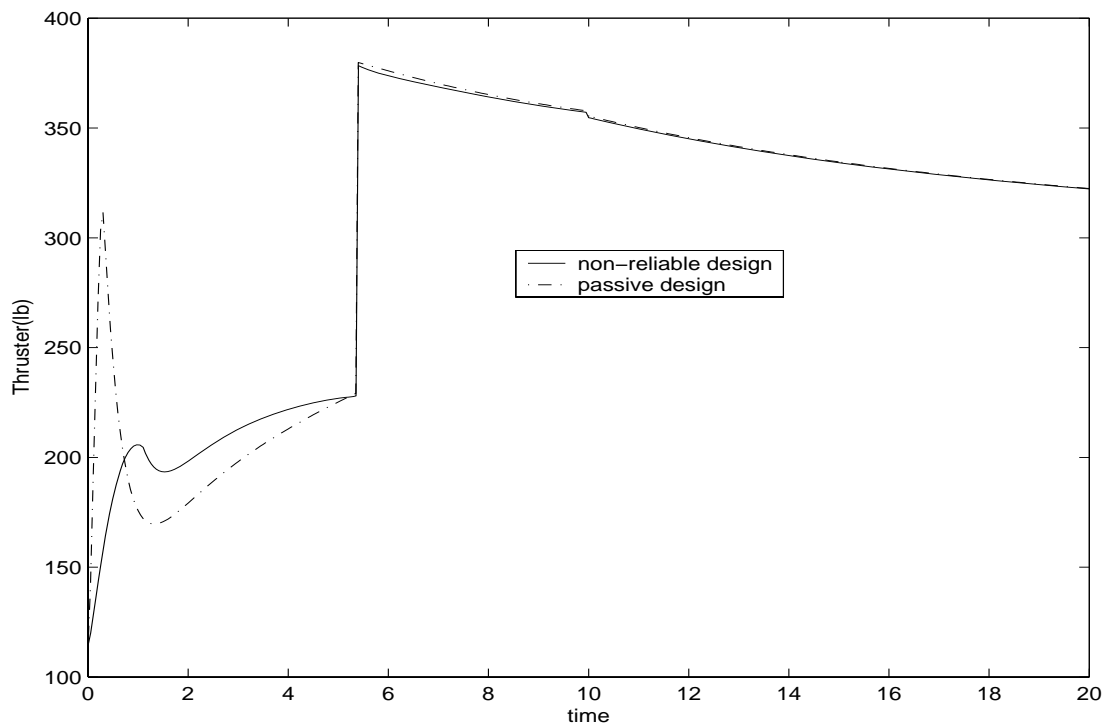


圖 6.32 所有致動器皆正常之下，針對 δ_3 故障所設計之被動式控制器與非容錯控制器其飛彈之推力

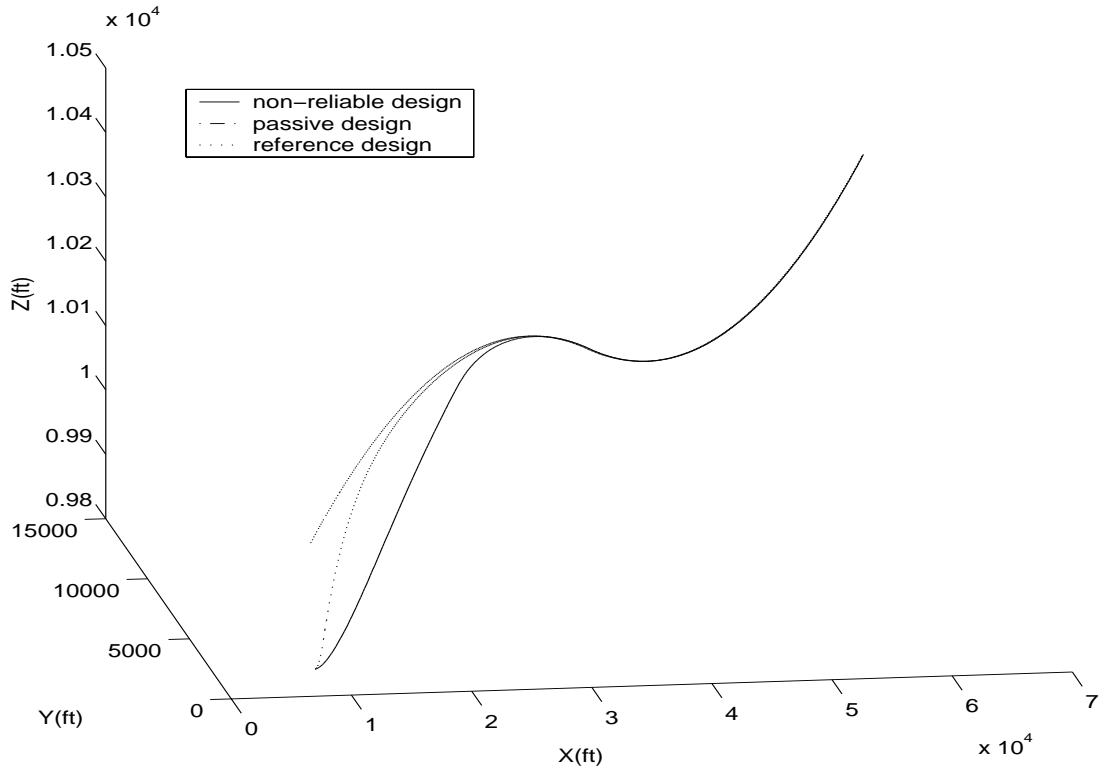


圖 6.33 所有致動器皆正常之下，針對 δ_4 故障所設計之被動式控制器與非容錯控制器其追蹤性能之表現

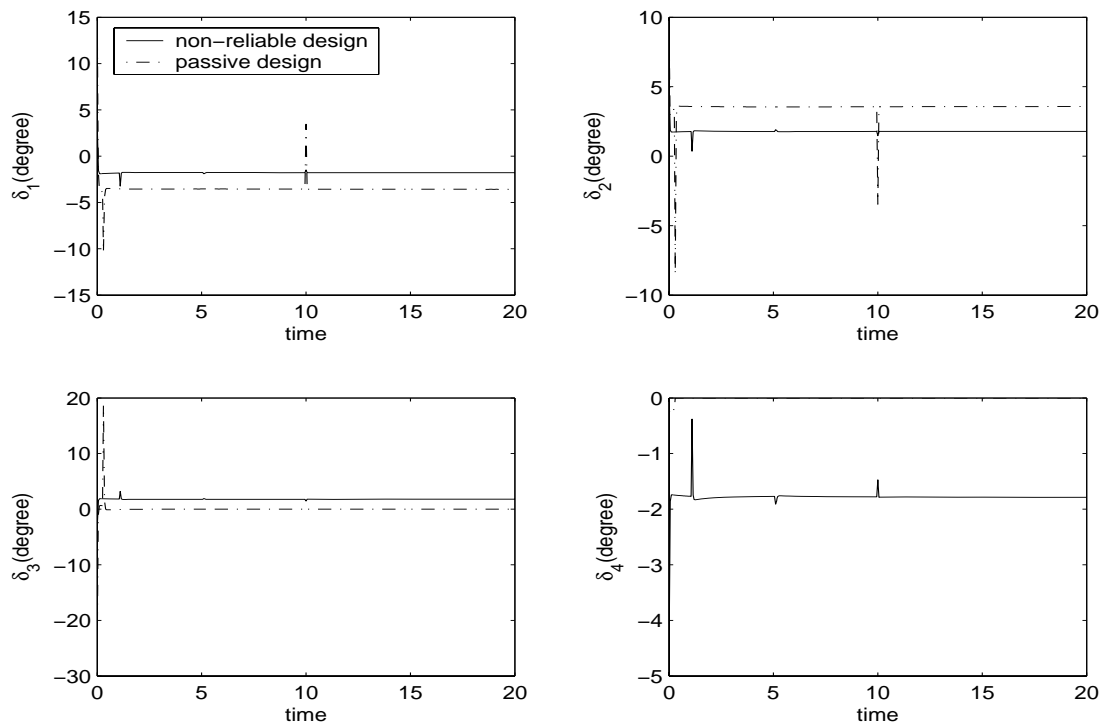


圖 6.34 所有致動器皆正常之下，針對 δ_4 故障所設計之被動式控制器與非容錯控制器其所有致動器之狀態

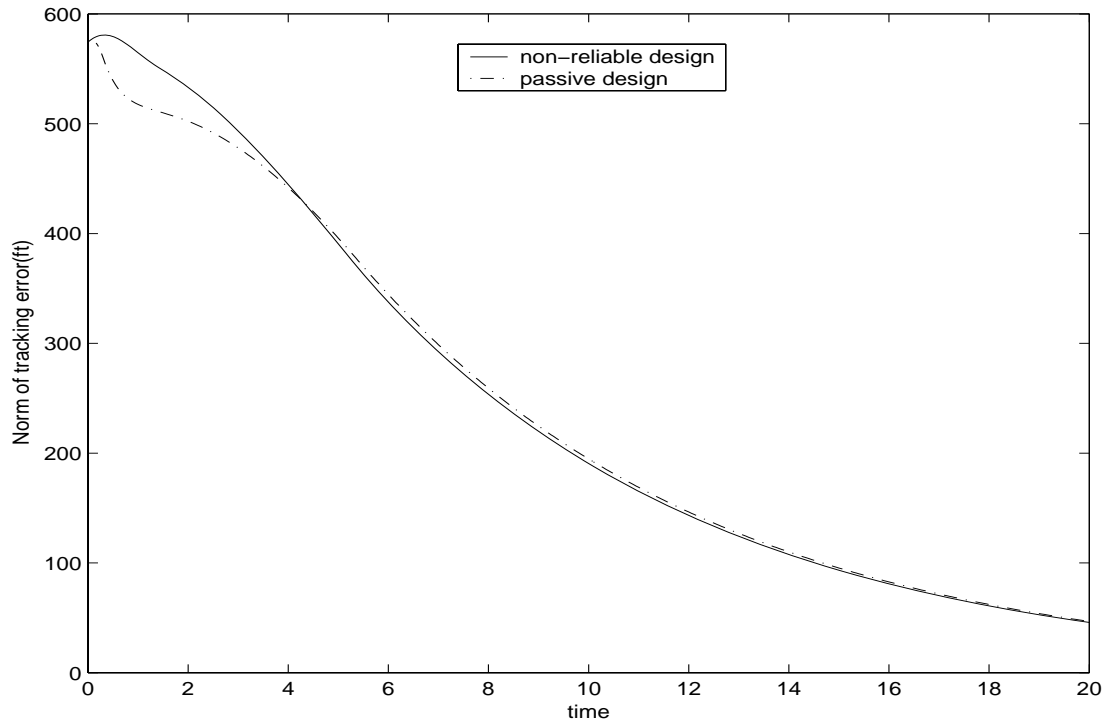


圖 6.35 所有致動器皆正常之下，針對 δ_4 故障所設計之被動式控制器與非容錯控制器其飛彈之追蹤誤差

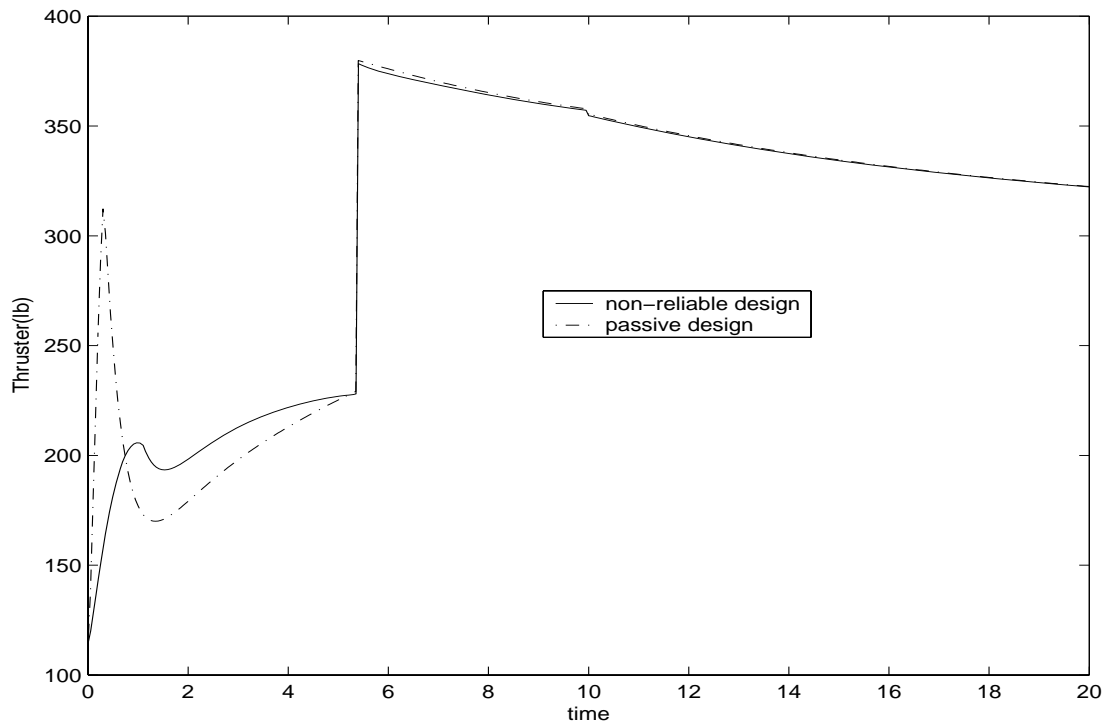


圖 6.36 所有致動器皆正常之下，針對 δ_4 故障所設計之被動式控制器與非容錯控制器其飛彈之推力

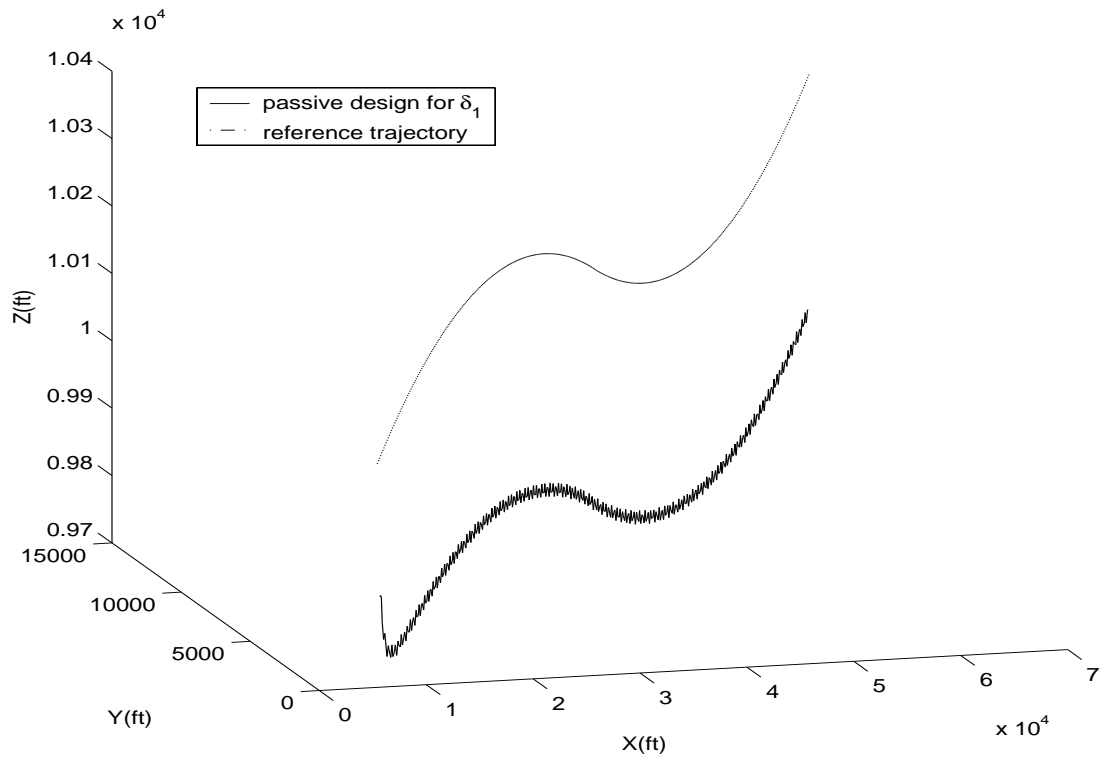


圖 6.37 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_2 發生故障之追蹤性能表現

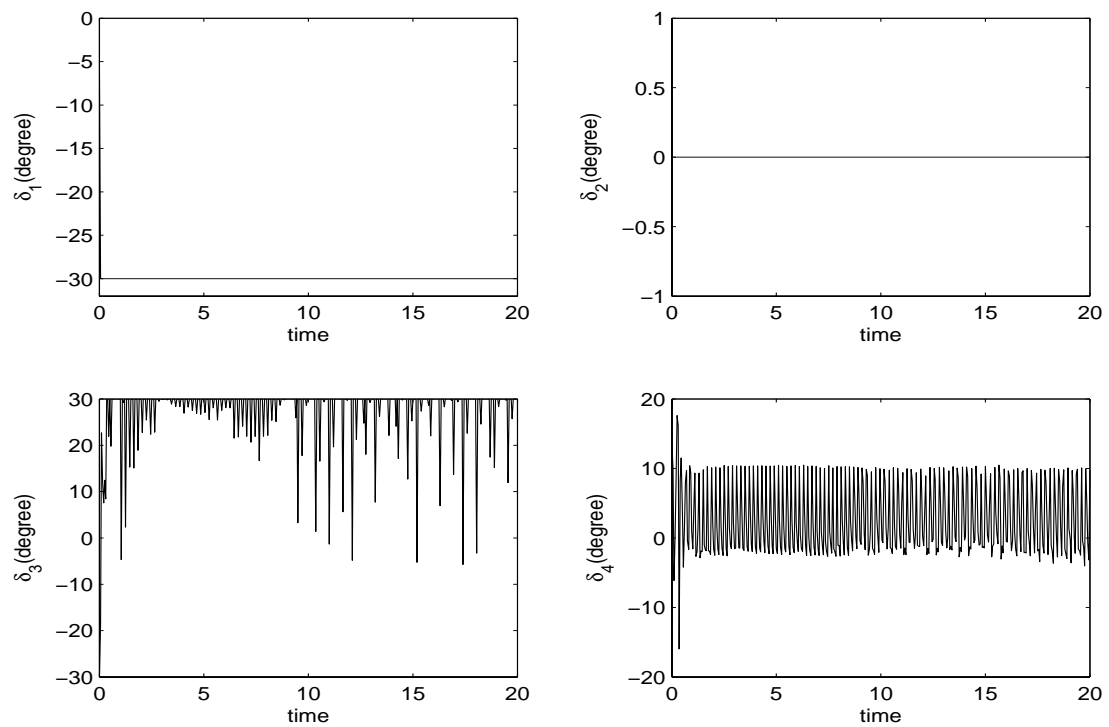


圖 6.38 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_2 發生故障之致動器狀態

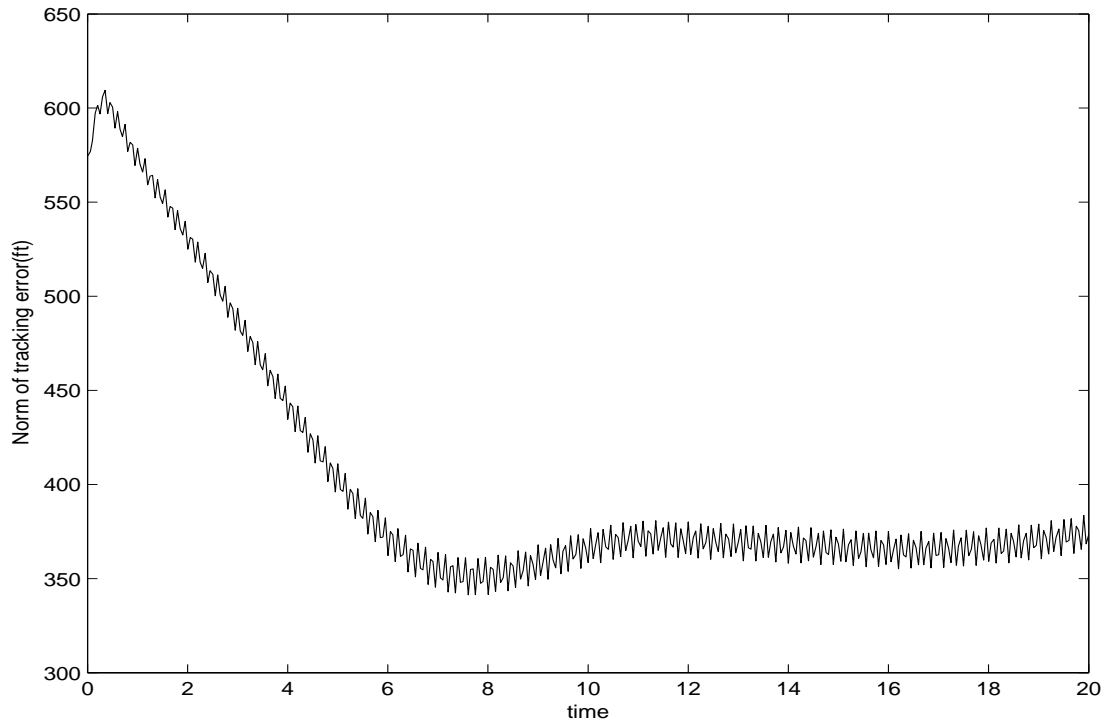


圖 6.39 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_2 發生故障之飛彈追蹤
誤差

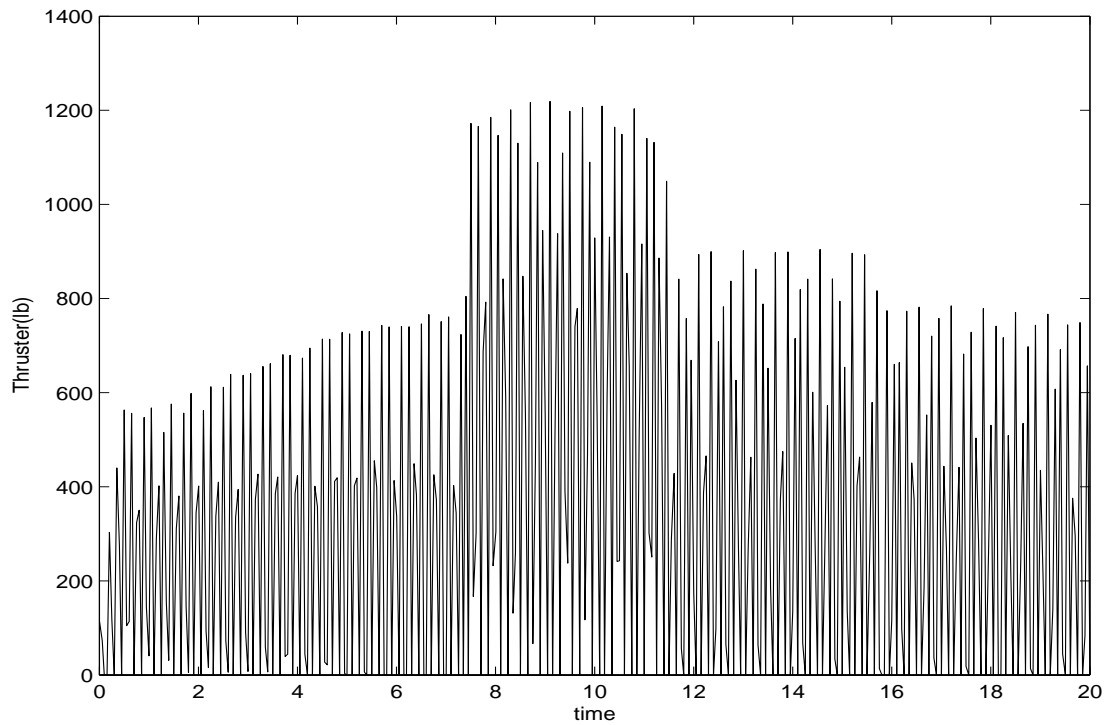


圖 6.40 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_2 發生故障之飛彈推力

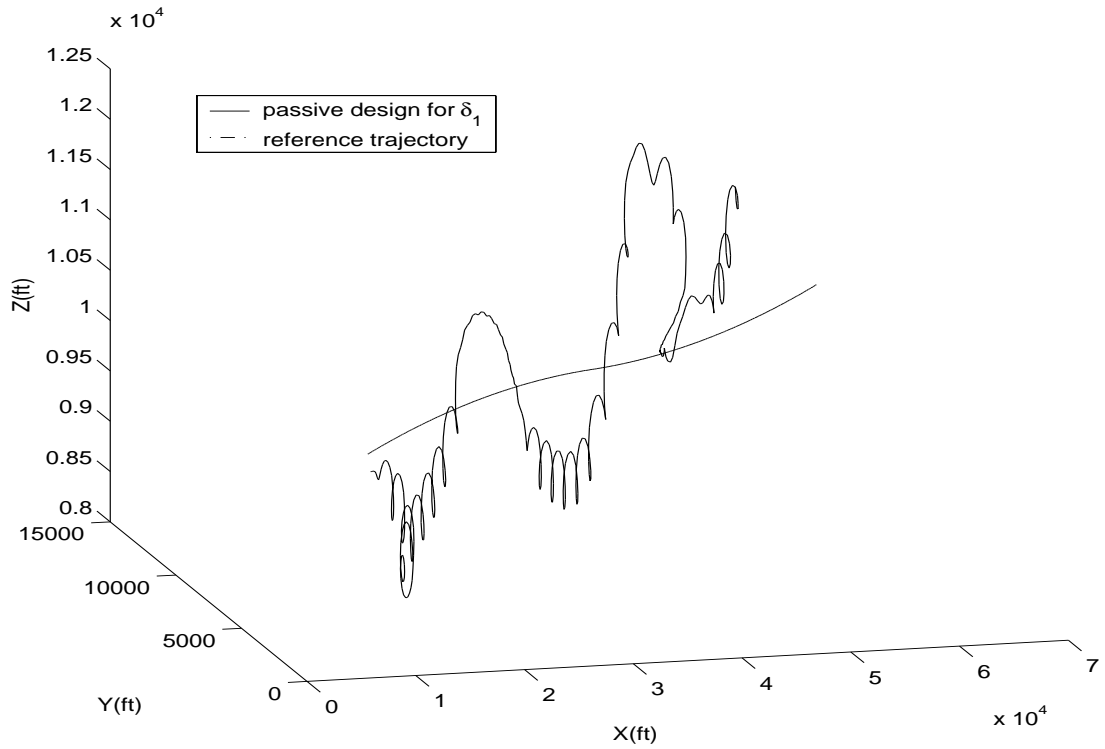


圖 6.41 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_3 發生故障之追蹤性能

表現

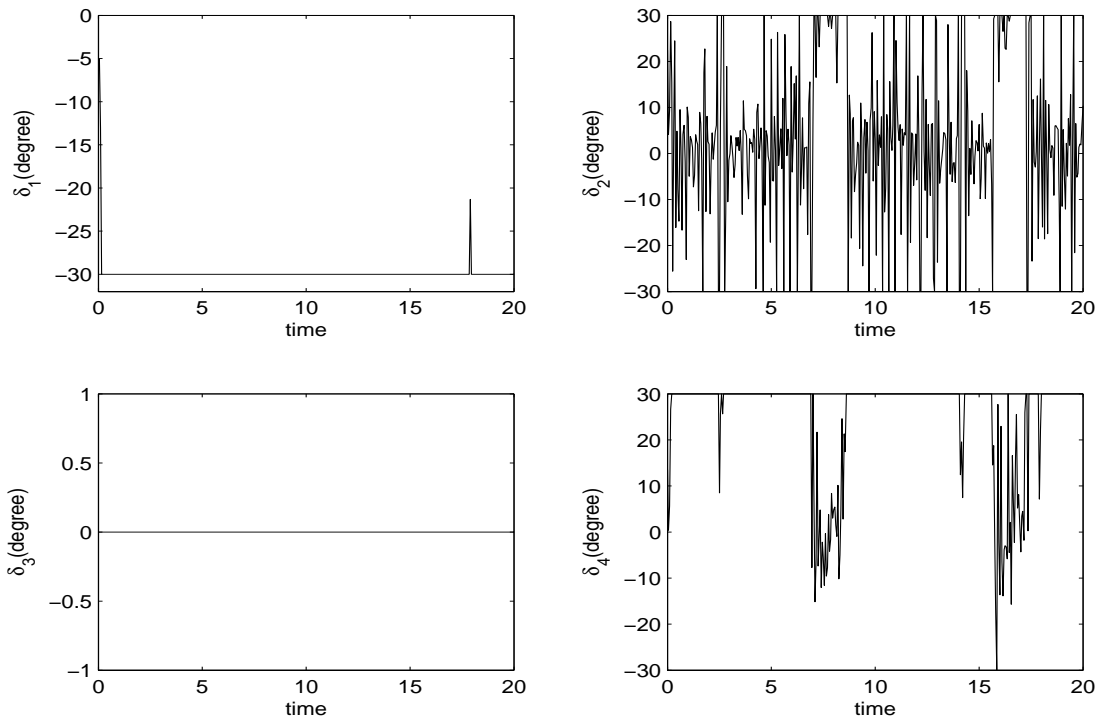


圖 6.42 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_3 發生故障之致動器狀

態

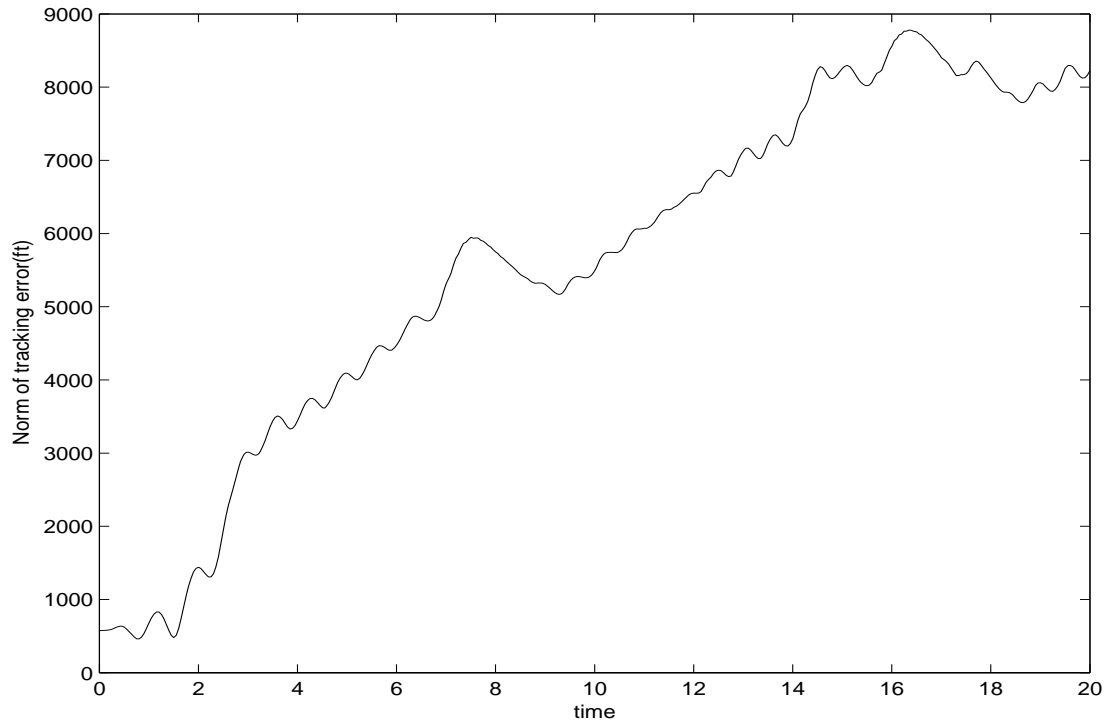


圖 6.43 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_3 發生故障之飛彈追蹤

誤差

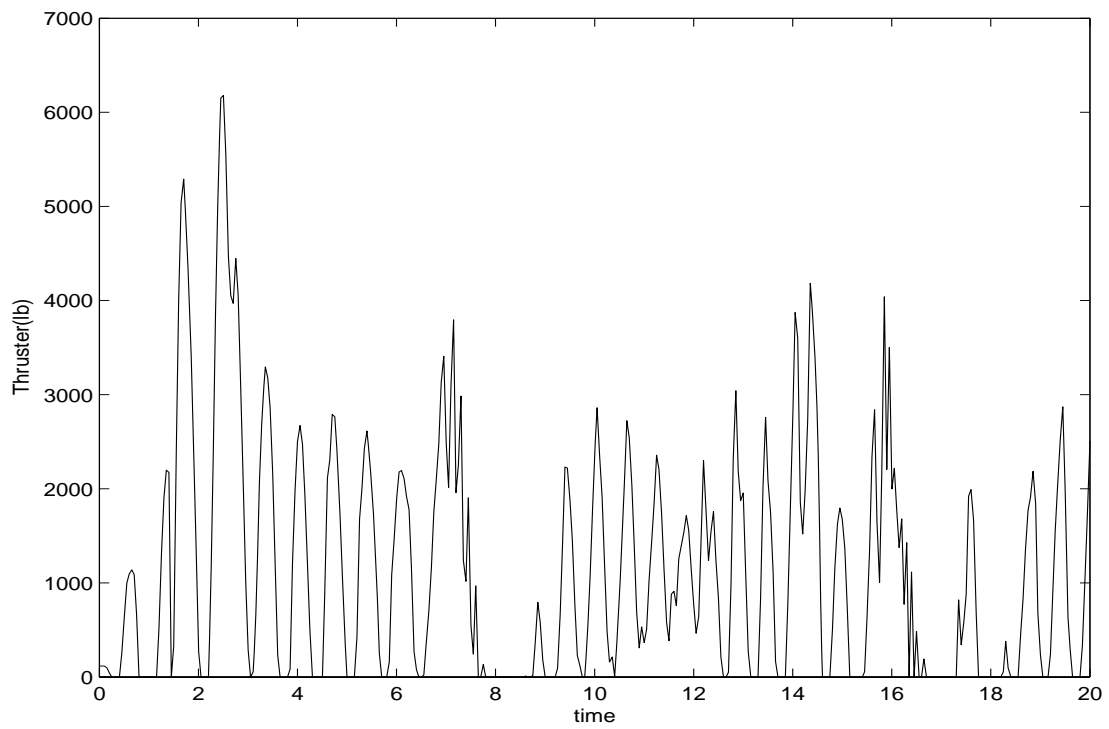


圖 6.44 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_3 發生故障飛彈推力

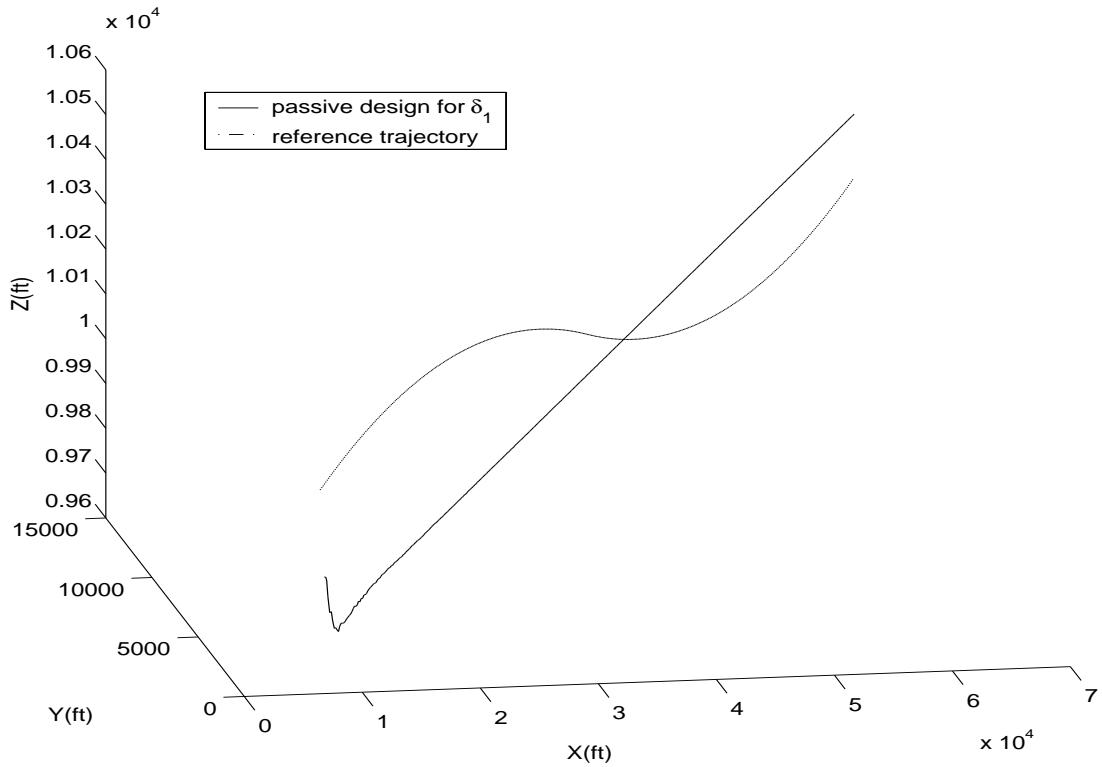


圖 6.45 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_4 發生故障之追蹤性能表現

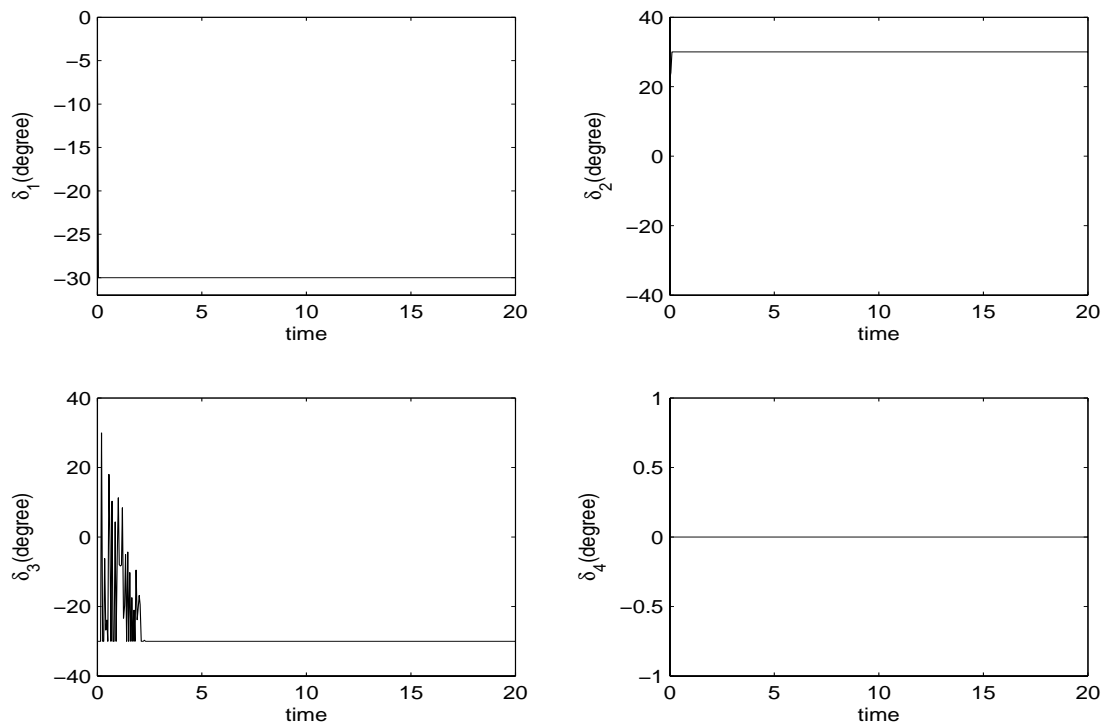


圖 6.46 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_4 發生故障之致動器狀態

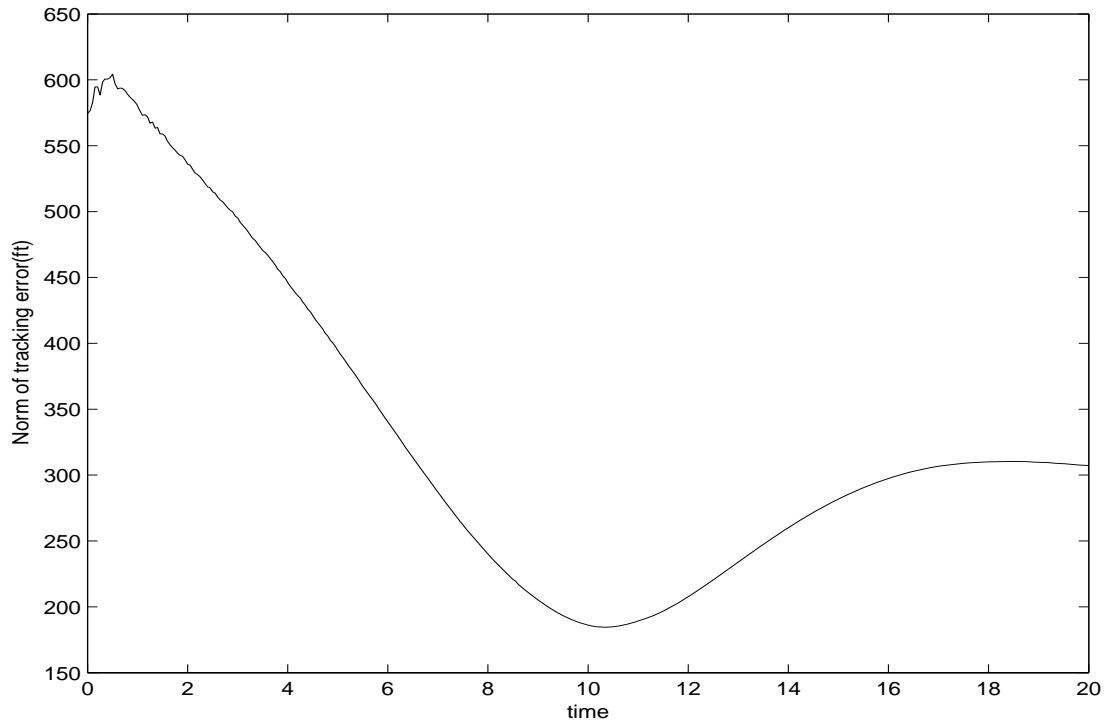


圖 6.47 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_4 發生故障之飛彈追蹤
誤差

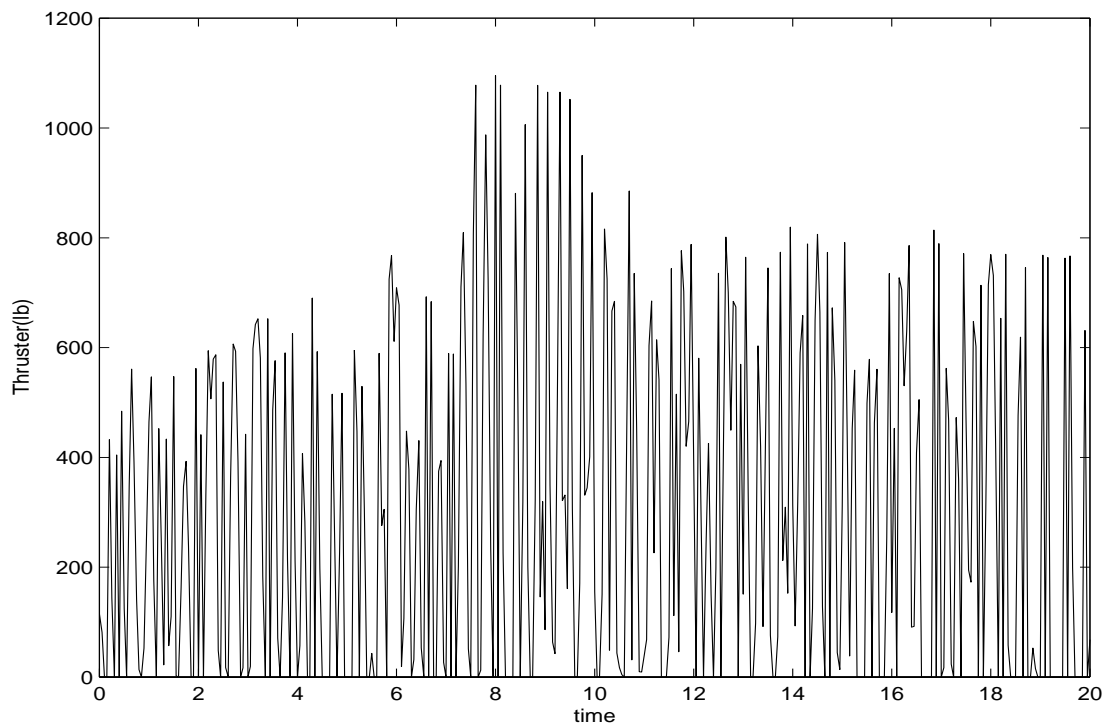


圖 6.48 針對 δ_1 故障所設計之被動式容錯控制器，模擬 δ_4 發生故障之飛彈推力

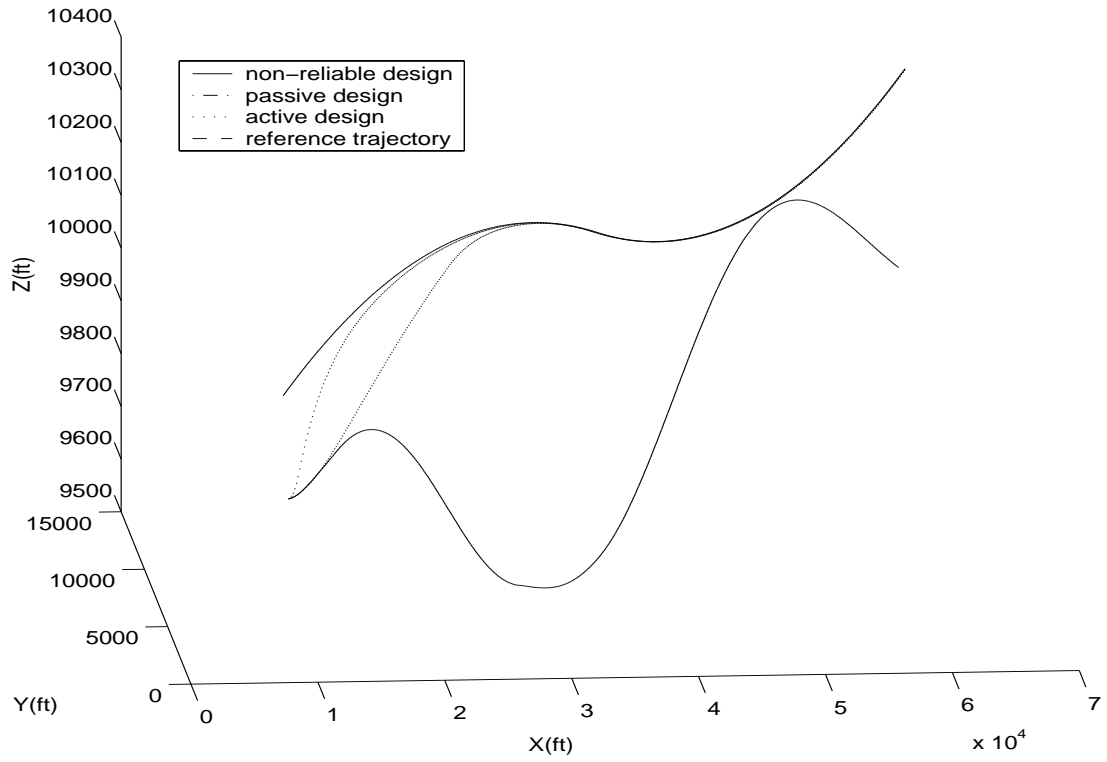


圖 6.49 針對 δ_1 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之追蹤性能

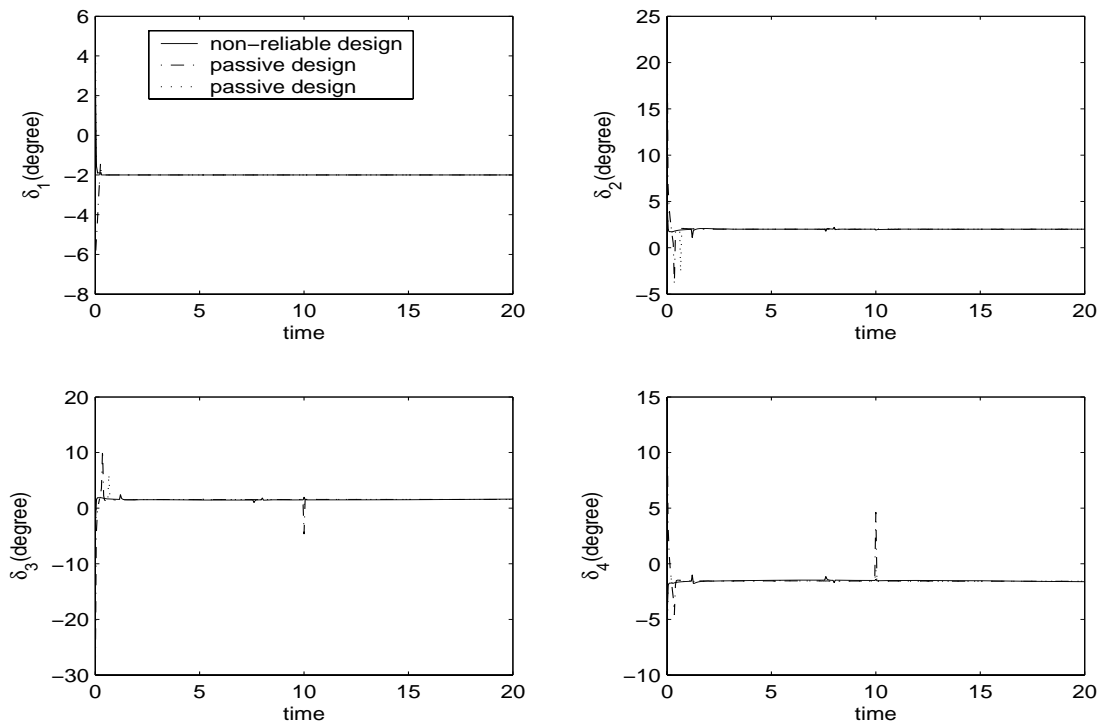


圖 6.50 針對 δ_1 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之致動器變化狀態

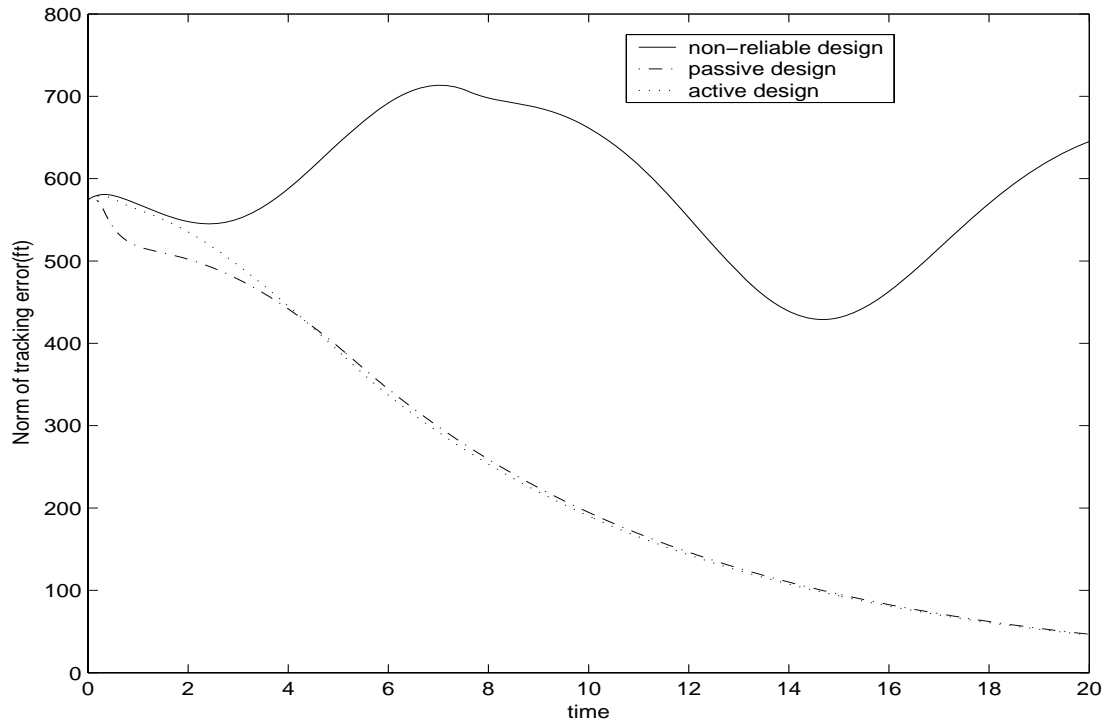


圖 6.51 針對 δ_1 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計，飛彈之
追蹤誤差

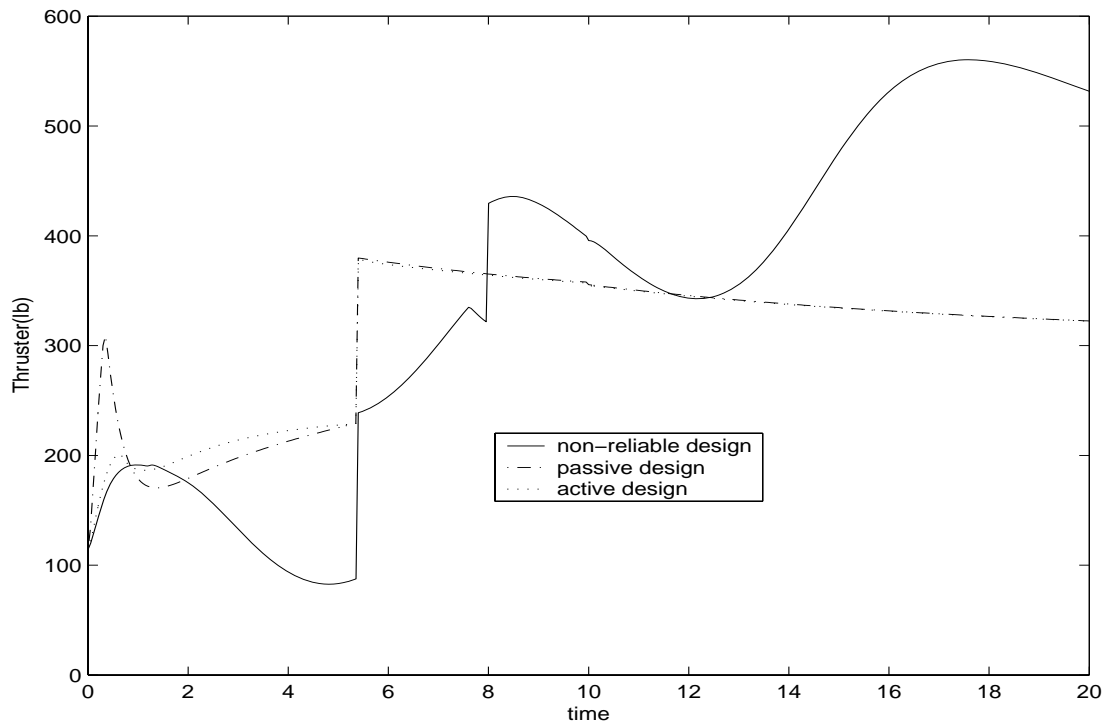


圖 6.52 針對 δ_1 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計，飛彈推
力曲線

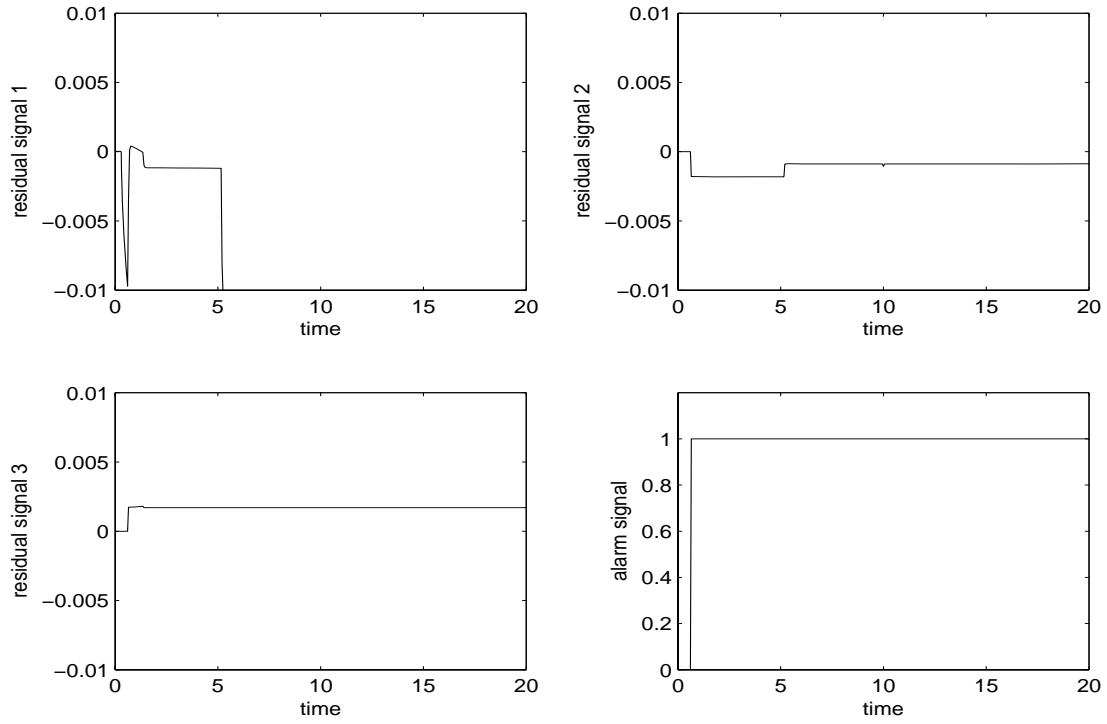


圖 6.53 針對 δ_1 於 0.3 秒時卡死時，錯誤偵測與診斷機制所提供之 residual signal 與 alarm signal

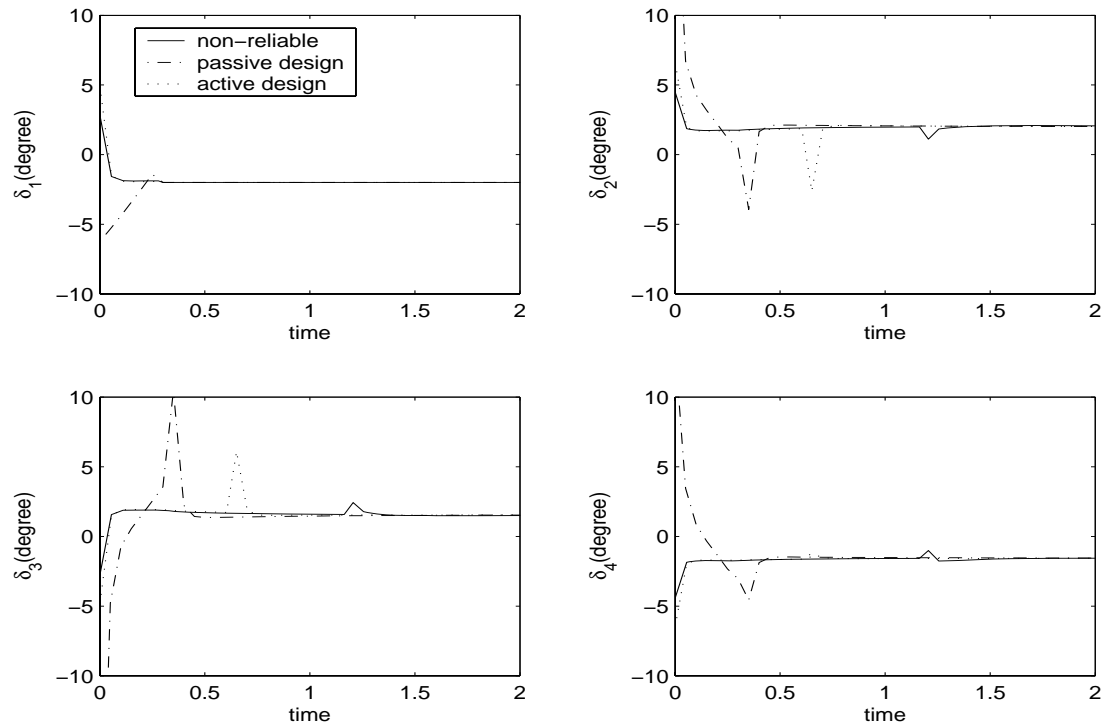


圖 6.54 針對 δ_1 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計，致動器在 0~2 秒之運作狀態

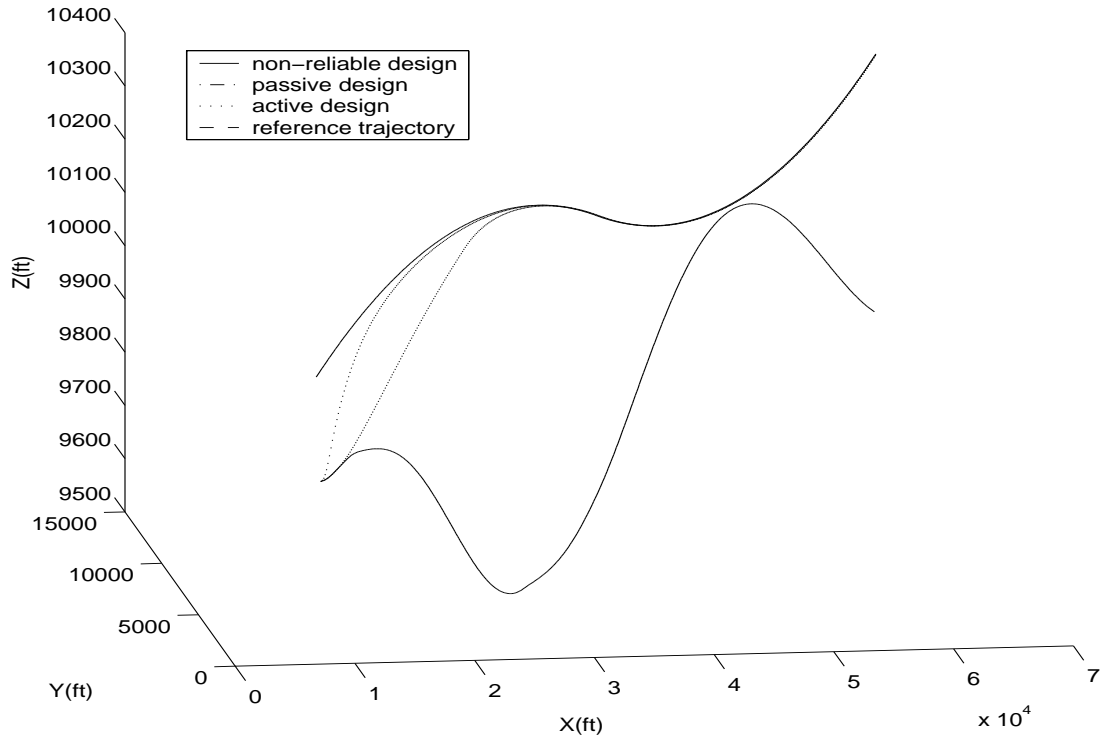


圖 6.55 針對 δ_2 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之追蹤性

能

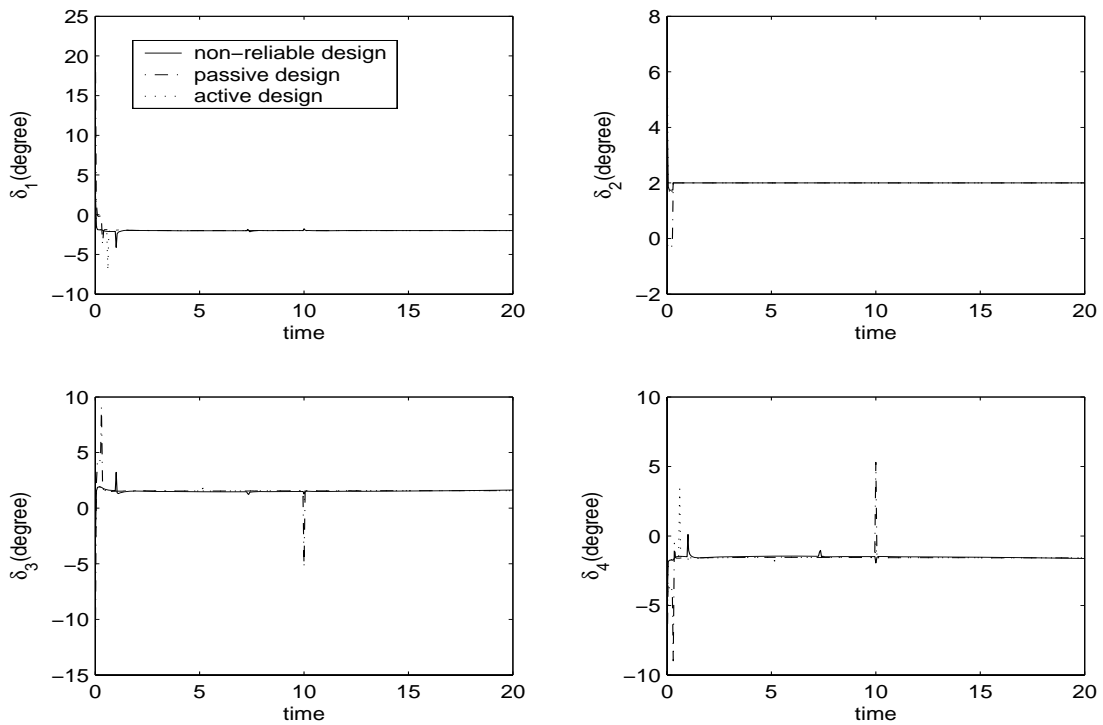


圖 6.56 針對 δ_2 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之

致動器變化狀態

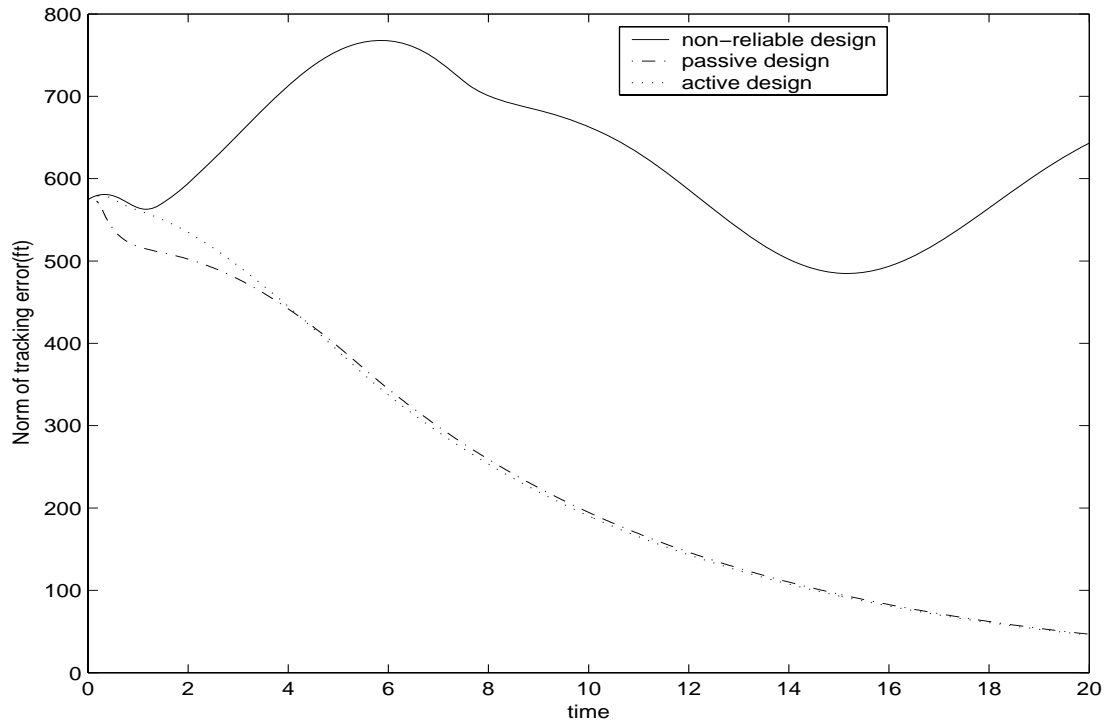


圖 6.57 針對 δ_2 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之飛彈追蹤誤差

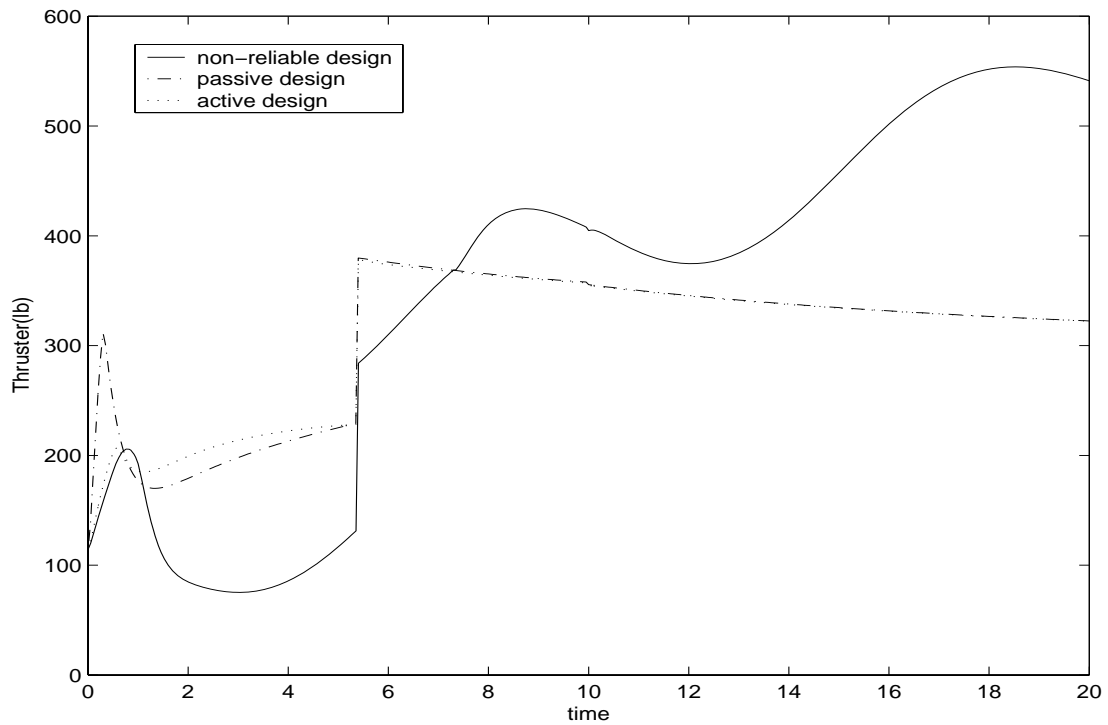


圖 6.58 針對 δ_2 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之飛彈推力曲線

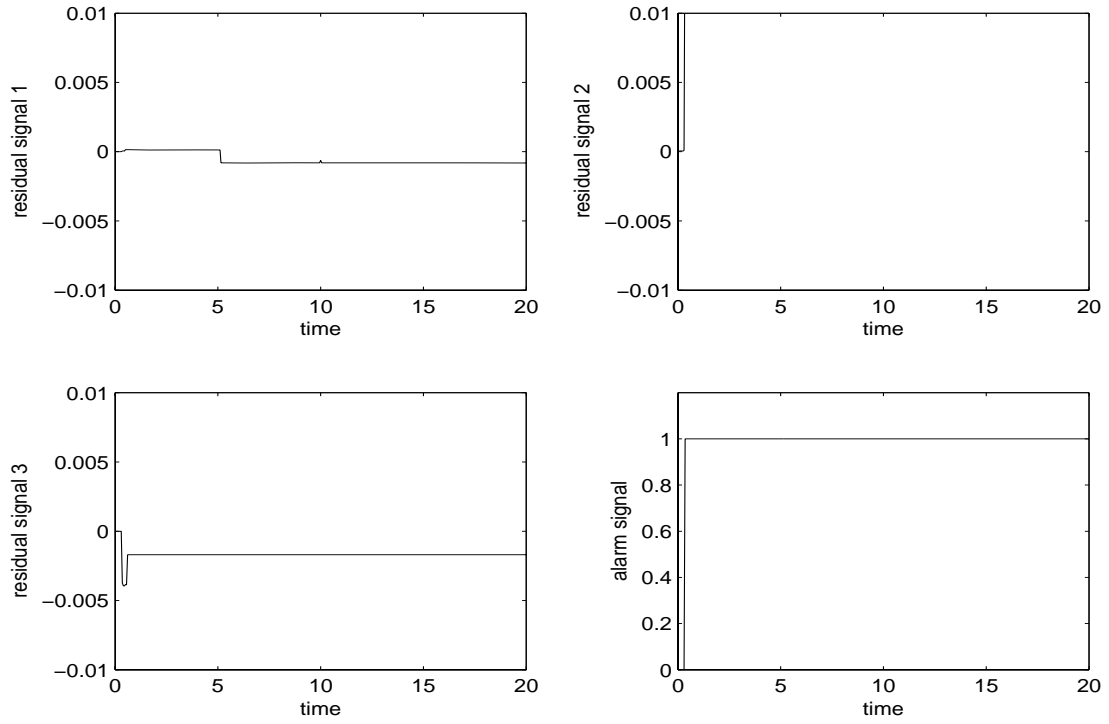


圖 6.59 針對 δ_2 於 0.3 秒時卡死時，錯誤偵測與診斷機制所提供之 residual signal 與 alarm signal

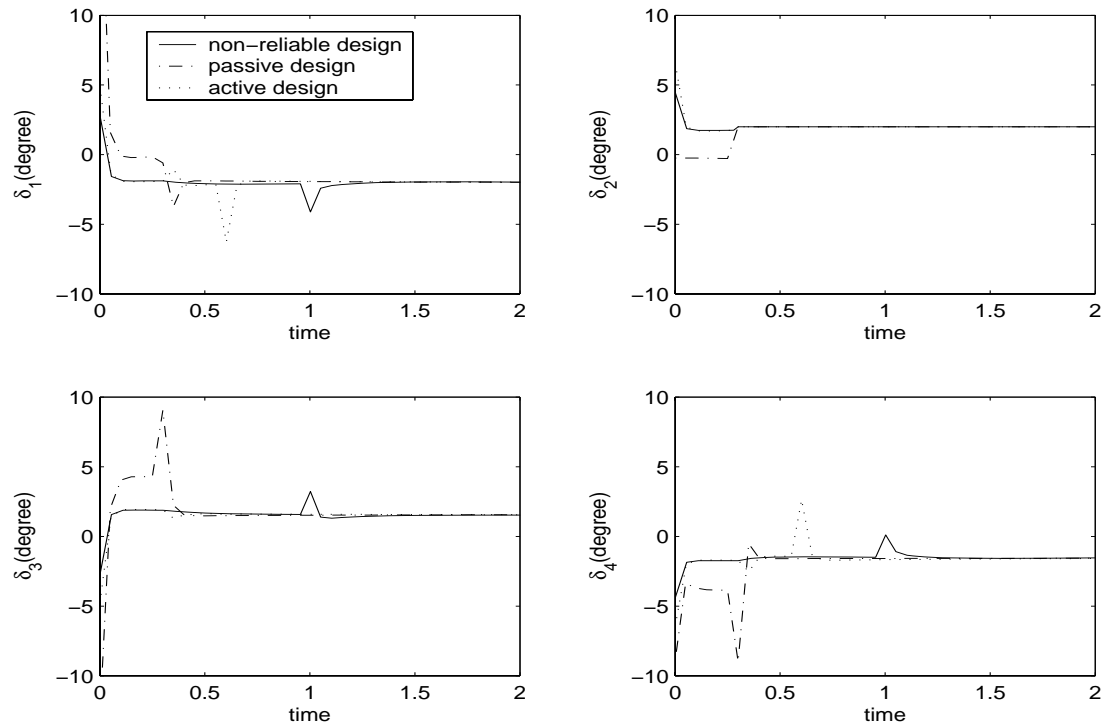


圖 6.60 針對 δ_2 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計，

致動器在 0~2 秒運作之狀態

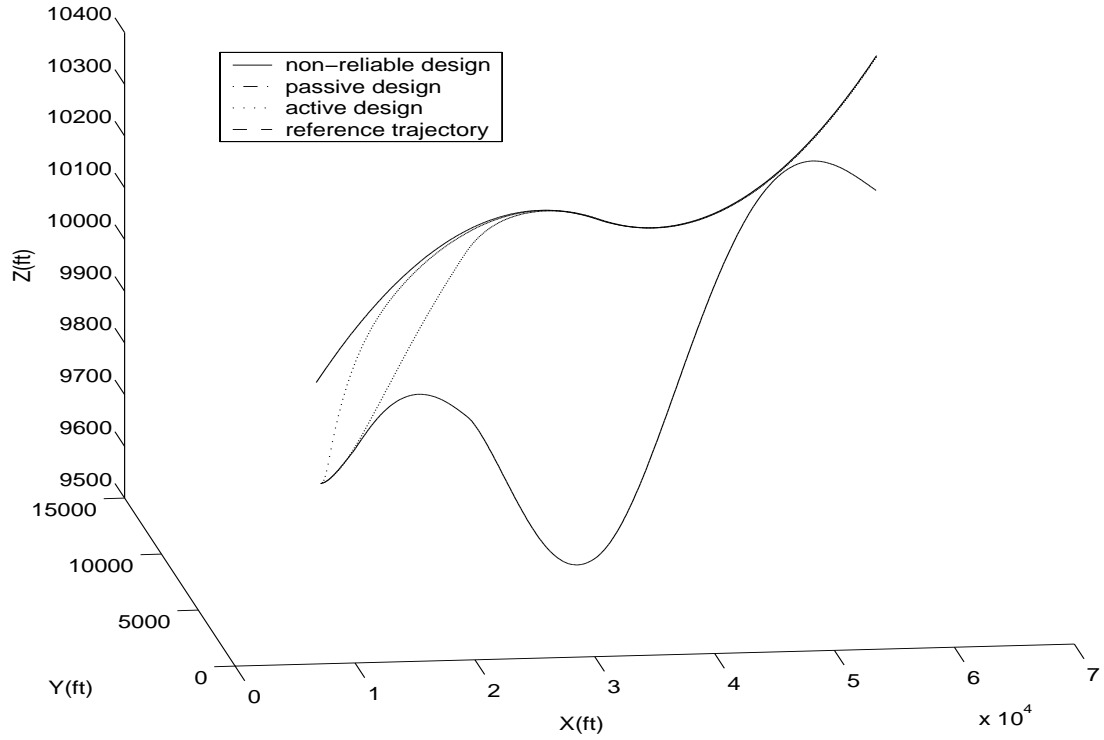


圖 6.61 針對 δ_3 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之追蹤性

能

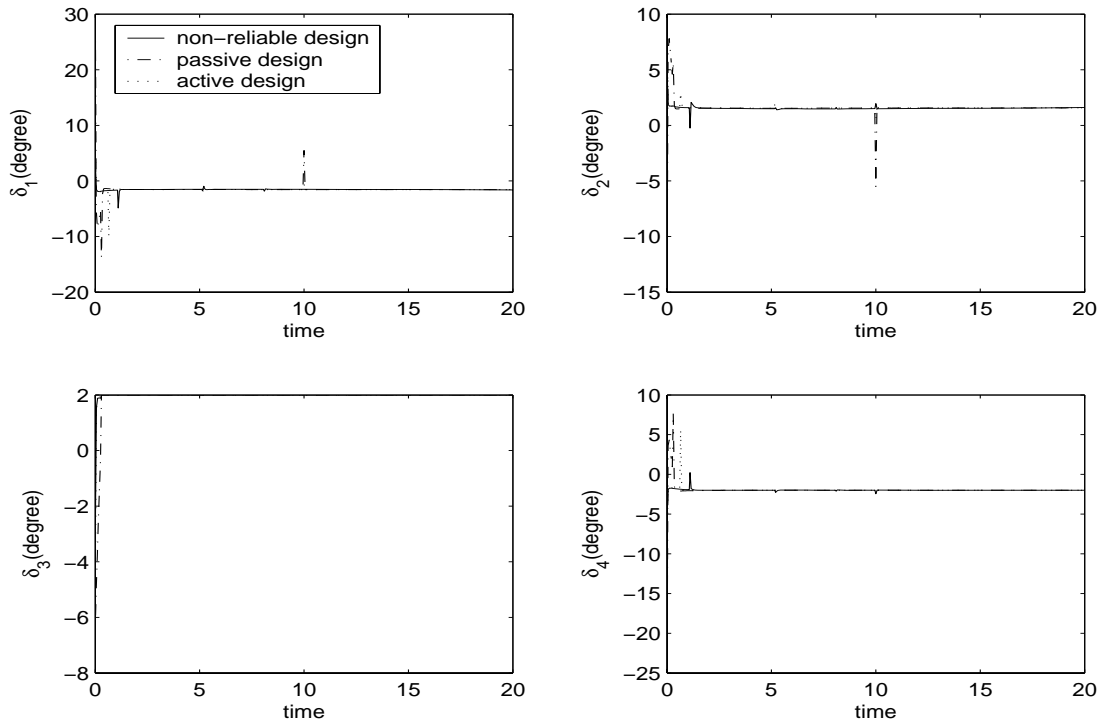


圖 6.62 針對 δ_3 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之

致動器變化狀態

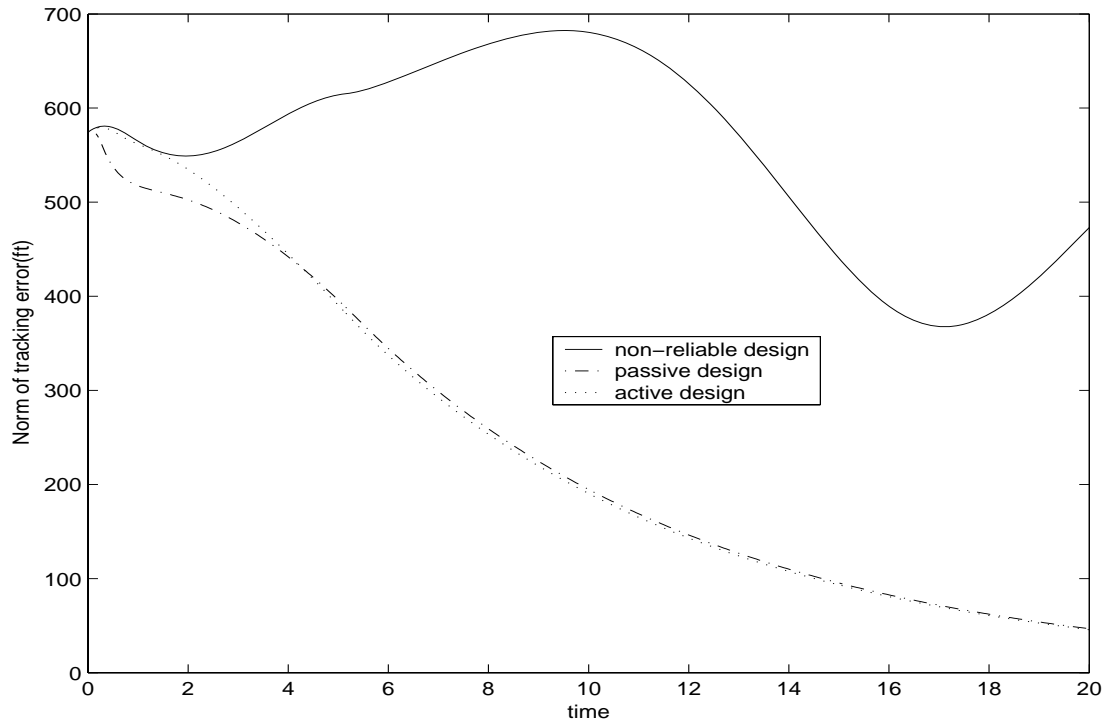


圖 6.63 針對 δ_3 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之飛彈追蹤誤差

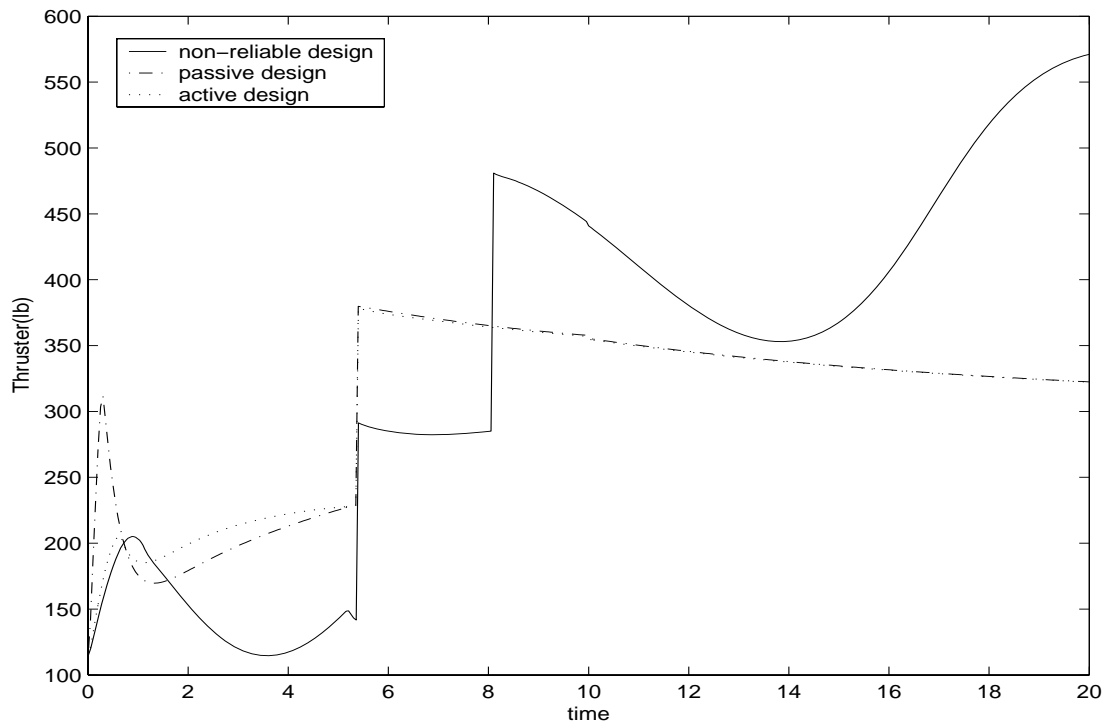


圖 6.64 針對 δ_3 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之飛彈推力

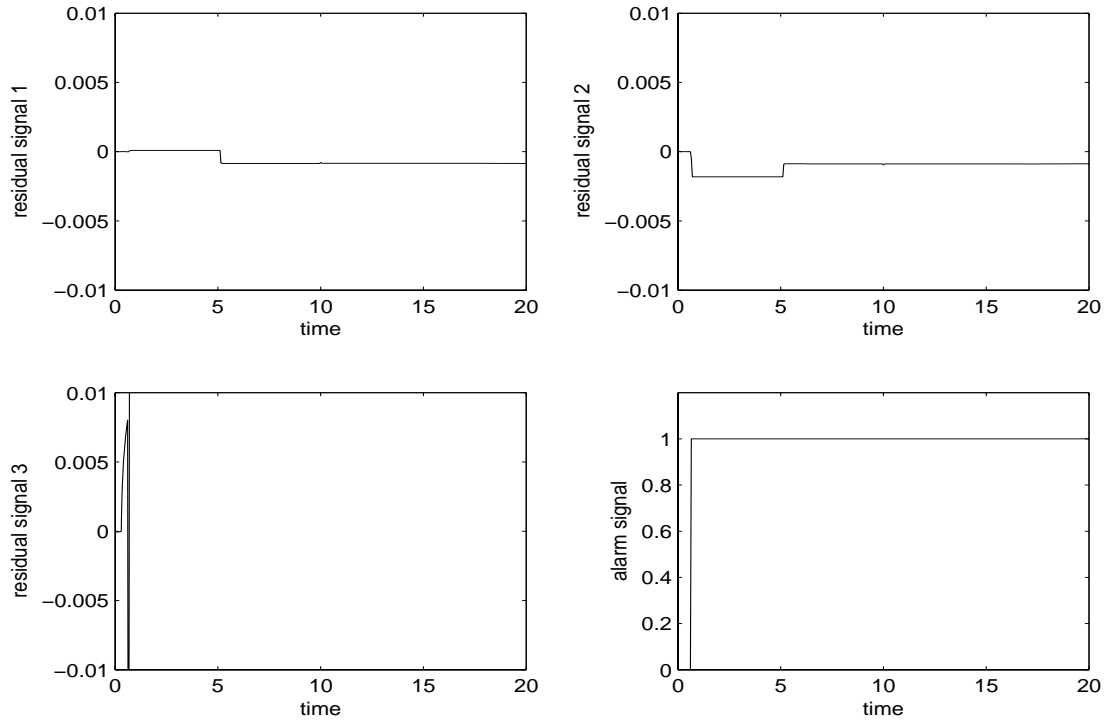


圖 6.65 針對 δ_3 於 0.3 秒時卡死時，錯誤偵測與診斷機制所提供之 residual signal 與 alarm signal

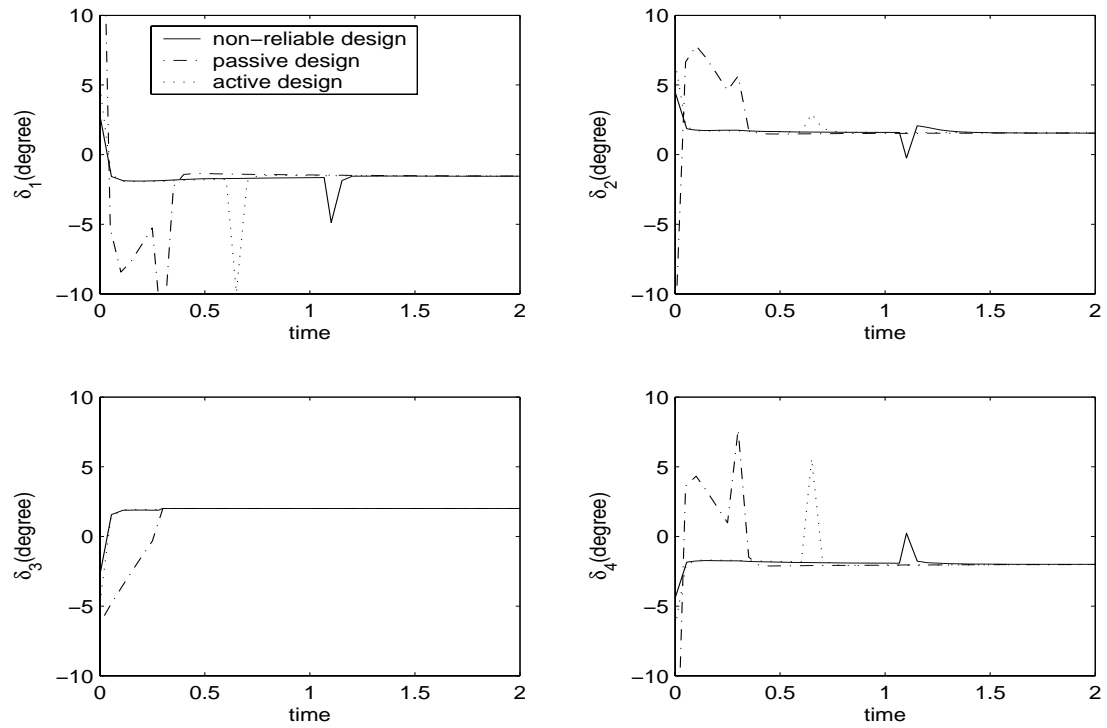


圖 6.66 針對 δ_3 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計，致動器於 0~2 秒運作之狀態

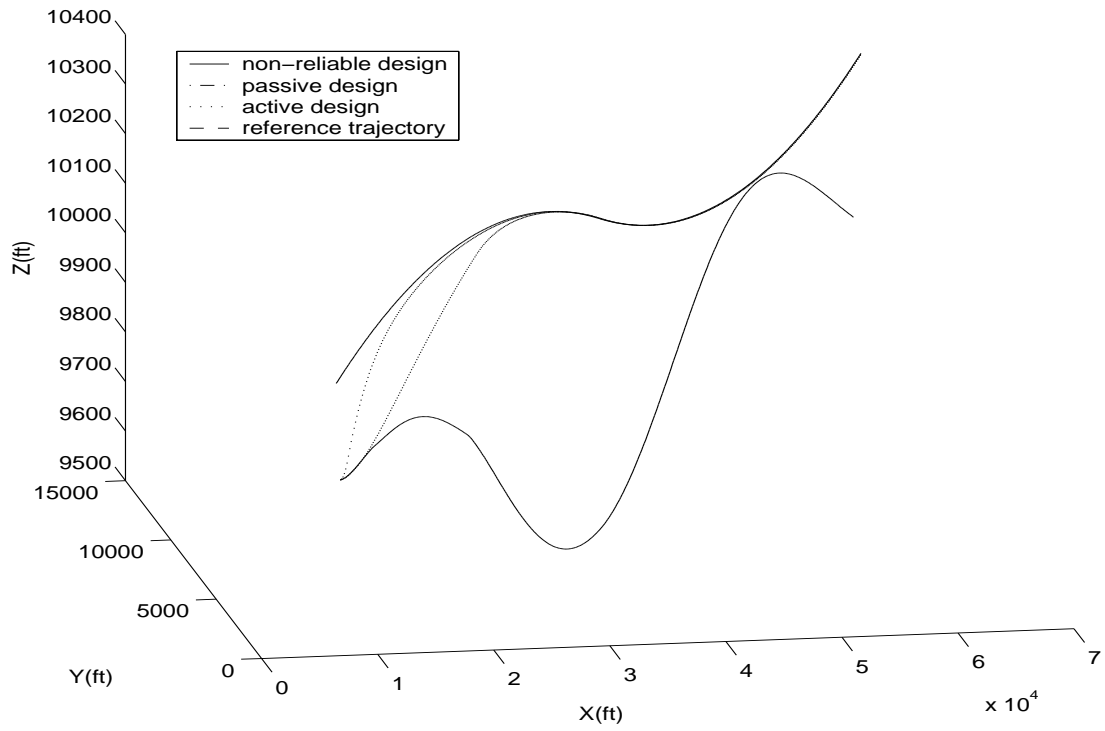


圖 6.67 針對 δ_4 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之追蹤性

能

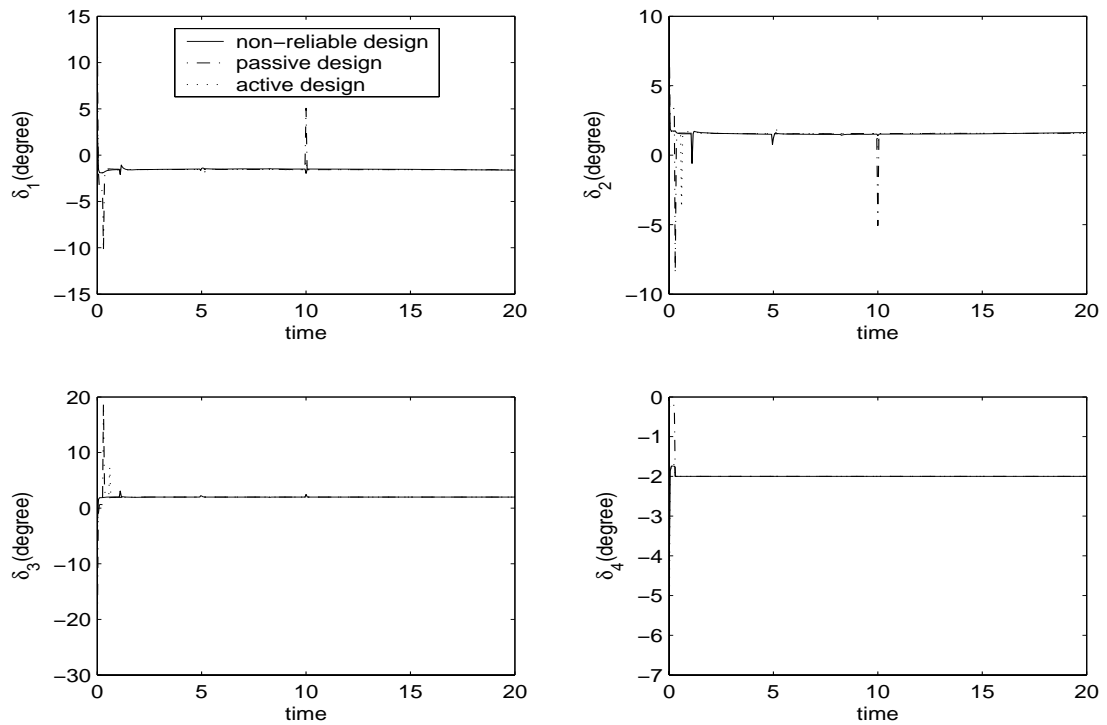


圖 6.68 針對 δ_4 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之

致動器變化狀態

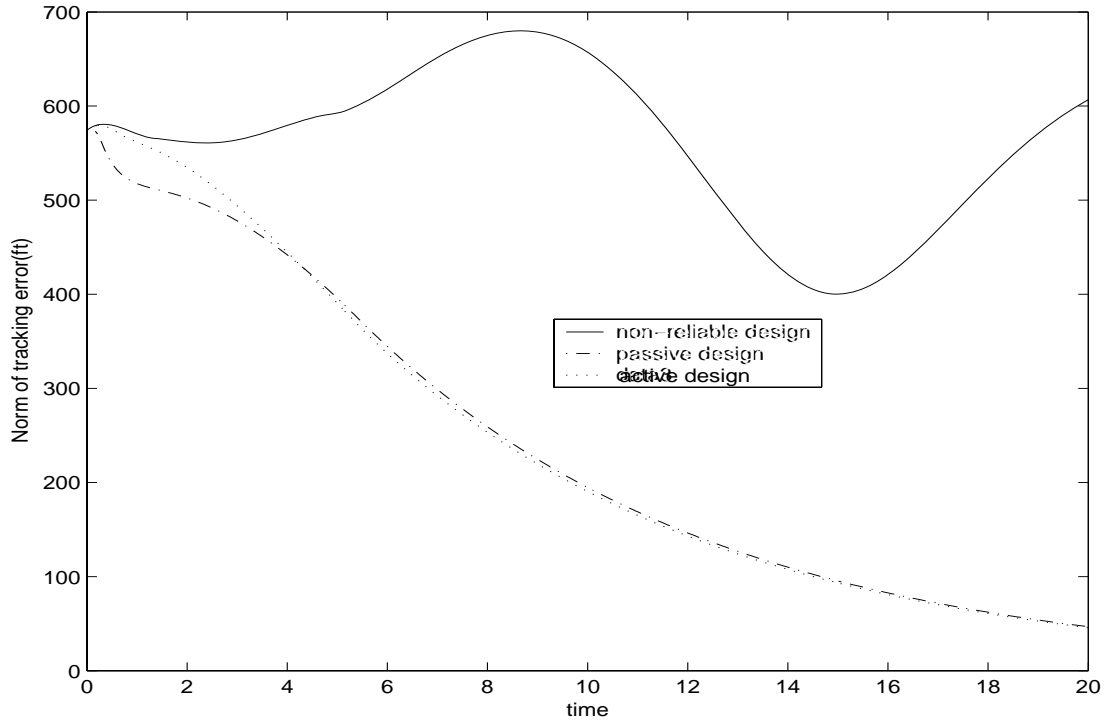


圖 6.69 針對 δ_4 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之飛彈追蹤誤差

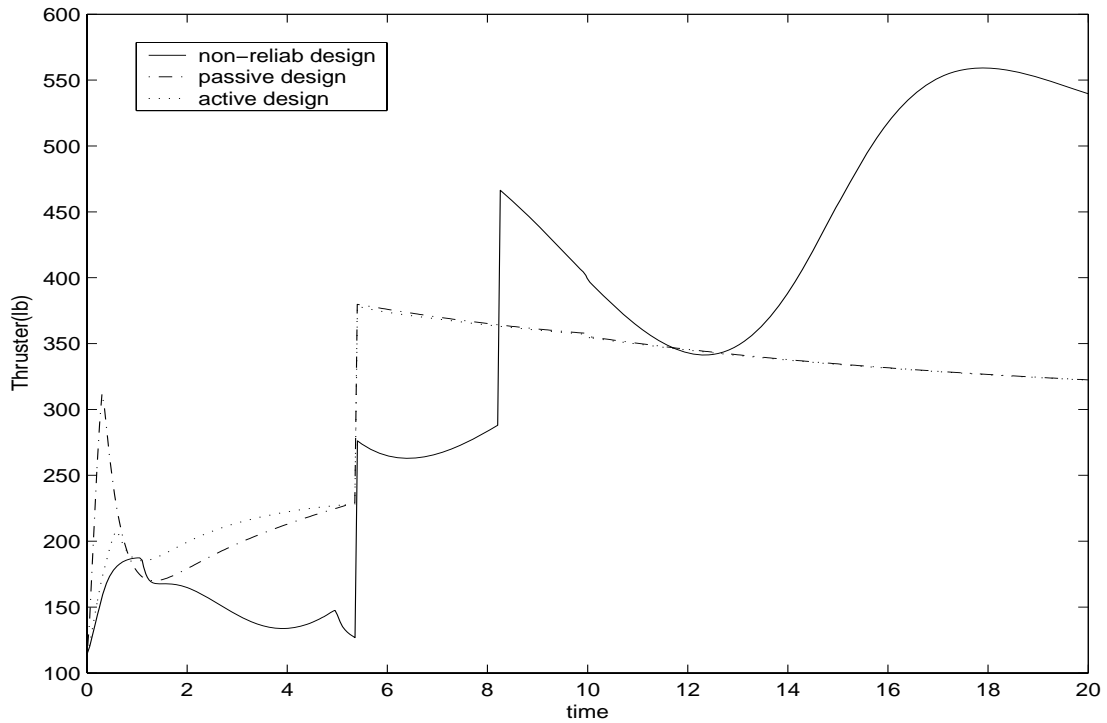


圖 6.70 針對 δ_4 於 0.3 秒時卡死時，非容錯、被動式與主動式容錯設計之飛彈推力

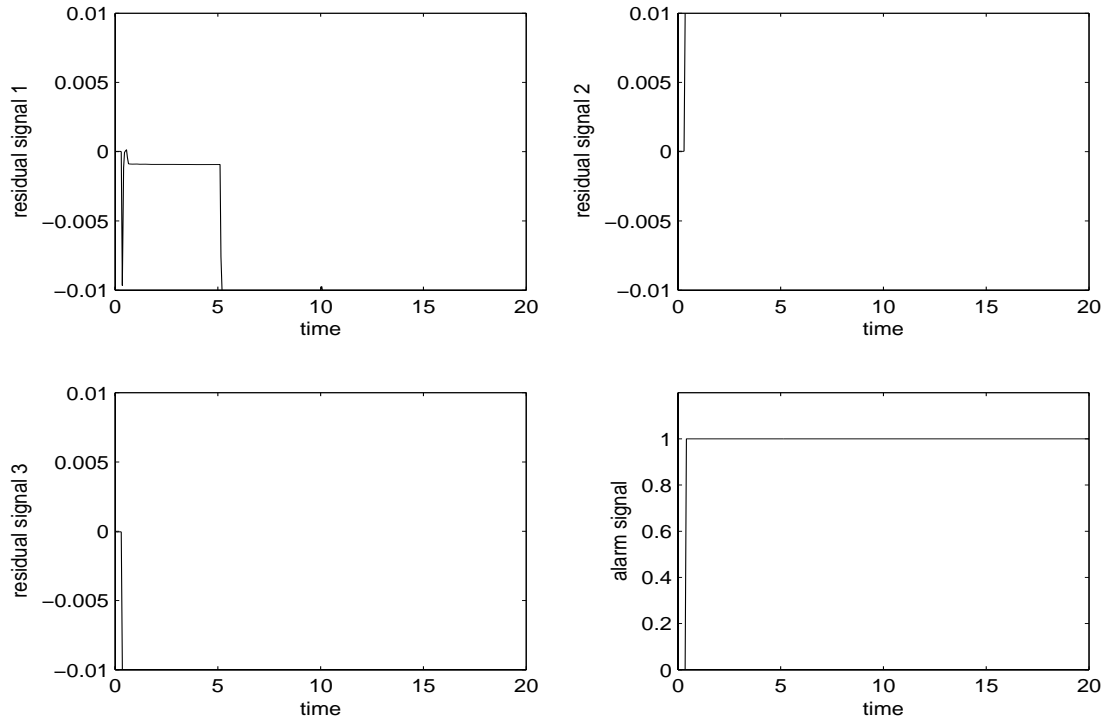


圖 6.71 針對 δ_4 於 0.3 秒時卡死時，錯誤偵測與診斷機制所提供之 residual signal 與 alarm signal

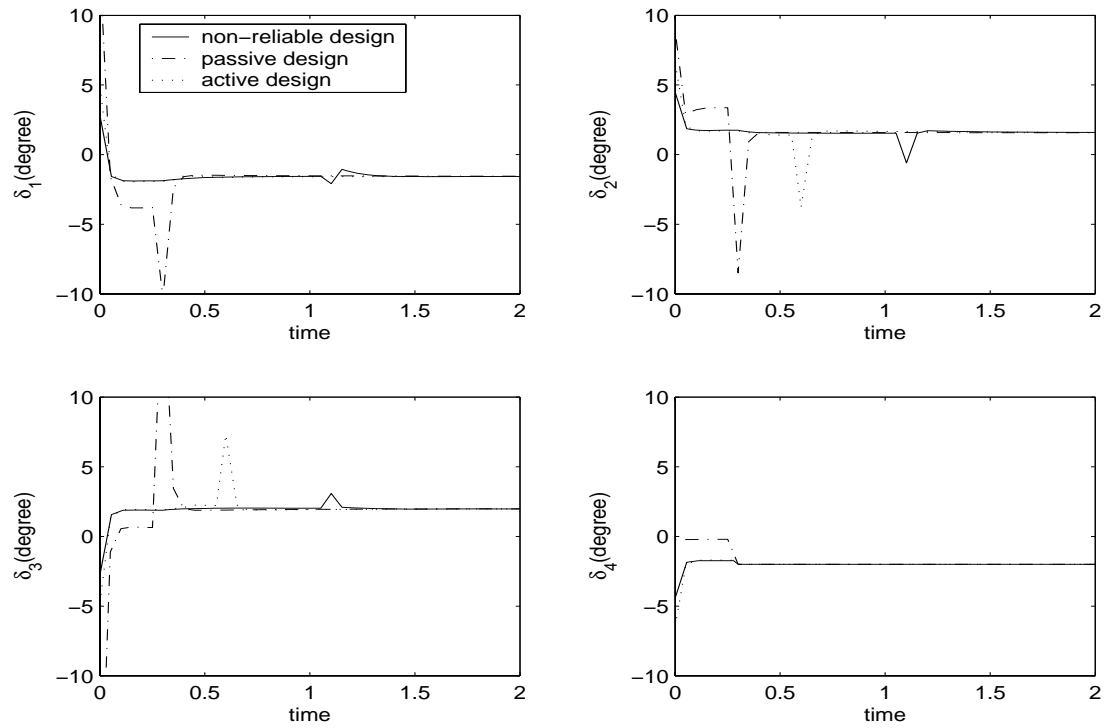


圖 6.72 針對 δ_4 於 0.3 秒時卡死時，非容錯、被動式與主動式控制器，致動器於 0~2 秒運作之狀態

第七章

結論

本計畫的主要目的在於探討如何精準的預估來襲彈道飛彈之飛行軌跡，以及如何使用容錯控制技術使我方攔截飛彈具有智慧型的容錯控制能力及高可靠性，達到快速且有效的攔截目的。在彈道預估方面，我們使用線上基因演算法(Genetic Algorithm)來預測彈道，即使來襲飛彈具有閃躲功能，軌跡估測器仍可在存有量測誤差的情況下，達到良好預估效果。在攔截導引律的設計方面，我們使用所謂的容錯控制技術，使攔截任務在特定致動器發生故障或異常的情況下仍能有效的被達成。此外，為了有效的執行容錯控制，本計畫也設計了一組錯誤診斷與偵測機制來反應系統的狀況，偵測並診斷飛彈尾翼之故障情況以提供主動式容錯控制即時切換控制器之目的，使整個攔截系統具有更佳的性能表現。

本計畫中我們已完成了下列事項：

- (1) 推導一般基因演算法則應用於來襲飛彈之彈道軌跡估測。
- (2) 建構適應性基因演算法則以應用於具有高度移動能力之來襲飛彈的彈道軌跡估測。
- (3) 經由估測來襲飛彈之彈道來決定我方攔截飛彈之攔截軌跡。
- (4) 推導彈體座標與地面座標之變換關係以執行攔截軌跡追蹤之任務。
- (5) 利用可變結構控制理論來設計達成攔截軌跡追蹤目的之被動式容錯控制法則。

- (6) 發展控制系統致動器線上即時錯誤偵測與診斷之機制。
- (7) 發展結合錯誤偵測結果之攔截軌跡追蹤主動式容錯控制法則。
- (8) 比較一般(非容錯)設計與兩容錯設計之閉迴路控制系統在致動器正常及異常下之性能表現。
- (9) 針對具有模式誤差或外界干擾之系統進行容錯控制之穩健性分析。
- (10) 進行電腦模擬來驗證所設計之容錯控制的有效性。

本計畫在彈道估測方面，我們混合了純RPTN和GA法則的轉換類型控制法則來減少追蹤能量，如果目標逃逸運動可被準確的預測，則飛彈攔截可達到比較好的效果。而在錯誤偵測與診斷部分，我們針對HAVE DASH II飛彈系統設計一觀察器，目的是希望一旦致動器發生故障時能即時偵測出來，並有效的診斷發生故障的尾翼與故障角度，以提供控制律切換的依據，使系統能夠穩定並達到預期的性能。由理論分析及數值模擬驗證了觀察器所產生的residual signals能夠成功且及時地偵測及診斷出發生故障的致動器，提供系統及時的警告資訊。另外在容錯控制方面，我們利用可變結構控制(VSC)的概念來設計容錯控制器，利用VSC來設計容錯控制器的好處為反應速度快、對系統數學模式不確定性及外界干擾的抑制性能佳，且可大幅減輕傳統容錯控制必須求解由最佳化衍生的Hamilton-Jacobi方程式之計算負擔。由模擬結果可看出無論是主動式或被動式容錯控制器都能有效讓系統達成良好的性能表現，雖然主動式及被動式容錯控制器都能有效讓系統達成良好的性能表現，然而被動式控制必須先設定不容許故障的致動器，在實際應用上較無法有效的應付各種可能的故障狀況。當實際故障的致動器發生在預設的範圍外時，系統往往無法達到預期的性能表現，相對的，由於主動式容錯控制包含有錯誤偵測與診斷之結果，因此，其所需耗費的控制能量及性能表現在系統正常及異常時通常都較被動式容錯控制佳。

參考書目

- [1] K. Argenz, "Proportional Navigation of Nonstationary Targets," *IEEE Transaction on Aerospace and Electronic Systems*, Vol.AES-6, no.4, pp. 455-457, 1970.
- [2] E.Z. Axelband and F.W. Hardy, "Quasi Optimal Proportional Navigation," *IEEE Trans. on Automatic Control*, Vol. AC-15,no.6, pp. 620-626, 1970.
- [3] K.Becker, "Closed-form Soution of Pure Proportional Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol.26,no.3 pp. 526-532, 1990.
- [4] Y.S. Kim,H.S. Cho,and Z. Bien "A New Guidance Law for Homing Missile," *Journal of Guidance, Control, and Dynamics*, Vol.8,no.3 pp. 402-404, 1985.
- [5] J.L. Meriam, *Dynamics*, 2nd Ed.,SI version, Wiley, New York,1975.
- [6] C.D Yang and F.B. Yeh "General Guidance Law for Homing Missile," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 25, no. 2, pp. 197-211, 1989.
- [7] C.D. Yang and C.C. Yang, "Analytical Solution of Three-Dimensional Realistic True Proportional Navigation," *Journal of Guidance, Control, and Dynamics*, Vol. 19, no.3, pp. 569-577 1996.
- [8] P.J. Yuan and J.S. Chern, "Solution of True Proportional Navigation for Maneuvering and Nonmaneuvering Targets," *Journal of Guidance, Control, and Dynamics*, Vol. 15, no.1, pp. 268-271, 1992.
- [9] P.J. Yuan and J.S. Chern, "Ideal Proportional Navigation," *Journal of Guidance, Control, and Dynamics*, Vol. 15, no.5, pp. 1161-1165, 1992.
- [10] J. Huang and C. F. Lin, "Application of Sliding Mode Control to Bank-To-Turn Missile Systems," *Proceedings of the First IEEE Regional Conference on Aerospace Control Systems*, pp. 3569-573, 1993.
- [11] P. Zarchan, "Proportional Navigation and Weaving Targets," *Journal of Guidance, Control, and Dynamics*, Vol. 18, no.5, pp. 969-974, 1995.
- [12] K.A. DeJong, "Adaptive system design: A genetic approach," *IEEE Trans. system Man Cybernet*,SMC-10(9) pp. 566-574, 1980.
- [13] D.E. Goldberg, "Genetic Algorithms ins Search, Optimization, and Machine Learning," *Addison-Wesley, Reading MA*,1989.
- [14] A. Varsek, T. Urbancic, and B. Filipic, "Genetic Algorithms ins controller design and tuning," *IEEE Trans. system Man Cybernet*, 23(5) pp. 1330-1339, 1993.

- [15] D.C. Liaw and J.T. Huang, "Contact Friction Compensation for Robots Using Genetic Learning Algorithms," *Journal of Intelligent and Robotic Systems*,7, pp. 331-349, 1998.
- [16] K. L. Lee and M. A. Langehough "Modern control bank-to-turn autopilot for HAVE DASH II missile" *IEEE*, pp. 214-219, 1992.
- [17] D.C. Liaw,Y.W. Liang,and C.C. Cheng, "Nonlinear Control for Missile Terminal Guidance," *Journal of Dynamic Systems, Measurement, and Control*, 2000.
- [18] H. Holland, "Adaptation in Natural and Artificial System," *Ann Arbor;The University of Michigan Press*, 1975.
- [19] M. Bodson and J.E. Groszkiewicz, "Multivariable adaptive algorithms for reconfigurable flight control," *IEEE Transactions on Control Systems Technology*, Vol.5 no.2 pp. 217-229, 1997.
- [20] I.J. Ha and S. Chong, "Design of a CLOS guidance law via feedback linearization," *IEEE Transactions on Aerospace and Electronics System*, Vol.28 no.1 pp. 51-63, 1992.
- [21] J. Huang and C.F. Lin, "Numerical approach to computing nonlinear control laws," *Journal of guidance, Control, and Dynamics*, Vol.18 no.5 pp. 989-994, 1995.
- [22] , "Design of reliable control systems possessing actuator redundancies," *Journal of Guidance, control, and dynamics*, Vol.23 no.4 pp. 706-710, 2000.
- [23] Y.W. Liang, D.C. Liaw,and T.C. Lee "Reliable Control of Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol.45 no.4 pp. 706-710, 2000.
- [24] F. Liao,J.L. Wang,and G.H. Yang, "Reliable robust flight tracking control: an LMI approach," *IEEE Transactions on Control Systems Technology*, Vol.10 no.1 pp. 76-89, 2002.
- [25] J.H. Oh and I.J. Ha, "Missile autopilot design via function inversion and time-scale transformation," *IEEE Transactions on Aerospace and Electronics System*, Vol.33 no.1 pp. 64-76, 1997.
- [26] R.J. Veillette,J.V. Medanic,and W.R. Perkins, "Design of reliable control systems," *IEEE Transactions on Automatic Control*, Vol.37 pp. 290-304, 1992.
- [27] R.J. Veillette, "Reliable linear-quadratic state-feedback control," *Automatica*, Vol.31 pp. 137-143, 1995.
- [28] G.H. Yang,J.L. Wang,and Y.C. Soh, "Reliable guaranteed cost control for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, Vol.45 no.11 pp. 2188-2192, 2000.

- [29] 遺傳演算法簡介，<http://cindy.cis.nctu.edu.tw/AI96/team12/GA/GA.html>,
- [30] 余國瑞和吳東軒，“應用遺傳演算法與類神經網路於混沌系統之辨識，”義守大學電機工程研究所。

附錄 HAVE DASH II 飛彈系統模擬之 SIMNON 程式

(1)傳統非容錯控制

```
CONTINUOUS SYSTEM normal1
normal control
=====
HAVE DASH II MISSILE MODEL SIMULATION
BY SLIDING MODE CONTROL
OUTPOT~THETA~PSI~PHI~Vm
=====symbol define=====
x1:P x2:Q x3:R x4:fi x5:thita x6:fine x7:Vm x8:afa x9:beta
=====
STATE x1 x2 x3 x4 x5 x6 x8 x9 Vm X Y Z
DER dx1 dx2 dx3 dx4 dx5 dx6 dx8 dx9 dVm dX dY dZ
TIME t
=====
parameter set
=====
pi:3.1415926
pp=-xd*xcdot-yd*ydot-zc*zdot
k1=IF pp>0 THEN 0.14 ELSE IF pp<0 THEN -0.14 ELSE 0
k2=-0.0005
k3=0.0005
xd=x-xc
dxd=dx-xcdot
ddxd=ddx-xcddot
yd=y-yc
dyd=dy-ydot
ddyd=ddy-yddot
zd=z-zc
dzd=dz-zdot
ddzd=ddz-zcddot

dxx11=d7*cos(x8)*cos(x9)*cos(x4)*cos(x5)
dxx12=-d4*Vm*cos(x8)*cos(x9)*sin(x4)*cos(x5)
dxx13=-Vm*d5*cos(x8)*cos(x9)*cos(x4)*sin(x5)
dxx1=dxx11+dxx12+dxx13
dxx2=d7*s9*(c4*s5*s6-s4*c6)
xxx1=-d4*s4*s5*s6+d5*c4*c5*s6+d6*c4*s5*c6-d4*c4*c6+d6*s4*s6
xxxx1=-dx4*s4*s5*s6+dx5*c4*c5*s6+dx6*c4*s5*c6-dx4*c4*c6+dx6*s4*s6
dxx3=Vm*s9*xxx1
dxx4=d7*s8*c9*(c4*s5*c6+s4*s6)
xxx2=-d4*s4*s5*c6+d5*c4*c5*c6-d6*c4*s5*s6+d4*c4*s6+dx6*s4*c6
xxxx2=-dx4*s4*s5*c6+dx5*c4*c5*c6-dx6*c4*s5*s6+dx4*c4*s6+dx6*s4*c6
dxx5=Vm*s8*c9*xxx2

ddx=dxx1+dxx2+dxx3+dxx4+dxx5

dyy1=d7*c8*c9*s4*c5+Vm*d4*c8*c9*c4*c5-d5*Vm*s4*s5*c8*c9
dyy2=d7*s9*(s4*s5*s6+c4*c6)
yyy1=d4*c4*s5*s6+d5*s4*c5*s6+d6*s4*s5*c6-d4*s4*c6-d6*c4*s6
yyyy1=dx4*c4*s5*s6+dx5*s4*c5*s6+dx6*s4*s5*c6-dx4*s4*c6-dx6*c4*s6
dyy3=Vm*s9*yyy1
dyy4=d7*s8*c9*(s4*s5*c6-c4*s6)
yyy2=d4*c4*s5*c6+d5*s4*c5*c6-d6*s4*s5*s6+d4*s4*s6-d6*c4*c6
yyyy2=dx4*c4*s5*c6+dx5*s4*c5*c6-dx6*s4*s5*s6+dx4*s4*s6-dx6*c4*c6
dyy5=Vm*s8*c9*yyy2

ddy=dyy1+dyy2+dyy3+dyy4+dyy5
```

```

dz1=-d7*c8*c9*s5-d5*Vm*c8*c9*c5+d7*s9*c5*s6
dz2=-d5*Vm*s9*s5*s6+d6*Vm*s9*c5*c6+d7*s8*c9*c5*c6
dz3=-d5*Vm*s8*c9*s5*c6-d6*Vm*s8*c9*c5*s6

ddz=dz1+dz2+dz3
=====
Vmc=(x*dot*x*dot+y*dot*y*dot+z*dot*z*dot)^0.5+k1*ee
Vmc*dot=(x*dot*x*dot*dot+y*dot*y*dot*dot+z*dot*z*dot*dot)/Vmc+k1*d*ee
Vmc*dot*dot1=(x*dot*x*dot*dot*dot+x*dot*dot*x*dot+y*dot*y*dot*dot*dot+y*dot*dot*y*dot*dot)
Vmc*dot*dot2=(z*dot*z*dot*dot*dot+z*dot*dot*z*dot*dot)
Vmc*dot*dot=(Vmc*dot*dot1+Vmc*dot*dot2)/Vmc-Vmc*dot*Vmc*dot/Vmc+k1*d*dee
=====
Tracking error
=====
ee=((x-xc)*(x-xc)+(y-yc)*(y-yc)+(z-zc)*(z-zc))^0.5
dee=((x-xc)*(dx-xc*dot)+(y-yc)*(dy-yc*dot)+(z-zc)*(dz-zc*dot))/ee
ddee1=x*d*ddx+dx*d*x-dx*x*dot-d*x*xc-d*x*xc*dot-x*x*dot*dot+x*dot*x*dot+xc*x*dot*dot
ddee2=y*d*ddy+dy*d*y-dy*y*dot-d*y*yc-d*y*yc*dot-y*y*dot*dot+y*dot*y*dot+yc*y*dot*dot
ddee3=x*d*ddz+dz*d*z-dz*z*dot-d*d*zc-d*z*zc*dot-z*z*dot*dot+z*dot*z*dot+zc*z*dot*dot
ddee4=-dee*d*ee/ee
ddee=(ddee1+ddee2+ddee3)/ee+ddee4
=====
Desired trajectory
=====
xc=8000+2620.5*t
yc=2000+462*t
zc=IF t<10 THEN 10245-5*(t-7)*(t-7) ELSE zc1
zc1=IF t>10 THEN 10155+5*(t-13)*(t-13) ELSE 10400

x*dot=2620.5
y*dot=462
z*dot=IF t<10 THEN -5*(2*t-14) ELSE zc1*dot
zc1*dot=IF t>10 THEN 5*(2*t-26) ELSE 0

x*dot*dot=0
y*dot*dot=0
z*dot*dot=IF t<10 THEN -10 ELSE zc1*dot*dot
zc1*dot*dot=IF t>10 THEN 10 ELSE 0

x*dot*dot*dot=0
y*dot*dot*dot=0
z*dot*dot*dot=0
=====
Desired trajectory and system state transformation
=====
yx=(x*dot*x*dot+y*dot*y*dot)^0.5
dyx=(x*dot*x*dot*dot+y*dot*y*dot*dot)/yx
ddyx1=(x*dot*x*dot*dot*dot+x*dot*dot*x*dot*dot+y*dot*y*dot*dot*dot+y*dot*dot*y*dot*dot)/yx
ddyx2=-dyx*(x*dot*x*dot*dot+y*dot*y*dot*dot)/yx/yx
ddyx=ddyx1+ddyx2

x4c=arcsin(y*dot/yx)+k2*yd
x4c*dot=(1/((1-yr*yr)^0.5))*(y*dot*dot/yx-y*dot*d*yx/yx/yx)+k2*d*yd
x4c*dot*dot=ddy*yr/(1-yr*yr)^0.5+d*yr*d*yr*yr*(1-yr*yr)^(-1.5)+k2*d*ddy

yr=y*dot/yx
dyr=y*dot*dot/yx-y*dot*d*yx/yx/yx
ddy1=y*dot*dot*dot/yx-y*dot*dot*d*yx/yx/yx-(y*dot*dot*d*yx+y*dot*dot*ddy)/yx/yx
ddy2=2*y*dot*dot*d*yx/yx/yx/yx
ddy=ddy1+ddy2

zyx=(z*dot/Vmc)
dzyx=z*dot*dot/Vmc-(z*dot*Vmc*dot)/Vmc/Vmc

```

```

ddzyx1=zcdot/Vmc
ddzyx2=-zcdot*Vmc/Vmc/Vmc
ddzyx3=-(Vmc*zcddot+Vmc*zcddot)/Vmc/Vmc
ddzyx4=2*Vmc*Vmc*zcddot/Vmc/Vmc/Vmc
ddzyx=ddzyx1+ddzyx2+ddzyx3+ddzyx4

x5c=x8-arcsin(zyx)+k3*zd
x5cdot=-dzyx/(1-zyx*zyx)^0.5+k3*dzd
x5cddot1=-ddzyx/(1-zyx*zyx)^0.5-zyx*dzyx*dzyx*(1-zyx*zyx)^(-1.5)
x5cddot2=k3*ddzd
x5cddot=x5cddot1+x5cddot2
=====
gx=-ma*g*sin(x5)
gy=ma*g*cos(x5)*sin(x6)
gz=ma*g*cos(x5)*cos(x6)
ax=km*10*Vm*Vm
af=kf*10*Vm*Vm

s4=sin(x4)
c4=cos(x4)
s5=sin(x5)
c5=cos(x5)
s6=sin(x6)
c6=cos(x6)
s8=sin(x8)
c8=cos(x8)
s9=sin(x9)
c9=cos(x9)
=====
Cx0=-0.57+0.0083*x8
Cy0=-0.21*x9
Cz1=-0.0015*(x8*x8*x8)+0.0125*(x8*x8)-0.5052*x8+0.0429
Cz2=0.0006*(x8*x8*x8)-0.0138*(x8*x8)+0.1230*x8-0.0191
Cz0=Cz1+Cz2*(Vm/c)

Fx=af*Cx0
Fy=af*Cy0
Fz=af*Cz0
=====
C10=0.116*x9
Cm1=-0.0055*((x8)^3)+0.2131*((x8)^2)-2.7419*x8-0.0381
Cm2=0.0014*((x8)^3)-0.0623*((x8)^2)+0.8715*x8-0.4041
Cm0=Cm1+Cm2*(Vm/c)
Cn0=0.08*x9
=====
d1=-(Izz-Iyy)/Ixx*x2*x3
d2=-(Ixx-Izz)/Iyy*x1*x3
d3=-(Iyy-Ixx)/Izz*x1*x2
d4=sin(x6)/cos(x5)*x2+cos(x6)/cos(x5)*x3
d5=cos(x6)*x2-sin(x6)*x3
d6=x1+sin(x6)*tan(x5)*x2+cos(x6)*tan(x5)*x3
Vm1=cos(x8)*cos(x9)*(Fx+gx)
Vm2=sin(x9)*(Fy+gy)+sin(x8)*cos(x9)*(Fz+gz)
d7=(Vm1+Vm2)/ma

d81=x2-(x1*cos(x8)+x3*sin(x8))*tan(x9)
d82=(1/(ma*Vm*cos(x9)))*(-sin(x8)*(Fx+gx)+cos(x8)*(Fz+gz))
d8=0*d81+d82
d91=x1*sin(x8)-x3*cos(x8)+1/(ma*Vm)*(-cos(x8)*s9*(Fx+gx))
d92=(1/(ma*Vm))*(cos(x9)*(Fy+gy)-sin(x8)*sin(x9)*(Fz+gz))
d9=0*d91+d92
=====
Cx0dot=0.0083*d8
Cy0dot=-0.21*d9

```

```

Cz1dot=(-0.0045*x8*x8+0.025*x8-0.5052)*d8
Cz2dot=(0.0018*x8*x8-0.0276*x8+0.123)*d8*(Vm/c)
Cz0dot=Cz1dot+Cz2dot
=====
gxdot=-ma*g*cos(x5)*d5
gydot=ma*g*(-sin(x5)*sin(x6)*d5+cos(x5)*cos(x6)*d6)
gzdot=-ma*g*(sin(x5)*cos(x6)*d5+cos(x5)*sin(x6)*d6)
=====
B1=d7

B21=d2*s6/c5+d3*c6/c5+ax*(Cm0/Iyy*c6+Cn0/Izz*c6)/c5
B22=d5*tan(x5)/c5*(x2*s6+x3*c6)+d6/c5*(x2*c6-x3*s6)
B2=B21+B22

B31=d2*c6-d6*x2*s6-d3*s6-d6*x3*c6
B32=ax*(cos(x6)/Iyy*Cm0-sin(x6)/Izz*Cn0)
B3=B31+B32

B41=d1+ax*C10/Ixx+d2*s6*tan(x5)+ax*Cm0/Iyy*s6*tan(x5)
B42=d3*c6*tan(x5)+ax*Cn0/Izz*c6*tan(x5)
B43=d6*(x2*c6*tan(x5)-x3*s6*tan(x5))
B44=d5/c5/c5*(x2*s6+x3*c6)
B4=B41+B42+B43+B44
=====
find A(x) and A(x) inverse
A^-1(x)=[a1 a2 a3;a4 a5 a6;a7 a8 a9];
=====
a11=c8*c9/ma
a12=0
a13=0
a14=0

a21=-k2*(yy1)*c8*c9/ma
a22=-ax*0.675*s6/c5/Iyy
a23=0
a24=c6/c5/Izz*(-0.584)*ax

a311=-k3*(-c8*c9*s5+s9*c5*s6+s8*c9*c5*c6)
xx1=(c8*c9*c4*c5+s9*(c4*s5*s6-s4*c6)+s8*c9*(c4*s5*c6+s4*s6))
yy1=(c8*c9*s4*c5+s9*(s4*s5*s6+c4*c6)+s8*c9*(s4*s5*c6-c4*s6))
zz1=(-c8*c9*s5+s9*c5*s6+s8*c9*c5*c6)
a312=((xx1*(x-xc)+yy1*(y-yc)+zz1*(z-zc))/ee)*zcdot/Vmc/Vmc
a313=a311-k1*a312/(1-zyx*zyx)^0.5
a31=a313*c8*c9/ma
a32=-0.675*ax*c6/Iyy
a33=0
a34=0.584*ax*s6/Izz

a41=0
a42=-0.675*ax*s6*tan(x5)/Iyy
a43=-0.127*ax/Ixx
a44=-0.584*ax*c6*tan(x5)/Izz
=====
find (A)^-1
=====
delta=-a11*a43*(a22*a34-a24*a32)

g11=(-a43*(a22*a34-a24*a32))/delta
g12=0
g13=0
g14=0

```

```

g21=-(a43*(a21*a34-a24*a31))/delta
g22=(a11*(-a34*a43))/delta
g23=-(a11*(-a24*a43))/delta
g24=0

g31=(a21*(a32*a44-a42*a34)-a31*(a22*a44-a24*a42))/delta
g32=-(a11*(a32*a44-a42*a34))/delta
g33=(a11*(a22*a44-a24*a42))/delta
g34=-(a11*(a22*a34-a24*a32))/delta

g41=-(a43*(a21*a32-a22*a31))/delta
g42=(a43*(a11*a32))/delta
g43=-(a43*(a11*a22))/delta
g44=0
=====
Choose Sliding surface
=====
e1=Vm-Vmc
e2=x4-x4c
e3=x5-x5c
e4=x6

e2dot=d4-x4c*dot
e3dot=d5-x5c*dot
e4dot=d6

s1=e1
s2=e2dot+k11*e2
s3=e3dot+k21*e3
ss4=e4dot+k31*e4

s11=-Vm*dot
s22=k11*e2dot-x4c*dot
s33=k21*e3dot-x5c*dot
s44=k31*d6
=====
calculate control law
=====
sat1=IF abs(s1/eps1)>1 THEN sign(s1/eps1) ELSE s1/eps1
sat2=IF abs(s2/eps2)>1 THEN sign(s2/eps2) ELSE s2/eps2
sat3=IF abs(s3/eps3)>1 THEN sign(s3/eps3) ELSE s3/eps3
sat4=IF abs(ss4/eps4)>1 THEN sign(ss4/eps4) ELSE ss4/eps4

t1=-s11-B1
t2=-s22-B2
t3=-s33-B3
t4=-s44-B4
=====actual control law delta_e delta_a delta_r
uleq=g11*t1+g12*t2+g13*t3+g14*t4
u2eq=g21*t1+g22*t2+g23*t3+g24*t4
u3eq=g31*t1+g32*t2+g33*t3+g34*t4
u4eq=g41*t1+g42*t2+g43*t3+g44*t4

tt1=-(0+eta1)*sat1
tt2=-(0+eta2)*sat2
tt3=-(0+eta3)*sat3
tt4=-(0+eta4)*sat4

ulre=g11*tt1+g12*tt2+g13*tt3+g14*tt4
u2re=g21*tt1+g22*tt2+g23*tt3+g24*tt4
u3re=g31*tt1+g32*tt2+g33*tt3+g34*tt4
u4re=g41*tt1+g42*tt2+g43*tt3+g44*tt4
=====
ull1=uleq+ulre

```

```

u222=u2eq+u2re
u333=u3eq+u3re
u444=u4eq+u4re

u1=u11
u2222=u22-u33+u44
u2=IF t<ts THEN u2222 ELSE -2
u3=-u22-u33+u44
u4=-u22-u33-u44
u5=u22-u33-u44

u11=IF u111<0 THEN 0 ELSE IF u111>600 THEN 600 ELSE u111
u22=IF u222>30 THEN 30 ELSE IF u222<-30 THEN -30 ELSE u222
u33=IF u333>30 THEN 30 ELSE IF u333<-30 THEN -30 ELSE u333
u44=IF u444>30 THEN 30 ELSE IF u444<-30 THEN -30 ELSE u444
=====
C1=C10-0.127*(0.25*(-u2-u3-u4-u5))
Cm=Cm0-0.675*(0.25*(u2-u3-u4+u5))
Cn=Cn0-0.584*(0.25*(u2+u3-u4-u5))

L=ax*C1
M=ax*Cm
N=ax*Cn
=====
Vm11=cos(x8)*cos(x9)*(Fx+gx+u1)
Vm22=sin(x9)*(Fy+gy)+sin(x8)*cos(x9)*(Fz+gz)
dx1=- (Izz-Iyy)/Ixx*x2*x3+L/Ixx
dx2=- (Ixx-Izz)/Iyy*x1*x3+M/Iyy
dx3=- (Iyy-Ixx)/Izz*x1*x2+N/Izz
dx4=sin(x6)/cos(x5)*x2+cos(x6)/cos(x5)*x3
dx5=cos(x6)*x2-sin(x6)*x3
dx6=x1+sin(x6)*tan(x5)*x2+cos(x6)*tan(x5)*x3
dVm=(Vm11+Vm22)/ma
dx8=d8
dx9=d9
=====
State on 3-D space
=====
dX11=Vm*c8*c9*c4*c5
dX12=Vm*s9*(c4*s5*s6-s4*c6)
dX13=Vm*s8*c9*(c4*s5*c6+s4*s6)
dX=dX11+dX12+dX13

dY11=Vm*c8*c9*s4*c5
dY12=Vm*s9*(s4*s5*s6+c4*c6)
dY13=Vm*s8*c9*(s4*s5*c6-c4*s6)
dY=dY11+dY12+dY13

dZ11=-Vm*c8*c9*s5+Vm*s9*c5*s6
dZ12=Vm*s8*c9*c5*c6
dZ=dZ11+dZ12
=====
true value (degree)
i1=x1*180/pi
i2=x2*180/pi
i3=x3*180/pi
i4=x4*180/pi
i5=x5*180/pi
i6=x6*180/pi
i8=x8*180/pi
i9=x9*180/pi
i4c=x4c*180/pi
i5c=x5c*180/pi
=====

```

ts:0.3
g:32.174
ma:9.89
c:968
kf:0.1534
km:0.0959
lo:0.0005124
lxx:1.1913
lyy:100.51
lzz:100.57

k11:50
k21:40
k31:18

eta1:15
eta2:10
eta3:1
eta4:1

eps1=1*pi/180
eps2=1*pi/180
eps3=1*pi/180
eps4=1*pi/180

=====

END

(2)被動式容錯控制

```
CONTINUOUS SYSTEM mpass1
passive control of u2(delta_1 outage)
=====
HAVE DASH II MISSILE MODEL SIMULATION
BY SLIDING MODE CONTROL
OUTPUT Vm PSI THETA PHI
=====symbol define=====
x1:P x2:Q x3:R x4:fi x5:thita x6:fine x7:Vm x8:afa x9:beta
=====
STATE x1 x2 x3 x4 x5 x6 x8 x9 Vm X Y Z
DER dx1 dx2 dx3 dx4 dx5 dx6 dx8 dx9 dVm dX dY dZ
TIME t
=====
parameter set
=====
pi:3.1415926

xd=x-xc
dx-dx-xcdot
ddxd=ddx-xcddot

yd=y-yc
dy-dy-ycdot
ddy-dy-ycddot

zd=z-zc
dz-dz-zcdot
ddzd=ddz-zcddot

dxx11=d7*cos(x8)*cos(x9)*cos(x4)*cos(x5)
dxx12=-d4*Vm*cos(x8)*cos(x9)*sin(x4)*cos(x5)
dxx13=-Vm*d5*cos(x8)*cos(x9)*cos(x4)*sin(x5)
dxx1=dxx11+dxx12+dxx13
dxx2=d7*s9*(c4*s5*s6-s4*c6)
xxx1=-d4*s4*s5*s6+d5*c4*c5*s6+d6*c4*s5*c6-d4*c4*c6+d6*s4*s6
xxxx1=-dx4*s4*s5*s6+dx5*c4*c5*s6+dx6*c4*s5*c6-dx4*c4*c6+dx6*s4*s6
dxx3=Vm*s9*xxx1
dxx4=d7*s8*c9*(c4*s5*c6+s4*s6)
xxx2=-d4*s4*s5*c6+d5*c4*c5*c6-d6*c4*s5*s6+d4*c4*s6+dx6*s4*c6
xxxx2=-dx4*s4*s5*c6+dx5*c4*c5*c6-dx6*c4*s5*s6+dx4*c4*s6+dx6*s4*c6
dxx5=Vm*s8*c9*xxx2

ddx=dxx1+dxx2+dxx3+dxx4+dxx5

dyy1=d7*c8*c9*s4*c5+Vm*d4*c8*c9*c4*c5-d5*Vm*s4*s5*c8*c9
dyy2=d7*s9*(s4*s5*s6+c4*c6)
yyy1=d4*c4*s5*s6+d5*s4*c5*s6+d6*s4*s5*c6-d4*s4*c6-d6*c4*s6
yyyy1=dx4*c4*s5*s6+dx5*s4*c5*s6+dx6*s4*s5*c6-dx4*s4*c6-dx6*c4*s6
dyy3=Vm*s9*yyy1
dyy4=d7*s8*c9*(s4*s5*c6-c4*s6)
yyy2=d4*c4*s5*c6+d5*s4*c5*c6-d6*s4*s5*s6+d4*s4*s6-d6*c4*c6
yyyy2=dx4*c4*s5*c6+dx5*s4*c5*c6-dx6*s4*s5*s6+dx4*s4*s6-dx6*c4*c6
dyy5=Vm*s8*c9*yyy2

ddy=dyy1+dyy2+dyy3+dyy4+dyy5

dz1=-d7*c8*c9*s5-d5*Vm*c8*c9*c5+d7*s9*c5*s6
dz2=-d5*Vm*s9*s5*s6+d6*Vm*s9*c5*c6+d7*s8*c9*c5*c6
dz3=-d5*Vm*s8*c9*s5*c6-d6*Vm*s8*c9*c5*s6

ddz=dz1+dz2+dz3
```



```

=====
Tracking error
=====
ee=((x-xc)*(x-xc)+(y-yc)*(y-yc)+(z-zc)*(z-zc))^(0.5)
dee=((x-xc)*(dx-xcdot)+(y-yc)*(dy-ycdot)+(z-zc)*(dz-zcdot))/ee
ddee1=x*ddx+dx*dx-dx*xcdot-ddx*xc-dx*xcdot-x*xcddot+xcdot*xcdot+xc*xcddot
ddee2=y*ddy+dy*dy-dy*ydot-ddy*yc-dy*ydot-y*yddot+ycdot*ydot+yc*yddot
ddee3=x*ddz+dz*dz-dz*zcdot-ddz*zc-dz*zcdot-z*zddot+zcdot*zdot+zc*zddot
ddee4=-dee*dee/ee
ddee=(ddee1+ddee2+ddee3)/ee+ddee4
=====
desired trajectory
=====
pp=-xd*xcdot-yd*ydot-zc*zcdot

k1=IF pp>0 THEN 0.14 ELSE IF pp<0 THEN -0.14 ELSE 0
k2=-0.0005
k3=0.0005

xc=8000+2620.5*t
yc=2000+462*t
zc=IF t<10 THEN 10245-5*(t-7)*(t-7) ELSE zc1
zc1=IF t>10 THEN 10155+5*(t-13)*(t-13) ELSE 10400

xcdot=2620.5
ycdot=462
zcdot=IF t<10 THEN -5*(2*t-14) ELSE zc1dot
zc1dot=IF t>10 THEN 5*(2*t-26) ELSE 0

xcddot=0
ycddot=0
zcddot=IF t<10 THEN -10 ELSE zc1ddot
zc1ddot=IF t>10 THEN 10 ELSE 0

xcddd=0
ycddd=0
zcddd=0

Vmc=(xcdot*xcdot+ycdot*ydot+zcdot*zcdot)^0.5+k1*ee
Vmcdot=(xcdot*xcddot+ycdot*yddot+zcdot*zddot)/Vmc+k1*dee
Vmcddot1=(xcdot*xcddd+xcddot*xcdot+ycdot*ydodd+ycddot*yddot)
Vmcddot2=(zcdot*zcddd+zcddot*zcdot)

Vmcddot=(Vmcddot1+Vmcddot2)/Vmc-Vmcdot*Vmcdot/Vmc+k1*ddee

x4c=arcsin(ycdot/yx)+k2*yd
x4cdot=(1/((1-yr*yr)^0.5))*(ycdot/yx-ydot*dyx/yx/yx)+k2*dyd
x4cddot=ddy/(1-yr*yr)^0.5+dyr*dyr*yr*(1-yr*yr)^(-1.5)+k2*ddy

x5c=x8-arcsin(zyx)+k3*zd
x5cdot=-dzyx/(1-zyx*zyx)^0.5+k3*dzd
x5cddot1=-ddzyx/(1-zyx*zyx)^0.5-zyx*dzyx*dzyx*(1-zyx*zyx)^(-1.5)
x5cddot2=k3*ddzd
x5cddot=x5cddot1+x5cddot2
=====
parameters of Vmc x4c x5c
=====
yx=(xcdot*xcdot+ycdot*ydot)^0.5
dyx=(xcdot*xcddot+ycdot*yddot)/yx
ddyx1=(xcdot*xcddd+xcddot*xcdot+ycdot*ydodd+ycddot*yddot)/yx
ddyx2=-dyx*(xcdot*xcddot+ycdot*yddot)/yx/yx
ddyx=ddyx1+ddyx2

```

```

yr=ycdot/yx
dyr=ycdotdot/yx-ycdot*dyx/yx/yx
ddyr1=ycdotddot/yx-ycdotdot*dyx/yx/yx-(ycdotdot*dyx+ycdot*ddyx)/yx/yx
ddyr2=2*ycdot*dyx*dyx/yx/yx/yx
ddyr=ddyr1+ddyr2

zyx=(zcdot/Vmc)
dzyx=zcdotdot/Vmc-(zcdot*Vmcdot)/Vmc/Vmc
ddzyx1=zcdotddot/Vmc
ddzyx2=-zcdotdot*Vmcdot/Vmc/Vmc
ddzyx3=-(Vmcdot*zcdotdot+Vmcddot*zcdot)/Vmc/Vmc
ddzyx4=2*Vmcdot*Vmcdot*zcdot/Vmc/Vmc/Vmc
ddzyx=ddzyx1+ddzyx2+ddzyx3+ddzyx4
=====
gx=-ma*g*sin(x5)
gy=ma*g*cos(x5)*sin(x6)
gz=ma*g*cos(x5)*cos(x6)
ax=km*1o*Vm*Vm
af=kf*1o*Vm*Vm

s4=sin(x4)
c4=cos(x4)
s5=sin(x5)
c5=cos(x5)
s6=sin(x6)
c6=cos(x6)
s8=sin(x8)
c8=cos(x8)
s9=sin(x9)
c9=cos(x9)
=====
Cx0=-0.57+0.0083*x8
Cy0=-0.21*x9
Cz1=-0.0015*(x8*x8*x8)+0.0125*(x8*x8)-0.5052*x8+0.0429
Cz2=0.0006*(x8*x8*x8)-0.0138*(x8*x8)+0.1230*x8-0.0191
Cz0=Cz1+Cz2*(Vm/c)

Fx=af*Cx0
Fy=af*Cy0
Fz=af*Cz0
=====
C10=0.116*x9
Cm1=-0.0055*((x8)^3)+0.2131*((x8)^2)-2.7419*x8-0.0381
Cm2=0.0014*((x8)^3)-0.0623*((x8)^2)+0.8715*x8-0.4041
Cm0=Cm1+Cm2*(Vm/c)
Cn0=0.08*x9
=====
d1=-(Izz-Iyy)/Ixx*x2*x3
d2=-(Ixx-Izz)/Iyy*x1*x3
d3=-(Iyy-Ixx)/Izz*x1*x2
d4=sin(x6)/cos(x5)*x2+cos(x6)/cos(x5)*x3
d5=cos(x6)*x2-sin(x6)*x3
d6=x1+sin(x6)*tan(x5)*x2+cos(x6)*tan(x5)*x3
Vm1=cos(x8)*cos(x9)*(Fx+gx)
Vm2=sin(x9)*(Fy+gy)+sin(x8)*cos(x9)*(Fz+gz)
d7=(Vm1+Vm2)/ma
d81=x2-(x1*cos(x8)+x3*sin(x8))*tan(x9)
d82=(1/(ma*Vm*cos(x9)))*(-sin(x8)*(Fx+gx)+cos(x8)*(Fz+gz))
d8=0" d81+d82
d91=x1*sin(x8)-x3*cos(x8)+1/(ma*Vm)*(-cos(x8)*s9*(Fx+gx))
d92=(1/(ma*Vm))*(cos(x9)*(Fy+gy)-sin(x8)*sin(x9)*(Fz+gz))
d9=0" d91+d92
=====
Cx0dot=0.0083*d8

```

```

Cy0dot=-0.21*d9
Cz1dot=(-0.0045*x8*x8+0.025*x8-0.5052)*d8
Cz2dot=(0.0018*x8*x8-0.0276*x8+0.123)*d8*(Vm/c)
Cz0dot=Cz1dot+Cz2dot
=====
gxdot=-ma*g*cos(x5)*d5
gydot=ma*g*(-sin(x5)*sin(x6)*d5+cos(x5)*cos(x6)*d6)
gzdot=-ma*g*(sin(x5)*cos(x6)*d5+cos(x5)*sin(x6)*d6)
=====
B1=d7

B21=d2*s6/c5+d3*c6/c5+ax*(Cm0/Iyy*c6+Cn0/Izz*c6)/c5
B22=d5*tan(x5)/c5*(x2*s6+x3*c6)+d6/c5*(x2*c6-x3*s6)
B2=B21+B22

B31=d2*c6-d6*x2*s6-d3*s6-d6*x3*c6
B32=ax*(cos(x6)/Iyy*Cm0-sin(x6)/Izz*Cn0)
B3=B31+B32

B41=d1+ax*C10/Ixx+d2*s6*tan(x5)+ax*Cm0/Iyy*s6*tan(x5)
B42=d3*c6*tan(x5)+ax*Cn0/Izz*c6*tan(x5)
B43=d6*(x2*c6*tan(x5)-x3*s6*tan(x5))
B44=d5/c5/c5*(x2*s6+x3*c6)
B4=B41+B42+B43+B44
"=====
" find A(x) for Tx&delta_e, a, r and H(x) for Tx&delta_1, 2, 3, 4
"A=[a11 a12 a13 a14      H=[h11 h12 h13 h14 h15
"   a21 a22 a23 a24      h21 h22 h23 h24 h25
"   a31 a32 a33 a34      h31 h32 h33 h34 h35
"   a41 a42 a43 a44]     h41 h42 h43 h44 h45]
"=====
a11=c8*c9/ma
a12=0
a13=0
a14=0

a21=-k2*(yy1)*c8*c9/ma
a22=-ax*0.675*s6/c5/Iyy
a23=0
a24=c6/c5/Izz*(-0.584)*ax

a311=-k3*(-c8*c9*s5+s9*c5*s6+s8*c9*c5*c6)
xx1=(c8*c9*c4*c5+s9*(c4*s5*s6-s4*c6)+s8*c9*(c4*s5*c6+s4*s6))
yy1=(c8*c9*s4*c5+s9*(s4*s5*s6+c4*c6)+s8*c9*(s4*s5*c6-c4*s6))
zz1=(-c8*c9*s5+s9*c5*s6+s8*c9*c5*c6)
a312=((xx1*(x-xc)+yy1*(y-yc)+zz1*(z-zc))/ee)*zcdot/Vmc/Vmc
a313=a311-k1*a312/(1-zyx*zyx)^0.5
a31=a313*c8*c9/ma
a32=-0.675*ax*c6/Iyy
a33=0
a34=0.584*ax*s6/Izz

a41=0
a42=-0.675*ax*s6*tan(x5)/Iyy
a43=-0.127*ax/Ixx
a44=-0.584*ax*c6*tan(x5)/Izz

h11=a11
h12=a12
h13=a13
h14=a14
h15=0

h21=a21

```

```

h22=0.25*(a22-a23+a24)
h23=0.25*(-a22-a23+a24)
h24=0.25*(-a22-a23-a24)
h25=0.25*(a22-a23-a24)

h31=a31
h32=0.25*(a32-a33+a34)
h33=0.25*(-a32-a33+a34)
h34=0.25*(-a32-a33-a34)
h35=0.25*(a32-a33-a34)

h41=a41
h42=0.25*(a42-a43+a44)
h43=0.25*(-a42-a43+a44)
h44=0.25*(-a42-a43-a44)
h45=0.25*(a42-a43-a44)
"=====
" find H^-1 for u2(delta_1)outage
"=====
delta1=h11*(h23*h34*h45+h25*h33*h44+h24*h43*h35)
delta2=h11*(-h43*h34*h25-h23*h44*h35-h33*h24*h45)
delta=delta1+delta2

g11=1/h11
g12=0
g13=0
g14=0

g21=-(h21*(h34*h45-h35*h44)-h31*(h24*h45-h25*h44))/delta
g22=h11*(h34*h45-h35*h44)/delta
g23=-h11*(h45*h24-h25*h44)/delta
g24=h11*(h24*h35-h25*h34)/delta

g31=(h21*(h33*h45-h35*h43)-h31*(h23*h45-h43*h25))/delta
g32=-h11*(h33*h45-h35*h43)/delta
g33=h11*(h23*h45-h25*h43)/delta
g34=-h11*(h23*h35-h25*h33)/delta

g41=-(h21*(h33*h44-h34*h43)-h31*(h23*h44-h24*h43))/delta
g42=h11*(h33*h44-h34*h43)/delta
g43=-h11*(h23*h44-h24*h43)/delta
g44=h11*(h23*h34-h24*h33)/delta
"=====
" choose silding surface
"=====
e1=Vm-Vmc
e2=x4-x4c
e3=x5-x5c
e4=x6

e2dot=d4-x4cdot
e3dot=d5-x5cdot
e4dot=d6

s1=e1
s2=e2dot+k11*e2
s3=e3dot+k21*e3
ss4=e4dot+k31*e4

s11=-Vmcdot
s22=k11*e2dot-x4cddot
s33=k21*e3dot-x5cddot
s44=k31*d6
ss5=s1*h12+s2*h22+s3*h32+ss4*h42

```

```

" =====
" calculate control law
" =====
sat1=IF abs(s1/eps1)>1 THEN sign(s1/eps1) ELSE s1/eps1
sat2=IF abs(s2/eps2)>1 THEN sign(s2/eps2) ELSE s2/eps2
sat3=IF abs(s3/eps3)>1 THEN sign(s3/eps3) ELSE s3/eps3
sat4=IF abs(ss4/eps4)>1 THEN sign(ss4/eps4) ELSE ss4/eps4
sat5=IF abs(ss5/eps5)>1 THEN sign(ss5/eps5) ELSE ss5/eps5

t1=-s11-B1
t2=-s22-B2
t3=-s33-B3
t4=-s44-B4
" =====actual control law delta_e delta_a delta_r
uleq=g11*t1+g12*t2+g13*t3+g14*t4
u2eq=0
u3eq=g21*t1+g22*t2+g23*t3+g24*t4
u4eq=g31*t1+g32*t2+g33*t3+g34*t4
u5eq=g41*t1+g42*t2+g43*t3+g44*t4

con:30
kc1=abs(con*h12)
kc2=abs(con*h22)
kc3=abs(con*h32)
kc4=abs(con*h42)

tt1=-(kc1+eta1)*sat1
tt2=-(kc2+eta2)*sat2
tt3=-(kc3+eta3)*sat3
tt4=-(kc4+eta4)*sat4

ulre=g11*tt1+g12*tt2+g13*tt3+g14*tt4
u2re=0
u3re=g21*tt1+g22*tt2+g23*tt3+g24*tt4
u4re=g31*tt1+g32*tt2+g33*tt3+g34*tt4
u5re=g41*tt1+g42*tt2+g43*tt3+g44*tt4
" =====
u11=u1eq+ulre
u22=-ss5*sat5
trob=IF t<ts THEN 1 ELSE (-2)/(-ss5*sat5)
u33=u3eq+u3re
u44=u4eq+u4re
u55=0" u5eq+u5re

u1=IF u11<0 THEN 0 ELSE u11
u2=IF u22>30 THEN 30 ELSE IF u22<-30 THEN -30 ELSE u22
u3=IF u33>30 THEN 30 ELSE IF u33<-30 THEN -30 ELSE u33
u4=IF u44>30 THEN 30 ELSE IF u44<-30 THEN -30 ELSE u44
u5=IF u55>30 THEN 30 ELSE IF u55<-30 THEN -30 ELSE u55
" =====
C1=C10-0.127*(0.25*(-u2-u3-u4-u5))
Cm=Cm0-0.675*(0.25*(u2-u3-u4+u5))
Cn=Cn0-0.584*(0.25*(u2+u3-u4-u5))

L=ax*C1
M=ax*Cm
N=ax*Cn

" =====
Vm11=cos(x8)*cos(x9)*(Fx+gx+u1)
Vm22=sin(x9)*(Fy+gy)+sin(x8)*cos(x9)*(Fz+gz)
dx1=- (Izz-Iyy)/Ixx*x2*x3+L/Ixx

```

```

dx2=- (Ixx-Izz)/Iyy*x1*x3+M/Iyy
dx3=- (Iyy-Ixx)/Izz*x1*x2+N/Izz
dx4=sin(x6)/cos(x5)*x2+cos(x6)/cos(x5)*x3
dx5=cos(x6)*x2-sin(x6)*x3
dx6=x1+sin(x6)*tan(x5)*x2+cos(x6)*tan(x5)*x3
dVm=(Vm11+Vm22)/ma
dx8=d8
dx9=d9

dX11=Vm*c8*c9*c4*c5
dX12=Vm*s9*(c4*s5*s6-s4*c6)
dX13=Vm*s8*c9*(c4*s5*c6+s4*s6)
dX=dX11+dX12+dX13

dY11=Vm*c8*c9*s4*c5
dY12=Vm*s9*(s4*s5*s6+c4*c6)
dY13=Vm*s8*c9*(s4*s5*c6-c4*s6)
dY=dY11+dY12+dY13

dZ11=-Vm*c8*c9*s5+Vm*s9*c5*s6
dZ12=Vm*s8*c9*c5*c6
dZ=dZ11+dZ12
" =====
" true value (degree)
i1=x1*180/pi
i2=x2*180/pi
i3=x3*180/pi
i4=x4*180/pi
i5=x5*180/pi
i6=x6*180/pi
i8=x8*180/pi
i9=x9*180/pi
i4c=x4c*180/pi
i5c=x5c*180/pi
" =====
ts:0.3
g:32.174
ma:9.89
c:968
kf:0.1534
km:0.0959
lo:0.0005124
Ixx:1.1913
Iyy:100.51
Izz:100.57

k11:50
k21:40
k31:18

eta1:15
eta2:10
eta3:1
eta4:1
eta5:1

eps1=1*pi/180
eps2=1*pi/180
eps3=1*pi/180
eps4=1*pi/180
eps5=1*pi/180

" =====
END

```

(3)主動式容錯控制

```
CONTINUOUS SYSTEM nfddl
active control
=====
HAVE DASH II MISSILE MODEL SIMULATION
BY SLIDING MODE CONTROL
OUTPUT Vm PSI THETA PHI
=====symbol define=====
x1:P x2:Q x3:R x4:fi x5:thita x6:fiine x7:Vm x8:afa x9:beta
=====
STATE x1 x2 x3 x4 x5 x6 x8 x9 Vm X Y Z o1 o2 o3 a11 a12 a13
DER dx1 dx2 dx3 dx4 dx5 dx6 dx9 dx8 dVm dX dY dZ do1 do2 do3 da1 da2 da3
TIME t
=====
parameter set
=====
pi=3.1415926
xd=x-xc
dxd=dx-xcdot
ddxd=ddx-xcddot

yd=y-yc
dyd=dy-ycdot
ddyd=ddy-ycddot

zd=z-zc
dzd=dz-zcdot
ddzd=ddz-zcddot

dxx11=d7*cos(x8)*cos(x9)*cos(x4)*cos(x5)
dxx12=-d4*Vm*cos(x8)*cos(x9)*sin(x4)*cos(x5)
dxx13=-Vm*d5*cos(x8)*cos(x9)*cos(x4)*sin(x5)
dxx1=dxx11+dxx12+dxx13
dxx2=d7*s9*(c4*s5*s6-s4*c6)
xxx1=-d4*s4*s5*s6+d5*c4*c5*s6+d6*c4*s5*c6-d4*c4*c6+d6*s4*s6
xxxx1=-dx4*s4*s5*s6+dx5*c4*c5*s6+dx6*c4*s5*c6-dx4*c4*c6+dx6*s4*s6
dxx3=Vm*s9*xxx1
dxx4=d7*s8*c9*(c4*s5*c6+s4*s6)
xxx2=-d4*s4*s5*s6+d5*c4*c5*c6-d6*c4*s5*s6+d4*c4*s6+dx6*s4*c6
xxxx2=-dx4*s4*s5*c6+dx5*c4*c5*c6-dx6*c4*s5*s6+dx4*c4*s6+dx6*s4*c6
dxx5=Vm*s8*c9*xxx2

ddx=dxx1+dxx2+dxx3+dxx4+dxx5

dyy1=d7*c8*c9*s4*c5+Vm*d4*c8*c9*c4*c5-d5*Vm*s4*s5*c8*c9
dyy2=d7*s9*(s4*s5*s6+c4*c6)
yyy1=d4*c4*s5*s6+d5*s4*c5*s6+d6*s4*s5*c6-d4*s4*c6-d6*c4*s6
yyyy1=dx4*c4*s5*s6+dx5*s4*c5*s6+dx6*s4*s5*c6-dx4*s4*c6-dx6*c4*s6
dyy3=Vm*s9*yyy1
dyy4=d7*s8*c9*(s4*s5*c6-c4*s6)
yyy2=d4*c4*s5*c6+d5*s4*c5*c6-d6*s4*s5*s6+d4*s4*s6-d6*c4*c6
yyyy2=dx4*c4*s5*c6+dx5*s4*c5*c6-dx6*s4*s5*s6+dx4*s4*s6-dx6*c4*c6
dyy5=Vm*s8*c9*yyy2

ddy=dyy1+dyy2+dyy3+dyy4+dyy5

dzz1=-d7*c8*c9*s5-d5*Vm*c8*c9*c5+d7*s9*c5*s6
dzz2=-d5*Vm*s9*s5*s6+d6*Vm*s9*c5*c6+d7*s8*c9*c5*c6
dzz3=-d5*Vm*s8*c9*s5*c6-d6*Vm*s8*c9*c5*s6

ddz=dzz1+dzz2+dzz3
```

```

"=====
"Tracking error
"=====
ee=((x-xc)*(x-xc)+(y-yc)*(y-yc)+(z-zc)*(z-zc))^(0.5)
dee=((x-xc)*(dx-xcdot)+(y-yc)*(dy-ycdot)+(z-zc)*(dz-zcdot))/ee
ddee1=x*ddx+dx*dx-dx*xcdot-ddx*xc-dx*xcdot-x*xcddot+xcdot*xcdot+xc*xcddot
ddee2=y*ddy+dy*dy-dy*ydot-ddy*yc-dy*ydot-y*yddot+ycdot*ydot+yc*yddot
ddee3=x*ddz+dz*dz-dz*zcdot-ddz*zc-dz*zcdot-z*zddot+zcdot*zdot+zc*zddot
ddee4=-dee*dee/ee
ddee=(ddee1+ddee2+ddee3)/ee+ddee4
"=====
"desired trajectory
"=====
pp=-xd*xcdot-yd*ydot-zc*zcdot

k1=IF pp>0 THEN 0.14 ELSE IF pp<0 THEN -0.14 ELSE 0
k2=-0.0005
k3=0.0005

xc=8000+2620.5*t
yc=2000+462*t
zc=IF t<10 THEN 10245-5*(t-7)*(t-7) ELSE zc1
zc1=IF t>10 THEN 10155+5*(t-13)*(t-13) ELSE 10400

xcdot=2620.5
ycdot=462
zcdot=IF t<10 THEN -5*(2*t-14) ELSE zc1dot
zc1dot=IF t>10 THEN 5*(2*t-26) ELSE 0

xcddot=0
ycddot=0
zcddot=IF t<10 THEN -10 ELSE zc1ddot
zc1ddot=IF t>10 THEN 10 ELSE 0

xcddd=0
ycddd=0
zcddd=0

Vmc=(xcdot*xcdot+ycdot*ydot+zcdot*zcdot)^0.5+k1*ee
Vmcdot=(xcdot*xcddot+ycdot*yddot+zcdot*zddot)/Vmc+k1*dee
Vmcddot1=(xcdot*xcddd+xcddot*xcdot+ycdot*yddd+ycddot*ydot)
Vmcddot2=(zcdot*zddd+zddot*zcdot)

Vmcddot=(Vmcddot1+Vmcddot2)/Vmc-Vmcdot*Vmcdot/Vmc+k1*ddee

x4c=arcsin(ycdot/yx)+k2*yd
x4cdot=(1/((1-yr*yr)^0.5))*(ycdot/yx-ydot*dyx/yx/yx)+k2*dyd
x4cddot=ddy/(1-yr*yr)^0.5+dyr*dyr*yr*(1-yr*yr)^(-1.5)+k2*ddy

x5c=x8-arcsin(zyx)+k3*zd
x5cdot=-dzyx/(1-zyx*zyx)^0.5+k3*dzd
x5cddot1=-ddzyx/(1-zyx*zyx)^0.5-zyx*dzyx*dzyx*(1-zyx*zyx)^(-1.5)
x5cddot2=k3*ddzd
x5cddot=x5cddot1+x5cddot2

"=====
"parameters of Vmc x4c x5c
"=====
yx=(xcdot*xcdot+ycdot*ydot)^0.5
dyx=(xcdot*xcddot+ycdot*yddot)/yx
ddyx1=(xcdot*xcddd+xcddot*xcdot+ycdot*yddd+ycddot*ydot)/yx
ddyx2=-dyx*(xcdot*xcddot+ycdot*yddot)/yx/yx
ddyx=ddyx1+ddyx2

```



```

yr=ycdot/yx
dyr=ycdotdot/yx-ycdot*dyx/yx/yx
ddyr1=ycdotddot/yx-ycdotdot*dyx/yx/yx-(ycdotdot*dyx+ycdot*ddyx)/yx/yx
ddyr2=2*ycdot*dyx*dyx/yx/yx/yx
ddyr=ddyr1+ddyr2

zyx=(zcdot/Vmc)
dzyx=zcdotdot/Vmc-(zcdot*Vmcdot)/Vmc/Vmc
ddzyx1=zcdotddot/Vmc
ddzyx2=-zcdotdot*Vmcdot/Vmc/Vmc
ddzyx3=-(Vmcdot*zcdotdot+Vmcddot*zcdot)/Vmc/Vmc
ddzyx4=2*Vmcdot*Vmcdot*zcdot/Vmc/Vmc/Vmc
ddzyx=ddzyx1+ddzyx2+ddzyx3+ddzyx4
" =====
gx=-ma*g*sin(x5)
gy=ma*g*cos(x5)*sin(x6)
gz=ma*g*cos(x5)*cos(x6)
ax=km*1o*Vm*Vm
af=kf*1o*Vm*Vm

s4=sin(x4)
c4=cos(x4)
s5=sin(x5)
c5=cos(x5)
s6=sin(x6)
c6=cos(x6)
s8=sin(x8)
c8=cos(x8)
s9=sin(x9)
c9=cos(x9)
" =====
Cx0=-0.57+0.0083*x8
Cy0=-0.21*x9
Cz1=-0.0015*(x8*x8*x8)+0.0125*(x8*x8)-0.5052*x8+0.0429
Cz2=0.0006*(x8*x8*x8)-0.0138*(x8*x8)+0.1230*x8-0.0191
Cz0=Cz1+Cz2*(Vm/c)

Fx=af*Cx0
Fy=af*Cy0
Fz=af*Cz0
" =====
C10=0.116*x9
Cm1=-0.0055*((x8)^3)+0.2131*((x8)^2)-2.7419*x8-0.0381
Cm2=0.0014*((x8)^3)-0.0623*((x8)^2)+0.8715*x8-0.4041
Cm0=Cm1+Cm2*(Vm/c)
Cn0=0.08*x9
" =====
d1=-(Izz-Iyy)/Ixx*x2*x3+ax*C10/Ixx
d2=-(Ixx-Izz)/Iyy*x1*x3+ax*Cm0/Iyy
d3=-(Iyy-Ixx)/Izz*x1*x2+ax*Cn0/Izz
d4=sin(x6)/cos(x5)*x2+cos(x6)/cos(x5)*x3
d5=cos(x6)*x2-sin(x6)*x3
d6=x1+sin(x6)*tan(x5)*x2+cos(x6)*tan(x5)*x3
Vm1=cos(x8)*cos(x9)*(Fx+gx)
Vm2=sin(x9)*(Fy+gy)+sin(x8)*cos(x9)*(Fz+gz)
d7=(Vm1+Vm2)/ma
d81=x2-(x1*cos(x8)+x3*sin(x8))*tan(x9)
d82=(1/(ma*Vm*cos(x9)))*(-sin(x8)*(Fx+gx)+cos(x8)*(Fz+gz))
d8=0"d81+d82
d91=x1*sin(x8)-x3*cos(x8)+1/(ma*Vm)*(-cos(x8)*s9*(Fx+gx))
d92=(1/(ma*Vm))*(cos(x9)*(Fy+gy)-sin(x8)*sin(x9)*(Fz+gz))
d9=0"d91+d92
" =====

```

```

Cx0dot=0.0083*d8
Cy0dot=-0.21*d9
Cz1dot=(-0.0045*x8*x8+0.025*x8-0.5052)*d8
Cz2dot=(0.0018*x8*x8-0.0276*x8+0.123)*d8*(Vm/c)
Cz0dot=Cz1dot+Cz2dot
" =====
gxdot=-ma*g*cos(x5)*d5
gydot=ma*g*(-sin(x5)*sin(x6)*d5+cos(x5)*cos(x6)*d6)
gzdot=-ma*g*(sin(x5)*cos(x6)*d5+cos(x5)*sin(x6)*d6)
" =====
B1=d7

B21=d2*s6/c5+d3*c6/c5
B22=d5*tan(x5)/c5*(x2*s6+x3*c6)+d6/c5*(x2*c6-x3*s6)
B2=B21+B22

B3=d2*c6-d6*x2*s6-d3*s6-d6*x3*c6

B41=d1+d2*s6*tan(x5)
B42=d3*c6*tan(x5)
B43=d6*(x2*c6*tan(x5)-x3*s6*tan(x5))
B44=d5/c5/c5*(x2*s6+x3*c6)
B4=B41+B42+B43+B44
" =====
" find A(x) for Tx&delta_e, a, r and H(x) for Tx&delta_1, 2, 3, 4
"A=[a11 a12 a13 a14      H=[h11 h12 h13 h14 h15
"   a21 a22 a23 a24      h21 h22 h23 h24 h25
"   a31 a32 a33 a34      h31 h32 h33 h34 h35
"   a41 a42 a43 a44]     h41 h42 h43 h44 h45]
" =====
a11=c8*c9/ma
a12=0
a13=0
a14=0

a21=-k2*(yy1)*c8*c9/ma
a22=-ax*0.675*s6/c5/Iyy
a23=0
a24=c6/c5/Izz*(-0.584)*ax

a311=-k3*(-c8*c9*s5+s9*c5*s6+s8*c9*c5*c6)
xx1=(c8*c9*c4*c5+s9*(c4*s5*s6-s4*c6)+s8*c9*(c4*s5*c6+s4*s6))
yy1=(c8*c9*s4*c5+s9*(s4*s5*s6+c4*c6)+s8*c9*(s4*s5*c6-c4*s6))
zz1=(-c8*c9*s5+s9*c5*s6+s8*c9*c5*c6)
a312=((xx1*(x-xc)+yy1*(y-yc)+zz1*(z-zc))/ee)*zcdot/Vmc/Vmc
a313=a311-k1*a312/(1-zyx*zyx)^0.5
a31=a313*c8*c9/ma
a32=-0.675*ax*c6/Iyy
a33=0
a34=0.584*ax*s6/Izz

a41=0
a42=-0.675*ax*s6*tan(x5)/Iyy
a43=-0.127*ax/Ixx
a44=-0.584*ax*c6*tan(x5)/Izz
" =====
" find (A)^-1
delta=-a11*a43*(a22*a34-a24*a32)

g11=(-a43*(a22*a34-a24*a32))/delta
g12=0
g13=0
g14=0

```

```

g21=-(a43*(a21*a34-a24*a31))/delta
g22=(a11*(-a34*a43))/delta
g23=-(a11*(-a24*a43))/delta
g24=0

g31=(a21*(a32*a44-a42*a34)-a31*(a22*a44-a24*a42))/delta
g32=-(a11*(a32*a44-a42*a34))/delta
g33=(a11*(a22*a44-a24*a42))/delta
g34=-(a11*(a22*a34-a24*a32))/delta

g41=-(a43*(a21*a32-a22*a31))/delta
g42=(a43*(a11*a32))/delta
g43=-(a43*(a11*a22))/delta
g44=0

h11=a11
h12=a12
h13=a13
h14=a14
h15=0

h21=a21
h22=0.25*(a22-a23+a24)
h23=0.25*(-a22-a23+a24)
h24=0.25*(-a22-a23-a24)
h25=0.25*(a22-a23-a24)

h31=a31
h32=0.25*(a32-a33+a34)
h33=0.25*(-a32-a33+a34)
h34=0.25*(-a32-a33-a34)
h35=0.25*(a32-a33-a34)

h41=a41
h42=0.25*(a42-a43+a44)
h43=0.25*(-a42-a43+a44)
h44=0.25*(-a42-a43-a44)
h45=0.25*(a42-a43-a44)
"=====
" find H^-1 for u2(delta_1)outage
"=====
delta1a=h11*(h23*h34*h45+h25*h33*h44+h24*h43*h35)
delta2a=h11*(-h43*h34*h25-h23*h44*h35-h33*h24*h45)
deltaa=delta1a+delta2a

g11a=1/h11
g12a=0
g13a=0
g14a=0

g21a=-(h21*(h34*h45-h35*h44)-h31*(h24*h45-h25*h44))/deltaa
g22a=h11*(h34*h45-h35*h44)/deltaa
g23a=-h11*(h45*h24-h25*h44)/deltaa
g24a=h11*(h24*h35-h25*h34)/deltaa

g31a=(h21*(h33*h45-h35*h43)-h31*(h23*h45-h43*h25))/deltaa
g32a=-h11*(h33*h45-h35*h43)/deltaa
g33a=h11*(h23*h45-h25*h43)/deltaa
g34a=-h11*(h23*h35-h25*h33)/deltaa

g41a=-(h21*(h33*h44-h34*h43)-h31*(h23*h44-h24*h43))/deltaa
g42a=h11*(h33*h44-h34*h43)/deltaa
g43a=-h11*(h23*h44-h24*h43)/deltaa
g44a=h11*(h23*h34-h24*h33)/deltaa

```

```

" =====
" find H^-1 for u3(delta_2)outage
" =====
delta1b=h11*(h22*h34*h45+h25*h32*h44+h24*h42*h35)
delta2b=h11*(-h42*h34*h25-h22*h44*h35-h32*h24*h45)
delta3b=delta1b+delta2b

g11b=1/h11
g12b=0
g13b=0
g14b=0

g21b=-(h21*(h34*h45-h35*h44)-h31*(h24*h45-h25*h44))/delta3b
g22b=h11*(h34*h45-h35*h44)/delta3b
g23b=-h11*(h45*h24-h25*h44)/delta3b
g24b=h11*(h24*h35-h25*h34)/delta3b

g31b=(h21*(h32*h45-h35*h42)-h31*(h22*h45-h42*h25))/delta3b
g32b=-h11*(h32*h45-h35*h42)/delta3b
g33b=h11*(h22*h45-h25*h42)/delta3b
g34b=-h11*(h22*h35-h25*h32)/delta3b

g41b=-(h21*(h32*h44-h34*h42)-h31*(h22*h44-h24*h42))/delta3b
g42b=h11*(h32*h44-h34*h42)/delta3b
g43b=-h11*(h22*h44-h24*h42)/delta3b
g44b=h11*(h22*h34-h24*h32)/delta3b
" =====
" find H^-1 for u4(delta_3)outage
" =====
delta1c=h11*(h22*h33*h45+h25*h32*h43+h23*h42*h35)
delta2c=h11*(-h42*h33*h25-h22*h43*h35-h32*h23*h45)
delta3c=delta1c+delta2c

g11c=1/h11
g12c=0
g13c=0
g14c=0

g21c=-(h21*(h33*h45-h35*h43)-h31*(h23*h45-h25*h43))/delta3c
g22c=h11*(h33*h45-h35*h43)/delta3c
g23c=-h11*(h45*h23-h25*h43)/delta3c
g24c=h11*(h23*h35-h25*h33)/delta3c

g31c=(h21*(h32*h45-h35*h42)-h31*(h22*h45-h42*h25))/delta3c
g32c=-h11*(h32*h45-h35*h42)/delta3c
g33c=h11*(h22*h45-h25*h42)/delta3c
g34c=-h11*(h22*h35-h25*h32)/delta3c

g41c=-(h21*(h32*h43-h33*h42)-h31*(h22*h43-h23*h42))/delta3c
g42c=h11*(h32*h43-h33*h42)/delta3c
g43c=-h11*(h22*h43-h23*h42)/delta3c
g44c=h11*(h22*h33-h23*h32)/delta3c
" =====
" find H^-1 for u5(delta_4)outage
" =====
delta1d=h11*(h22*h33*h44+h23*h34*h42+h24*h43*h32)
delta2d=h11*(-h42*h33*h24-h22*h43*h34-h32*h23*h44)
delta3d=delta1d+delta2d

g11d=1/h11
g12d=0
g13d=0
g14d=0

```

```

g21d=-(h21*(h33*h44-h34*h43)-h31*(h23*h44-h24*h43))/del tad
g22d=h11*(h33*h44-h34*h43)/del tad
g23d=-h11*(h44*h23-h24*h43)/del tad
g24d=h11*(h23*h34-h24*h33)/del tad

g31d=(h21*(h32*h44-h34*h42)-h31*(h22*h44-h42*h24))/del tad
g32d=-h11*(h32*h44-h34*h42)/del tad
g33d=h11*(h22*h44-h24*h42)/del tad
g34d=-h11*(h22*h34-h24*h32)/del tad

g41d=-(h21*(h32*h43-h33*h42)-h31*(h22*h43-h23*h42))/del tad
g42d=h11*(h32*h43-h33*h42)/del tad
g43d=-h11*(h22*h43-h23*h42)/del tad
g44d=h11*(h22*h33-h23*h32)/del tad
" =====
Choose sliding surface
=====
e1=Vm-Vmc
e2=x4-x4c
e3=x5-x5c
e4=x6

e2dot=d4-x4cdot
e3dot=d5-x5cdot
e4dot=dx6

s1=e1
s2=e2dot+k11*e2
s3=e3dot+k21*e3
ss4=e4dot+k31*e4

s11=-Vmcdot
s22=k11*e2dot-x4cddot
s33=k21*e3dot-x5cddot
s44=k31*dx6
" =====
" calculate control law
" =====
sat1=IF abs(s1/eps1)>1 THEN sign(s1/eps1) ELSE s1/eps1
sat2=IF abs(s2/eps2)>1 THEN sign(s2/eps2) ELSE s2/eps2
sat3=IF abs(s3/eps3)>1 THEN sign(s3/eps3) ELSE s3/eps3
sat4=IF abs(ss4/eps4)>1 THEN sign(ss4/eps4) ELSE ss4/eps4

t1=-s11-B1-(0+eta1)*sat1
t2=-s22-B2-(0+eta2)*sat2
t3=-s33-B3-(0+eta3)*sat3
t4=-s44-B4-(0+eta4)*sat4

t1a=-B1-s11-h12*uo2
t2a=-B2-s22-h22*uo2
t3a=-B3-s33-h32*uo2
t4a=-B4-s44-h42*uo2

t1b=-B1-s11-h13*uo3
t2b=-B2-s22-h23*uo3
t3b=-B3-s33-h33*uo3
t4b=-B4-s44-h43*uo3

t1c=-B1-s11-h14*uo4
t2c=-B2-s22-h24*uo4
t3c=-B3-s33-h34*uo4
t4c=-B4-s44-h44*uo4

t1d=-B1-s11-h15*uo5

```

```

t2d=-B2-s22-h25*uo5
t3d=-B3-s33-h35*uo5
t4d=-B4-s44-h45*uo5
"=====actual control law delta_e delta_a delta_r
u1ln=g11*t1+g12*t2+g13*t3+g14*t4
u22n=g21*t1+g22*t2+g23*t3+g24*t4
u33n=g31*t1+g32*t2+g33*t3+g34*t4
u44n=g41*t1+g42*t2+g43*t3+g44*t4
"=====
"normal
"=====
un1=u1ln
un2=u22n-u33n+u44n
un3=-u22n-u33n+u44n
un4=-u22n-u33n-u44n
un5=u22n-u33n-u44n
"=====
"u2 outage
"=====
u1eqa=g11a*t1a+g12a*t2a+g13a*t3a+g14a*t4a
u3eqa=g21a*t1a+g22a*t2a+g23a*t3a+g24a*t4a
u4eqa=g31a*t1a+g32a*t2a+g33a*t3a+g34a*t4a
u5eqa=g41a*t1a+g42a*t2a+g43a*t3a+g44a*t4a
"=====
"u3 outage
"=====
u1eqb=g11b*t1b+g12b*t2b+g13b*t3b+g14b*t4b
u2eqb=g21b*t1b+g22b*t2b+g23b*t3b+g24b*t4b
u4eqb=g31b*t1b+g32b*t2b+g33b*t3b+g34b*t4b
u5eqb=g41b*t1b+g42b*t2b+g43b*t3b+g44b*t4b
"=====
"u4 outage
"=====
u1eqc=g11c*t1c+g12c*t2c+g13c*t3c+g14c*t4c
u2eqc=g21c*t1c+g22c*t2c+g23c*t3c+g24c*t4c
u3eqc=g31c*t1c+g32c*t2c+g33c*t3c+g34c*t4c
u5eqc=g41c*t1c+g42c*t2c+g43c*t3c+g44c*t4c
"=====
"u5 outage
"=====
u1eqd=g11d*t1d+g12d*t2d+g13d*t3d+g14d*t4d
u2eqd=g21d*t1d+g22d*t2d+g23d*t3d+g24d*t4d
u3eqd=g31d*t1d+g32d*t2d+g33d*t3d+g34d*t4d
u4eqd=g41d*t1d+g42d*t2d+g43d*t3d+g44d*t4d
"=====
con=2
kc1a=abs(con*h12)
kc2a=abs(con*h22)
kc3a=abs(con*h32)
kc4a=abs(con*h42)

kc1b=abs(con*h13)
kc2b=abs(con*h23)
kc3b=abs(con*h33)
kc4b=abs(con*h43)

kc1c=abs(con*h14)
kc2c=abs(con*h24)
kc3c=abs(con*h34)
kc4c=abs(con*h44)

kc1d=abs(con*h15)
kc2d=abs(con*h25)
kc3d=abs(con*h35)

```

```

kc4d=abs(con*h45)

tt1a=-(kc1a+eta1)*sat1
tt2a=-(kc2a+eta2)*sat2
tt3a=-(kc3a+eta3)*sat3
tt4a=-(kc4a+eta4)*sat4

tt1b=-(kc1b+eta1)*sat1
tt2b=-(kc2b+eta2)*sat2
tt3b=-(kc3b+eta3)*sat3
tt4b=-(kc4b+eta4)*sat4

tt1c=-(kc1c+eta1)*sat1
tt2c=-(kc2c+eta2)*sat2
tt3c=-(kc3c+eta3)*sat3
tt4c=-(kc4c+eta4)*sat4

tt1d=-(kc1d+eta1)*sat1
tt2d=-(kc2d+eta2)*sat2
tt3d=-(kc3d+eta3)*sat3
tt4d=-(kc4d+eta4)*sat4

ulrea=g11a*tt1a+g12a*tt2a+g13a*tt3a+g14a*tt4a
u3rea=g21a*tt1a+g22a*tt2a+g23a*tt3a+g24a*tt4a
u4rea=g31a*tt1a+g32a*tt2a+g33a*tt3a+g34a*tt4a
u5rea=g41a*tt1a+g42a*tt2a+g43a*tt3a+g44a*tt4a

ulreb=g11b*tt1b+g12b*tt2b+g13b*tt3b+g14b*tt4b
u2reb=g21b*tt1b+g22b*tt2b+g23b*tt3b+g24b*tt4b
u4reb=g31b*tt1b+g32b*tt2b+g33b*tt3b+g34b*tt4b
u5reb=g41b*tt1b+g42b*tt2b+g43b*tt3b+g44b*tt4b

ulrec=g11c*tt1c+g12c*tt2c+g13c*tt3c+g14c*tt4c
u2rec=g21c*tt1c+g22c*tt2c+g23c*tt3c+g24c*tt4c
u3rec=g31c*tt1c+g32c*tt2c+g33c*tt3c+g34c*tt4c
u5rec=g41c*tt1c+g42c*tt2c+g43c*tt3c+g44c*tt4c

ulred=g11d*tt1d+g12d*tt2d+g13d*tt3d+g14d*tt4d
u2red=g21d*tt1d+g22d*tt2d+g23d*tt3d+g24d*tt4d
u3red=g31d*tt1d+g32d*tt2d+g33d*tt3d+g34d*tt4d
u4red=g41d*tt1d+g42d*tt2d+g43d*tt3d+g44d*tt4d

u11a=u1eqa+ulrea "For u2 outage
u22a=fin
u33a=u3eqa+u3rea
u44a=u4eqa+u4rea
u55a=u5eqa+u5rea

u11b=u1eqb+ulreb "For u3 outage
u22b=u2eqb+u2reb
u33b=fin
u44b=u4eqb+u4reb
u55b=u5eqb+u5reb

u11c=u1eqc+ulrec "For u4 outage
u22c=u2eqc+u2rec
u33c=u3eqc+u3rec
u44c=fin
u55c=u5eqc+u5rec

u11d=u1eqd+ulred "For u5 outage
u22d=u2eqd+u2red
u33d=u3eqd+u3red
u44d=u4eqd+u4red

```

```

u555d=fin
" =====
C1=C10-0.127*(0.25*(-u2-u3-u4-u5))
Cm=Cm0-0.675*(0.25*(u2-u3-u4+u5))
Cn=Cn0-0.584*(0.25*(u2+u3-u4-u5))

L=ax*C1
M=ax*Cm
N=ax*Cn
" =====

gp=1.3096e-6*Vm*Vm
gq=8.2498e-8*Vm*Vm
gr=7.1335e-8*Vm*Vm

Vm11=cos(x8)*cos(x9)*(Fx+gx+u1)
Vm22=sin(x9)*(Fy+gy)+sin(x8)*cos(x9)*(Fz+gz)
dx1=d1+gp*(u2+u3+u4+u5)
dx2=d2+gq*(-u2+u3+u4-u5)
dx3=d3+gr*(-u2-u3+u4+u5)
dx4=sin(x6)/cos(x5)*x2+cos(x6)/cos(x5)*x3
dx5=cos(x6)*x2-sin(x6)*x3
dx6=x1+sin(x6)*tan(x5)*x2+cos(x6)*tan(x5)*x3
dVm=(Vm11+Vm22)/ma
dx8=d8
dx9=d9

dX11=Vm*c8*c9*c4*c5
dX12=Vm*s9*(c4*s5*s6-s4*c6)
dX13=Vm*s8*c9*(c4*s5*c6+s4*s6)
dX=dX11+dX12+dX13

dY11=Vm*c8*c9*s4*c5
dY12=Vm*s9*(s4*s5*s6+c4*c6)
dY13=Vm*s8*c9*(s4*s5*c6-c4*s6)
dY=dY11+dY12+dY13

dZ11=-Vm*c8*c9*s5+Vm*s9*c5*s6
dZ12=Vm*s8*c9*c5*c6
dZ=dZ11+dZ12
" =====
" true value (degree)

i1=x1*180/pi
i2=x2*180/pi
i3=x3*180/pi
i4=x4*180/pi
i5=x5*180/pi
i6=x6*180/pi
i8=x8*180/pi
i9=x9*180/pi
i4c=x4c*180/pi
i5c=x5c*180/pi
" =====
" obsever desired
" =====

mp=3.8195e5/Vm/Vm
mq=0.0606e8/Vm/Vm
mr=0.07009e8/Vm/Vm

z1=mp*x1-mq*x2
z2=mq*x2-mr*x3
z3=mp*x1+mr*x3

```



```

do1=mp*d1-mq*d2+un2+un5+kk1*r1
do2=mq*d2-mr*d3+un3-un5+kk2*r2
do3=mp*d1+mr*d3+un4+un5+kk3*r3

r1=z1-o1
r2=z2-o2
r3=z3-o3

"effective when normal case
uo1=un1
uo2=IF (0.9<CA and CA<1.1) THEN 0 ELSE un2+r1*kk1
uo3=IF (1.9<CA and CA<2.1) THEN 0 ELSE un3+r2*kk2
uo4=IF (2.9<CA and CA<3.1) THEN 0 ELSE un4+r3*kk3
uo5=IF (3.9<CA and CA<4.1) THEN 0 ELSE un5

da1=IF a11>1.1 THEN 0 ELSE IF abs(r1)>bnd_a11 THEN 100 ELSE 0
da2=IF a12>1.1 THEN 0 ELSE IF abs(r2)>bnd_a12 THEN 100 ELSE 0
da3=IF a13>1.1 THEN 0 ELSE IF abs(r3)>bnd_a13 THEN 100 ELSE 0
"=====
" Outage selection
"=====

ulac=un1

"u2ac=un2
u2ac=IF t<ts THEN un2 ELSE fin

u3ac=un3
"u3ac=IF t<ts THEN un3 ELSE fin

u4ac=un4
"u4ac=IF t<ts THEN un4 ELSE fin

u5ac=un5
"u5ac=IF t<ts THEN un5 ELSE fin

u1=IF un1<0 THEN 0 ELSE un1
"=====
CA =IF (a11<1 AND a12<1 AND a13<1) THEN 0 ELSE C_1
C_1=IF (a11>1 AND a12<1 AND a13<1) THEN 1 ELSE C_2
C_2=IF (a11<1 AND a12>1 AND a13<1) THEN 2 ELSE C_3
C_3=IF (a11<1 AND a12<1 AND a13>1) THEN 3 ELSE 4

u2=IF (CA<0.1) THEN u2ac ELSE u2_1
u2_1=IF (0.9<CA and CA<1.1) THEN u222a ELSE u2_2
u2_2=IF (1.9<CA and CA<2.1) THEN u222b ELSE u2_3
u2_3=IF (2.9<CA and CA<3.1) THEN u222c ELSE u2_4
u2_4=IF (3.9<CA and CA<4.1) THEN u222d ELSE 100

u3=IF (CA<0.1) THEN u3ac ELSE u3_1
u3_1=IF (0.9<CA and CA<1.1) THEN u333a ELSE u3_2
u3_2=IF (1.9<CA and CA<2.1) THEN u333b ELSE u3_3
u3_3=IF (2.9<CA and CA<3.1) THEN u333c ELSE u3_4
u3_4=IF (3.9<CA and CA<4.1) THEN u333d ELSE 100

u4=IF (CA<0.1) THEN u4ac ELSE u4_1
u4_1=IF (0.9<CA and CA<1.1) THEN u444a ELSE u4_2
u4_2=IF (1.9<CA and CA<2.1) THEN u444b ELSE u4_3
u4_3=IF (2.9<CA and CA<3.1) THEN u444c ELSE u4_4
u4_4=IF (3.9<CA and CA<4.1) THEN u444d ELSE 100

u5=IF (CA<0.1) THEN u5ac ELSE u5_1
u5_1=IF (0.9<CA and CA<1.1) THEN u555a ELSE u5_2

```

```

u5_2=IF (1.9<CA and CA<2.1) THEN u555b ELSE u5_3
u5_3=IF (2.9<CA and CA<3.1) THEN u555c ELSE u5_4
u5_4=IF (3.9<CA and CA<4.1) THEN u555d ELSE 100

"=====
m1=2
m2=2
m3=2

fin:-2

ts:0.3

g:32.174
ma:9.89
c:968
kf:0.1534
km:0.0959
lo:0.0005124
lxx:1.1913
lyy:100.51
lzz:100.57

k11:50
k21:40
k31:18

etal:15
eta2:1
eta3:1
eta4:1

eps1=1*pi/180
eps2=1*pi/180
eps3=1*pi/180
eps4=1*pi/180

kk1:10          "<==converge rate of observer
kk2:10
kk3:10

bnd_a11=0.051   "<==boundary of alarm
bnd_a12=0.05
bnd_a13=0.05

a1=a11          "<==alarm value output
a2=a12
a3=a13

ss=(s1*s1+s2*s2+s3*s3+ss4*ss4)^0.5
"=====

END

```