

# **A Fault-Tolerant Neural Aided Controller for Aircraft**

## **Auto-landing**

A. A. Pashilkar, N. Sundararajan\* and P. Saratchandran

School of Electrical and Electronic Engineering,

Nanyang Technological University,

Singapore, 639798

\* Corresponding Author :

Phone : 65 – 6790 5027

Fax : 65 – 6792 0415

Email: {Abhay, ensundara, epsarat} @ntu.edu.sg

# **A Fault-Tolerant Neural Aided Controller for Aircraft**

## **Auto-landing**

A. A. Pashilkar, N. Sundararajan and P. Saratchandran

School of Electrical and Electronic Engineering,

Nanyang Technological University,

Singapore, 639798

Email: {Abhay, ensundara, epsarat} @ntu.edu.sg

### ***Abstract***

This paper presents a neural-aided controller that enhances the fault tolerant capabilities of a high performance fighter aircraft during the landing phase when subjected to severe winds and failures such as stuck control surfaces. The controller architecture uses a neural controller aiding an existing conventional controller. The neural controller uses a feedback error learning mechanism and employs a dynamic Radial Basis Function neural network called Extended Minimal Resource Allocating Network (EMRAN), which uses only on-line learning and does not need a priori training. The conventional controller is designed using a classical design approach to achieve the desired autonomous landing profile with tight touchdown dispersions called herein as the pillbox. This design is carried out for no failure conditions but with the aircraft being subjected to winds. The failure scenarios considered in this study are: i. Single faults of either aileron or elevator stuck at certain deflections and ii. Double fault cases where both the aileron and elevator

are stuck at different deflections. Simulation studies indicate that the designed conventional controller has only a limited failure handling ability. However, neural controller augmentation considerably improves the ability to handle large faults and meet the strict touchdown dispersion requirements, thus enlarging the fault-tolerance envelope.

**Keywords:** RBF neural network; EMRAN; fault tolerant; actuator failure; auto landing; flight control;

### ***1. Introduction***

The auto-landing control systems in modern aircraft are designed to give a satisfactory performance under nominal operating conditions and are generally unable to cope with failures such as control surfaces being stuck at certain deflections. However, if one can build some intelligence into the existing auto-landing controllers, they can react quickly to such failures and reconfigure the control system to achieve a safe landing with the desired performance requirements. Neural networks provide a fast mechanism to achieve this because of their ability to learn on-line and adapt the aircraft control systems to the sudden changes in the environment as well as sensor and actuator failures [3, 11]. Napolitano et. al. [11] present the investigation of on-line learning neural controllers in the context of the aircraft autopilot functions and of stability augmentation systems for both longitudinal and lateral directional dynamics. In Johnson and Calise [6] a particular architecture is proposed for the use of neural networks in flight control. It also uses Pseudo Control Hedging (PCH), which is essentially a method of modifying the command signal when the control surface is close to saturation.

An early use of neural networks in auto-landing is given in [7] where the neural network was trained off-line to generate the desired trajectories for landing under wind disturbances and worked in conjunction with a conventional PID landing controller. A feed-forward network neural network, trained off-line is used as an auto-landing controller in [5]. Here, the neural network replaced the original PID controller and similar performance has been observed.

In some of the above work [7, 5], a feed-forward neural network with back propagation learning algorithm has been used. The main drawback of such a scheme is that the neural network requires a priori training on normal and faulty operating data. Also, the size of the neural network needs to be fixed beforehand. An alternate neural network is the Radial Basis Function Network (RBFN) with Gaussian functions, which have good local interpolation and global generalization ability [8, 9 10, 14]. In this paper, we use an on-line learning radial basis function network (RBFN) that decides its size automatically for auto-landing control purposes under failures and we evaluate its fault tolerance capabilities.

A sequential learning RBFN called Minimal Resource Allocation Network (MRAN) has been proposed by Lu Yingwei *et. al* [10]. In this work hidden neurons are added and removed to maintain a compact network. MRAN has been used for several applications varying from function approximation to nonlinear system identification and its application in flight control was reported in [14]. An improvement to MRAN called

Extended MRAN (EMRAN), which increases the speed of the algorithm by updating parameters of only the nearest neuron, has been described in [8].

An auto-landing controller based on MRAN aided  $H_\infty$  controller was proposed in [9] for the aircraft model described in [7]. In this scheme, a simple architecture originating from Kawato's feedback-error-learning scheme (Gomi and Kawato [4]) has been utilized. This control architecture uses a conventional PID/  $H_\infty$  controller in the inner loop to stabilize the system, and the MRAN neuro-controller acts as an aid to the conventional controller. The performance of the neural controller has been evaluated under a microburst and partial loss of control effectiveness and has been found to be better than conventional control schemes.

In this paper, we use a Baseline Trajectory Following Controller (BTFC) designed using conventional methods in the inner loop and the neural controller is used to aid the BTFC during failures. This is also realistic since if one wants to improve the fault-tolerance of existing controllers without a complete redesign, then the neural network controller can be used as an add-on. The aircraft model used here is that of a high performance fighter aircraft [12]. The BTFC controller in the inner loop plays an important role in this strategy. It is not only used to stabilize the overall system, but also provides the signals to train the EMRAN network on-line.

The overall scheme for the neuro-controller is shown in Figure 1. The landing task is autonomous, hence there is a navigation function incorporated in the block called

‘Tracking Command generator’. The output of this block consists of reference commands (labeled as ‘ $\mathbf{r}$ ’ in the Figure), which are input to the BTFC called “Classical Feedback Controller” in the figure. Under normal conditions, the BTFC is designed to cause the aircraft outputs ‘ $\mathbf{y}$ ’ to follow the reference vector ‘ $\mathbf{r}$ ’. The neural controller uses the reference signals and the aircraft outputs to generate its command signal. It also uses the output of the BTFC to learn the inverse dynamics of the plant (in this case the aircraft) as in the feedback error learning scheme [4].

The auto-landing problem studied in this paper consists of a high performance fighter aircraft executing four phases of flight segments consisting of a wing-level flight, a coordinated turn, glide slope descent and finally the flare maneuver. The trajectory segments corresponding to these four phases have to be flown with severe winds and specified trajectory deviations have to be met. The touch condition conditions are given with tight specifications, named for convenience as the *touchdown pillbox* (Table 1). The BTFC controller is first designed to meet all these specifications for all these four phases under no failure conditions of the actuators. We wish to point out that the above trajectory for landing may be more appropriate for an Unmanned Air Vehicle (UAV) and may be severe for an aircraft, nonetheless, we have used it here for the evaluation of the neural controller.

The fault scenario studied consists of control surfaces stuck at different deflections. In this case, they correspond to elevator and aileron surfaces stuck at different deflections either alone or in combination. The occurrence of these faults has been studied for all the

segments; however, the results are given in this paper for the failures that occur at the most critical phase. The critical phase is before the turn and descent maneuvers.

When the above failures are introduced in the landing phase, it was found that the BTFC controller is unable to meet the strict touchdown dispersions (pill box conditions) except for actuator stuck faults for small aileron deflections. When the neural controller was introduced in the control scheme, it was found that these stuck deflections could be made large thereby enhancing the fault tolerance envelope for meeting the strict touch down pillbox conditions for all the failure cases.

The paper is organized as follows. Section 2 deals with the aircraft, actuator model and the landing task (including winds) used in this study. Section 3 describes the Baseline trajectory Following Controller (BTFC) design and its performance with and without severe winds and with no failures. Its performance is then evaluated under single and multiple control surface failures. Section 4 briefly describes the EMRAN algorithm. Section 5 presents the main results of this paper showing the impact of the neural controllers under large failures and severe winds. Section 6 summarizes the conclusions from this study.

## ***2. Auto-Landing Problem***

The aircraft model used in this study is that of a high performance fighter aircraft. Details of the model can be found in [12]. For the purposes of this study, the elevator and aileron control surface aerodynamic data has been split into two parts corresponding to left and right surfaces using CFD computations, similar to [13]. The aerodynamic model also

contains the ground effects. The aircraft has two elevators (-25 to 25 deg deflection), which can be moved together or in differential mode. It also has a pair of ailerons (-20 to 20deg deflection) and a rudder (-30 to 30 deg deflection). The engine model (without dynamics) completes the six-degree of freedom simulation. The aircraft has hydraulic actuators, which drive the primary control surfaces that are modeled as first order lags with a time constant of 50msec. The rate limits for the actuators is set at 60deg/sec. The entire mathematical model of the aircraft and controllers was implemented in *Simulink/Matlab*® environment on a P4 2.4GHz PC.

The landing task to be executed by the aircraft is shown in Figure 2. It comprises six distinct phases:

- Segment 1: level flight at 600m, heading -90deg (from East to West). Velocity is maintained at 83 m/s.
- Segment 2: A coordinated right turn with bank angle 40deg at 600m to align with the runway 0 deg (heading North). Velocity is maintained at 83m/s.
- Segment 3: A level flight at 600m heading 0 deg (towards North). Velocity is maintained at 83m/s.
- Segment 4: Descent on glide-slope of -6deg to altitude of 300m
- Segment 5: Descent at glide-slope of -3deg to 12m.
- Segment 6: Flare and touch down. Velocity is reduced from 83m/s to 79m/s during the flare and sink rate from about 4m/s to practically zero.

For the BTFC design, a linear model is used for the longitudinal and lateral-directional axes separately. This is generated based on the level flight Segment 1. This condition is

adequate for the entire flight envelope because the trajectory does not demand significant changes in angle of attack, sideslip or Mach Number.

The aircraft is subjected to winds while flying this trajectory. The winds are a combination of Dryden turbulence along the x-body axis and deterministic winds in the other two axes (Figure 3). The sharp step changes in the  $v_w$  (at 470m and 190m) and  $w_w$  (150m and 90m) are particularly noticeable. These profiles represent a large horizontal and vertical wind shear respectively.

The success of the landing mission is evaluated based on a successful touchdown in the landing ‘pill box’. The ‘pill box’ conditions are defined in Table 1.

Two types of hard over failures are studied in this paper. The first type is a failure of a single control surface and the second type is the failure of two dissimilar control surfaces. Further, the surface can achieve any of the hard over position from the full range of deflections (-25 to 25 deg for the elevator and -20 to 20 deg for the aileron). In this paper only left elevator and left aileron are considered for study as we expect similar result for the right control surfaces from symmetry considerations. In case of the second category (two surface hard over failures), we have considered left elevator and left aileron failure separate from left elevator and right aileron failures. Again the results for the other two combination failures, namely, right elevator and left aileron, right elevator and right aileron will be similar.

Before we proceed with simulations to assess the fault tolerance, it is important to note that not all possible failures are feasible. This is because some surface failures result in residual moments, which cannot be trimmed out using the remaining healthy surfaces. Rudder failure falls in this category and hence is not considered in this study. To determine the regions in which a single and two failures can be tolerated, the open loop aircraft was trimmed for

- Region of level flight trim ( $p=q=r=\gamma=0$ , 6dof accelerations = 0)
- Region of level descent trim ( $p=q=r=0$ ,  $\gamma=-6\text{deg}$ , 6dof accelerations = 0)
- Region of level turning trim ( $\phi=40\text{deg}$ , 6dof accelerations = 0)

The intersection of these trim regions was used to obtain the *feasible region* from where the faults can be recovered. It is to be noted that this computation is still conservative, since in a realistic situation, the aircraft has to have control authority over and above the trim to maneuver (e.g. for disturbance rejection). There is no margin for maneuvering at the boundaries of these regions. In realistic conditions, the actual boundaries are expected to lie within the feasible region. Furthermore, the trim regions have been computed for no winds and may differ somewhat for the varying wind conditions imposed during the flight.

### ***3. Baseline Trajectory Following Controller (BTFC) Design***

The BTFC is designed using the classical loop shaping SISO design techniques. The trajectory following control design task consists of two parts – a tracking command generator that generates the command signals based on trajectory deviations which is then used by a classical feedback controller as its inputs. In the design of the controller a

worst-case delay of 40msec (twice the controller sample rate) was introduced in the loop in all the simulations and analysis.

The tracking command generator in Figure 1, determines the offset of the aircraft from the desired ground track for each segment of the flight. It computes the reference commands (labeled ' $\mathbf{r}$ ' in Figure 1) consisting of Altitude, Velocity and Cross Distance from the desired track and the angular error of the aircraft velocity vector from the desired track vector. The segments of trajectory are either straight lines or arcs of circles. Thus, the cross distance is the length of the perpendicular in case of line segments. In case of the circular arcs, this quantity is the difference between the distance to the center of the circular arc and the radius of the turn. Similarly, the angular error of the velocity vector can be calculated using the components of the aircraft velocity in the X-Y plane and the direction of the desired trajectory nearest to the aircraft. Once these quantities are known, the velocity and altitude references are obtained by linear interpolation between the beginning and end point values of each segment.

The classical feedback controller processes the four reference signals generated by the tracking command generator in the following way.

- Altitude control is achieved by generating the desired attitude command. The flight path angle error is also used for feedback in order to provide damping to the altitude signal. The inner most loop is the pitch rate loop (Fig. 4).
- Airspeed control is achieved by manipulating the throttle (Fig. 4).

- Tracking of trajectories in the X-Y plane is achieved using heading angle as inner loop and the details can be found in [1]. This loop in turn commands the bank angle loop with the roll rate loop as the innermost loop.
- Sideslip is minimized by commanding the rudder using the estimated sideslip rate given by  $(r - p \cdot \tan \alpha)$  and lateral acceleration as feed back signals.

All the loops have been designed with a minimum gain margin of 6dB and a phase margin of at least 45 deg. Simulation of Landing trajectory is shown in Figure 5 under winds and no failures. Fig. 5 (a) shows the sideslip, altitude and velocity time histories. The numbers at the top of Fig. 5 (a) indicate the different segments of the trajectory:

**1** represents the first straight line segments of level flight, followed by **2** showing the right turn. **3** represents level flight after the turn and so on. Fig. 5 (b) shows the X-Y plot of the trajectory. The simulation was stopped as soon as the landing gear touched the ground. In the figures for velocity, altitude and X-Y, the reference signal is also plotted for comparison. With no failures the controller is able to achieve the desired trajectory and the touch down point meeting strict touch down specifications given in ('pill box') Table 1. There is an initial jump in the velocity at  $t=0$ , because the model was trimmed for straight level flight and the gust shown in Figure 3 was introduced at  $t=0$  resulting in an instantaneous increase in forward velocity. This perturbation died out within 5 secs. Subsequently, the velocity shows a dip in Segment 2 due to the initial loss of lift at the beginning of the turn. Velocity deviation seen at the beginning of the 4<sup>th</sup> and 5<sup>th</sup> segments is due to change in the glide slope. The velocity disturbances at about 120 and 130 sec are due to the appearance of wind shear of magnitude  $-12\text{m/s}$  and  $23\text{m/s}$  ( $-12\text{m/s}$  to  $11\text{m/s}$ ) respectively. Lateral deviation in ground track (Fig. 5 (b)) appears in the X-Y plane

during the turn in the 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> segments, eventually being corrected in the 5<sup>th</sup> segment. There is also a corresponding small but noticeable deviation in sideslip during these phases, which is corrected rapidly. In the 4<sup>th</sup> and 5<sup>th</sup> segments the sideslip shows a large deviation from the desired (zero) value due to the side gusts of magnitude +10m/s and -20m/s (+10m/s to -10m/s) respectively.

For failure studies, a series of simulation were conducted at one to two degree intervals of control surface deflections to obtain the region in which a single and two surface failures can be tolerated. The failures were injected at 10sec for elevator and 8sec for the aileron irrespective whether a single or two failure cases were being simulated. These points are very close to the initiation of turn. Therefore, it was felt that any potential instabilities would be exacerbated due to the turning maneuver. Further, forcing the controller to work through the three maneuvers would be closer to the rationale behind the feasible region computation. This makes up a good basis to compare the simulation results with the feasible region computed by the trim calculations described in the previous section. Based on the above failure simulations, it was found that BTFC has no tolerance to elevator failure under severe winds when the ‘pill box’ conditions are required to be met.

Before describing the neural controller and its performance, a brief description of EMRAN is given in the next section.

#### ***4. Extended Minimum Resource Allocating Network (EMRAN)***

EMRAN is a fast implementation of the sequential learning RBF network of Lu *et. al*

[10] referred to as MRAN. MRAN is a sequential learning algorithm and utilises a compact RBF neural network. A brief description of MRAN and its extension EMRAN is given here. For more details see Li Yan et.al [8].

The outputs of an RBF network with Gaussian function  $\Phi$  are given by

$$f(\xi_n) = a_0 + \sum_{i=1}^h \Phi_i(\xi_n) \quad \xi \in R^m, f \in R^p \quad (1)$$

where,

$$\Phi_i(\xi_n) = \alpha_i \exp\left(-\frac{1}{\sigma_i^2} \|\xi_n - \mu_i\|^2\right)$$

where  $\xi$  is the input vector of the network,  $h$  indicates the total number of hidden neurons,  $\mu_i$ ,  $\sigma_i$  refers to the centre and width of the  $i^{\text{th}}$  hidden neuron and  $n$  is the time index.  $\|\cdot\|$  denotes the Euclidean norm. The function  $f$  is the output of RBF network, which represents the network approximation to the desired output  $y_n$ . In the feedback error learning architecture, the desired output is that of the BTFC ( $u_c$ ) as shown in Figure 1. The coefficient  $\alpha_i$  is the connection weights of the  $i^{\text{th}}$  hidden neuron to the output layer and  $a_0$  is the bias term and both are vectors. The network starts with no hidden neurons. As input data is received sequentially, the network adds or prunes hidden neurons based on certain criteria.

The sequential learning algorithm for MRAN, is summarized as follows [10]:

1. Obtain an input and calculate the network outputs and the corresponding error.
2. Create a new RBF hidden neuron if the following three conditions are satisfied

- The error exceeds a minimum threshold value,
  - The mean square error of the network for a series of past data has been above a certain threshold value, and
  - The new input is sufficiently far from the existing hidden neurons.
3. If condition (2) is not met, adjust the weights and widths of the existing RBF network using the Extended Kalman Filter Algorithm.

As discussed previously, EMRAN differs from the MRAN in this last step [8]. Instead of updating all the parameters (representing weights, centers and widths) of all the hidden neurons, it only updates the parameters of the nearest neuron. There is only slight difference between the performance of EMRAN and MRAN in terms of approximation error but in terms of speed EMRAN outperforms MRAN significantly.

In addition to the above three steps, a pruning strategy is also adopted in MRAN/EMRAN:

1. If a hidden neuron's normalized contribution to the output for a certain number of consecutive inputs is found to be below a threshold value, that hidden neuron is pruned.
2. If two hidden neurons are found to be close to each other, as defined by a threshold value, then they are combined into a single hidden neuron.

The dimensions of the EKF are adjusted and the next input is processed. The pruning strategy ensures that the resulting radial basis network uses minimum number of neurons.

The lower the number of neurons makes the algorithm computationally efficient.

Consider the aircraft dynamics represented by the equations

$$\dot{x} = f(x, u) \tag{2}$$

with  $f$  assumed to be smooth and having bounded first derivatives in the neighborhood of the trajectory. Bugajski and Enns [2] have shown that it is possible to invert the nonlinear dynamic equations for an aircraft and generate a robust nonlinear controller. The inversion process begins with calculation of the control surface deflections based on the desired angular accelerations by inverting the three angular rate equations simultaneously. The desired angular accelerations are computed using the inversion of the force equations. This process eventually gives a controller, which is able to receive, the desired velocity, altitude and heading as reference signals and generate the elevator, aileron, rudder and throttle control signals to track the reference signals. One may represent the inversion by the equations:

$$u = f^{-1}(\dot{x}, x) \tag{3}$$

where,  $f^{-1}$  represents the inversion of the equations of motion. Thus, we must synthesize this set of multivariable functions using the states and their derivatives to obtain the control inputs required to make the aircraft follow the desired trajectory. Further, if this function representing the inverse aircraft dynamics is changing over a period of time, we can exploit the learning ability of the neural network to generate immediate corrective action when such changes take place. In this paper we make the assumption that the aerodynamic control inputs (elevator, aileron and rudder) do not have significant coupling. This means that the elevator primarily generates pitching moments, the aileron contributes the rolling moment and the rudder mainly results in yawing moments. This assumption allows us to postulate two inversion controllers for the aircraft one each for the longitudinal and lateral-directional axes.

With this as the motivation, we have utilized the feedback error learning strategy of Gomi and Kowato [4]. This postulates the classical feedback controller output as the signal to be learned ( $u_c = u_t - u_m$ ). Over a period of time EMRAN learns the total control signal ( $u_m \rightarrow u_t$ ) which results in driving the BTFC control output to zero. This means that EMRAN has generated the inverse of the plant by learning the inverse functions represented by equation (3). There are two such blocks used in the trajectory simulation one each for the longitudinal and lateral-directional axes. As seen in the previous section, the BTFC has very poor failure tolerance compared to the feasible region. If EMRAN learns from the classical controller, there is the possibility that it will not achieve anything better than this controller. To overcome this, we have modified the error signal used by the EMRAN by adding scaled trajectory error signal to the classical controller output to remedy this drawback. Figure 6 shows the scaling factors used for the longitudinal EMRAN block. Similar scheme is also used for the lateral EMRAN block. Taken together, the longitudinal and lateral-directional EMRAN blocks represent the neuro-controller in Figure 1.

##### ***5. Performance Evaluation of BTFC and Neural Aided BTFC***

In this section, results are presented for both no failure and failure cases which show the ability of neural aided controller to meet the tight touchdown specifications while the BTFC is unable to meet this under failures. Simulations were conducted at one to two degree intervals of surface deflections for both single and double control surface failures

of actuators similar to the BTFC simulation. Figures 7-13 present the results for these cases that are analyzed in detail below.

### **5.1 No Surface Failure :**

First the neural aided BTFC controller was flown for the auto landing task with severe winds and no control surface failures. Figure 7a shows the lateral position (Y), sideslip angle (beta), altitude and the velocity during the landing phase for both the BTFC and the neural aided BTFC for this case. The control signals from the neural aided BTFC are shown in Fig. 7b. It is seen that the trajectory following is practically identical for the two cases, except for some differences in the velocity. The neural network for the elevator gives a correction which is opposite in sign from that of the BTFC. The aileron control signal from the neural network shows comparatively less learning. These responses are due to the small magnitudes of the control signals involved for this landing task under no failure conditions. However, it is seen that the longitudinal and lateral-directional neural networks are stable.

### **5.2 Single surface failure**

In this scenario the aircraft was subjected to a single failure of either elevator or aileron. Figure 8a shows the lateral position (Y), sideslip angle (beta), altitude and the velocity during the landing phase for both the BTFC and the neural aided BTFC for the failure case of left elevator stuck at  $-10$  degrees at 10sec.

Figure 8b shows the elevator and aileron deflections along with the control signals from the BTFC and EMRAN and also the neuron history for EMRAN. Fig. 8 (b) i shows the deflection of the left elevator ( $\delta_{e-left-t}$ ) and right elevator ( $\delta_{e-right-t}$ ) along with the BTFC ( $\delta_{e-right-c}$ ) and NN components ( $\delta_{e-right-nn}$ ) of the control signals. The neuron history ( $nn_{\delta_e}$ ) based on EMRAN for the right elevator is also shown. Fig. 8(b) ii shows similar signals for the ailerons. Fig. 8c shows the right elevator and right aileron signals in more detail to illustrate the learning in the presence of the neural network.

From Figure 8a, it can be seen that BTFC alone is unable to cope with this failure as the altitude drops around 50 sec whereas the neural aided control scheme is able to follow the reference trajectory closely. While the BTFC is unable to complete the landing task, the neural aided controller is not only able to land the aircraft but also achieve the touchdown performance requirements. The aircraft response with the neural aided BTFC is similar to the BTFC with no failures (Fig. 5) except that the deviations in velocity and sideslip are larger. The asymmetry due to left elevator being stuck, results in a persistent positive sideslip deviation subsequent to the occurrence of the failure. In Figure 8b the contribution of the BTFC and the neural network is shown. The shape and magnitude of the commands generated by the neural network is similar to that of the BTFC and the total control signal indicates that the neural network is successfully learning the system dynamics and is sharing the load of the BTFC. Eventually at the end of the trajectory when the neural network has learnt the changing dynamics, it takes over the total control load driving the BTFC load to zero. The neural learning is demonstrated in Fig. 8c, where the elevator and aileron control signals for this test case have been shown in more detail.

The learning is particularly marked for the right aileron control signal shown in 8c. The sharp changes seen in the control signals after 100 seconds are the result of the wind-shear and side gust encounters. For learning, EMRAN requires about 30 neurons for the elevator and 15 neurons for the aileron.

The simulation traces were examined for failure conditions of elevator and aileron in the feasible region where the aircraft was unable to land or could not meet the touch down requirements. On analysis, it was found the main reasons for this are the following:

- Elevator failure induces an un-commanded rolling moment due to the lateral displacement of its center of pressure from the roll axis.
- During the final descent segment before the flare the sudden change in vertical wind causes the aircraft trajectory to deviate from the desired touchdown point.

Both the above conditions can result in control surface saturation of one or more of the healthy control surfaces depending on the controller gains and feedback structure employed.

Figure 9 (a) shows the fault tolerant capabilities of BTFC and neural aided BTFC for a single surface failure (in this case, elevator). The points in the figure are generated by running a complete landing simulation with the failure at a particular deflection and checking whether it meets the landing pillbox requirements. BTFC did not meet the pillbox requirements for the entire range of failures from -20 deg to + 25 deg and hence no points are shown in the figure. On the other hand, the neural aided controller was able to meet the requirements in a significant range of deflections (-12 to 12 degrees).

However, at  $-8$ ,  $0$  and  $8$  degrees of elevator deflections the pillbox requirements are not met by the neural aided controller. To analyze this in detail, Figure 9b, shows the actual touch down points in the x-y plane along with the pillbox (between the limits  $100\text{m} \leq x \leq 300\text{m}$  and  $|y| \leq 5\text{m}$ ) for all the simulations. We note that with the neural aided controller three of the points (corresponding to  $-8, 0, 8$ ) lie just outside the pillbox. In Table 2, we have listed the main touch down performance variables for all these three cases. Clearly,  $-8$ ,  $0$  and  $8$  degree cases violate only one condition of the pillbox, namely the y-deviation at touchdown. Further, the amount of deviation from the pillbox condition is not large, being less than  $1.5\text{m}$ .

Figures 10 (a-b) show similar results for the case of aileron failure. From Figure 10 (a) we note that the BTFC is able to meet the pillbox requirement in the range of aileron stuck deflections of  $-7$  to  $4$  degrees. Neural aided controller has improved on this performance range to  $-7$  to  $20$  degrees with the exception of failure at  $2$  degrees. For this case also, the ‘gap’ in the figure is due to the inability of the neural aided controller to meet the y-deviation at touchdown (Fig. 10b). In this figure, we can note that the BTFC also has a few touchdowns outside the pillbox.

### **5.3 Two (dissimilar) surface failures**

In this scenario, the aircraft was subjected to a combination of dissimilar surface failures (one elevator and aileron). Figure 11 shows the feasible region based on the trim considerations along with the regions in the left elevator- left aileron plane that result in

successful touchdowns for the BTFC and the neural aided BTFC controllers. From the figure, it is seen that the boundaries of these regions are irregular, particularly for the neural controller. Further, for this controller, there are gaps in the bounded region representing combination of failures, where the pillbox requirements could not be met. In contrast, the BTFC controller has a uniform coverage of its bounded region, but smaller in size. It is also noted that the BTFC controller favors the region with higher positive elevator deflection, while the neural controller favors moderately positive and large negative deflections of the elevator. An analysis of these regions reveals the following features:

- Neural controller increases the region of failure tolerance compared to BTFC. In percentage terms, BTFC' has a 12% success rate whereas the neural controller has a 35% success rate. Here we define success rate as the ratio of number of simulations that resulted in a successful achievement of the pillbox criteria to the total number of feasible points tested.
- The 'gaps' in the region covered by the neural controller are directly attributed to not meeting the  $y$ -deviation among the pillbox requirements. In a couple of cases, the flight path angle requirement was also violated marginally.
- The part where the neural controller does not cover BTFC controller region is mainly due to its inability to meet the tight pillbox requirements but even in this region the neural controller is able to land the aircraft successfully. However, the converse is not true, namely the region where the BTFC controller does not work and the neural controller is successful (particularly, in the large negative elevator range), the former is unable to complete the landing task.

In Figure 12, we have repeated the results of Figure 11 but this time the pillbox condition for lateral deviation has been relaxed to  $|y| \leq 10\text{m}$ . We note that the coverage is now more uniform, indicating that the amongst all the pillbox criteria, the lateral deviation at touch down is the most demanding for these controllers. Still, the neural controller produces a larger envelope compared to BTFC. Similar results are shown in Figure 13, for combination failures of left elevator and right aileron. It can be seen that the conclusions are essentially similar to those for the left elevator and left aileron case.

The results clearly show that the neural aided feedback learning strategy is capable of significantly improving the fault tolerance capability of the aircraft-landing task.

## **6. Conclusions**

In this paper, a neural aided landing control scheme has been described for increasing the fault tolerance capabilities of existing aircraft controllers without the aid of an active Fault Detection and Isolation (FDI) schemes. Based on this study, the following conclusions emerge:

- Fixed gain classical controller, which was not designed for failure tolerance, has a limited failure-handling ability.
- For single surface failure cases, the BTFC is unable to meet the stringent touch down dispersions under elevator failures. For aileron stuck conditions, it meets the requirements only for small deflections (-8 to 4 degrees aileron failure).
- For the same case of single surface failure, the neural aided controller meets the strict touch down dispersions for both the aileron and elevator surfaces stuck at large deflections (-8 to 20 degrees for aileron and -12 to 12 degrees for elevator). Thus,

the neural aided control proposed in this paper enhances the fault tolerance capabilities of the existing controller.

- For two surface failure cases like the combined elevator and aileron failures, the fault tolerance envelope of the neural aided controller is significantly enlarged compared to that of the existing BTFC controller.

It should be noted that the neural aided controller acts as an add on to an existing controller and hence this can be easily implemented on a real aircraft without modifying the existing controllers. In real operational aircraft systems, the above neural controller can be easily implemented to handle sudden unknown failures. Also, these schemes can be easily flight tested to evaluate its performance under real surface failures.

### ***Acknowledgements***

This work was supported by a grant from Defence Science Organization (DSO) of Singapore (DSOCLO1144) and Nanyang Technological University (RGM 34/01). We wish to thank Dr. Poh Eng Kee and Dr Li Dong of DSO and Dr. Wang Jianlianf of NTU for many useful discussions and their help during this study.

## **References**

- [1] Azinheira J. R., de Paiva E. C., Ramos J. J. G, and Bueno, S. S., Mission Path Following for an Autonomous Unmanned Airship. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, Detroit, MI* (2000).
- [2] D. J. Bugajski and D. F. Enns, “Nonlinear control law with application to high angle-of-attack flight”, *J. Guid., Contr., Dynam.*, vol. 15, no. 3, pp. 761–767, 1992.
- [3] A.J. Calise, Neural Networks in Non-Linear Aircraft Flight Control. *IEEE Aerospace and Electronic Systems Magazine* 11 7 (1996) 5-10.
- [4] H. Gomi and M. Kawato, Neural network control for a closed-loop system using feedback-error-learning. *Neural Networks* 6 7 (1993) 933-946.
- [5] Iiguni, Y., Akiyoshi, H., and Adachi, N., An Intelligent Landing System Based on a Human Skill Model. *IEEE Transactions on Aerospace and Electronic Systems* 34 3 (1998) 877-882.
- [6] Johnson, E., and Calise, A., Neural Network Adaptive Control of Systems with Input Saturation. In: *Proceedings of the American Control Conference, Inst. of Electrical and Electronics Engineers, Piscataway, NJ* (2001) 3527–3532.
- [7] Jorgensen, C. C., and Scheley, C., Neural Network Baseline Problem for Control of Aircraft Flare and Touchdown, in: *Neural Networks for Control*, Cambridge, MA: MIT Press, 1990, pp. 402-425.
- [8] Li Y., Sundararajan N. and Saratchandran P., Analysis of Minimal Radial Basis Function Network Algorithm for Real-time Identification of Nonlinear Dynamic Systems. *IEE Proceedings on Control Theory Applications* 147 4 (2000) 476-484.

[9] Li Y., Sundararajan N., Saratchandran P., and Z. Wang Robust Neuro- $H_\infty$  Controller Design for Aircraft Auto-Landing. *IEEE Transactions on Aerospace and Electronic Systems*, 40 1 (2004) 158-167.

[10] Lu Yingwei., Sundararajan N. and Saratchandran P., A Sequential Learning Scheme for Function Approximation Using minimal Radial Basis Function Neural Networks. *Neural Computation* 9 (1997) 461-478.

[11] M.R. Napolitano, S. Naylor, C. Neppach and V. Casdorff, On-Line Learning Nonlinear Direct Neurocontrollers for Restructurable Control Systems. *AIAA Journal of Guidance, Control and Dynamics* 18 1 (1995) 170-176.

[12] Nguyen L. T., Ogburn M. E., Gilbert W. P., Kibler K. S., Brown P. W., Deal P. L., Simulator Study of Stall/Post-Stall Characteristics of a Fighter Airplane with Relaxed Longitudinal Static Stability. NASA Technical Paper 1538, Dec. 1979.

[13] Perhinschi M.G., Campa G., Napolitano M. R., Lando M., Massotti L., Fravolini M.L., Modeling and Simulation of Failures for Primary Control Surfaces, AIAA 2002-4786. In: *AIAA Modeling and Simulation Technologies Conference and Exhibit, Monterey, CA* (2002).

[14] Sundararajan N., Saratchandran P. and Li Y., Fully Tuned Radial Basis Function Neural Networks for Flight Control, Kulwer Academic Publishers, Boston, 2001.

**Table 1. Touchdown Specifications ( pillbox conditions )**

X-distance	$-100\text{m} \leq x \leq 300\text{m}$
Y-distance	$ y  \leq 5\text{m}$
Total Velocity	$V_T \geq 60\text{m/s}$
Sink Rate	$\dot{h} \leq 1.0\text{m/s}$
Bank Angle	$ \phi  \leq 10\text{deg}$

**Table 2. Pillbox Parameters with Left Elevator Failure.**

$\delta_e$ (deg)	$V_T$ (m/s)	$X$ (m)	$Y$ (m)	$\dot{h}$ (m/s)	$\phi$ (deg)
-8.0	80.90	94.74	5.35	-0.61	-1.34
0.0	80.80	120.05	-6.20	-0.80	0.53
8.0	79.70	110.59	-5.74	-0.77	-0.35

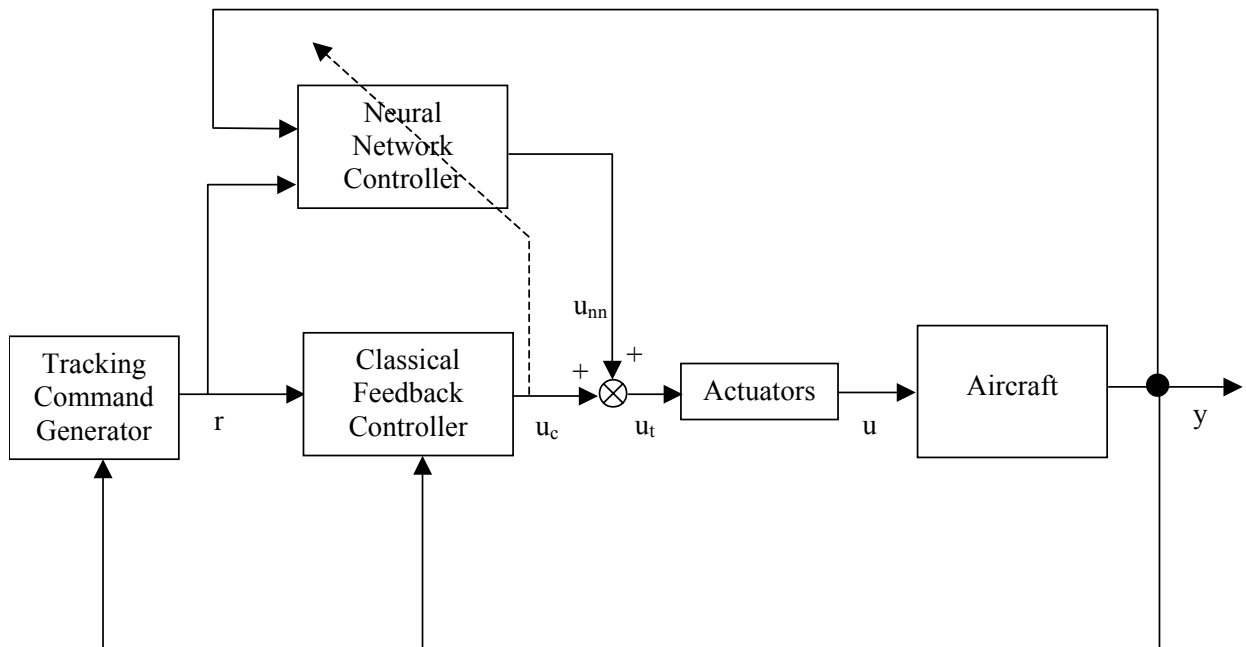


Figure 1. Neural Aided Controller Architecture.

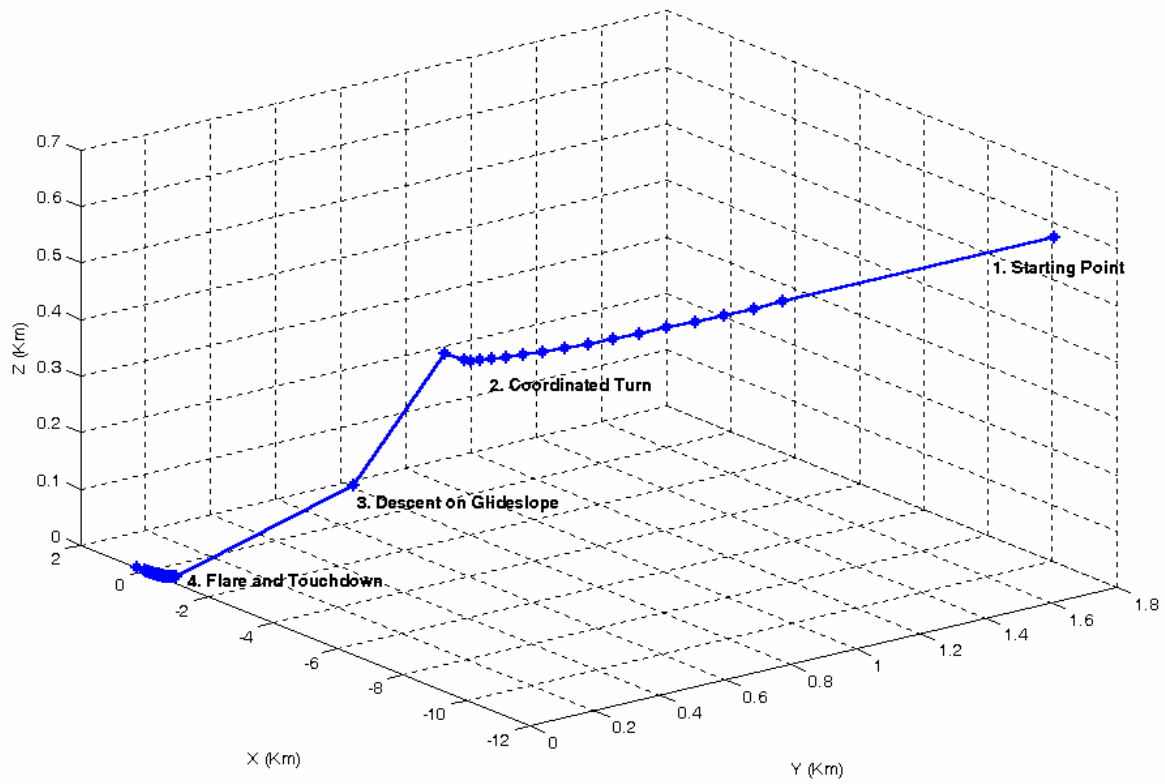


Figure 2. Landing Trajectory.

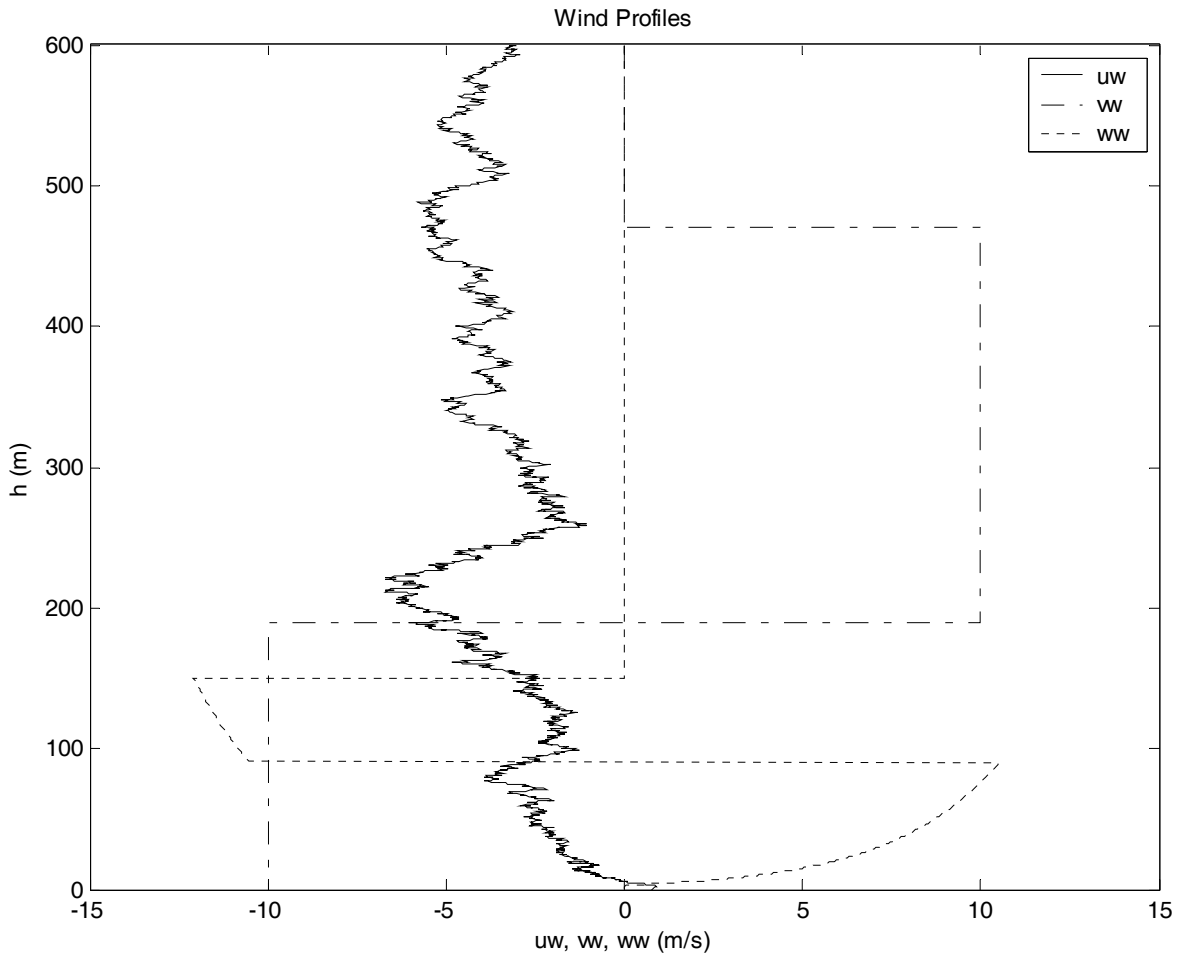


Figure 3. Wind Profiles during landing.

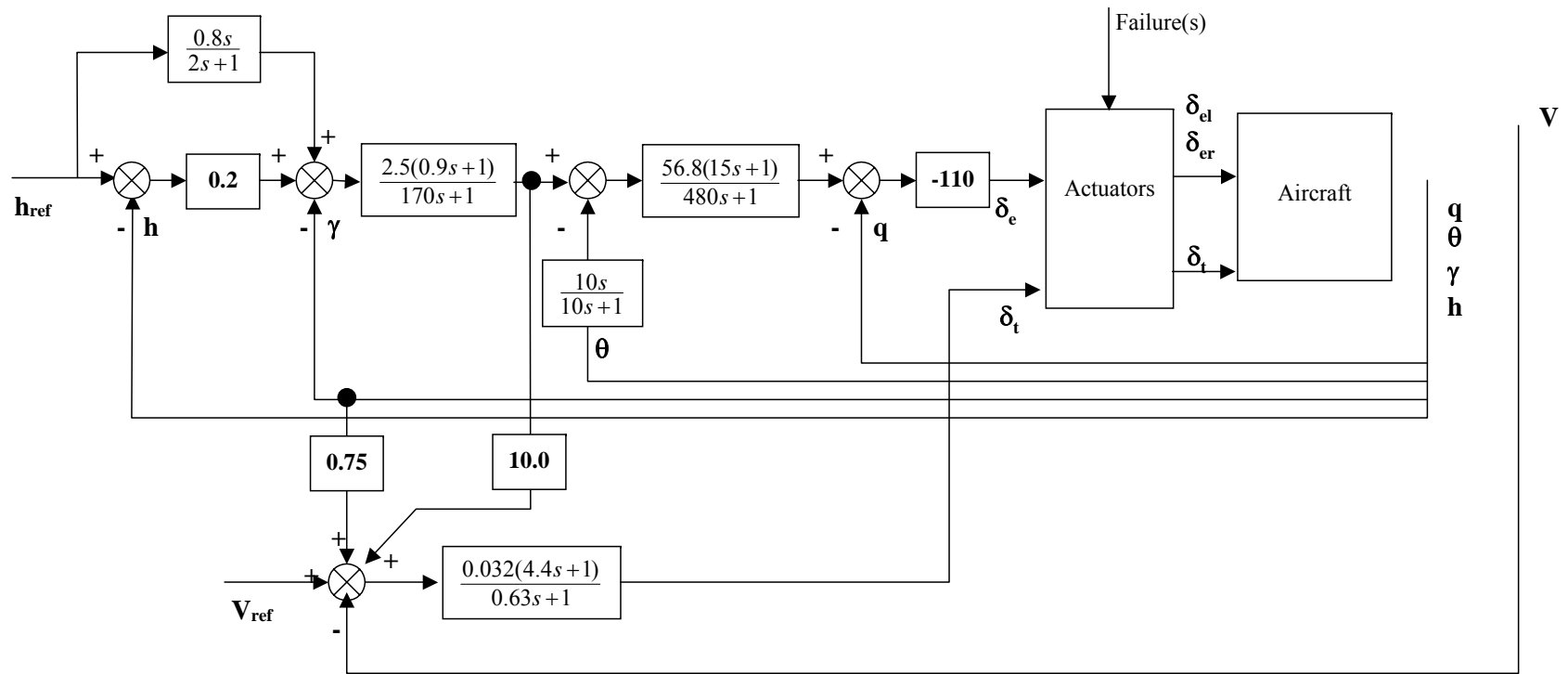


Figure 4. Longitudinal Baseline Trajectory Following Controller ( BTFC )  
 ( $h$ -meters,  $\gamma$ -deg,  $\theta$ -rad,  $q$ -rad/s,  $V$ -m/s, all surface deflections in degrees).

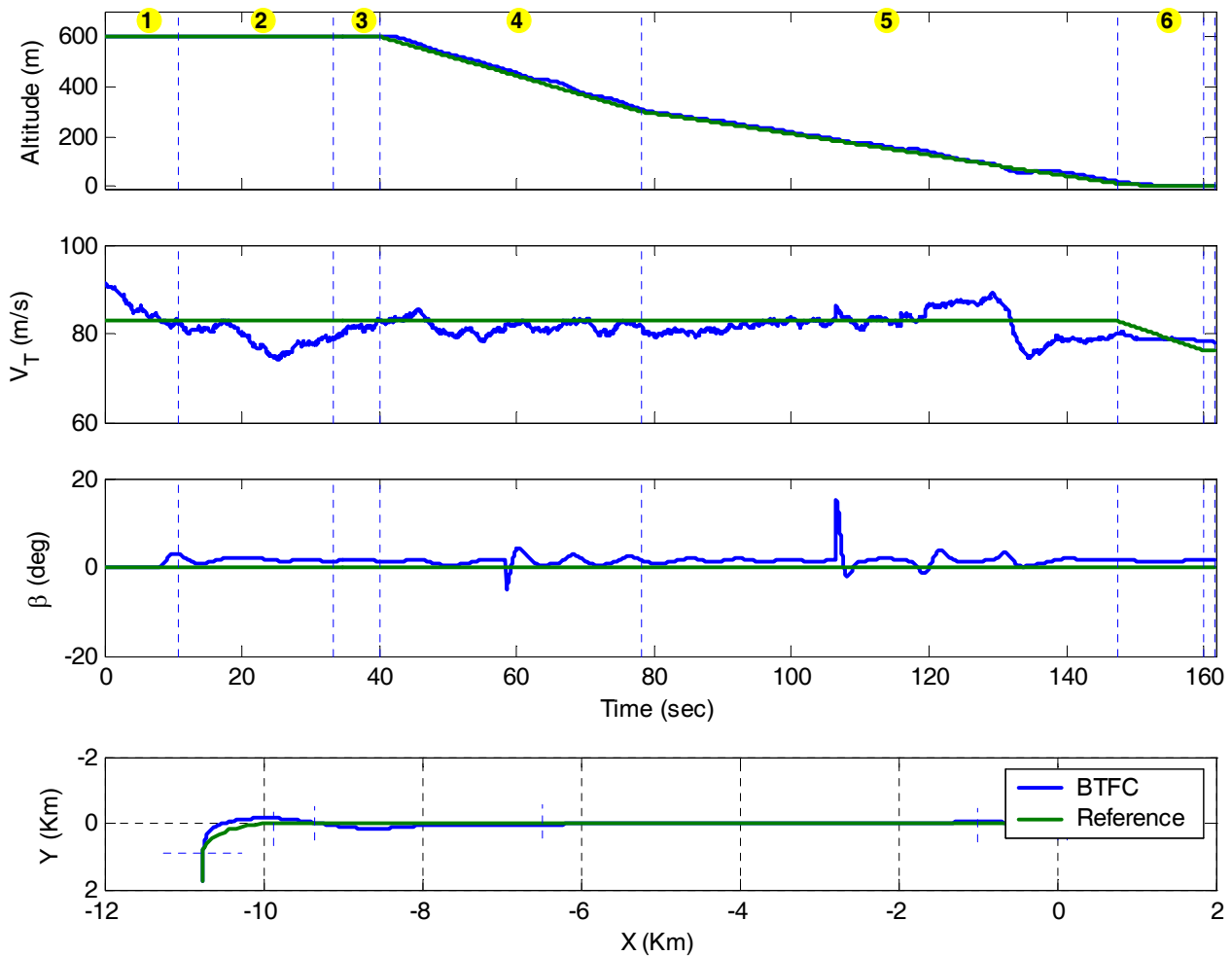


Figure 5. BTFC Controller Performance: Winds and No failures.

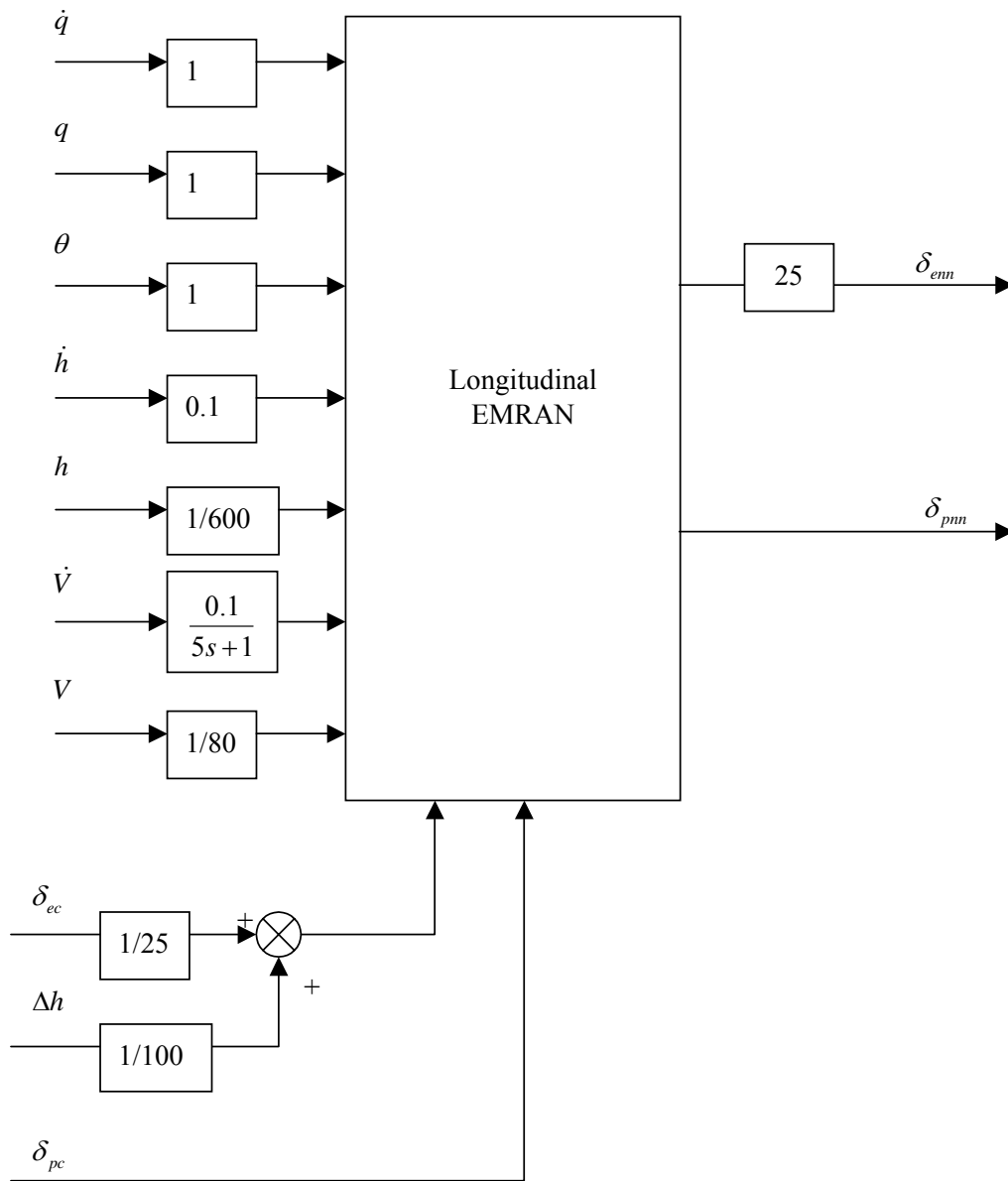


Figure 6. EMRAN Implementation for the Longitudinal Axis.

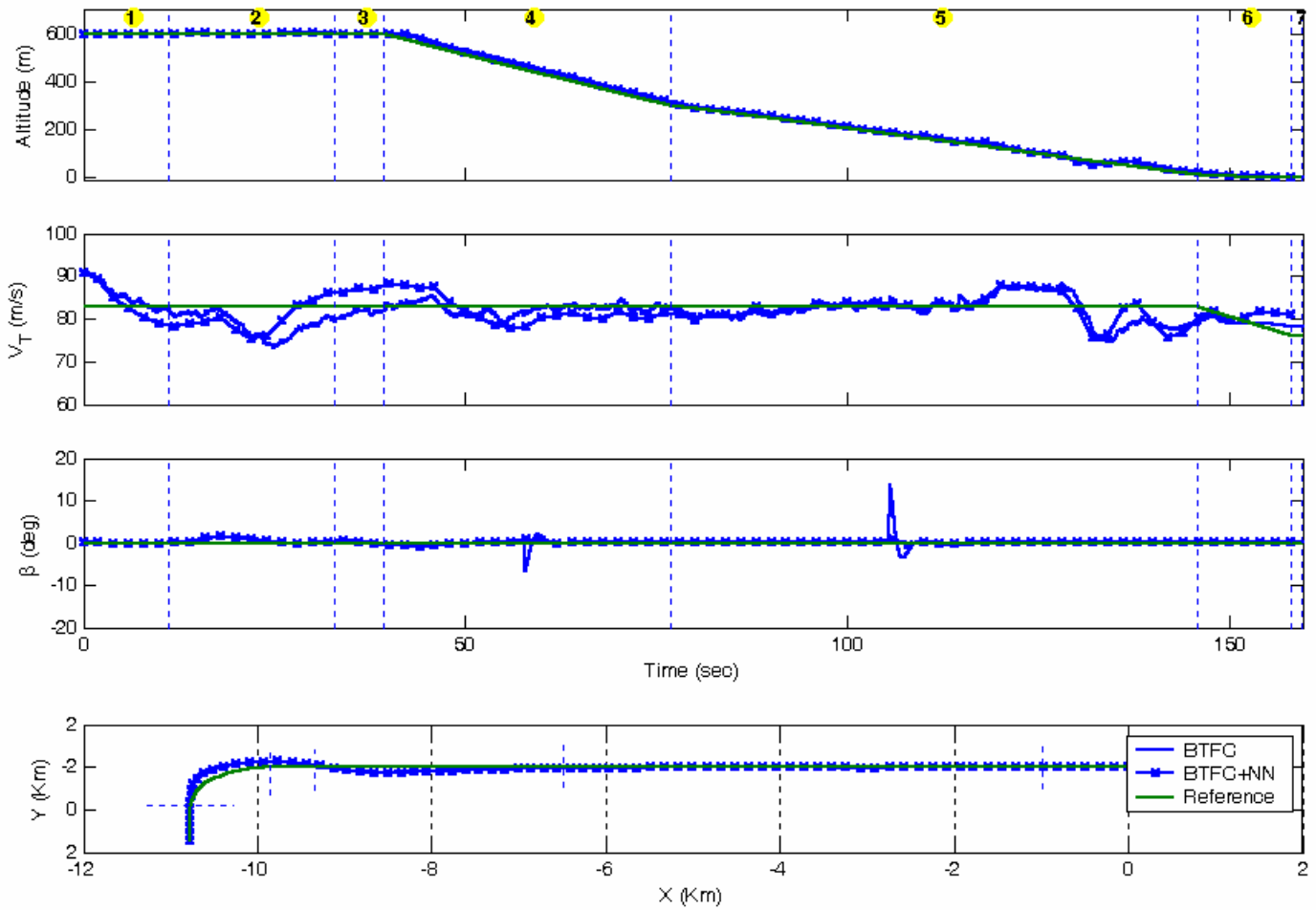


Figure 7a. Comparison of BTFC and Neural Aided BTFC Performance: Winds and No Failures.

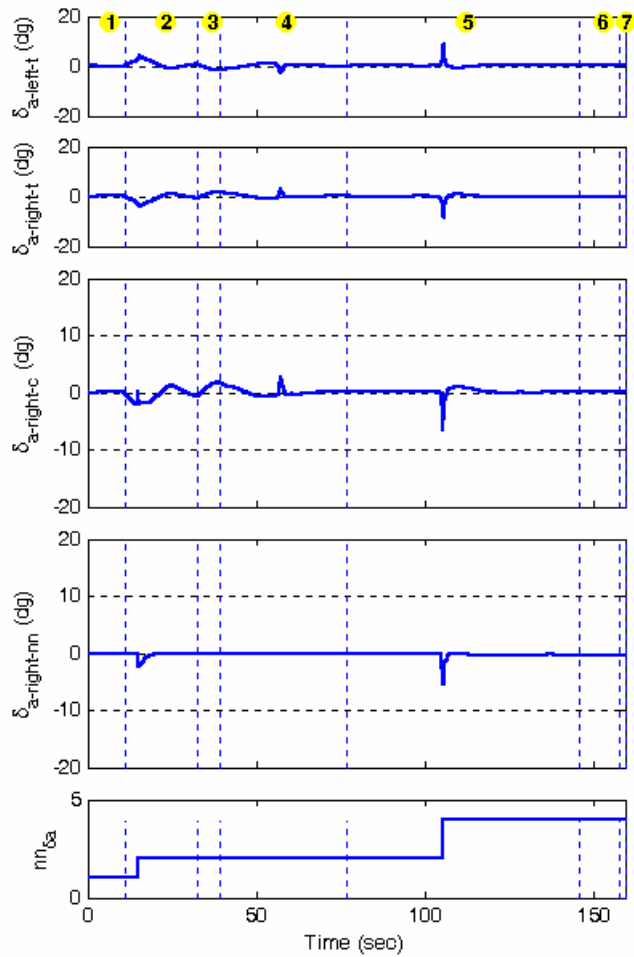
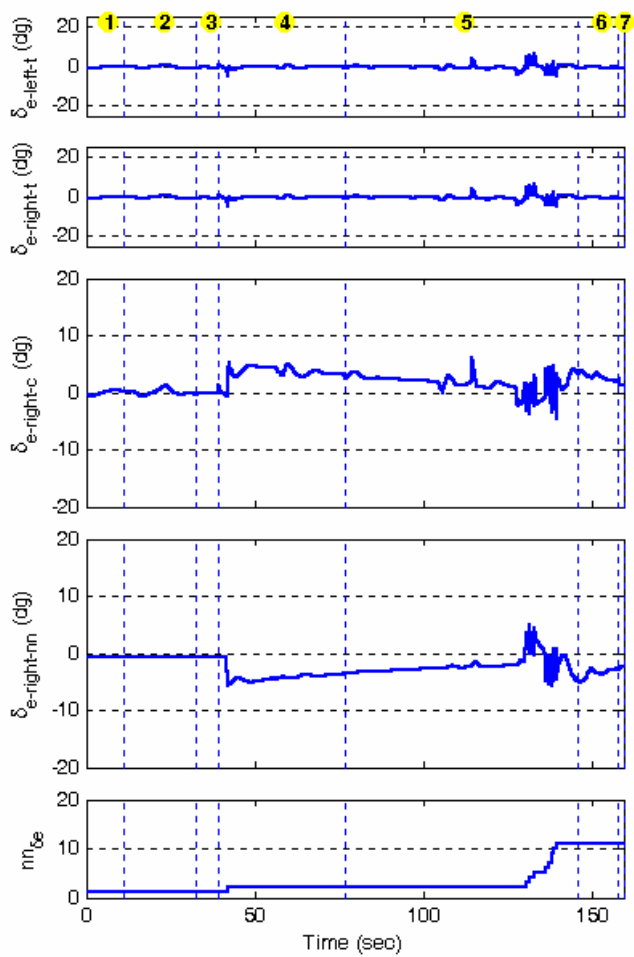


Figure 7b. Control Signals for Neural Aided BTFC for No Failures And Winds.

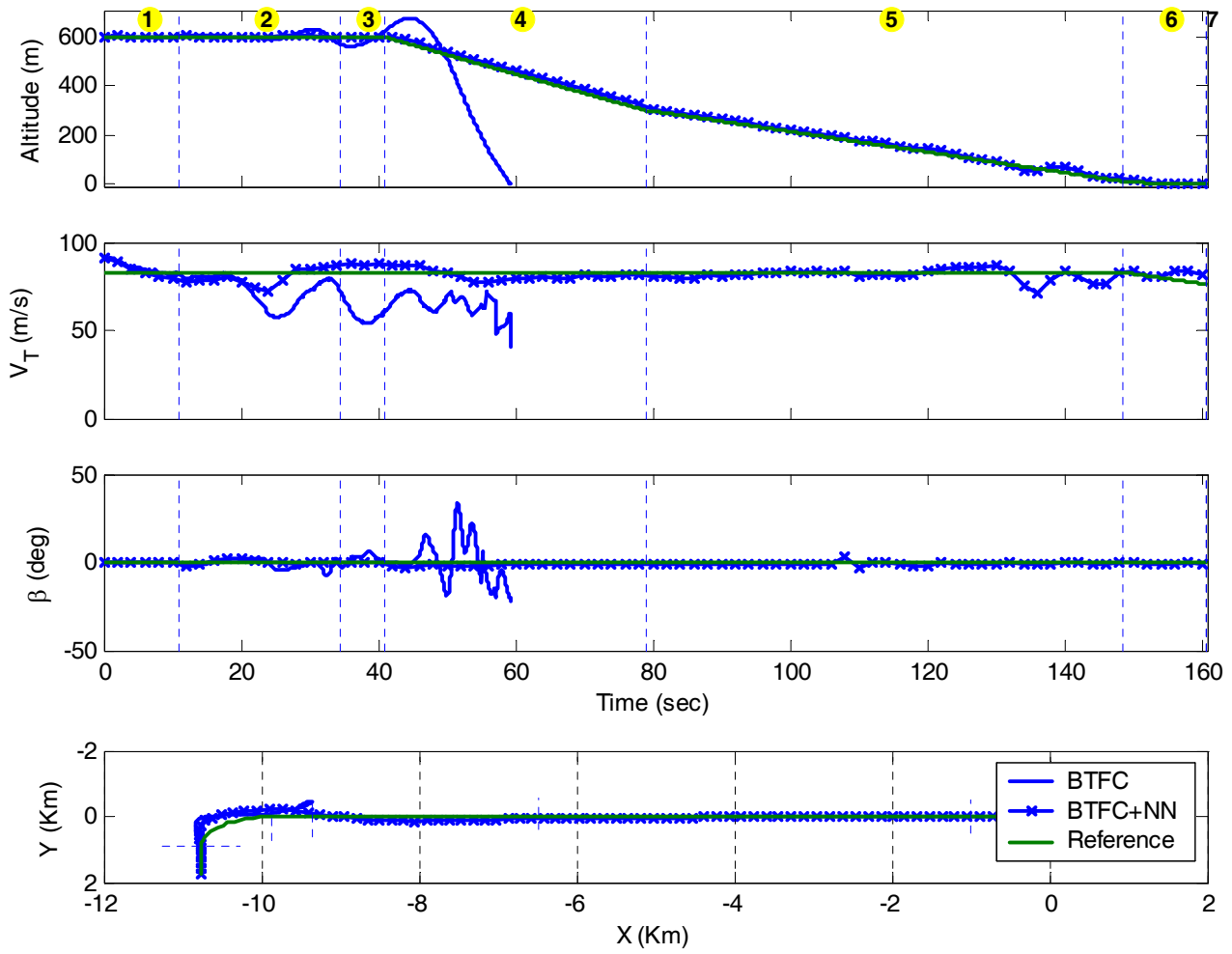


Figure 8a. Comparison of BTFC and Neural Aided BTFC Performance: Winds and Single failure (left elevator: -10deg).

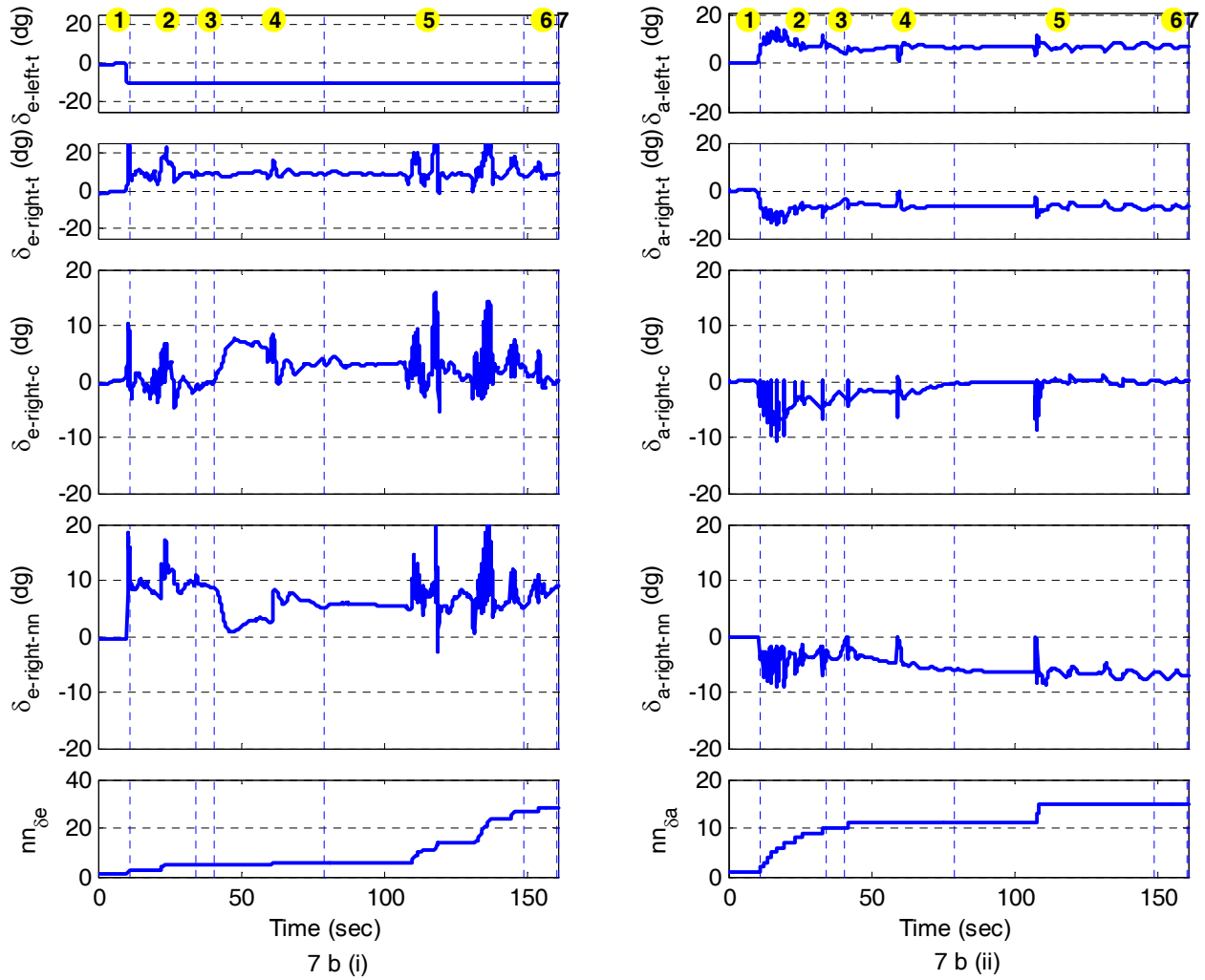


Figure 8b. Control Signals for Neural Aided BTFC for Single Failure of left elevator at -10 deg. And winds.

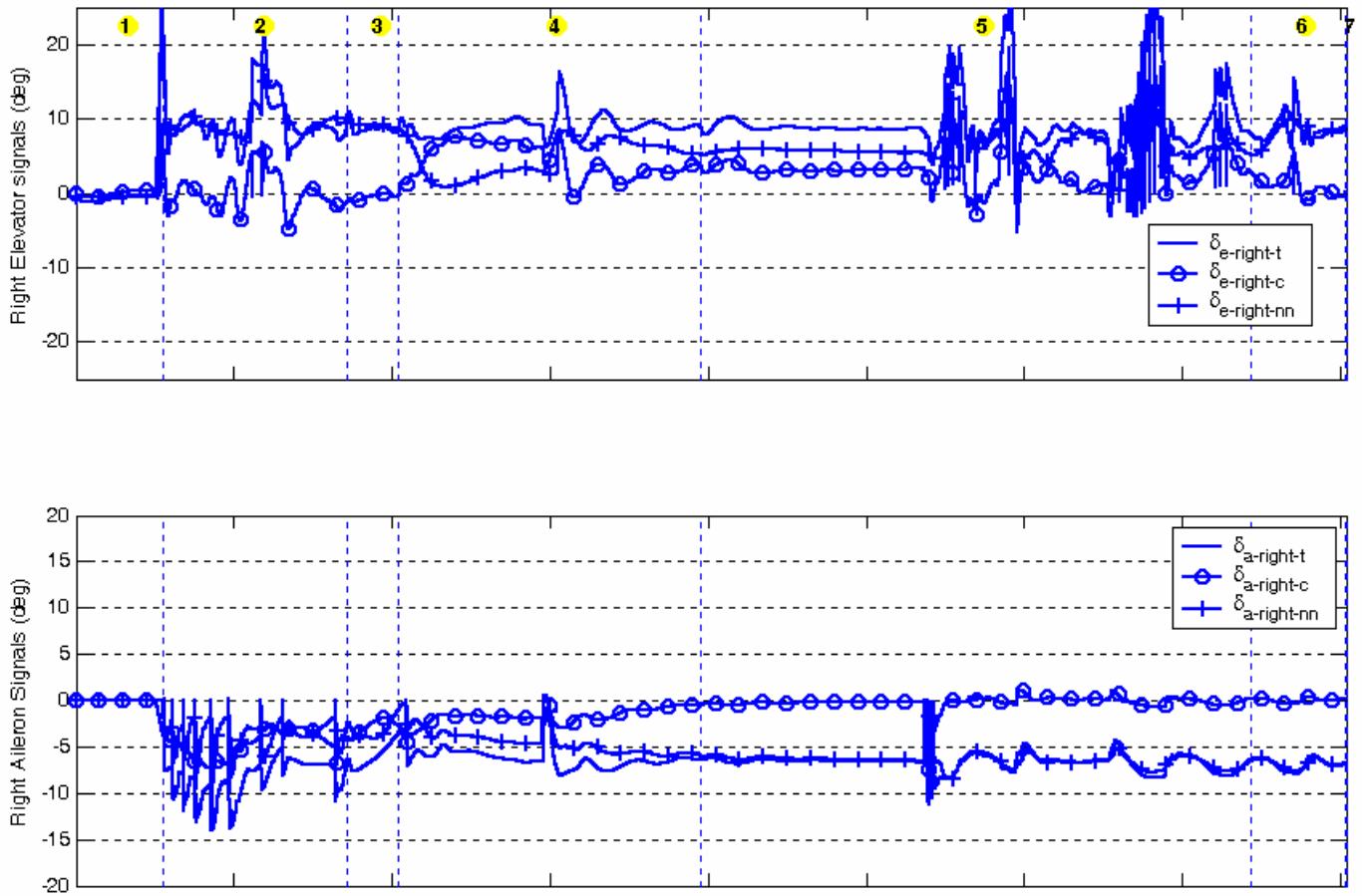
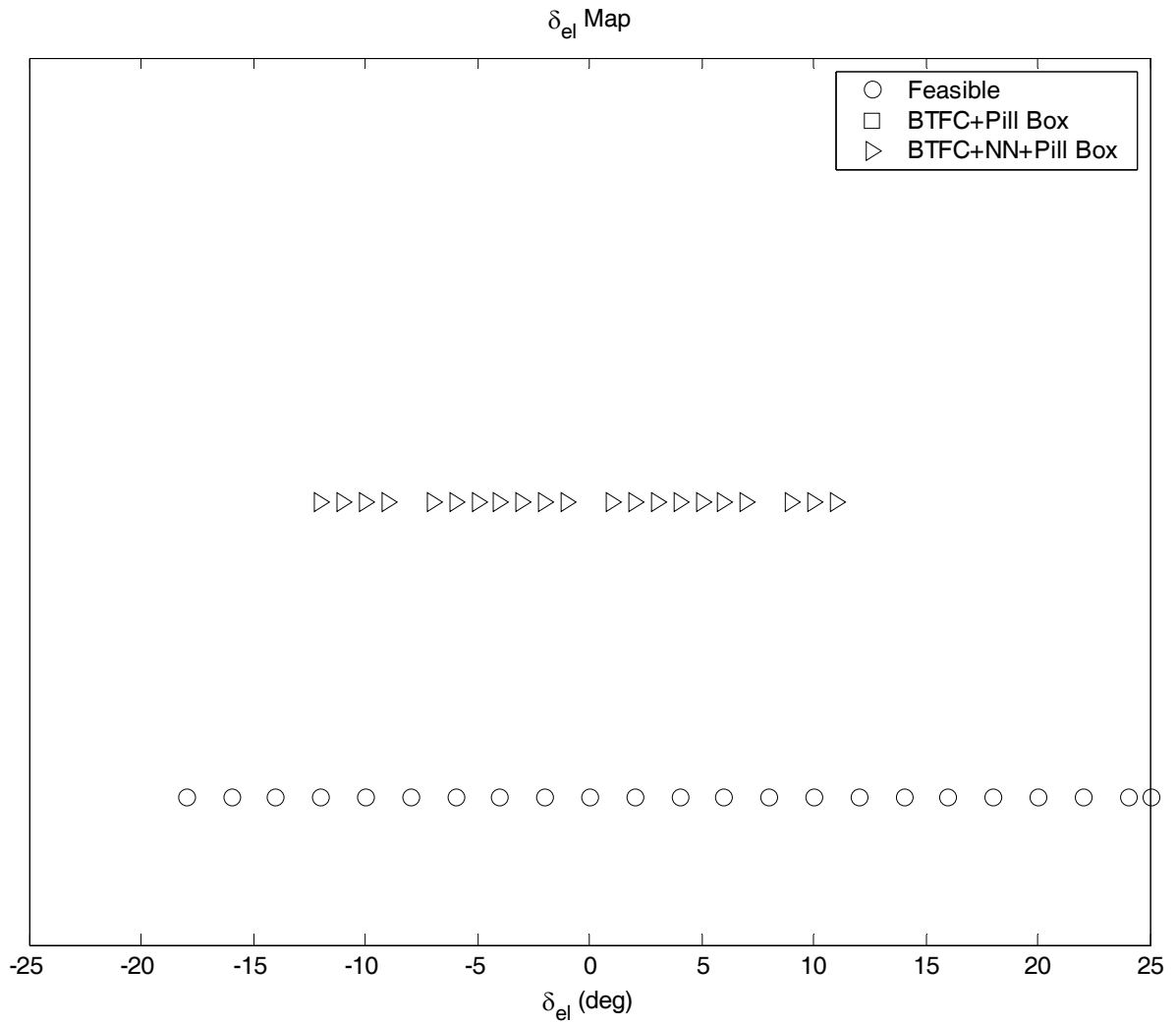


Figure 8c. Right Elevator and Aileron Control Signals for Neural Aided BTFC

Demonstrating Neural Learning for Single Failure of left elevator at -10 deg. And winds.



Note: BTFC is unable to meet pillbox for any failure position.

Figure 9a. Failure Tolerance of BTFC and Neural Aided BTFC : Left elevator stuck conditions.

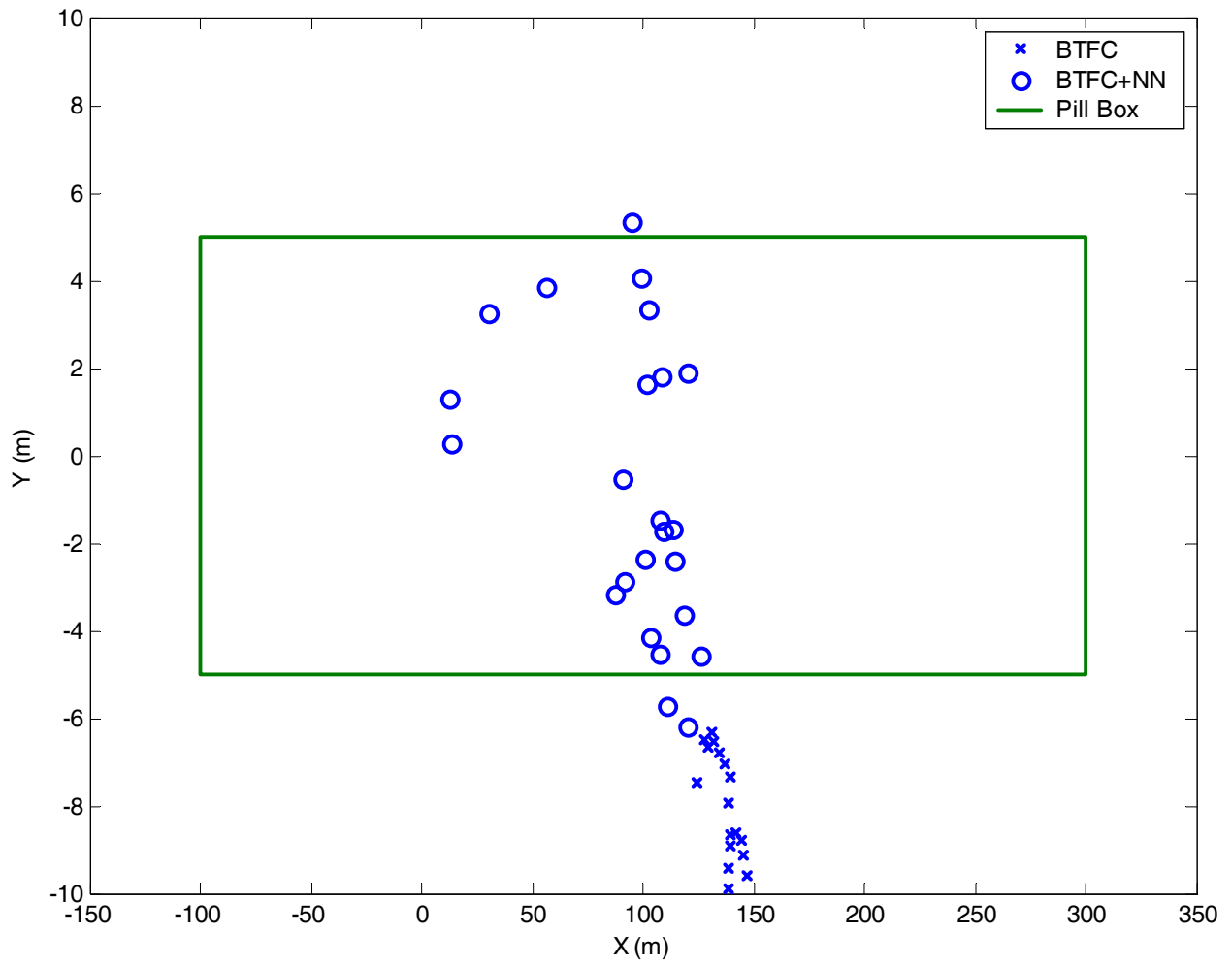


Figure 9b. Touch Down Points for BTFC and Neural Aided BTFC : Left elevator stuck conditions.

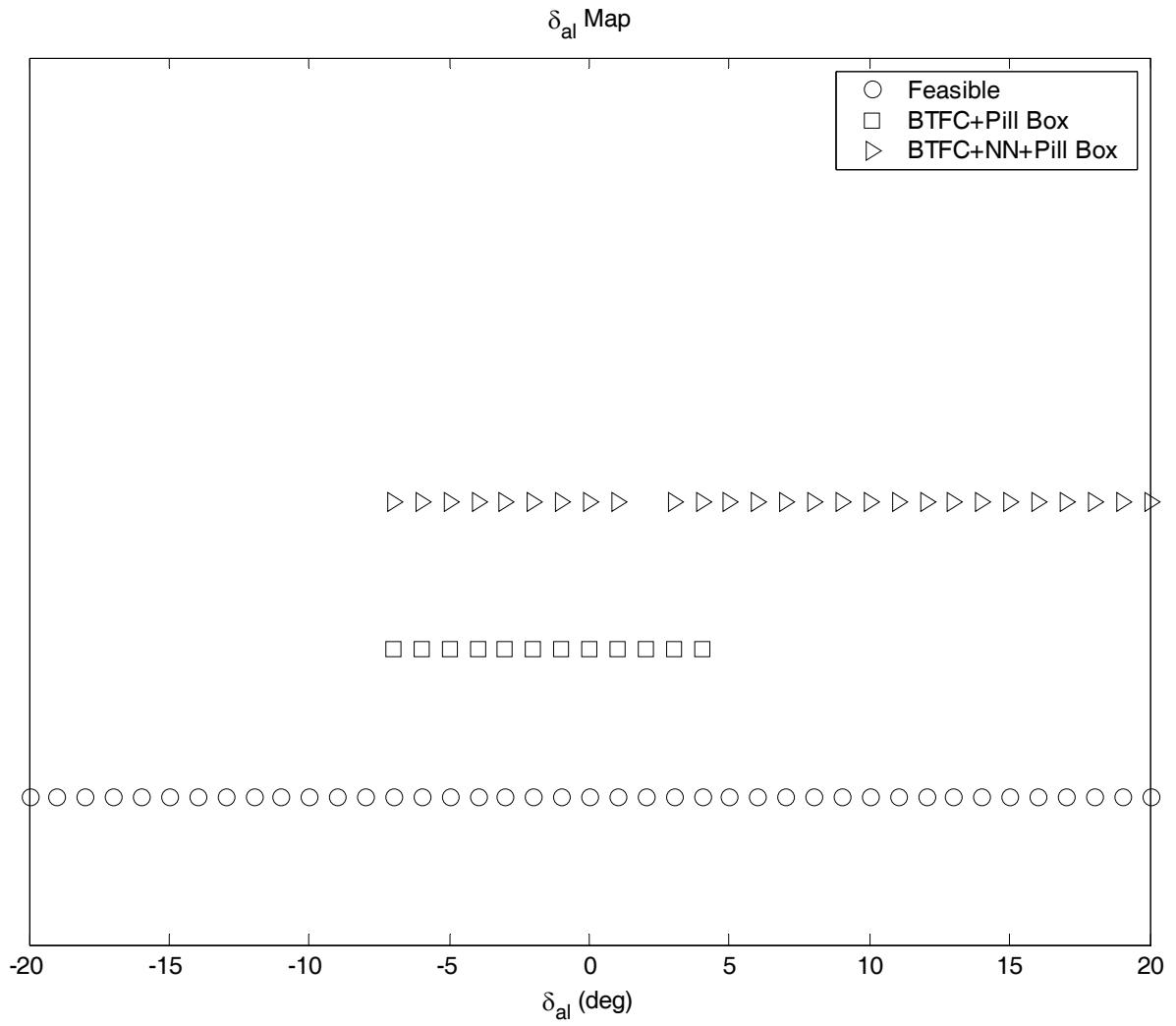


Figure 10a. . Failure Tolerance of BTFC and Neural Aided BTFC : Left aileron stuck conditions.

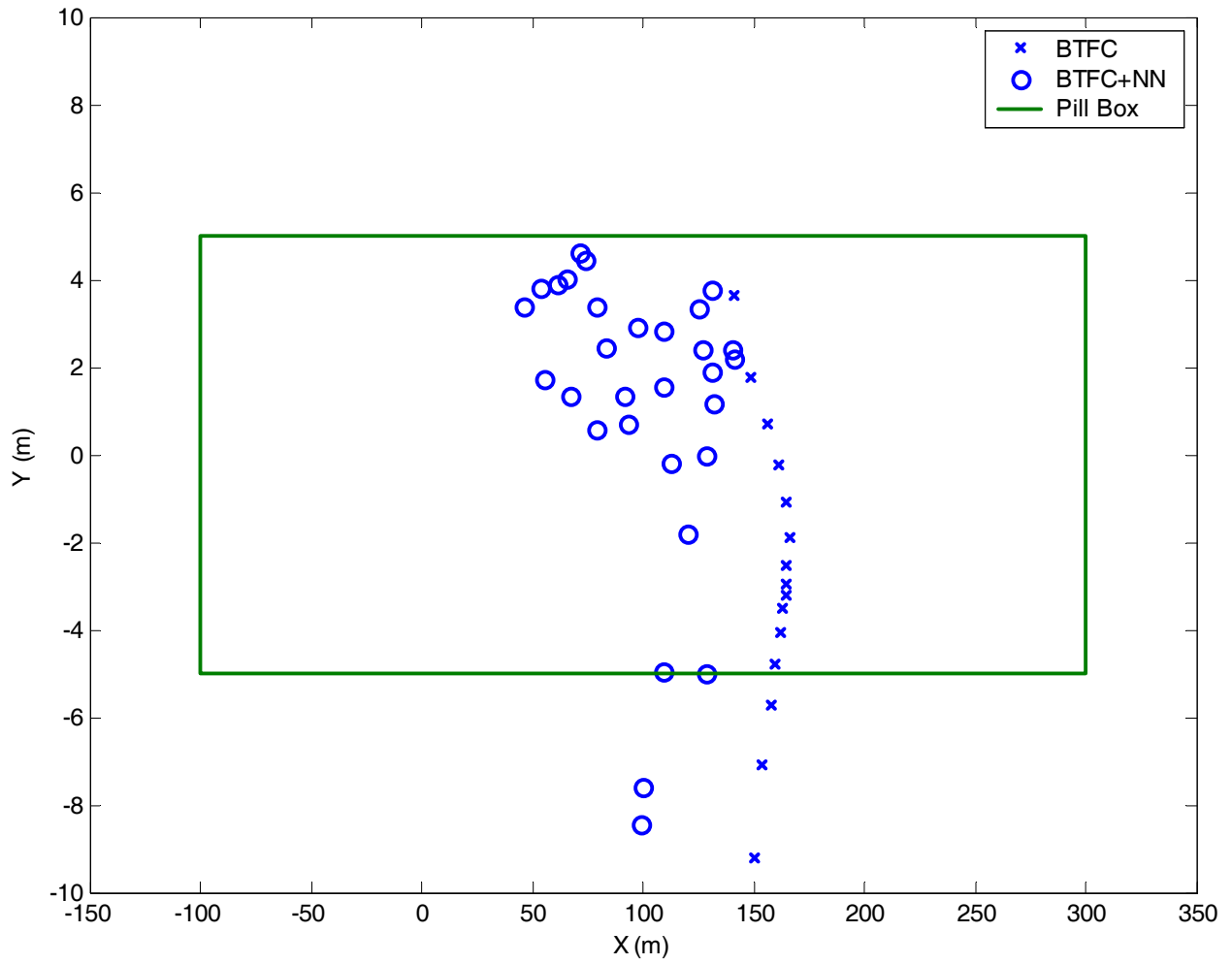


Figure 10b. Touch Down Points for BTFC and Neural Aided BTFC : Left aileron stuck conditions.

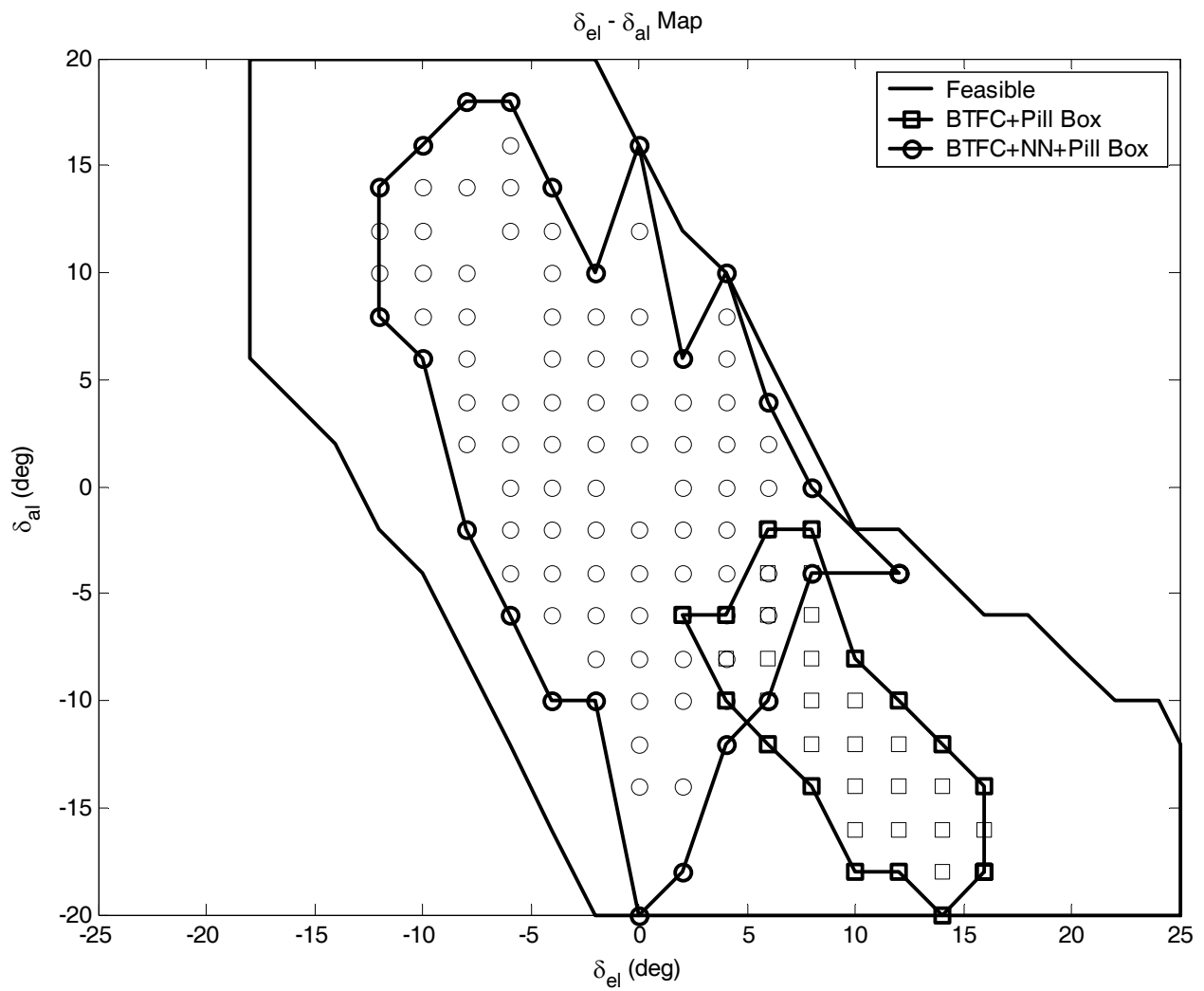


Figure 11. Failure Tolerance of BTFC and Neural Aided BTFC : Left Elevator and Left Aileron stuck conditions.

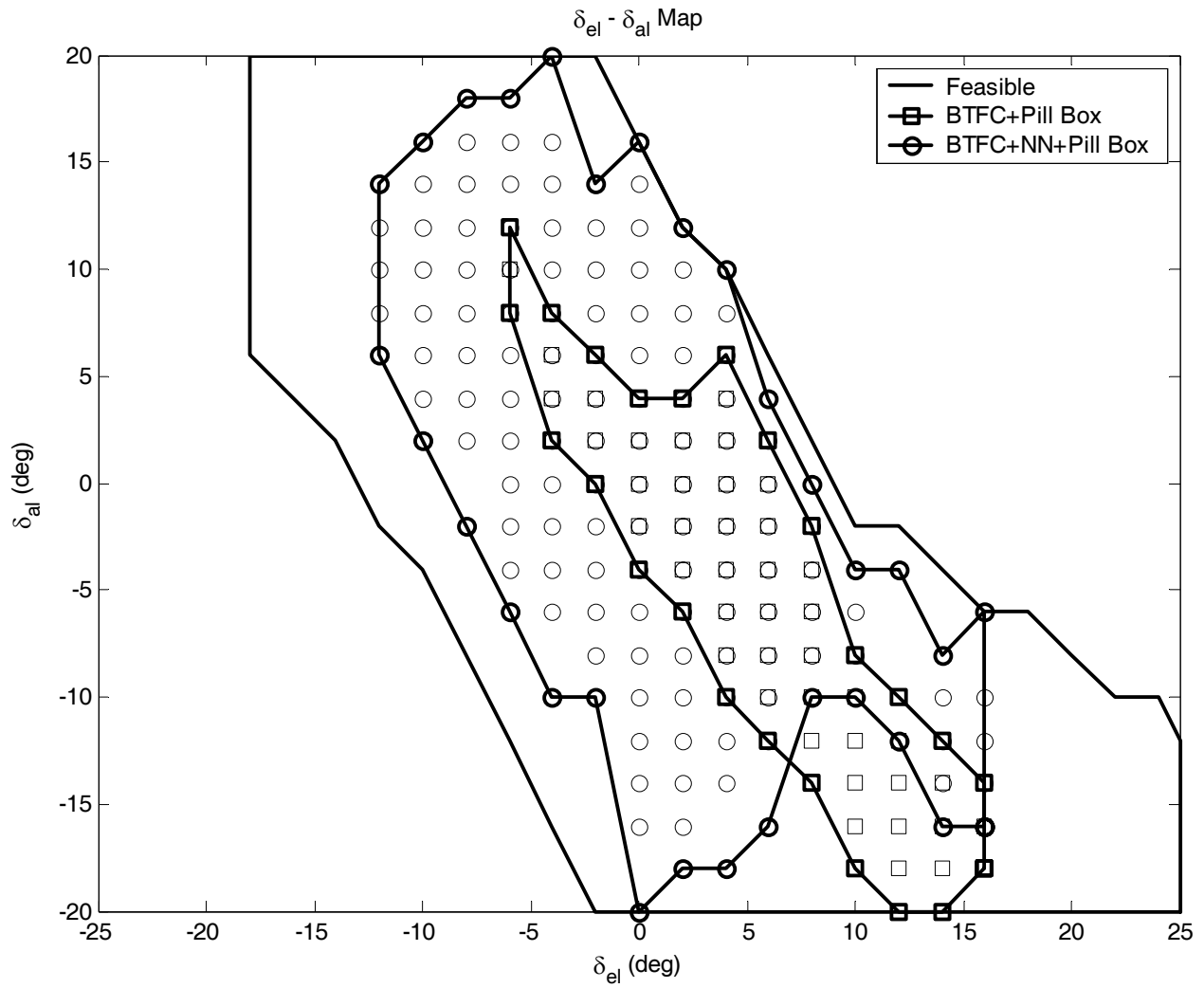


Figure 12. Failure Tolerance of BTFC and Neural Aided BTFC : Left Elevator and Left Aileron stuck conditions with y deviations relaxed to  $\pm 10$  mrs.

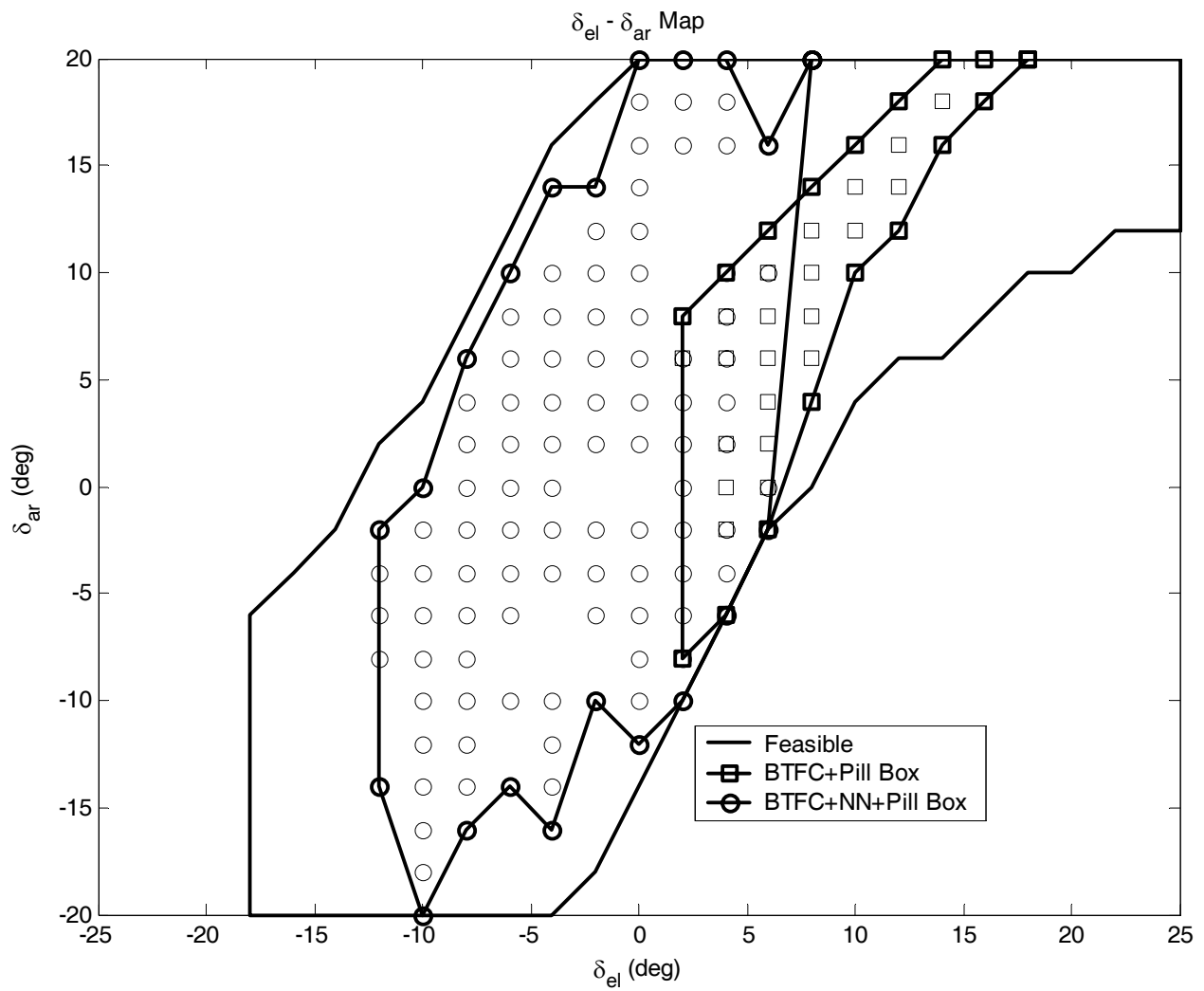


Figure 13. . Failure Tolerance of BTFC and Neural Aided BTFC : Left Elevator and Right Aileron stuck conditions.