



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

## Διπλωματική εργασία:

Μοντελοποίηση και διάγνωση βλαβών  
αεριοστροβίλων σε περιβάλλον  
MATLAB/Simulink

Ονοματεπώνυμο: Ιωάννης-Παναγιώτης  
Φιλιππόπουλος

Αριθμός Μητρώου: 02117070

Τομέας: Ρευστών

Επιβλέπων: Νικόλαος Αρετάκης, Αναπληρωτής  
Καθηγητής ΕΜΠ

Αθήνα, 2022



NATIONAL TECHNICAL UNIVERSITY OF ATHENS  
SCHOOL OF MECHANICAL ENGINEERING

# Diploma thesis:

**Gas turbine fault modeling and diagnosis in  
MATLAB/Simulink**

Name: **Ioannis-Panagiotis Filippopoulos**

Student Number: **02117070**

Section: **Fluids**

Supervisor: **Nikolaos Aretakis, Associate Professor NTUA**

Athens, 2022

--κενή σελίδα --

Έχω διαβάσει και κατανοήσει τους κανόνες για τη λογοκλοπή και τον τρόπο σωστής αναφοράς των πηγών που περιέχονται στον οδηγό συγγραφής Διπλωματικών Εργασιών. Δηλώνω ότι, από όσα γνωρίζω, το περιεχόμενο της παρούσας Διπλωματικής Εργασίας είναι προϊόν δικής μου εργασίας και υπάρχουν αναφορές σε όλες τις πηγές που χρησιμοποίησα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτή τη Διπλωματική εργασία είναι του συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις της Σχολής Μηχανολόγων Μηχανικών ή του Εθνικού Μετσόβιου Πολυτεχνείου.

Ιωάννης-Παναγιώτης Φιλιππόπουλος

## Περιεχόμενα

Σύνοψη . . . . .	7
Abstract . . . . .	8
Αρκτικόλεξα . . . . .	9
<b>1 Εισαγωγή . . . . .</b>	<b>10</b>
<b>2 Simulink και T-MATS . . . . .</b>	<b>11</b>
2.1 Έτοιμα blocks Simulink . . . . .	11
2.2 Blocks της βιβλιοθήκης T-MATS . . . . .	14
<b>3 Ανάπτυξη εφαρμογής . . . . .</b>	<b>18</b>
3.1 Μοντέλο Simulink . . . . .	18
3.1.1 Μεταβλητές σημείου λειτουργίας . . . . .	21
3.1.2 Διαθέσιμες μετρήσεις του κινητήρα . . . . .	23
3.1.3 Μεταβλητή εισόδου ελέγχου . . . . .	24
3.1.4 Σφάλματα των μετρήσεων . . . . .	25
3.1.5 Επιλογή σημάτων . . . . .	26
3.1.6 Επιλύτης . . . . .	28
3.1.7 Παράμετροι υγείας . . . . .	29
3.1.8 Αποτελέσματα στον χώρο εργασίας (workspace) . . . . .	30
3.2 Εφαρμογή . . . . .	31
3.2.1 Πλαίσιο εισόδων . . . . .	34
3.2.2 Πλαίσιο παραμέτρων υγείας . . . . .	35
3.2.3 Πλαίσιο παραμέτρων απόδοσης . . . . .	35
3.2.4 Ομάδα καρτελών . . . . .	36
3.2.5 Πλαίσιο σχηματικού . . . . .	40
3.2.6 Πλαίσιο ελέγχου . . . . .	40
3.3 Παραδείγματα χρήσης της εφαρμογής . . . . .	44
3.3.1 Προσομοίωση ενός σημείου . . . . .	44
3.3.2 Προσομοίωση πολλαπλών σημείων . . . . .	46
3.3.3 Διάγνωση . . . . .	48
3.4 Ιακωβιανός πίνακας και αριθμός κατάστασης . . . . .	50
3.5 Αρχικές συνθήκες επιλύτη . . . . .	51
<b>4 Διερεύνηση μεθόδων διάγνωσης μέσω Νευρωνικών Δικτύων 55</b>	<b>55</b>
4.1 Αλγόριθμος διάγνωσης . . . . .	55
4.2 Δημιουργία προτύπων . . . . .	55
4.3 Θόρυβος . . . . .	57
4.4 Τύποι Νευρωνικών Δικτύων . . . . .	57
4.4.1 Πιθανολογικό Νευρωνικό Δίκτυο . . . . .	57
4.4.2 Λοιπά ΝΔ πρόσθιας τροφοδότησης . . . . .	59
4.5 Εκπαίδευση των ΝΔ . . . . .	59
4.6 Εκτίμηση παραμέτρων υγείας . . . . .	62

4.7	Διερεύνηση . . . . .	65
4.7.1	Σύγκριση ΝΔ . . . . .	65
4.7.2	Διαφορετικά σημεία λειτουργίας . . . . .	67
4.7.3	Γενική χειροτέρευση (Deterioration) . . . . .	69
4.7.4	Συνδυασμός γενικής χειροτέρευσης με βλάβη συνιστώσας . .	70
4.7.5	Βλάβη 2 συνιστωσών . . . . .	71
<b>5</b>	<b>Ανακεφαλαίωση/Συμπεράσματα . . . . .</b>	<b>73</b>
	<b>Κατάλογος Αναφορών . . . . .</b>	<b>78</b>
	<b>Παραρτήματα . . . . .</b>	<b>79</b>
I	Κώδικας εφαρμογής . . . . .	79
II	Κώδικας Νευρωνικών Δικτύων . . . . .	93

## Σύνοψη

Η ανάπτυξη υπολογιστικών μοντέλων για την πραγματοποίηση προσομοίωσης και διάγνωσης σε αεριοστροβίλους είναι αρκετά διαδεδομένη τόσο σε ακαδημαϊκό, όσο και σε βιομηχανικό επίπεδο. Στην παρούσα εργασία αναπτύχθηκε μια εφαρμογή στο περιβάλλον Matlab/Simulink με σκοπό ο χρήστης της να μπορεί να πραγματοποιήσει προσομοίωση και διάγνωση σε έναν αεριοστρόβιλο. Επιπροσθέτως, δημιουργήθηκαν κώδικες που χρησιμοποιούν Νευρωνικά Δίκτυα για την διάγνωση βλαβών ενός αεροπορικού κινητήρα αξιοποιώντας την Εργαλειοθήκη Βαθιάς Μάθησης (Deep Learning Toolbox) της Matlab. Για τον σκοπό αυτό αναπτύχθηκε ένα μοντέλο κινητήρα στο περιβάλλον Simulink χρησιμοποιώντας την βιβλιοθήκη συνιστωσών αεριοστροβίλων T-MATS της NASA και στη συνέχεια, μέσω του App Designer της Matlab, δημιουργήθηκε η διεπαφή χρήστη της εφαρμογής. Η εφαρμογή που αναπτύχθηκε μπορεί να χρησιμοποιηθεί ως ένα έτοιμο εργαλείο προσομοίωσης και διάγνωσης ενός αεριοστροβίλου, παρέχοντας στον χρήστη έλεγχο του σημείου λειτουργίας και δίνοντάς του πλήρη εποπτεία της κατάστασής του. Τέλος, βρέθηκε πως τα Νευρωνικά Δίκτυα που αναπτύχθηκαν είχαν ποσοστό επιτυχούς ταξινόμησης προτύπων πάνω από 90%.

## Abstract

Development of computational models to conduct simulations and diagnoses of gas turbines is fairly prevalent both in the academic and the industrial level. In the present thesis an app was developed in the Matlab/Simulink environment aiming for the user to be able to perform simulations and diagnoses on a gas turbine. Furthermore, scripts that employ Neural Networks were created in order to perform aircraft engine diagnosis utilizing Matlab's Deep Learning Toolbox. For this purpose, an engine model was developed in the Simulink environment using NASA's gas turbine component library T-MATS and subsequently a user interface was created using Matlab's App Designer. The developed app can be used as a ready-to-use simulation and diagnosis tool of a gas turbine, providing the user control of the operating point and allowing for a full overview of its state. Finally, the developed Neural Networks had a rate of successful pattern classification of over 90%.



## Αρκτικόλεξα

**HPC** High Pressure Compressor

**HPT** High Pressure Turbine

**LPC** Low Pressure Compressor

**LPT** Low Pressure Turbine

**NASA** National Aeronautics and Space Administration

**PNN** Probabilistic Neural Network

**T-MATS** Toolbox for the Modeling and Analysis of Thermodynamic Systems

**ΝΔ** Νευρωνικό Δίκτυο

# 1 Εισαγωγή

Η ανάπτυξη ενός υπολογιστικού μοντέλου αποτελεί ένα αναγκαίο πρώτο βήμα για την πραγματοποίηση προσομοίωσης και διάγνωσης σε έναν αεριοστροβίλο. Η διάγνωση ορίζεται ως η διαδικασία εντοπισμού βλάβης σε μία συνιστώσα ή ένα σύστημα ώστε να πραγματοποιηθεί κατάλληλη συντήρηση του[4]. Μία μέθοδος διάγνωσης που έχει απασχολήσει την βιβλιογραφία τα τελευταία χρόνια είναι η διάγνωση μέσω Νευρωνικών Δικτύων [4, 8, 9]. Σύμφωνα με τον [5], ένα Νευρωνικό Δίκτυο ορίζεται ως ένας μαζικά παράλληλος και κατανεμημένος επεξεργαστής αποτελούμενος από απλές μονάδες επεξεργασίας, ο οποίος διαθέτει φυσική τάση για την αποθήκευση βιωματικής γνώσης σε μορφή διαθέσιμη προς χρήση.

Στην παρούσα διπλωματική εργασία αναπτύχθηκε μία εφαρμογή στο περιβάλλον Matlab/Simulink μέσω του App Designer. Για τον σκοπό αυτό δημιουργήθηκε ένα μοντέλο Simulink, το οποίο βασίζεται στην κινητήρα JT9D της εταιρείας Pratt & Whitney, αξιοποιώντας την βιβλιοθήκη συνιστωσών αεριοστροβίλων T-MATS της NASA[2]. Επιπλέον, πραγματοποιήθηκε διερεύνηση των Νευρωνικών Δικτύων ως μέθοδο διάγνωσης αεροπορικών κινητήρων για διάφορα είδη Νευρωνικών Δικτύων πρόσθιας τροφοδότησης, όπως τα Πιθανολογικά Νευρωνικά Δίκτυα.

Εδώ αξίζει να σημειωθεί ότι η εφαρμογή αναπτύχθηκε μόνο για ένα τύπο κινητήρα, ενώ για λόγους απλότητας δεν δόθηκε έμφαση στη βελτιστοποίηση των Νευρωνικών Δικτύων που εξετάστηκαν. Ακόμη, έγινε χρήση μόνο δεδομένων που προέκυψαν μέσω του μοντέλου Simulink και όχι πραγματικών μετρήσεων. Τέλος, δεν διερευνήθηκε η δυνατότητα εντοπισμού συγκεκριμένων βλαβών, αλλά μόνο της συνιστώσας στην οποία αυτές εντοπίζονται και τις αντίστοιχες παραμέτρους υγείας της.

Στο πρώτο κεφάλαιο του κυρίου μέρους της παρούσας εργασίας, αφότου σχημαγραφηθούν οι κύριες δυνατότητες του περιβάλλοντος Simulink, περιγράφονται τα έτοιμα blocks που χρησιμοποιήθηκαν, καθώς και εκείνα της βιβλιοθήκης T-MATS. Στη συνέχεια, στο δεύτερο κεφάλαιο δίνεται η διαδικασία ανάπτυξης του μοντέλου στο περιβάλλον Simulink για την πραγματοποίηση προσομοιώσεων και διάγνωσης. Ενώ, παράλληλα, περιγράφεται η μεθοδολογία ανάπτυξης της εφαρμογής στο περιβάλλον App Designer της MATLAB. Στο τελευταίο κεφάλαιο αρχικά περιγράφονται ο αλγόριθμος διάγνωσης μέσω ενός Νευρωνικού Δικτύου και η διαδικασία δημιουργίας προτύπων για την εκπαίδευση και δοκιμή του. Εν συνεχεία, έπειτα από την παρουσίαση των διαφόρων τύπων Νευρωνικών Δικτύων που εξετάστηκαν, περιγράφεται η διαδικασία εκπαίδευσής τους και δίνεται ο τρόπος ταξινόμησης βλαβών σε προκαθορισμένες κλάσεις, καθώς και ο τρόπος εκτίμησης των αντίστοιχων παραμέτρων υγείας. Τέλος, παρουσιάζονται τα αποτελέσματα της πραγματοποίησης διάγνωσης μέσω Νευρωνικών Δικτύων σε διάφορες περιπτώσεις, όπως σε διάφορα σημεία λειτουργίας, σε περίπτωση ύπαρξης γενικής χειροτέρευσης (deterioration) και σε περιπτώσεις συνδυασμένης βλάβης.

## 2 Simulink και T-MATS

Το Simulink είναι ένα περιβάλλον ανάπτυξης σχηματικών διαγραμμάτων (block diagrams), το οποίο χρησιμοποιείται για τον σχεδιασμό συστημάτων μέσω της δημιουργίας μοντέλων και της πραγματοποίησης προσομοιώσεων σε αυτά. Έχει αναπτυχθεί από την MathWorks και λειτουργεί σε συνδυασμό με την MATLAB, δίνοντας στον χρήστη τη δυνατότητα να χρησιμοποιήσει έτοιμα blocks, είτε για να δημιουργήσει δικά του στην γλώσσα MATLAB ή και σε γλώσσες όπως η C και η C++, για τον σχεδιασμό, την ανάλυση και την δοκιμή συστημάτων σε μια πληθώρα πεδίων εφαρμογής, όπως τα ηλεκτρονικά, τα συστήματα ελέγχου και τα αεροθερμodynamικά συστήματα [6].

### 2.1 Έτοιμα blocks Simulink

Στο παρόν υποκεφάλαιο περιγράφονται τα έτοιμα block του Simulink τα οποία χρησιμοποιήθηκαν στην παρούσα εργασία. Κάθε block διαθέτει παραμέτρους που καθορίζουν την συμπεριφορά τους. Ο χρήστης έχει την δυνατότητα να ρυθμίσει τις παραμέτρους αυτές μέσω ενός μενού το οποίο ανοίγει κάνοντας διπλό κλικ στο επιθυμητό block. Εδώ αξίζει να σημειωθεί ότι σε όλη την έκταση της παρούσας εργασίας, οι λέξεις στις οποίες χρησιμοποιείται η γραμματοσειρά Latin Modern Sans Typewriter αναφέρονται σε μεταβλητές, συναρτήσεις, προγράμματα κλπ. της Matlab ή του Simulink.

Αρχικά, το Constant block δημιουργεί ένα σταθερό σήμα με τιμή που καθορίζεται από την παράμετρο Constant value, ενώ το Terminator block "σταματάει" ένα οποιοδήποτε σήμα (βλ. **Σχήμα 1**).



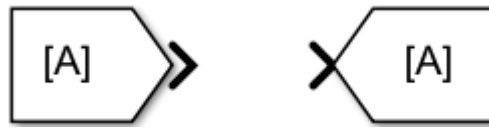
Σχήμα 1: Constant και Terminator blocks

Στη συνέχεια, τα Sum, Subtract και Product blocks χρησιμοποιούνται για την πρόσθεση, αφαίρεση ή πολλαπλασιασμό αντίστοιχα δύο ή περισσότερων σημάτων (βλ. **Σχήμα 2**). Ο αριθμός των σημάτων και τα πρόσημά τους στην αντίστοιχη πράξη καθορίζονται ρυθμίζοντας κατάλληλα την παράμετρο του αντίστοιχου block.



Σχήμα 2: Sum, Subtract και Product blocks

Ακόμη, τα From και Goto blocks χρησιμοποιούνται για την λήψη και την αποστολή αντίστοιχα σημάτων ανάμεσα σε ένα ζευγάρι από αυτά με την ίδια ετικέτα, η οποία καθορίζεται ως παράμετρος (βλ. **Σχήμα 3**). Η χρήση τους κρίνεται απαραίτητη ιδιαίτερα σε περίπλοκα σχηματικά διαγράμματα, καθώς απλοποιούν τις συνδέσεις ανάμεσα στα διάφορα blocks και καθιστούν το διάγραμμα πιο ευανάγνωστο.



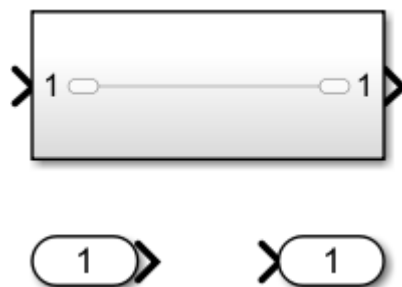
Σχήμα 3: From και Goto blocks

Επιπλέον, τα Mux και Demux blocks χρησιμεύουν για την ένωση ή τον διαχωρισμό, αντίστοιχα, ενός αριθμού σημάτων που καθορίζεται μέσω της κατάλληλης παραμέτρου τους. Δηλαδή, τα blocks αυτά δημιουργούν ή διαχωρίζουν ένα διάνυσμα. Παράλληλα, το BusSelector block χρησιμοποιείται για την επιλογή ενός ή περισσότερων σημάτων από ένα σήμα τύπου bus, το οποίο είναι ένα σύνολο "επώνυμων" σημάτων. Τα παραπάνω blocks φαίνονται στο **Σχήμα 4**.



Σχήμα 4: Mux, Demux και BusSelector blocks

Επιπροσθέτως, το Subsystem block είναι ιδιαίτερα χρήσιμο καθώς δίνει τη δυνατότητα ομαδοποίησης ενός υποσυνόλου blocks ενός σχηματικού διαγράμματος. Με αυτόν τον τρόπο το υποσύνολο αυτό μπορεί να θεωρηθεί ως ένα "μαύρο κουτί" (black box) με εισόδους και εξόδους καθοριζόμενες εισάγοντας στο Subsystem κατάλληλα Inport και Outport blocks (βλ. **Σχήμα 5**). Χρησιμοποιώντας με κατάλληλο τρόπο Subsystem blocks ένα σχηματικό διάγραμμα απλοποιείται σημαντικά και γίνεται πιο εύκολα κατανοητό.



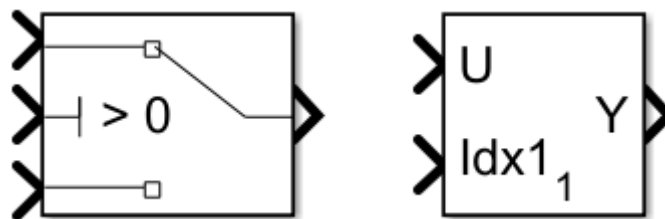
Σχήμα 5: Subsystem, Inport και Outport blocks

Ακόμη, το Function block (βλ. **Σχήμα 6**) δίνει τη δυνατότητα εισαγωγής μιας συνάρτησης MATLAB σε ένα σχηματικό διάγραμμα.



Σχήμα 6: Function block

Στη συνέχεια, το Switch block έχει ως έξοδο το σήμα που είναι συνδεδεμένο στην πρώτη θύρα εισόδου αν το σήμα της δεύτερης εισόδου ικανοποιεί την συνθήκη που καθορίζεται από την παράμετρο του block, ενώ σε αντίθετη περίπτωση το σήμα που είναι συνδεδεμένο στην τρίτη θύρα εισόδου. Δηλαδή, το block αυτό χρησιμεύει για την επιλογή ανάμεσα σε δύο σήματα σύμφωνα με μία συνθήκη. Παράλληλα, το Selector block χρησιμοποιείται για την επιλογή ή την αναδιάταξη των στοιχείων ενός διανύσματος που εισάγεται στην πρώτη του θύρα σύμφωνα με το διάνυσμα δεικτών που εισάγεται στην δεύτερη. Τα δύο αυτά blocks παρουσιάζονται στο **Σχήμα 7**.



Σχήμα 7: Switch και Selector blocks

Τέλος, το Display block δίνει την δυνατότητα εποπτείας της τιμής ενός σήματος κατά την διάρκεια μίας προσομοίωσης στο περιβάλλον Simulink, ενώ το ToWorkspace block στέλνει την τιμή ενός σήματος σε κάθε βήμα εκτέλεσης μιας προσομοίωσης στον χώρο εργασίας (workspace) της MATLAB με όνομα καθορισμένο από τον χρήστη (βλ. **Σχήμα 8**).

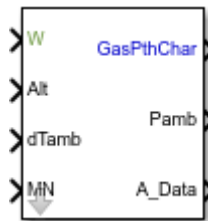


Σχήμα 8: Display και ToWorkspace blocks

## 2.2 Blocks της βιβλιοθήκης T-MATS

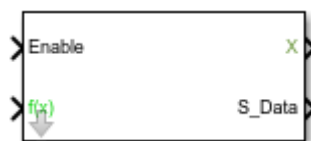
Το Simulink διαθέτει μια πληθώρα βιβλιοθηκών (libraries), οι οποίες είναι συλλογές από blocks. Στην παρούσα εργασία αξιοποιήθηκε η βιβλιοθήκη T-MATS, η οποία έχει αναπτυχθεί από την NASA και αποτελεί ένα πακέτο blocks για θερμοδυναμικές προσομοιώσεις δυναμικού χαρακτήρα. Δίνει τη δυνατότητα στον χρήστη να δημιουργήσει προσομοιώσεις θερμοδυναμικών συστημάτων, όπως οι αεροστρόβιλοι, αλλά και ελεγκτών σε ένα μοναδικό εργαλείο. Η δημιουργία τέτοιων προσομοιώσεων μπορεί να επιτευχθεί κάνοντας χρήση δεδομένων, είτε από πραγματικά συστήματα, είτε από μοντέλα μόνιμης κατάστασης στα οποία εισάγονται οι κατάλληλες δυναμικές [1].

Στην παρούσα εργασία αξιοποιήθηκαν τα ακόλουθα blocks από την βιβλιοθήκη T-MATS: Αρχικά, το Ambient block (βλ. **Σχήμα 9**) χρησιμοποιείται για την μετατροπή των περιβαλλοντικών συνθηκών (αριθμός Mach, θερμοκρασία περιβάλλοντος) στα χαρακτηριστικά της ροής του αερίου. Οι είσοδοί του είναι με τη σειρά: η παροχή αερίου, το υψόμετρο, η διαφορά θερμοκρασίας από την ατμοσφαιρική και ο αριθμός Mach. Οι έξοδοι του είναι με τη σειρά: τα χαρακτηριστικά του αερίου (παροχή, ενθαλπία, ολική θερμοκρασία, ολική πίεση και λόγος καυσίμου-αέρα), η στατική ατμοσφαιρική πίεση και τα διάφορα λοιπά δεδομένα.



Σχήμα 9: Ambient block

Ακόμη, το SS NR Solver w Jacobian Calc block (βλ. **Σχήμα 10**) είναι ένας επιλύτης Newton-Raphson μόνιμης κατάστασης που χρησιμοποιεί την Ιακωβιανή ενός συστήματος μεταβλητών με σκοπό τον μηδενισμό των σφαλμάτων τους. Οι είσοδοί του είναι μία σταθερά, η οποία αν είναι ίση με 1 τότε ο επιλύτης είναι ενεργός, και ένα διάνυσμα με τα σφάλματα του συστήματος προς μηδενισμό (εξαρτημένες μεταβλητές). Οι έξοδοι του είναι οι μεταβλητές που αποτελούν την λύση του συστήματος (ανεξάρτητες μεταβλητές) και τα δεδομένα του επιλύτη (κατάσταση σύγκλισης, Ιακωβιανή).



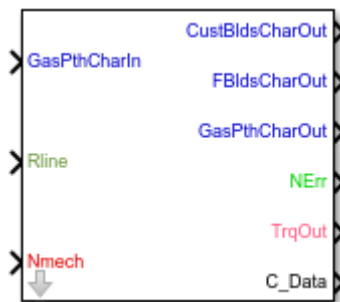
Σχήμα 10: SS NR Solver w Jacobian Calc block

Στη συνέχεια, το Inlet block (βλ. **Σχήμα 11**) προσομοιώνει τη συμπεριφορά ενός αγωγού εισόδου. Ως εισόδους έχει τα χαρακτηριστικά της ροής στη είσοδό του και την στατική ατμοσφαιρική πίεση, ενώ ως έξοδο έχει τα χαρακτηριστικά της ροής στην έξοδό του.



Σχήμα 11: Inlet block

Παράλληλα, το Compressor block (βλ. **Σχήμα 12**) προσομοιώνει τη συμπεριφορά ενός συμπιεστή. Οι εισοδοί του είναι τα χαρακτηριστικά της ροής στην είσοδο του, η θέση στον χάρτη, οι στροφές και προαιρετικά, μετά από ενεργοποίηση από το μενού των παραμέτρων του, οι παράμετροι υγείας του. Οι έξοδοι του είναι οι απομαστεύσεις για τις ανάγκες του αεροσκάφους (customer bleeds), οι απομαστεύσεις για τις ανάγκες του κινητήρα (fractional bleeds), τα χαρακτηριστικά της ροής στην έξοδό του, το κανονικοποιημένο σφάλμα, η ροπή και τα λοιπά δεδομένα.



Σχήμα 12: Compressor block

Επιπλέον, το Splitter block (βλ. **Σχήμα 13**) χρησιμοποιείται για τον διαχωρισμό μίας ροής σε πρωτεύουσα και δευτερεύουσα. Ως εισόδους έχει τα χαρακτηριστικά της ροής στην είσοδό του και τον λόγο παράκαμψης. Ως εξόδους έχει τα δύο ρεύματα με πρώτο το δευτερεύων.



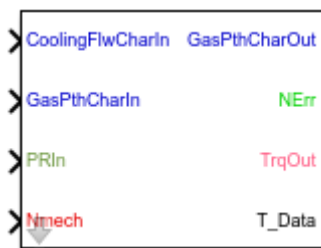
Σχήμα 13: Splitter block

Στη συνέχεια, το Burner block (βλ. **Σχήμα 14**) προσομοιώνει τη συμπεριφορά ενός θαλάμου καύσης. Οι εισοδοί του είναι η παροχή καυσίμου και τα χαρακτηριστικά της ροής στην είσοδό του. Οι έξοδοι του είναι τα χαρακτηριστικά της ροής στην έξοδό του και τα λοιπά δεδομένα.



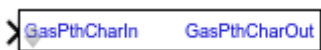
Σχήμα 14: Burner block

Ακόμη, το Turbine NPSS block (βλ. **Σχήμα 15**) προσομοιώνει τη συμπεριφορά ενός στροβίλου για εφαρμογές μόνιμης κατάστασης. Ως εισόδους έχει τα χαρακτηριστικά της ροής ψύξης, τα χαρακτηριστικά της ροής στην είσοδό του, τον λόγο πίεσης και τις στροφές. Ως εξόδους έχει τα χαρακτηριστικά της ροής στην έξοδό του, το κανονικοποιημένο σφάλμα, την ροπή και τα λοιπά δεδομένα.



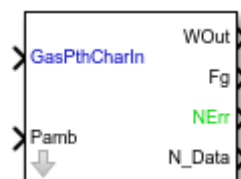
Σχήμα 15: Turbine NPSS block

Επιπλέον, το Duct block (βλ. **Σχήμα 16**) προσομοιώνει τη συμπεριφορά ενός αγωγού και έχει ως είσοδο και έξοδο τα χαρακτηριστικά της ροής στην είσοδο και έξοδο του αντίστοιχα.



Σχήμα 16: Duct block

Επιπροσθέτως, το Nozzle block (βλ. **Σχήμα 17**) προσομοιώνει τη συμπεριφορά ενός ακροφυσίου, το οποίο μπορεί να είναι συγκλίνον ή συγκλίνον-αποκλίνον. Οι εισοδοί του είναι τα χαρακτηριστικά της ροής στην είσοδό του και η στατική ατμοσφαιρική πίεση. Οι εξοδοί του είναι η παροχή, η μικτή ώση, το κανονικοποιημένο σφάλμα και τα λοιπά δεδομένα.



Σχήμα 17: Nozzle block



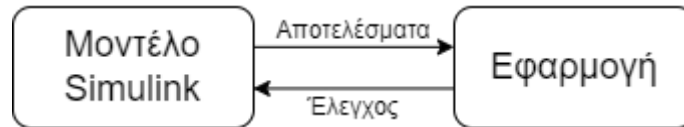
Τέλος, το Shaft block (βλ. **Σχήμα 18**) προσομοιώνει τη συμπεριφορά ενός άξονα. Ως εισόδους έχει την ροπή εισόδου των στροβιλοσυνιστωσών, την απομαστευόμενη ισχύ και τις στροφές. Ως εξόδους έχει τις στροφές και την παράγωγο των στροφών, η οποία κατά την επίλυση στην μόνιμη κατάσταση πρέπει να τείνει στο μηδέν.



Σχήμα 18: Shaft block

### 3 Ανάπτυξη εφαρμογής

Σε αυτό το κεφάλαιο παρουσιάζεται η μεθοδολογία ανάπτυξης του μοντέλου Simulink και της διεπαφής χρήστη της εφαρμογής. Η εφαρμογή δίνει τη δυνατότητα στον χρήστη να ελέγξει το μοντέλο ώστε να πραγματοποιηθεί η επιθυμητή προσομοίωση ή διάγνωση. Το μοντέλο με τη σειρά του επιστρέφει στην διεπαφή χρήστη τα αποτελέσματα, όπως φαίνεται στο **Σχήμα 19**.



Σχήμα 19: Διάγραμμα επικοινωνίας ανάμεσα στην εφαρμογή και στο μοντέλο Simulink

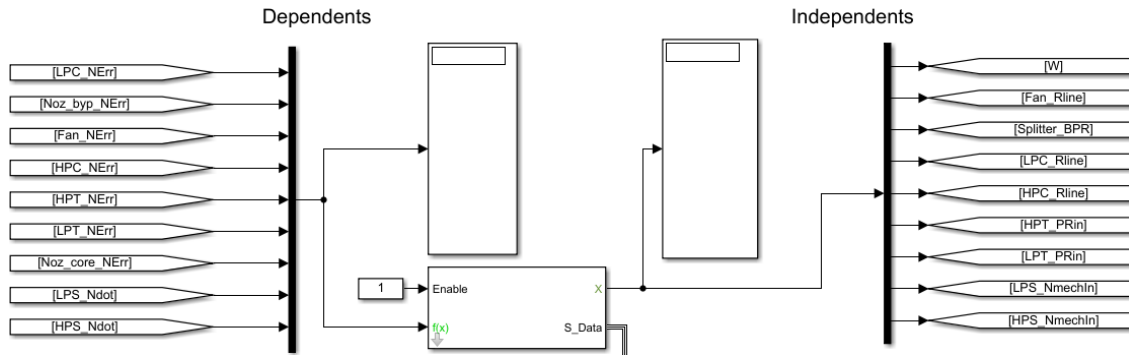
#### 3.1 Μοντέλο Simulink

Πριν την ανάπτυξη της εφαρμογής είναι απαραίτητη η δημιουργία ενός μοντέλου στο λογισμικό Simulink, το οποίο αποτελεί τον πυρήνα της εφαρμογής. Κάνοντας χρήση του μοντέλου αυτού και μέσω της εφαρμογής που αναπτύχθηκε, ο χρήστης έχει την δυνατότητα να πραγματοποιήσει προσομοιώσεις και διάγνωση στον κινητήρα στον οποίο βασίζεται το μοντέλο. Το μοντέλο που αναπτύχθηκε βασίζεται στο μοντέλο JT9D\_Model\_SS που παρέχεται από τους δημιουργούς της βιβλιοθήκης T-MATS ως παράδειγμα[2, 3], το οποίο απαιτεί να είναι φορτωμένο στο χώρο εργασίας (workspace) το αρχείο MWS\_SS.mat ώστε να αρχικοποιηθούν όλες οι απαραίτητες μεταβλητές. Το μοντέλο αυτό βασίζεται στον κινητήρα JT9D της εταιρείας Pratt & Whitney και είναι μοντέλο τύπου μόνιμης κατάστασης. Ο κινητήρας αυτός αποτελείται από τις ακόλουθες αεροθερμοδυναμικές συνιστώσες:

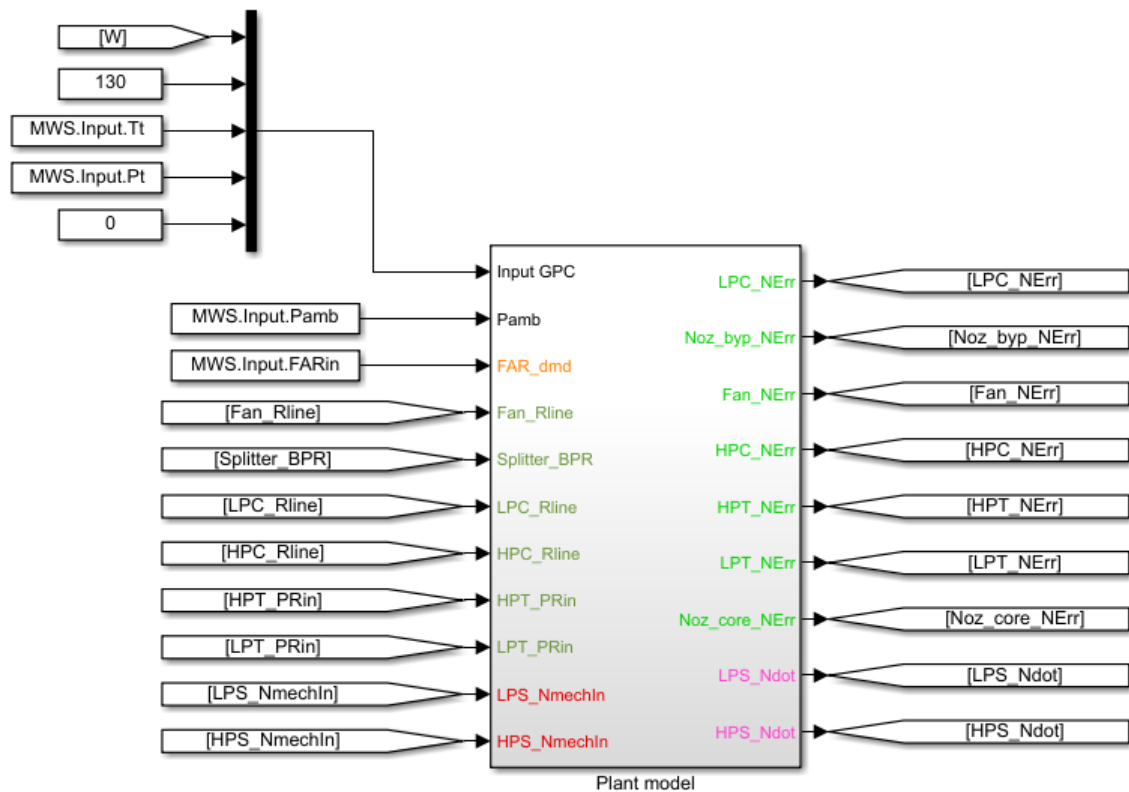
- Ανεμιστήρας (Fan)
- Συμπιεστής χαμηλής πίεσης (Low Pressure Compressor - LPC)
- Συμπιεστής υψηλής πίεσης (High Pressure Compressor - HPC)
- Στρόβιλος υψηλής πίεσης (High Pressure Turbine - HPT)
- Στρόβιλος χαμηλής πίεσης (Low Pressure Turbine - LPT)

Το μοντέλο περιέχει αρχικά έναν επιλύτη Newton-Raphson (βλ. **Σχήμα 20**), ο οποίος έχει ως εισόδους τις εξαρτημένες μεταβλητές που με τη σειρά τους προέρχονται από το υπο-μοντέλο του κινητήρα (Plant Model Subsystem βλ. **Σχήμα 21**) και ως εξόδους τις ανεξάρτητες μεταβλητές που καθορίζουν την συμπεριφορά του κινητήρα (παροχή, στροφές, θέσεις στους χάρτες και λόγος παράκαμψης). Οι εξαρτημένες μεταβλητές είναι τα σφάλματα που προέρχονται από τις στροβιλοσυνιστώσες

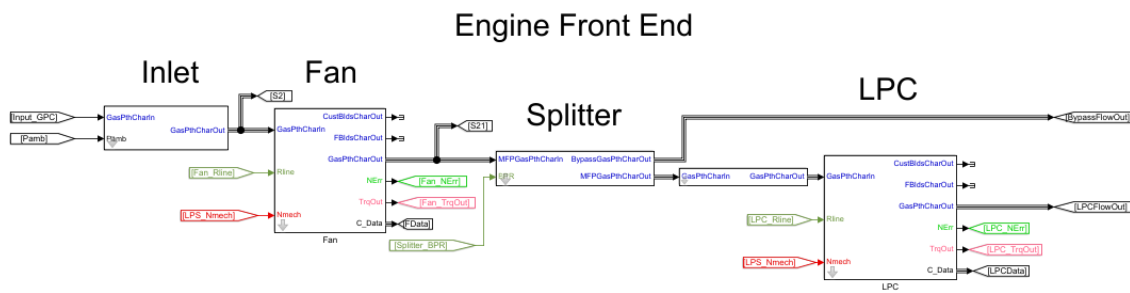
και οι παράγωγοι των στροφών, γεγονός που σημαίνει ότι στην μόνιμη κατάσταση πρέπει να τείνουν στο μηδέν. Τέλος, μέσα στο Plant Model Subsystem μοντελοποιούνται οι διάφορες συνιστώσες του κινητήρα, όπως φαίνεται στα Σχήματα 22 έως 26.



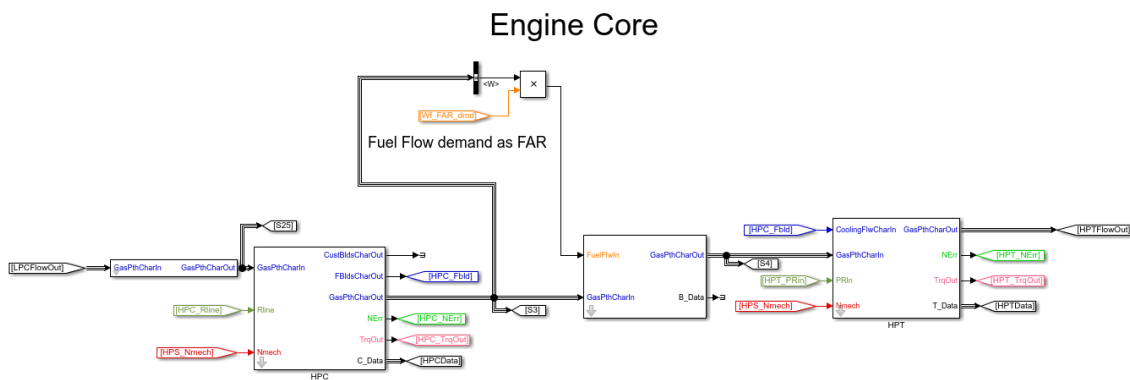
Σχήμα 20: Σχηματικό διάγραμμα του επιλύτη στο αρχικό μοντέλο Simulink



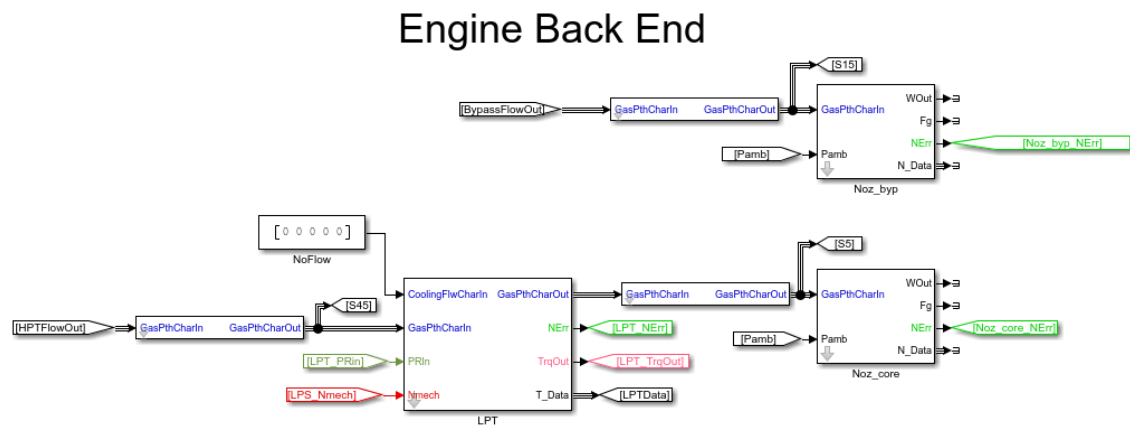
Σχήμα 21: Σχηματικό διάγραμμα του υπο-μοντέλου του κινητήρα στο αρχικό μοντέλο Simulink



Σχήμα 22: Σχηματικό διάγραμμα του εμπρός μέρους του κινητήρα στο αρχικό μοντέλο Simulink

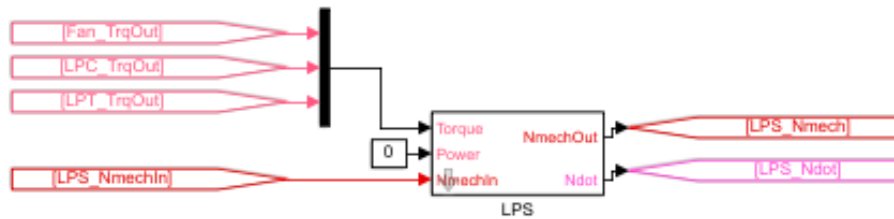


Σχήμα 23: Σχηματικό διάγραμμα του πυρήνα του κινητήρα στο αρχικό μοντέλο Simulink



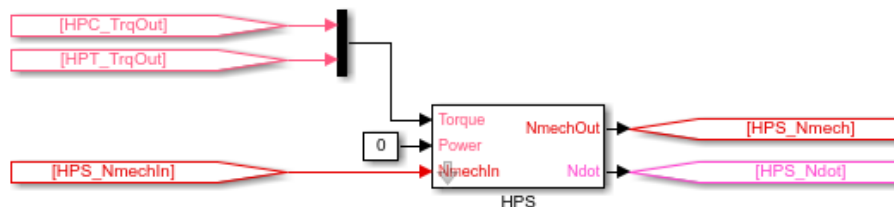
Σχήμα 24: Σχηματικό διάγραμμα του πίσω μέρους του κινητήρα στο αρχικό μοντέλο Simulink

## LP Shaft Model



Σχήμα 25: Σχηματικό διάγραμμα του άξονα χαμηλής πίεσης του κινητήρα στο αρχικό μοντέλο Simulink

## HP Shaft Model



Σχήμα 26: Σχηματικό διάγραμμα του άξονα υψηλής πίεσης του κινητήρα στο αρχικό μοντέλο Simulink

Για να έχει η εφαρμογή που αναπτύχθηκε την απαιτούμενη λειτουργικότητα, ήταν απαραίτητο να προστεθούν στο μοντέλο Simulink οι ακόλουθες δυνατότητες:

1. Καθορισμός του σημείου λειτουργίας μέσω του υψόμετρου, του αριθμού Mach και της διαφοράς θερμοκρασίας με εκείνη της τυπικής μέρας
2. Επιλογή μιας μεταβλητής ως "μεταβλητή ελέγχου" και καθορισμός της τιμής της στο σημείο λειτουργίας
3. Δημιουργία υπογραφής βλαβών του κινητήρα
4. Πραγματοποίηση διάγνωσης βλαβών με την χρήση των παραμέτρων υγείας των στροβιλοσυνιστωσών

Οι τροποποιήσεις που πραγματοποιήθηκαν στο μοντέλο αυτό παρουσιάζονται στις παραγράφους που ακολουθούν.

### 3.1.1 Μεταβλητές σημείου λειτουργίας

Αρχικά, καθώς ήταν θεμιτό οι μεταβλητές που καθορίζουν το σημείο λειτουργίας να είναι το υψόμετρο, η διαφορά θερμοκρασίας με εκείνη της τυπικής ημέρας και ο

αριθμός Mach, προστέθηκε στο μοντέλο ένα Ambient block, καθώς και 3 Constant blocks, ένα για καθεμία από τις μεταβλητές αυτές. Το όνομα του κάθε Constant block και η τιμή των μεταβλητών αυτών στο σημείο σχεδιασμού[1], η οποία τίθεται στην ιδιότητα Value του, παρουσιάζονται στον **Πίνακα 1**.

Πίνακας 1: Όνομα block και τιμή στο σημείο σχεδιασμού για τις μεταβλητές του σημείου λειτουργίας

Μεταβλητή	Όνομα	Τιμή
Υψόμετρο	input_alt	34000 ft
Διαφορά θερμοκρασίας	input_dt	0 °R
Αριθμός Mach	input_mach	0.8

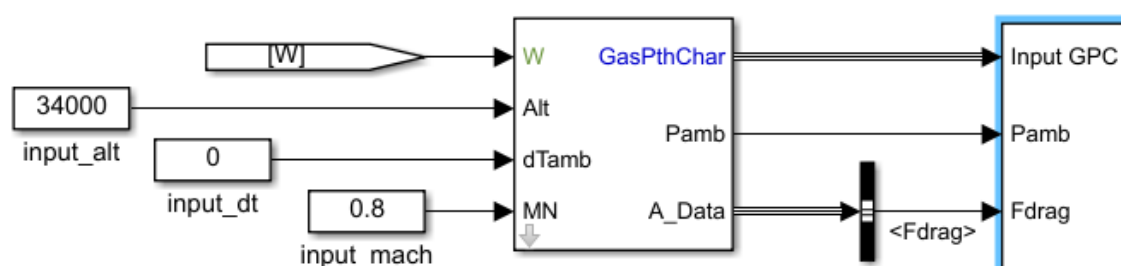
Στις εισόδους του Ambient block συνδέθηκαν:

- Στην θύρα εισόδου W ένα From block το οποίο αναφέρεται στο Goto block με όνομα [W] των ανεξάρτητων μεταβλητών και αποτελεί την τιμή της παροχής που εισέρχεται στον κινητήρα
- Το κάθε Constant block στην κατάλληλη θύρα εισόδου

Οι έξοδοι του Ambient block συνδέθηκαν:

- Η θύρα εξόδου GasPthChar με την θύρα εισόδου Input GPC του Plant Model Subsystem
- Η θύρα εξόδου Pamb με τη θύρα εισόδου με το ίδιο όνομα του Plant Model Subsystem
- Η θύρα εξόδου A\_Data, μέσω ενός Bus Selector block στο οποίο επιλέγεται η μεταβλητή <Fdrag>, στην θύρα Fdrag, η οποία δημιουργήθηκε τοποθετώντας ένα Inport block μέσα στο Plant Model Subsystem με το κατάλληλο όνομα

Τα παραπάνω παρουσιάζονται στο **Σχήμα 27**.

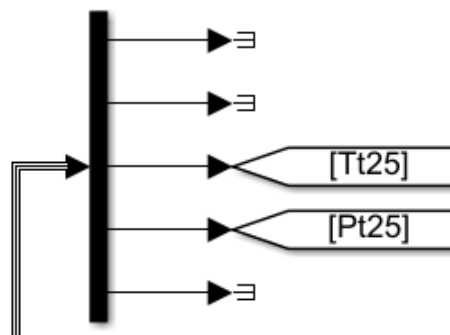


Σχήμα 27: Σχηματικό διάγραμμα της εφαρμογής των μεταβλητών σημείου λειτουργίας στο Simulink

### 3.1.2 Διαθέσιμες μετρήσεις του κινητήρα

Στη συνέχεια, δημιουργήθηκε το "σήμα" των διαθέσιμων μετρήσεων του κινητήρα με σκοπό την πραγματοποίηση διάγνωσης και την υλοποίηση της μεταβλητής εισόδου ελέγχου, όπως θα περιγραφεί σε επόμενη παράγραφο. Οι μεταβλητές ("μετρήσεις") που επιλέχθηκαν ώστε να δημιουργηθεί η υπογραφή του κινητήρα παρατίθενται στον **Πίνακα 2**.

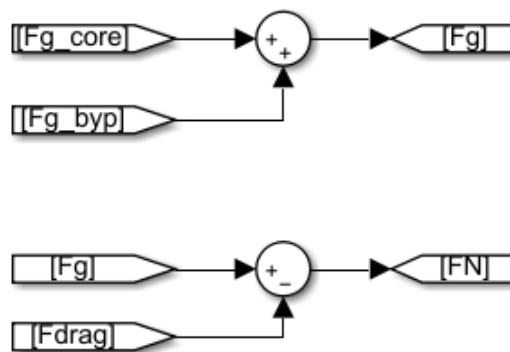
Για να ληφθούν οι τιμές των απαιτούμενων ολικών θερμοκρασιών και πιέσεων, καθώς και της παροχής στην είσοδο του κινητήρα, το σήμα σε κάθε μία από τις αντίστοιχες θέσεις (όπως περιγράφεται στον **Πίνακα 2**) εισήχθη σε ένα Demux block, όπου στην ιδιότητα Number of outputs εισήχθη ο αριθμός 5, καθώς το σήμα αποτελεί ένα Bus με 5 μεταβλητές κατά σειρά: παροχή, ενθαλπία, ολική θερμοκρασία, ολική πίεση και λόγος καυσίμου-αέρα. Οπότε, στις εξόδους των Demux blocks συνδέθηκαν Goto blocks με κατάλληλο όνομα για τις μεταβλητές που ανήκουν στην υπογραφή και Terminator blocks σε εκείνες που δεν ανήκουν σε αυτή με σκοπό την αποφυγή παρουσίας σφάλματος από το λογισμικό Simulink (πχ. βλ. **Σχήμα 28**). Ακόμη, για να ληφθεί η παροχή καυσίμου, το σήμα που εισέρχεται στην είσοδο FuelFlwIn του θαλάμου καύσης συνδέθηκε με ένα Goto block, ενώ τα απαραίτητα Goto blocks για τις στροφές υπάρχουν ήδη στο μοντέλο (LPS\_Nmech και HPS\_Nmech). Τέλος, η καθαρή ώση προέκυψε προσθέτοντας το σήμα που προέρχεται από την θύρα εξόδου Fg του κύριου ακροφυσίου με το αντίστοιχο του δευτερεύοντος και αφαιρώντας από αυτό το σήμα Fdrag που εισάγεται στο Plant Model Subsystem. Αυτή η πράξη πραγματοποιήθηκε μέσω ενός Sum block, όπου στην ιδιότητα List of signs εισήχθη "|++-" και συνδέθηκαν τα σήματα στις κατάλληλες εισόδους. Το αποτέλεσμα εισήχθη σε ένα Goto block, όπως φαίνεται στο **Σχήμα 29**. Οπότε, για τα 10 Goto blocks, εισάγονται στο Plant Model Subsystem τα αντίστοιχα From blocks και τα σήματά τους συνδέθηκαν με την σειρά που παρουσιάστηκαν στον **Πίνακα 2** σε ένα Mux block, του οποίου η ιδιότητα Number of inputs τέθηκε ίση με 10. Η έξοδος αυτού συνδέθηκε με ένα Outport block με όνομα signatures, ώστε να δημιουργηθεί μία ακόμη θύρα εξόδου στο Plant Model Subsystem, η χρησιμότητα της οποίας θα περιγραφεί σε παρακάτω παράγραφο.



Σχήμα 28: Παράδειγμα λήψης μεταβλητών υπογραφής από μία θέση του κινητήρα

Πίνακας 2: Διαθέσιμες μετρήσεις του κινητήρα

Μεταβλητή	Περιγραφή
N_LP	Στροφές άξονα χαμηλής πίεσης
N_HP	Στροφές άξονα υψηλής πίεσης
Tt25	Ολική θερμοκρασία ανάμεσα στον LPC και HPC
Pt25	Ολική πίεση ανάμεσα στον LPC και HPC
Tt3	Ολική θερμοκρασία ανάμεσα στον HPC και στον θάλαμο καύσης
Pt3	Ολική πίεση ανάμεσα στον HPC και στον θάλαμο καύσης
Tt45	Ολική θερμοκρασία ανάμεσα στον HPT και LPT
Wf	Παροχή καυσίμου
FN	Καθαρή ώση
W	Παροχή αέρα στην είσοδο του κινητήρα



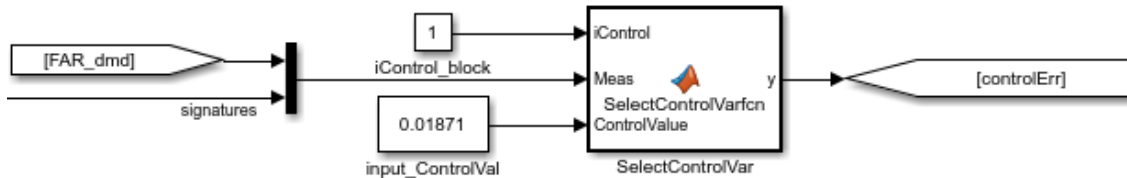
Σχήμα 29: Σχηματικό διάγραμμα του υπολογισμού της καθαρής ώσης στο Simulink

### 3.1.3 Μεταβλητή εισόδου ελέγχου

Επόμενο βήμα ήταν η υλοποίηση της μεταβλητής εισόδου ελέγχου ώστε να υπολογιστεί ο απαιτούμενος λόγος καυσίμου-αέρα σε κάθε χρονικό βήμα εκτέλεσης του μοντέλου. Για τον σκοπό αυτό αρχικά προστέθηκε στο μοντέλο ένα From block με όνομα controlErr στο Mux block των ανεξάρτητων μεταβλητών, το οποίο αποτελεί το σήμα του σφάλματος της μεταβλητής ελέγχου, καθώς και ένα Goto block με όνομα FAR\_dmd στο Demux block των εξαρτημένων μεταβλητών, το οποίο αποτελεί το σήμα του υπολογιζόμενου απαιτούμενου λόγου καυσίμου-αέρα. Στη συνέχεια, το σήμα από ένα From block με όνομα FAR\_dmd και το σήμα από την θύρα εξόδου signatures του Plant Model Subsystem συνδέθηκαν με αυτή τη σειρά σε ένα Mux block με 2 εισόδους. Ακολούθως, προστέθηκε στο μοντέλο ένα MATLAB Function block, η συνάρτηση (function) του οποίου είναι η SelectControlVarfcn (βλ. **Παράρτημα Ι**), ώστε να υπολογιστεί το σφάλμα της μεταβλητής εισόδου ελέγχου σε κάθε χρονικό βήμα εκτέλεσης του μοντέλου. Ακόμη, προστέθηκαν 2 Constant blocks με όνομα iControl\_block και input\_ControlVal με αρχική τιμή 1 και 0.01871 (όπως δίνεται στο MWS) αντίστοιχα. Ενώ, επιπλέον, συνδέθηκαν στις θύρες εισόδου του παραπάνω MATLAB Function block, iControl και ControlValue



αντίστοιχα. Τα δύο blocks αυτά αντιπροσωπεύουν αντίστοιχα τον δείκτη και την τιμή της μεταβλητής ελέγχου. Τέλος, συνδέθηκε: το σήμα του Mux block στην είσοδο Meas του MATLAB Function block, η έξοδος  $y$  σε ένα Goto block με όνομα controlErr και ένα From block με όνομα FAR\_dmd στην θύρα εισόδου FAR\_dmd του Plant Model Subsystem, όπως φαίνεται στο **Σχήμα 30**.

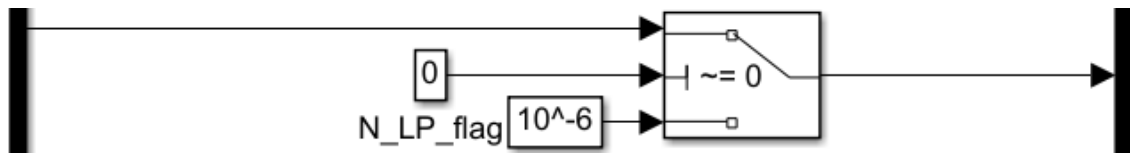


Σχήμα 30: Σχηματικό διάγραμμα της εφαρμογής της μεταβλητής ελέγχου στο Simulink

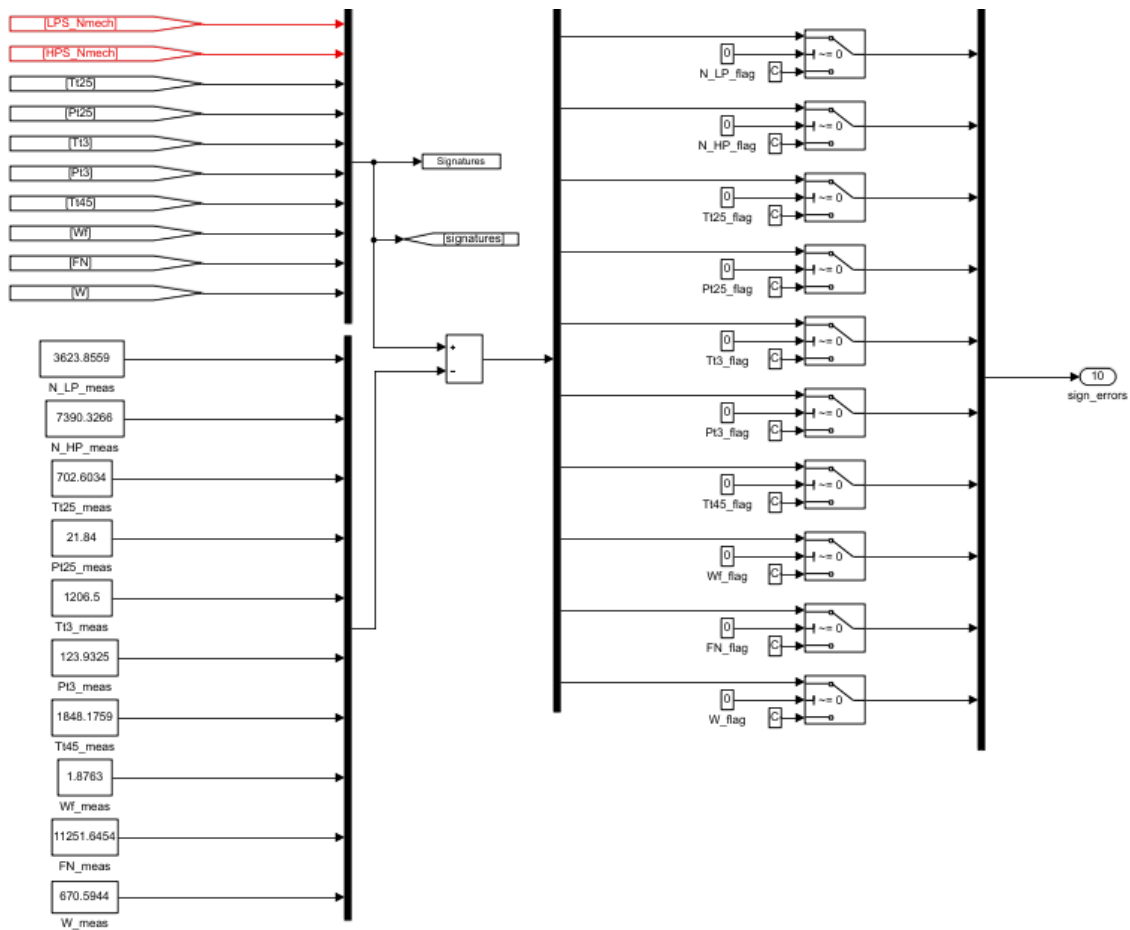
### 3.1.4 Σφάλματα των μετρήσεων

Ακολούθως, υλοποιήθηκε ο υπολογισμός των σφαλμάτων των μετρήσεων ώστε να μπορούν να χρησιμοποιηθούν κατά την διάγνωση. Αρχικά, προστέθηκε στο Plant Model Subsystem ένα Constant block για κάθε μεταβλητή με όνομα το σύμβολό της με κατάληξη "\_meas" (π.χ. Tt25\_meas). Σε αυτά ως τιμή τίθεται, κατά περίπτωση, η "πραγματική μέτρηση" για την εκάστοτε μεταβλητή ώστε να μπορεί να πραγματοποιηθεί διάγνωση χρησιμοποιώντας αυτή την τιμή. Τα σήματα των Constant blocks συνδέθηκαν με τη σειρά που παρουσιάζονται στον **Πίνακα 2** σε ένα Mux block, το σήμα εξόδου του οποίου συνδέθηκε στην αρνητική θύρα εισόδου ενός Subtract block. Ενώ, στην θετική του θύρα συνδέθηκε το σήμα του Mux block των μετρήσεων. Συνακόλουθα, με σκοπό να καταστεί δυνατή η επιλογή των επιθυμητών μετρήσεων για την πραγματοποίηση διάγνωσης, υλοποιήθηκαν τα ακόλουθα. Κατ' αρχάς, το σήμα από το Subtract block συνδέθηκε σε ένα Demux block με 10 εξόδους. Κάθε μία από τις εξόδους αυτές συνδέθηκε στην πρώτη θύρα ενός Switch block, όπου στην ιδιότητα Criteria for passing first input επιλέχθηκε η επιλογή " $u2 \sim 0$ ". Στην τρίτη θύρα καθενός από τα αυτά συνδέθηκε ένα Constant block με τιμή ίση με  $10^{-6}$ . Στην δεύτερη θύρα συνδέθηκε από ένα ακόμη Constant block με όνομα το σύμβολο της κάθε μεταβλητής με κατάληξη "\_flag" (π.χ. N\_LP\_flag) (βλ. **Σχήμα 31**). Το σήμα αυτού του Constant block χρησιμοποιείται ως "σήμα ελέγχου". Πιο συγκεκριμένα, αν η τιμή του είναι διάφορη του μηδενός, το σήμα εξόδου του αντίστοιχου Switch block είναι αυτό που δέχεται από την πρώτη θύρα, ενώ αν η τιμή είναι ίση με 0 το σήμα εξόδου είναι ίσο με αυτό της τρίτης θύρας, δηλαδή είναι θα είναι ίσο με  $10^{-6}$ . Έτσι, τα σήματα εξόδου των Switch blocks περιέχουν σημαντικό σφάλμα μόνο για τις μεταβλητές που επιθυμεί ο χρήστης, ενώ για τις υπόλοιπες το σφάλμα έχει μια απειροστή τιμή και μπορεί να θεωρηθεί σαν έχει συγκλίνει από τον επιλύτη, όπως θα περιγραφεί παρακάτω. Τέλος, τα σήματα εξόδου των Switch blocks συνδέονται με τη σειρά σε ένα Mux block, η έξοδος του οποίου ενώνεται με ένα Outport block με όνομα sign\_errors, δημιουργώντας έτσι μία επιπλέον θύρα εξόδου στο Plant Model Subsystem (βλ. **Σχήμα 32**). Το

σήμα της θύρας εξόδου αυτής μέσω ενός ζεύγους Goto και From blocks συνδέθηκε στο Mux block των εξαρτημένων μεταβλητών.



Σχήμα 31: Υλοποίηση επιλογής σφαλμάτων στο Simulink



Σχήμα 32: Υλοποίηση των σφαλμάτων των μετρήσεων στο Simulink

### 3.1.5 Επιλογή σημάτων

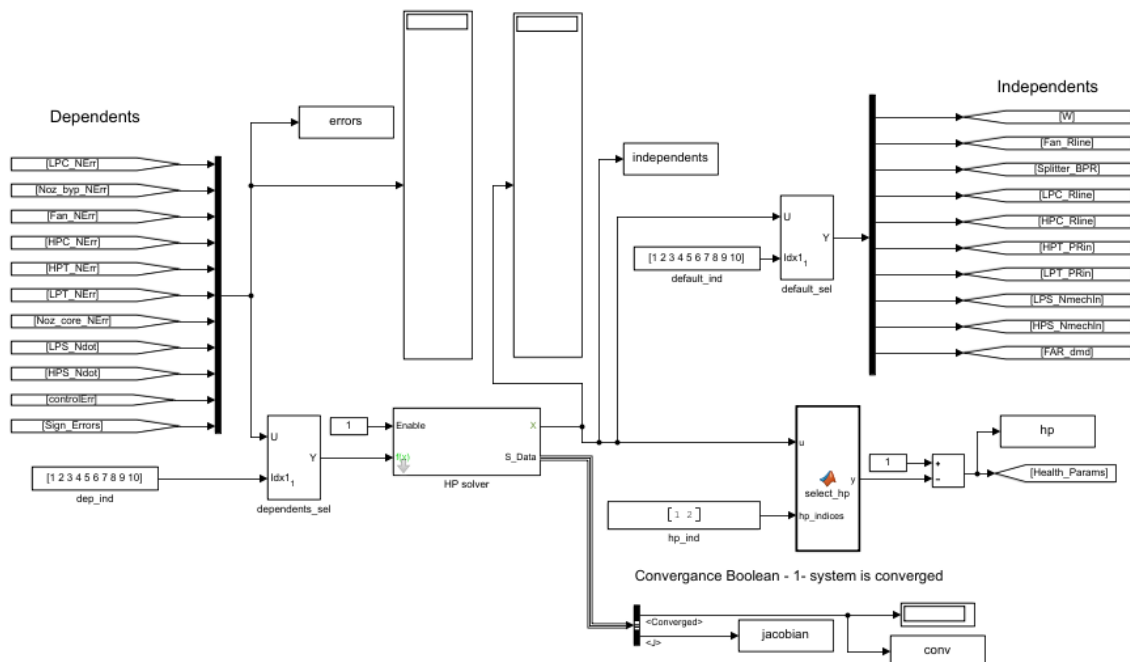
Συνακόλουθα, με σκοπό να καταστεί δυνατή η επιλογή τόσο των σημάτων που εισέρχονται στον επιλύτη (solver), δηλαδή τις εξαρτημένες μεταβλητές, όσο και των παραμέτρων υγείας που είναι επιθυμητό να εκτιμηθούν κατά τη διάγνωση, προστέθηκαν τα ακόλουθα στο μοντέλο (βλ. **Σχήμα 33**). Αρχικά, εισήχθηκε ένα Selector block με παραμέτρους:

- Number of input dimensions ίση με 1
- Index mode επιλέχθηκε η επιλογή "One-based"
- Index option επιλέχθηκε η επιλογή "Index vector (port)"
- Input port size ίση με -1

Στην θύρα εισόδου U συνδέθηκε το σήμα του Mux block των εξαρτημένων μεταβλητών, ενώ στη Idx1 ένα Constant block με όνομα dep\_ind και αρχική τιμή "[1 2 3 4 5 6 7 8 9 10]". Το block αυτό δίνει την δυνατότητα επιλογής από τον χρήστη των σημάτων που είναι θεμιτό να χρησιμοποιηθούν από τον επιλύτη. Η αρχική τιμή που έχει τεθεί είναι εκείνη για την πραγματοποίηση προσομοίωσης. Σε αυτή ο χρήστης καλείται να προσθέσει τους κατάλληλους αριθμούς (11 - 20) ώστε να επιλέξει τις απαραίτητες μετρήσεις για την διάγνωση. Για παράδειγμα, αν είναι επιθυμητή η διάγνωση κάνοντας χρήση των στροφών χαμηλής πίεσης και της καθαρής ώσης (1η και 9η μεταβλητή της υπογραφής), η τιμή που πρέπει να τεθεί στο dep\_ind είναι "[1 2 3 4 5 6 7 8 9 10 11 19]", δηλαδή προσθέτοντας τον αριθμό 10 στη σειρά που παρουσιάζεται η κάθε μεταβλητή στον **Πίνακα 2** προκύπτει ο κατάλληλος δείκτης (index) που πρέπει να εισαχθεί. Ακόμη, το σήμα εξόδου του dep\_ind συνδέθηκε στη θύρα εισόδου f(x) του επιλύτη. Στη συνέχεια, ήταν απαραίτητος ο διαχωρισμός των σημάτων που αναφέρονται στις παραμέτρους υγείας από το σήμα εξόδου X του επιλύτη (ανεξάρτητες μεταβλητές). Το σήμα αυτό συνδέθηκε στην θύρα εισόδου U ενός νέου Selector block με τις ίδιες τιμές παραμέτρων με το προηγούμενο, ενώ στην Idx1 ένα Constant block με την ίδια τιμή, ώστε το σήμα εξόδου να περιλαμβάνει μόνο τις ανεξάρτητες μεταβλητές που δεν είναι παράμετροι υγείας. Το σήμα αυτό συνδέθηκε στο Demux block των ανεξάρτητων μεταβλητών. Ακόμη, προστέθηκε ένα MATLAB Function block, η συνάρτηση του οποίου είναι η select\_hp (βλ. **Παράρτημα I**). Η συνάρτηση αυτή, ανάλογα με τους δείκτες (indices) που λαμβάνει ως είσοδο, θέτει σε κάθε στοιχείο του σήματος εξόδου της, το οποίο έχει μέγεθος ίσο με τον συνολικό αριθμό των πιθανών παραμέτρων υγείας: την τιμή της κάθε παραμέτρου υγείας, όπως αυτή προκύπτει μέσω του επιλύτη, που έχει επιλεγεί, στον σωστό δείκτη (index) του σήματος εξόδου και τιμή ίση με 1 στις μη επιλεγμένες παραμέτρους. Επιπλέον, στην θύρα εισόδου u του MATLAB Function block συνδέθηκε το σήμα εξόδου X του επιλύτη, ενώ στην hp\_indices ένα Constant block με όνομα hp\_ind. Σε αυτό ο χρήστης, κατά τη διάγνωση, καλείται να εισάγει έναν μονοδιάστατο πίνακα (array) που περιέχει τους δείκτες των απαραίτητων παραμέτρων υγείας. Οι παράμετροι υγείας που επιλέχθηκε να χρησιμοποιηθούν στην παρούσα εργασία παρουσιάζονται στον **Πίνακα 3**, ενώ ως δείκτης της κάθε παραμέτρου θεωρείται ο αύξων αριθμός της σε αυτόν. Οι παράμετροι υγείας ορίζονται ως η ποσοστιαία μεταβολή της διορθωμένης παροχής ή του βαθμού απόδοσης εκφραζόμενοι ως κλασματικοί αριθμοί. Έπειτα, το σήμα εξόδου της συνάρτησης συνδέθηκε στην αρνητική θύρα εισόδου ενός Subtract block, ενώ στη θετική ένα Constant block με τιμή 1. Η τιμή αυτή θα επεξηγηθεί παρακάτω. Τέλος, η έξοδος του, μέσω ενός ζεύγους Goto και From blocks, συνδέθηκε στην θύρα εισόδου Health\_Params του Plant Model Subsystem, η οποία δημιουργήθηκε εισάγοντας ένα ακόμη Inport block με το κατάλληλο όνομα σε αυτό.

Πίνακας 3: Παράμετροι υγείας του κινητήρα

A/A	Μεταβλητή	Περιγραφή
1	Fan SW	Διορθωμένη παροχή του Fan
2	Fan SE	Βαθμός απόδοσης του Fan
3	LPC SW	Διορθωμένη παροχή του LPC
4	LPC SE	Βαθμός απόδοσης του LPC
5	HPC SW	Διορθωμένη παροχή του HPC
6	HPC SE	Βαθμός απόδοσης του HPC
7	HPT SW	Διορθωμένη παροχή του HPT
8	HPT SE	Βαθμός απόδοσης του HPT
9	LPT SW	Διορθωμένη παροχή του LPT
10	LPT SE	Βαθμός απόδοσης του LPT



Σχήμα 33: Σχηματικό διάγραμμα της υλοποίησης της επίλυσης του συστήματος στο Simulink

### 3.1.6 Επιλύτης

Σε αυτό το σημείο κρίνεται απαραίτητο να αναφερθούν οι παράμετροι του επιλύτη (SS NR Solver w JacobianCalc block). Κατ' αρχάς, στις αρχικές συνθήκες του αλγόριθμου Newton-Raphson (παράμετρος SNR\_IC\_M) τίθεται ένας μονοδιάστατος πίνακας που περιέχει τιμές για τις ανεξάρτητες μεταβλητές (εκτός των παραμέτρων υγείας) που είναι "κοντά" στην αναμενόμενη τελική λύση ώστε να συγκλίνει ο αλγόριθμος. Σχετικά με τις παραμέτρους υγείας, στο τέλος του πίνακα αυτού προστίθεται

ο αριθμός 1 τόσες φορές όσες είναι ο αριθμός των παραμέτρων υγείας που είναι επιθυμητό να εκτιμηθούν. Μολονότι οι συνθήκες τιμές των παραμέτρων υγείας είναι δεκαδικές τιμές κοντά στο 0, εντούτοις επιλέχθηκε ως αρχική εκτίμηση η τιμή 1 για λόγους διευκόλυνσης της σύγκλισης του αλγορίθμου Newton-Raphson. Όμως, για να παρέχονται οι σωστές τιμές των παραμέτρων υγείας στις συνιστώσες του κινητήρα, οι τιμές που προκύπτουν από τον επιλύτη αφαιρούνται από 1, όπως περιγράφτηκε παραπάνω. Επιπλέον, στο μέγεθος διαταραχής (παραμέτρος SJac\_Per\_M) ως τιμή τίθεται ένας δισδιάστατος πίνακας με 2 γραμμές και αριθμό στηλών ίσο με το μήκος του πίνακα των αρχικών συνθηκών. Στην πρώτη γραμμή τίθεται η τιμή 0.01 και στην δεύτερη -0.01. Δηλαδή, σε κάθε μεταβλητή πραγματοποιούνται 2 διαταραχές: μία με τιμή 1% μεγαλύτερη και μία με 1% μικρότερη. Οι υπόλοιπες παράμετροι του επιλύτη συνοψίζονται στον **Πίνακα 4**. Τέλος, με σκοπό την διευκόλυνση της σύγκλισης το χρονικό βήμα (Fixed-step size) τέθηκε ίσο με 0.02 στις ρυθμίσεις του επιλύτη του μοντέλου Simulink (Configuration Parameters/Solver).

Πίνακας 4: Λοιπές παράμετροι επιλύτη

Παράμετρος	Τιμή	Περιγραφή
SNR_DX_M	1	Ποσοστιαίο μέγεθος βήματος Newton-Raphson
SNR_Mx_M	$10^6$	Άνω όριο Newton-Raphson
SNR_Mn_M	$10^{-5}$	Κάτω όριο Newton-Raphson
SJacobianAttempts_M	20	Προσπάθειες Newton-Raphson πριν τον επαναυπολογισμό της ιακωβιανής
Cond_Limits_M	$10^{-4}$	Όριο συνθήκης επανάληψης

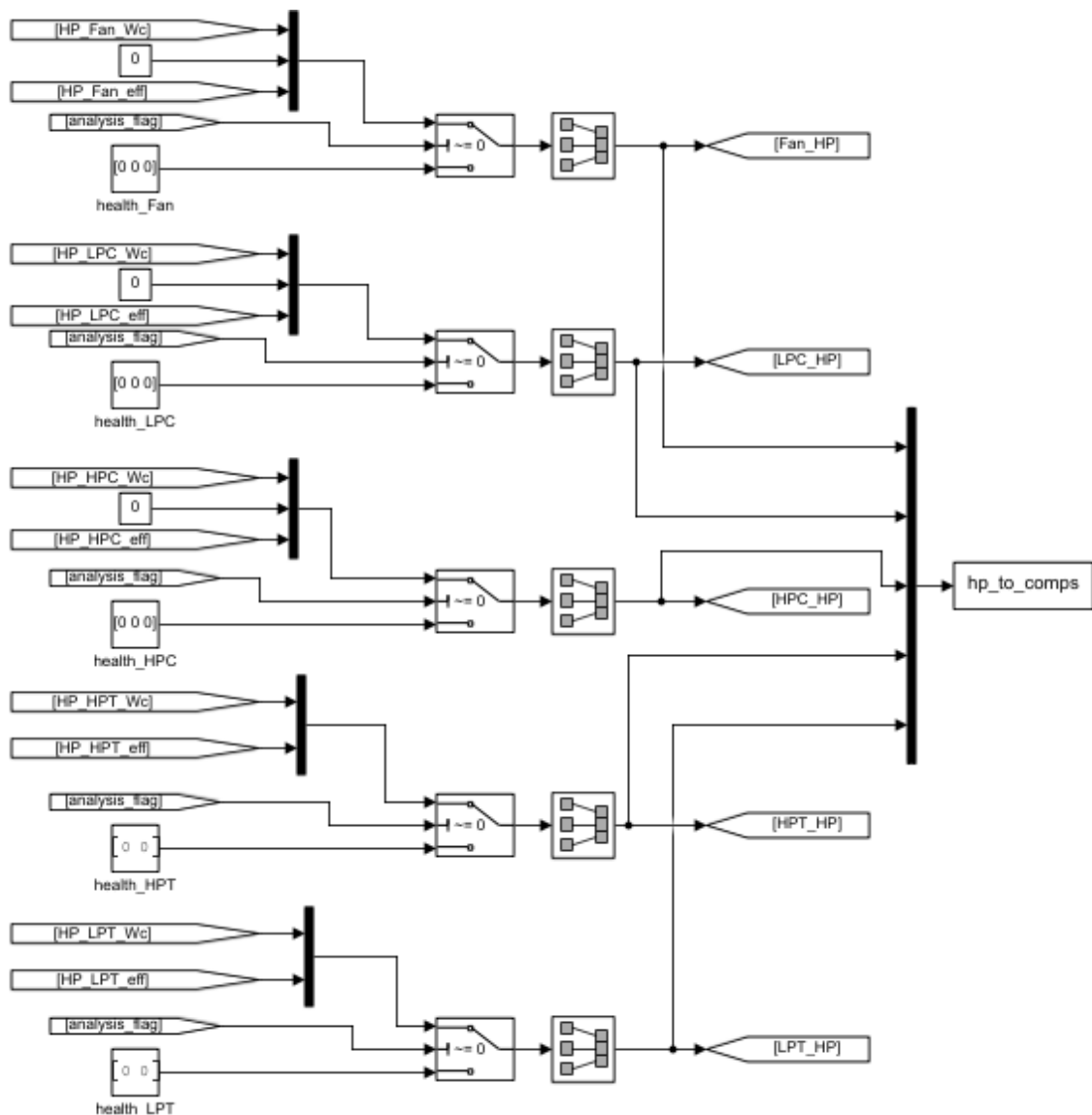
### 3.1.7 Παράμετροι υγείας

Επόμενο βήμα ήταν η τροφοδότηση των παραμέτρων υγείας στις κατάλληλες συνιστώσες κατά την προσομοίωση και την διάγνωση (βλ. **Σχήμα 34**). Αρχικά, τοποθετήθηκε ένα Inport block με όνομα Analysis\_Flag στο Plant Model Subsystem, στην έξοδο του οποίου συνδέθηκε ένα From block με όνομα analysis\_flag. Στην θύρα εισόδου που μόλις δημιουργήθηκε, συνδέθηκε ένα Constant block με όνομα analysis\_flag, με το οποίο δίνεται η δυνατότητα στον χρήστη να επιλέξει αν θα πραγματοποιηθεί προσομοίωση ή διάγνωση. Αν η τιμή του είναι μηδενική τότε στο μοντέλο πραγματοποιείται προσομοίωση, ενώ, σε αντίθετη περίπτωση, διάγνωση. Στη συνέχεια, το σήμα από την είσοδο Health\_Params συνδέθηκε σε ένα Demux block με 10 εξόδους, η κάθε μία από αυτές συνδέθηκε με ένα Goto block με όνομα που να καταδεικνύει σε ποια παράμετρο υγείας αναφέρεται. Έπειτα, εισήχθησαν στο Plant Model Subsystem ένα Mux block για κάθε συνιστώσα του κινητήρα που δέχεται παραμέτρους υγείας, δηλαδή 5. Για τις συνιστώσες που είναι συμπιεστές το Mux block έχει 3 εισόδους, ενώ για εκείνες που είναι στρόβιλοι 2. Σε αυτά, τα From blocks που αντιστοιχούν στα Goto blocks των παραμέτρων υγείας συνδέθηκαν ως εξής: Για την περίπτωση των συμπιεστών, εκείνα που αναφέρονται στην διορθωμένη παροχή συνδέθηκαν στην πρώτη είσοδο, εκείνα που αναφέρονται στον βαθμό

απόδοσης στην τρίτη, ενώ στη δεύτερη ένα Constant block με τιμή 0, καθώς ο λόγος πίεσης δεν χρησιμοποιήθηκε ως παράμετρος υγείας στην παρούσα εργασία. Ενώ, για την περίπτωση των στροβίλων, εκείνα που αναφέρονται στην διορθωμένη παροχή συνδέθηκαν στην πρώτη είσοδο και εκείνα που αναφέρονται στον βαθμό απόδοσης στην δεύτερη. Επιπλέον, η έξοδος καθενός από τα παραπάνω Mux blocks συνδέθηκε με την πρώτη θύρα ενός Switch block, στην παράμετρο Criteria for passing first input του οποίου επιλέχθηκε η επιλογή "u2 ~= 0". Στις δεύτερες θύρες συνδέθηκαν From blocks με όνομα analysis\_flag και στην τρίτη Constant blocks με αρχική τιμή έναν μονοδιάστατο πίνακα με μηδενικές τιμές και μήκος 3 για τους συμπιεστές και 2 για τους στροβίλους, ώστε να μπορεί ο χρήστης να θέτει τις τιμές που επιθυμεί στις παραμέτρους υγείας κατά την προσομοίωση. Στα Constant blocks αυτά δόθηκε ως όνομα το όνομα της συνιστώσας στην οποία αναφέρεται με κατάληξη \_health (π.χ. Fan\_health). Έπειτα, το σήμα εξόδου κάθε Switch block, μέσω ενός Signal Conversion block στην παράμετρο Output το οποίου επιλέχθηκε η επιλογή "Signal Copy", συνδέθηκε με ένα Goto block με όνομα που καταδεικνύει την συνιστώσα στην οποία αναφέρεται. Τέλος, αφότου ενεργοποιήθηκε η παράμετρος HP\_M σε κάθε συνιστώσα, συνδέθηκε το κάθε αντίστοιχο From block στην θύρα εισόδου HP\_Mods της κατάλληλης συνιστώσας.

### 3.1.8 Αποτελέσματα στον χώρο εργασίας (workspace)

Τελευταίο βήμα αποτέλεσε η προσθήκη κατάλληλων ToWorkspace blocks ώστε τα αποτελέσματα του εκάστοτε τρεξίματος του μοντέλου να μπορούν να αξιοποιηθούν από την εφαρμογή. Αρχικά, From blocks με ονόματα Splitter\_BPR, Wf, FN και Input\_GPC συνδέθηκαν με ToWorkspace blocks με ονόματα BPRData, Fuel, FNData και Input\_GPC αντίστοιχα στο Plant Model Subsystem. Τέλος, ToWorkspace blocks συνδέθηκαν με το σήμα που εισέρχεται στην θύρα εισόδου Health\_Params του Plant Model Subsystem και στο σήμα <Converged> του BusSelector block που είναι συνδεδεμένο στην θύρα εξόδου S\_Data του επιλύτη, με ονόματα hp και conv αντίστοιχα.



Σχήμα 34: Σχηματικό διάγραμμα της υλοποίησης των παραμέτρων υγείας στο Simulink

### 3.2 Εφαρμογή

Η εφαρμογή που αναπτύχθηκε έχει ως βάση το πρότυπο (template) του App Designer της Matlab "2-Panel App with Auto-Reflow". Στο σημείο αυτό αξίζει να σημειωθεί ότι ο πηγαίος κώδικας όλων των συναρτήσεων (functions) βρίσκεται στο **Παράρτημα Ι**. Η διαδικασία ανάπτυξης της εφαρμογής παρουσιάζεται παρακάτω.

Αρχικά, διαγράφηκε η υπάρχουσα συνάρτηση `updateAppLayout` και προστέθηκαν οι Callback συναρτήσεις `startupFcn` στην συνιστώσα της εφαρμογής (app component) και `UIFigureCloseRequest` στην συνιστώσα `UIFigure`. Η πρώτη συνάρτηση καλείται κατά την εκκίνηση της εφαρμογής και η λειτουργία της θα περιγραφεί

φεί αναλυτικά σε επόμενη παράγραφο, ενώ η δεύτερη καλείται κατά το κλείσιμο της εφαρμογής και έχει σκοπό το κλείσιμο όλων των συνιστωσών της, όπως π.χ. την διεπαφή χρήστη (user interface), το μοντέλο Simulink κλπ. Στη συνέχεια, στην παράμετρο WindowState του UIFigure επιλέχθηκε η επιλογή "maximized", ενώ στην συνιστώσα LeftPanel πραγματοποιήθηκαν αλλαγές στις ακόλουθες παραμέτρους:

- Title τέθηκε η τιμή "Input Data"
- FontSize τέθηκε η τιμή 20
- FontWeight τέθηκε η τιμή "bold"

Μέσω της συνάρτησης startupFcn πραγματοποιούνται με τη σειρά οι ακόλουθες ενέργειες:

1. Αρχικοποιούνται οι βασικές πληροφορίες της εφαρμογής
2. Δημιουργείται το πλαίσιο των εισόδων (Input Panel)
3. Δημιουργείται το πλαίσιο των παραμέτρων υγείας (Health Parameters Panel)
4. Δημιουργείται το πλαίσιο των παραμέτρων απόδοσης (Performance Parameters Panel)
5. Δημιουργείται το πλαίσιο ελέγχου (Control Panel)
6. Δημιουργείται μία ομάδα καρτελών (TabGroup) που περιέχει μια καρτέλα (tab) για κάθε αεροθερμοδυναμική συνιστώσα του κινητήρα όπου σχεδιάζεται ο χάρτης της, μία καρτέλα που περιέχει έναν πίνακα όπου παρουσιάζονται τα αποτελέσματα των προσομοιώσεων, μία καρτέλα που περιέχει την κονσόλα της MATLAB και μία καρτέλα που περιέχει ένα γράφημα με τις διάφορες παραμέτρους του κινητήρα
7. Δημιουργείται το πλαίσιο του σχηματικού του κινητήρα (Schematic Panel)

Οι ιδιότητες (properties) της εφαρμογής που είτε αρχικοποιούνται μέσω της συνάρτησης startupFcn, είτε έχουν σκοπό την διάδοση πληροφορίας ανάμεσα στις διάφορες συνιστώσες της παρουσιάζονται στον **Πίνακα 5**.

Πίνακας 5: Ιδιότητες της εφαρμογής

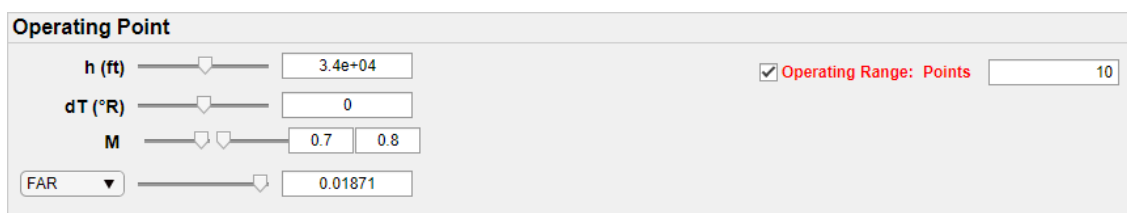
Ιδιότητα	Περιγραφή
Model_Name	Όνομα μοντέλου Simulink
model_type	Τύπος μοντέλου (μόνιμης ή μεταβατικής κατάστασης)
health_info	Πληροφορίες σχετικά με τις παραμέτρους υγείας



Ιδιότητα	Περιγραφή
inputs	Ονόματα των blocks των μεταβλητών εισόδου
input_names	Ονόματα των μεταβλητών εισόδου
input_vals_file	Όνομα του αρχείου που περιέχει τις αρχικές τιμές των μεταβλητών εισόδου
input_units	Μονάδες μέτρησης των μεταβλητών εισόδου
input_limits	Όρια τιμών των μεταβλητών εισόδου
stations	Ονόματα των blocks των θέσεων του κινητήρα
results_tbl	Δεδομένα του πίνακα αποτελεσμάτων
sign_names	Ονόματα των μεταβλητών της υπογραφής του κινητήρα
sign_labels	Ετικέτες των μεταβλητών της υπογραφής του κινητήρα για τα γραφήματα
dd_item_limits	Όρια τιμών των μεταβλητών ελέγχου
dd_items	Ονόματα μεταβλητών ελέγχου
dd_data	Αρχικές τιμές των μεταβλητών ελέγχου
meas_names	Ονόματα blocks των μετρήσεων
health_names	Ονόματα παραμέτρων υγείας
cmp_names	Ονόματα αεροθερμοδυναμικών συνιστωσών
cmp_types	Τύποι αεροθερμοδυναμικών συνιστωσών
misc_names	Ονόματα λοιπών παραμέτρων απόδοσης
misc_units	Μονάδες μέτρησης των λοιπών παραμέτρων απόδοσης
default_cb	Πλαίσια ελέγχου (checkboxes) που επιλέγονται αρχικά
h_label_style	Παράμετροι ετικετών (labels) των παραμέτρων υγείας
h_slider_style	Παράμετροι ολισθητών (sliders) των παραμέτρων υγείας
h_input_style	Παράμετροι πλαισίων εισόδου (input fields) των παραμέτρων υγείας
panel_style	Παράμετροι πλαισίων
NRIC	Αρχικές συνθήκες επιλύτη Newton-Raphson
NumDefVars	Αριθμός μεταβλητών εκτός εκείνων που περιέχονται στην υπογραφή του κινητήρα
dep_ind	Όνομα block για την επιλογή εξαρτημένων μεταβλητών
solver	Όνομα block του επιλύτη
hp_ind	Όνομα block για την επιλογή παραμέτρων υγείας
analysis_flag	Όνομα block για την επιλογή προσομοίωσης ή διάγνωσης
iControl	Όνομα block για την επιλογή δείκτη της παραμέτρου ελέγχου
JFile	Όνομα αρχείου που περιέχει τον πίνακα της ιακωβιανής
NumOpts	Αριθμός επιλογών κατά την εύρεση μεταβλητών διάγνωσης
dfile	Όνομα αρχείου στο οποίο αποθηκεύεται ό,τι γράφεται στην κονσόλα
tmr	Χρονόμετρο για την ανανέωση της κονσόλας
def_inputs	Τιμές των μεταβλητών εισόδου
sch_path	Διαδρομή αρχείου (file path) της εικόνας του σχηματικού του κινητήρα
sel_dd_item	Επιλεγμένη μεταβλητή εισόδου ελέγχου

### 3.2.1 Πλαίσιο εισόδων

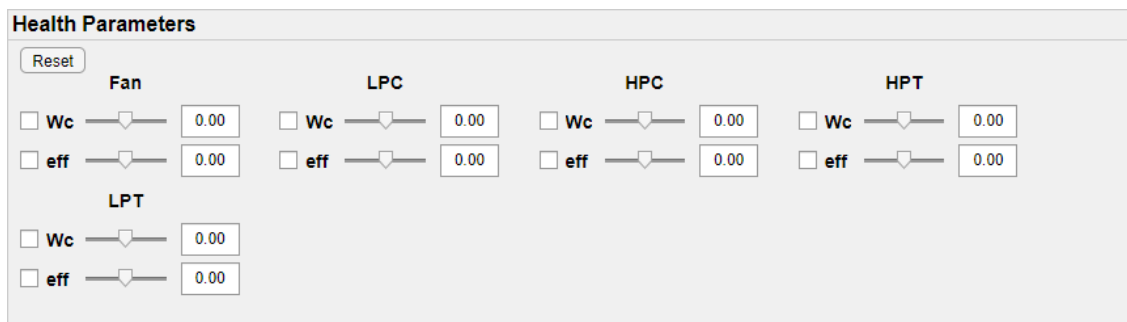
Το πλαίσιο εισόδων (Input Panel), όπως και όλα τα πλαίσια της εφαρμογής, είναι ένα αντικείμενο (object) τύπου `uipanel` και έχει ως αντικείμενο-γονέα (parent object) το `LeftPanel` (βλ. **Σχήμα 35**). Για την δημιουργία του καλείται από την `startupFcn` η συνάρτηση `setup_inputs`. Αυτή, μέσω της `setup_single_input`, προσθέτει στο Input Panel για κάθε είσοδο εκτός της μεταβλητής εισόδου ελέγχου: μία ετικέτα (`uicontrol`), έναν ολισθητή (`uislider`) και ένα πεδίο (`uieditfield`). Ενώ, αν πρόκειται για την μεταβλητή εισόδου ελέγχου, στην θέση της ετικέτας, εισάγει ένα μενού (`uidropdown`) με επιλογές τις μεταβλητές ελέγχου που έχουν επιλεγεί. Επιπλέον, όταν ο χρήστης αλλάξει τιμή είτε στο πεδίο, είτε στον ολισθητή, μέσω της `update_input_pair`, η τιμή που θέτει τίθεται στον ολισθητή ή στο πεδίο αντίστοιχα, ενώ όταν επιλέξει μία διαφορετική μεταβλητή ελέγχου τα κατάλληλα αντικείμενα της διεπαφής χρήστη ενημερώνονται μέσω της `update_input_dd`. Τέλος, η συνάρτηση `setup_inputs` προσθέτει ένα πλαίσιο ελέγχου (`uicheckbox`), μέσω του οποίου η εφαρμογή εναλλάσσεται ανάμεσα σε προσομοίωση ενός σημείου και πολλαπλών σημείων. Όταν αλλάζει η τιμή του πλαισίου ελέγχου αυτού καλείται η συνάρτηση `toggle_range`. Όταν το πλαίσιο επιλέγεται, τότε, αρχικά, προστίθεται δίπλα σε αυτό ένα πεδίο στο οποίο ο χρήστης μπορεί να εισάγει τον αριθμό των σημείων που επιθυμεί να τρέξει. Αφότου ο χρήστης εισάγει τιμή στο πεδίο αυτό ελέγχεται αν η τιμή αυτή είναι από 1 και πάνω μέσω της συνάρτησης `checkValid`. Στη συνέχεια, αφού διαγραφεί η προηγούμενη διεπαφή χρήστη, προστίθενται στο Input Panel για κάθε μεταβλητή εισόδου, μέσω της συνάρτησης `setup_double_input`, τα εξής: μία ετικέτα ή ένα μενού όπως προηγουμένως και από δύο πεδία και ολισθητές για την εισαγωγή της ελάχιστης και μέγιστης τιμής της μεταβλητής κατά την προσομοίωση. Τα πεδία και οι ολισθητές χρησιμοποιούν την συνάρτηση `update_input_pair` όπως περιγράφηκε παραπάνω. Η παράμετρος με την οποία επιθυμεί ο χρήστης να πραγματοποιήσει προσομοίωση πολλαπλών σημείων επιλέγεται αλλάζοντας την τιμή ενός από τους αντίστοιχους δύο ολισθητές. Όταν πραγματοποιηθεί αυτό, καλείται η συνάρτηση `lock_param`, η οποία για τις μη επιλεγμένες μεταβλητές διαγράφει τα υπάρχοντα αντικείμενα και δημιουργεί καινούρια μέσω της `setup_single_input`, ενώ για την επιλεγμένη θέτει ως συνάρτηση η οποία καλείται όταν αλλάζει η τιμή των δύο ολισθητών (ιδιότητα `ValueChangedFcn`) την `updateLimits`. Η τελευταία αλλάζει κατάλληλα τα όρια των ολισθητών ώστε ο χρήστης να μπορεί να επιλέξει ένα διάστημα προσομοίωσης σε όλο το επιτρεπτό διάστημα τιμών. Ενώ, όταν το πλαίσιο αποεπιλέγεται, τότε διαγράφονται τα υπάρχοντα αντικείμενα της διεπαφής χρήστη και δημιουργούνται καινούρια μέσω της `setup_single_input`.



Σχήμα 35: Πλαίσιο εισόδων

### 3.2.2 Πλαίσιο παραμέτρων υγείας

Το πλαίσιο των παραμέτρων υγείας (Health Panel) είναι ένα αντικείμενο τύπου `uipanel` και έχει ως αντικείμενο-γονέα το `LeftPanel` (βλ. **Σχήμα 36**). Για την δημιουργία του καλείται από την `startupFcn` η συνάρτηση `setup_health`. Αυτή, μέσω της συνάρτησης `setup_health_ui`, προσθέτει τα ακόλουθα στοιχεία για κάθε αεροθερμοδυναμική συνιστώσα του κινητήρα: μία ετικέτα με το όνομα της συνιστώσας και για τις δύο παραμέτρους υγείας της (διορθωμένη παροχή και βαθμός απόδοσης): ένα πλαίσιο ελέγχου, μία ετικέτα με το όνομα της παραμέτρου, έναν ολισθητή και ένα πεδίο. Όπως και στο `Input Panel`, κάθε ζευγάρι ολισθητή και πεδίου χρησιμοποιεί την συνάρτηση `update_input_pair`. Επιπλέον, η συνάρτηση `setup_health` δημιουργεί ένα κουμπί (`uibutton`) στο `Health Panel`, το οποίο όταν πατηθεί επαναφέρει τις τιμές των ολισθητών και των πεδίων του πλαισίου αυτού στις αρχικές μηδενικές τιμές μέσω της συνάρτησης `reset_health`.



Σχήμα 36: Πλαίσιο παραμέτρων υγείας

### 3.2.3 Πλαίσιο παραμέτρων απόδοσης

Το πλαίσιο των παραμέτρων απόδοσης (Perf Panel) είναι ένα αντικείμενο τύπου `uipanel` και έχει ως αντικείμενο-γονέα το `LeftPanel` (βλ. **Σχήμα 37**). Για την δημιουργία του καλείται από την `startupFcn` η συνάρτηση `setup_perf`. Αρχικά, αυτή για κάθε θέση του κινητήρα που περιέχεται στην ιδιότητα `stations` της εφαρμογής και μέσω της συνάρτησης `add_perf_item` προσθέτει τα ακόλουθα στοιχεία: ένα πλαίσιο ελέγχου, μία ετικέτα με το όνομα της παραμέτρου και ένα πεδίο. Αυτό πραγματοποιείται δύο φορές ανά θέση: μία φορά για την ολική πίεση και μία φορά για την ολική θερμοκρασία. Τα ίδια στοιχεία προστίθεται και για τις λοιπές παραμέτρους απόδοσης, οι οποίες περιέχονται στην ιδιότητα `misc_names` της εφαρμογής. Οι αρχικές τιμές που τίθενται στα πεδία που δημιουργήθηκαν προέρχονται από ένα τρέξιμο του μοντέλου με τις αρχικές τιμές των διαφόρων παραμέτρων του μέσω της κλήσης της συνάρτησης `sim_and_send_to_workspace` από την `startupFcn`. Η συνάρτηση αυτή τρέχει το μοντέλο του κινητήρα, στέλνει τα αποτελέσματα στον χώρο εργασίας της Matlab (`workspace`) και ενημερώνει τον χρήστη αν ο επιλύτης συνέκλινε. Οπότε, οι τιμές των ολικών θερμοκρασιών και πιέσεων λαμβάνονται απευθείας από τον χώρο εργασίας, ενώ οι λοιπές παράμετροι απόδοσης μέσω της συνάρτησης

get\_misc\_perfs, η οποία επιστρέφει τις τιμές τους αφού τις "διαβάσει" από τον χώρο εργασίας. Στην παρούσα εργασία οι λοιπές παράμετροι απόδοσης είναι οι ακόλουθες:

- Οι στροφές του άξονα χαμηλής πίεσης (N\_LP)
- Οι στροφές του άξονα υψηλής πίεσης (N\_HP)
- Η παροχή αέρα στην είσοδο του κινητήρα (W)
- Η παροχή καυσίμου (Wf)
- Ο λόγος παράκαμψης (BPR)
- Η καθαρή ώση (FN)

Στη συνέχεια, η setup\_perf διαγράφει από την διεπαφή χρήστη τα πλαίσια ελέγχου των μεταβλητών που δεν περιέχονται στην υπογραφή του κινητήρα, καθώς δεν είναι χρήσιμα για αυτές. Τέλος, όταν ο χρήστης αλλάζει την τιμή ενός πλαισίου ελέγχου καλείται η συνάρτηση toggle\_sign, η οποία θέτει στο κατάλληλο Constant block με κατάληξη \_flag στο μοντέλο την τιμή 1 αν είναι επιλεγμένο ή την τιμή 0 αν είναι αποεπιλεγμένο.

Performance Parameters		
<input checked="" type="checkbox"/> Pt15 (psia) 8.3253:9.0342	<input type="checkbox"/> Pt3 (psia) 121.8398:126.	<input type="checkbox"/> N_LP (rpm) 3758.0239:366
<input checked="" type="checkbox"/> Tt15 (°R) 517.6263:527.	<input type="checkbox"/> Tt3 (°R) 1204.5303:121	<input type="checkbox"/> N_HP (rpm) 7375.8337:746
<input checked="" type="checkbox"/> Pt2 (psia) 5.0361:5.5344	<input type="checkbox"/> Pt4 (psia) 115.1386:119.	<input type="checkbox"/> W (pps) 634.8682:678.
<input checked="" type="checkbox"/> Tt2 (°R) 436.6683:448.	<input type="checkbox"/> Tt4 (°R) 2370.3091:237	<input type="checkbox"/> Wf (pps) 1.8453:1.9108
<input checked="" type="checkbox"/> Pt21 (psia) 8.3882:9.1025	<input type="checkbox"/> Pt45 (psia) 43.1886:44.44	<input type="checkbox"/> BPR 4.8578:5.0421
<input checked="" type="checkbox"/> Tt21 (°R) 517.6263:527.	<input type="checkbox"/> Tt45 (°R) 1850.8688:185	<input type="checkbox"/> FN (lbf) 11455.5252:11
<input type="checkbox"/> Pt25 (psia) 21.8614:22.48	<input type="checkbox"/> Pt5 (psia) 8.6901:9.0198	
<input type="checkbox"/> Tt25 (°R) 705.7913:707.	<input type="checkbox"/> Tt5 (°R) 1278.491:1284	

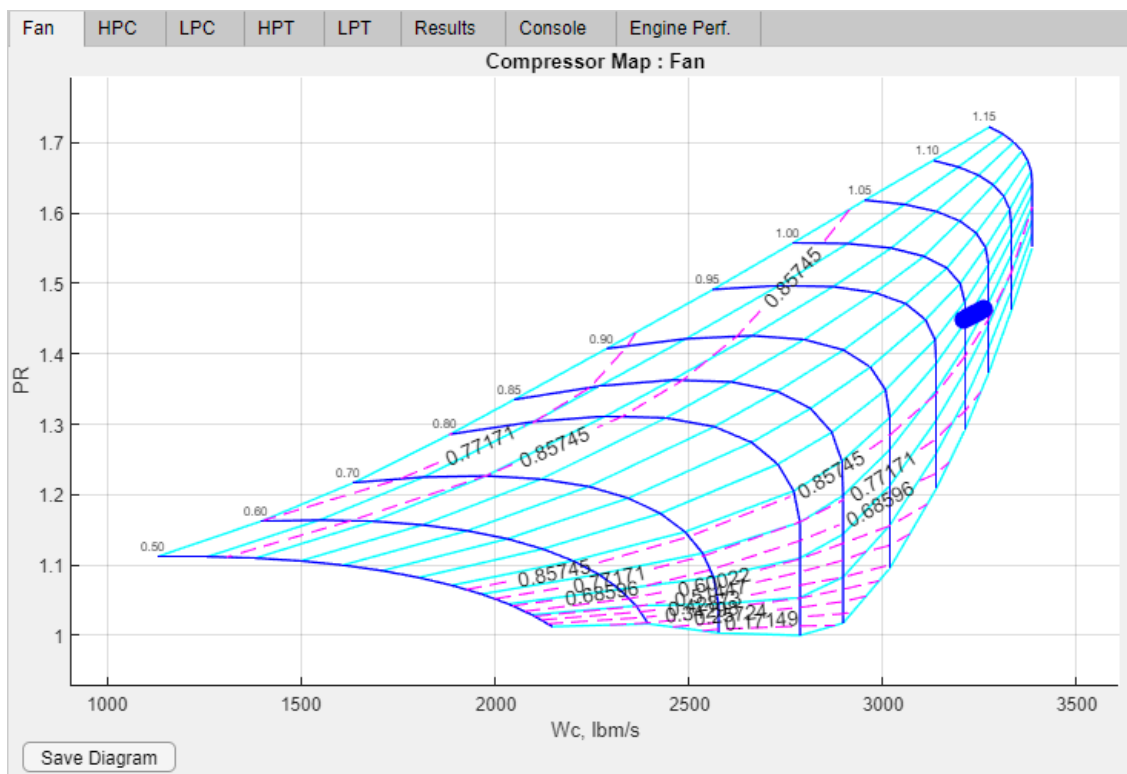
Σχήμα 37: Πλαίσιο παραμέτρων απόδοσης

### 3.2.4 Ομάδα καρτελών

Η ομάδα καρτελών (PlotTabGroup) είναι ένα αντικείμενο τύπου uitabgroup και έχει ως αντικείμενο-γονέα το RightPanel. Αρχικά, η startupFcn καλεί την συνάρτηση plot\_results\_SS, η οποία προέκυψε με κατάλληλες τροποποιήσεις στην συνάρτηση TMATS\_SSplot της βιβλιοθήκης T-MATS ώστε να είναι συμβατή με τον App Designer της Matlab. Συγκεκριμένα, για κάθε αεροθερμοδυναμική συνιστώσα του κινητήρα προσθέτει (αν δεν υπάρχει ήδη) στο PlotTabGroup μία καρτέλα (uitab) και μέσα σε αυτή ένα γράφημα (uiaxes). Σε αυτές σχεδιάζονται οι χάρτες των συνιστωσών αυτών και τα σημεία του κινητήρα που έχουν τρεχτεί (πχ. ο χάρτης του Fan

φαίνεται στο **Σχήμα 38**). Οι χάρτες των συμπιεστών και των στροβίλων σχεδιάζονται μέσω των συναρτήσεων `PlotCMap_mod` και `PlotTMap_mod` αντίστοιχα, οι οποίες προέκυψαν μέσω κατάλληλων τροποποιήσεων στις συναρτήσεις `TMATS.PlotCMap` και `TMATS.PlotTMap` της βιβλιοθήκης T-MATS ώστε να είναι συμβατές με τον App Designer της Matlab. Ακόμη, σε κάθε μία από τις καρτέλες αυτές προστίθεται και ένα κουμπί, το οποίο όταν πατηθεί αποθηκεύει το εκάστοτε γράφημα.

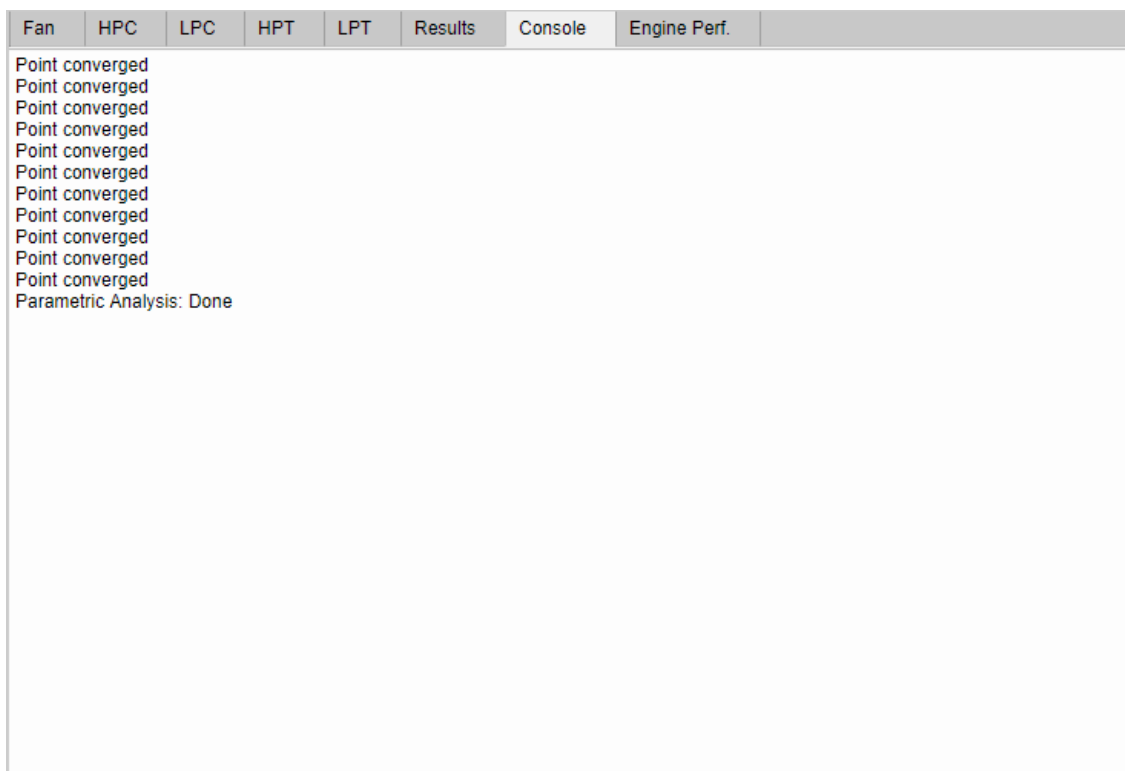
Στη συνέχεια, προστίθεται μία ακόμη καρτέλα στο `PlotTabGroup`, στην οποία εισάγεται ένα αντικείμενο πίνακα (`uitable`), όπου παρουσιάζονται οι μεταβλητές εισόδου, οι παράμετροι υγείας και οι παράμετροι απόδοσης των διαφόρων προσομοιώσεων που πραγματοποιεί ο χρήστης (βλ. **Σχήμα 39**). Ακόμη, προστίθεται δύο κουμπιά. Το πρώτο πραγματοποιεί εκκαθάριση των δεδομένων του πίνακα μέσω της συνάρτησης `clearTable`, ενώ το δεύτερο αποθηκεύει τον πίνακα σε ένα υπολογιστικό φύλλο μέσω της συνάρτησης `saveTable`.



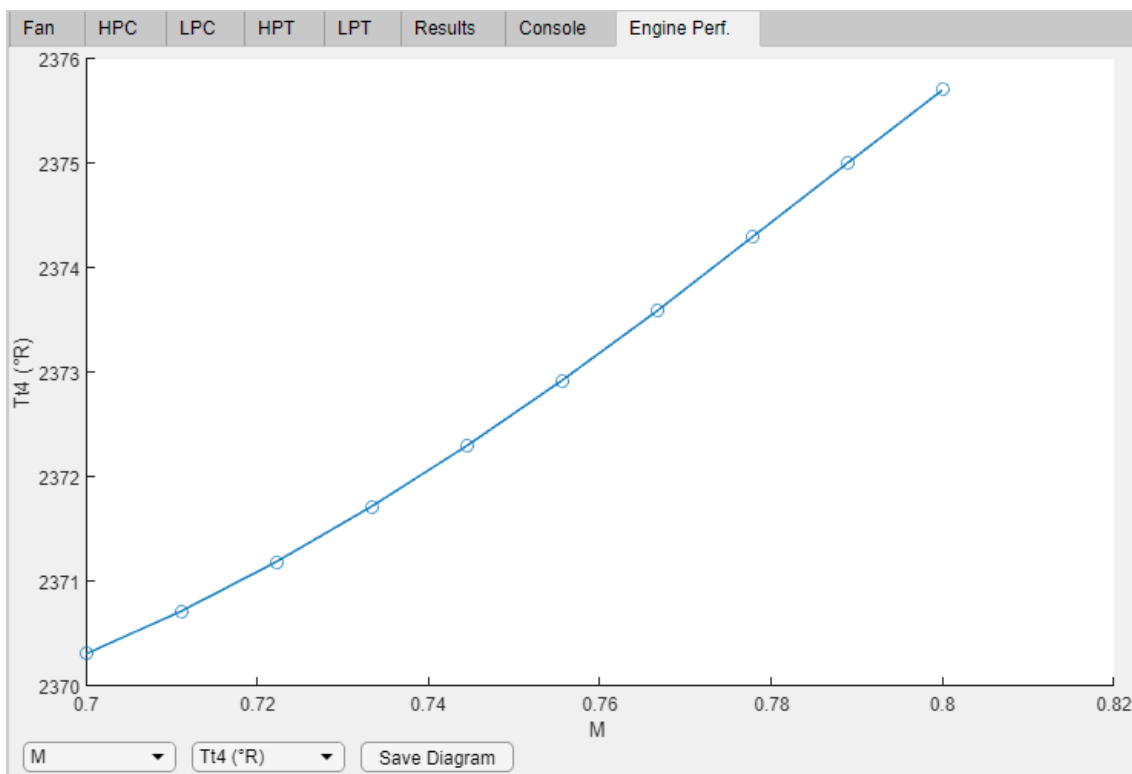
Σχήμα 38: Καρτέλα χάρτη Fan

Έπειτα, εισάγεται μια επιπλέον καρτέλα στο `PlotTabGroup`, στην οποία εισάγεται ένα αντικείμενο περιοχής κειμένου (`uitextarea`), όπου περιέχεται ό,τι "γράφεται" στην κονσόλα (`console`) της Matlab (βλ. **Σχήμα 40**). Το περιεχόμενο της περιοχής κειμένου ανανεώνεται κάθε 2 δευτερόλεπτα μέσω της συνάρτησης `refreshConsole`, η οποία διαβάζει τα περιεχόμενα της κονσόλας της Matlab μέσω ενός αρχείου (`diary`).





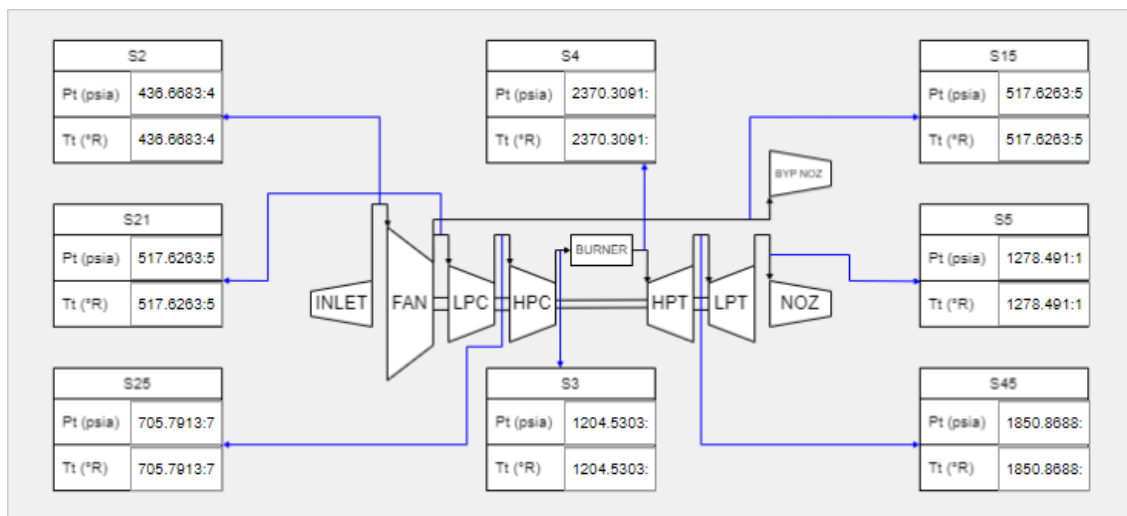
Σχήμα 40: Καρτέλα κονσόλας



Σχήμα 41: Καρτέλα γραφήματος αποτελεσμάτων

### 3.2.5 Πλαίσιο σχηματικού

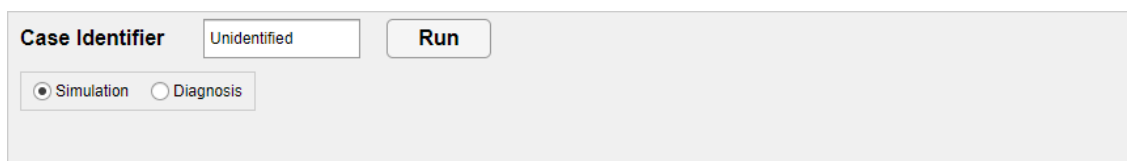
Το πλαίσιο του σχηματικού (Schematic Panel) του κινητήρα είναι ένα αντικείμενο τύπου `uipanel` και έχει ως αντικείμενο-γονέα το `RightPanel` (βλ. **Σχήμα 42**). Σε αυτό εισάγεται ένα αντικείμενο εικόνας (`uiimage`) που περιέχει μια εικόνα της σχηματικής αναπαράστασης του κινητήρα. Μέσω της συνάρτησης `setup_schematic_data` εισάγονται σε κατάλληλες θέσεις πεδία στα οποία παρουσιάζονται οι τιμές των ολικών πιέσεων και θερμοκρασιών στις διάφορες θέσεις του κινητήρα μετά το πέρας της προσομοίωσης.



Σχήμα 42: Πλαίσιο σχηματικού

### 3.2.6 Πλαίσιο ελέγχου

Το πλαίσιο ελέγχου (Control Panel) είναι ένα αντικείμενο τύπου `uipanel` και έχει ως αντικείμενο-γονέα το `LeftPanel` (βλ. **Σχήμα 43**). Αρχικά, εισάγεται σε αυτό μέσω της `startupFcn` μία ετικέτα και ένα πεδίο στο οποίο ο χρήστης έχει τη δυνατότητα να δώσει ένα χαρακτηριστικό όνομα (`case identifier`) στην προσομοίωση που επιθυμεί να τρέξει, το οποίο εισάγεται και στην πρώτη στήλη του πίνακα των αποτελεσμάτων. Στη συνέχεια, προστίθεται ένα κουμπί που όταν πατηθεί πραγματοποιείται η προσομοίωση ή η διάγνωση (`run_btn`). Τέλος, εισάγεται ένα αντικείμενο τύπου "ομάδας κουμπιών" (`uibuttongroup`), στο οποίο προστίθενται δύο κουμπιά επιλογής (`uiradiobutton`), ένα για την επιλογή της λειτουργίας προσομοίωσης και ένα για την επιλογή της λειτουργίας διάγνωσης.



Σχήμα 43: Πλαίσιο ελέγχου



Όταν ο χρήστης αλλάζει λειτουργία στην εφαρμογή, δηλαδή από προσομοίωση σε διάγνωση ή το αντίθετο, καλείται η συνάρτηση `toggle_mode`. Αυτή, στην πρώτη περίπτωση καλεί την συνάρτηση `setup_diag_btns`, η οποία προσθέτει 2 ζευγάρια ενός κουμπιού και ενός μενού, ενώ επίσης επιλέγει τα αρχικά πλαίσια ελέγχου, όπως υποδεικνύεται από την ιδιότητα `default_cb` της εφαρμογής. Το ένα κουμπί χρησιμεύει για την εύρεση των κατάλληλων μετρήσεων για δεδομένες παραμέτρους υγείας, ενώ το άλλο για την εύρεση των κατάλληλων παραμέτρων υγείας για δεδομένες μετρήσεις. Όταν ο χρήστης πατήσει ένα από τα δύο αυτά κουμπιά καλείται η συνάρτηση `setup_cp_dd`, η οποία αρχικά βρίσκει ποιες παραμέτρους υγείας ή μετρήσεις έχει επιλέξει ο χρήστης μέσω των πλαισίων ελέγχου. Στην συνέχεια, μέσω των παρόμοιων μεταξύ τους συναρτήσεων `getHPcases` και `getMcases`, βρίσκεται ο αριθμός κατάστασης του κατάλληλου μέρους της ιακωβιανής του συστήματος κάθε πιθανού συνδυασμού μετρήσεων ή παραμέτρων υγείας αντίστοιχα, ανάλογα με το πόσες έχει επιλέξει ο χρήστης. Έπειτα, οι συνδυασμοί αυτοί κατατάσσονται από τον συνδυασμό με τον χαμηλότερο αριθμό κατάστασης στον μεγαλύτερο. Από αυτούς οι πρώτοι, όσοι υπαγορεύονται από την ιδιότητα `NumOpts` της εφαρμογής, αφότου αντιστοιχιστούν οι δείκτες που περιέχονται σε αυτούς (βλ. **Πίνακες 2 και 3**) σε ονόματα μεταβλητών και μετατραπούν σε σειρά χαρακτήρων (`string`) με κόμματα ανάμεσά τους (μέσω της συνάρτησης `mapAndJoin`), καθώς και με την προσθήκη της τιμής του αριθμού κατάστασής τους στο τέλος της σειράς αυτής, εισάγονται ως επιλογές στο κατάλληλο μενού. Ενώ, μέσω της συνάρτησης `selectCBs`, επιλέγονται τα κατάλληλα πλαίσια ελέγχου, δηλαδή εκείνα που αντιστοιχούν στις μεταβλητές που περιέχονται στον πρώτο συνδυασμό, εκείνον με τον χαμηλότερο αριθμό κατάστασης. Στο σημείο αυτό αξίζει να σημειωθεί ότι ο τρόπος υπολογισμού της ιακωβιανής και του αριθμού κατάστασής της περιγράφεται στο **υποκεφάλαιο 3.4**. Η συνάρτηση `selectCBs` καλείται επίσης όταν ο χρήστης επιλέξει μία επιλογή στο μενού ώστε να επιλεχθούν και πάλι τα κατάλληλα πλαίσια ελέγχου. Τέλος, στην περίπτωση που πραγματοποιηθεί αλλαγή της λειτουργίας της εφαρμογής από διάγνωση σε προσομοίωση, η συνάρτηση `toggle_mode` διαγράφει τα κουμπιά και τα μενού που είχαν δημιουργηθεί και αποεπιλέγει τυχόν επιλεγμένα πλαίσια ελέγχου.

Ακόμη, όταν ο χρήστης πατήσει το κουμπί `run_btn` καλείται η συνάρτηση `run_model`, η οποία, αφότου θέσει στο μοντέλο την κατάλληλη τιμή στο `iControl block`, ελέγχει αν η εφαρμογή είναι σε λειτουργία προσομοίωσης ή διάγνωσης.

Για την περίπτωση της λειτουργίας προσομοίωσης, αρχικά η συνάρτηση `run_model` θέτει την κατάλληλη τιμή στις διάφορες παραμέτρους του μοντέλου. Στη συνέχεια, ελέγχει αν η εφαρμογή έχει τεθεί σε λειτουργία ενός ή πολλαπλών σημείων. Για την περίπτωση προσομοίωσης ενός σημείου, ελέγχεται πρώτα αν ο χρήστης έχει θέσει μη μηδενική τιμή σε οποιαδήποτε από τις παραμέτρους υγείας μέσω της συνάρτησης `checkHealthValues`. Αν υπάρχουν μη μηδενικές τιμές, τότε προστίθεται, αν δεν υπάρχει ήδη, μία επιπλέον καρτέλα στο `PlotTabGroup` και σε αυτή ένα αντικείμενο γραφήματος με σκοπό την σχεδίαση του γραφήματος της υπογραφής του κινητήρα. Ακολούθως, πραγματοποιείται προσομοίωση ενός σημείου μέσω της συνάρτησης `simulate_point`. Η συνάρτηση αυτή, αφού αρχικά θέσει τις τιμές των τεσσάρων μεταβλητών εισόδου στο μοντέλο `Simulink`, θέτει σε αυτό και τις τιμές των

παραμέτρων υγείας στα κατάλληλα blocks μέσω της συνάρτησης `update_health`. Στη συνέχεια, τίθεται οι αρχικές συνθήκες του επιλύτη (ιδιότητα `SNR_IC_M`) στο μοντέλο, οι οποίες υπολογίζονται μέσω της συνάρτησης `getNRIC`. Η συνάρτηση αυτή, πραγματοποιώντας κατάλληλες παρεμβολές σε πίνακες που δίνουν κάθε μεταβλητή των αρχικών συνθηκών συναρτήσει του αριθμού `Mach` και της διορθωμένης καθαρής ώσης, δίνει μία εκτίμηση για τις αρχικές συνθήκες για δεδομένες εισόδους, η οποία είναι κοντινή στις αναμενόμενες τελικές τιμές των μεταβλητών αυτών, ώστε να διευκολυνθεί η σύγκλιση του επιλύτη. Ο τρόπος δημιουργίας των πινάκων αυτών, καθώς και η μεθοδολογία της παρεμβολής από την οποία προκύπτουν οι αρχικές συνθήκες του επιλύτη περιγράφονται στο **υποκεφάλαιο 3.5**. Τέλος, αφότου πραγματοποιηθεί ένα τρέξιμο του μοντέλου μέσω της `sim_and_send_to_workspace`, σχεδιάζεται το σημείο στους χάρτες των συνιστωσών, προστίθεται στον πίνακα των αποτελεσμάτων μία επιπλέον γραμμή με τις τιμές των μεταβλητών που προέκυψαν, ενημερώνονται οι τιμές των πεδίων του `Perf Panel` και ενημερώνεται το γράφημα των αποτελεσμάτων μέσω της συνάρτησης `update_perf_plot`. Επόμενο βήμα στη συνάρτηση `run_model` είναι η αποθήκευση σε έναν πίνακα (array) των τιμών των μεταβλητών της υπογραφής μέσω της συνάρτησης `getSignatures`, η οποία λαμβάνει τις τιμές αυτές από τον χώρο εργασίας της Matlab. Έπειτα, καλείται η συνάρτηση `simulate_point_zero_hp`, η οποία είναι παρόμοια με την `simulate_point`, με την διαφορά ότι θέτει μηδενικές τιμές σε όλες τις παραμέτρους υγείας κατά το τρέξιμο του μοντέλου και ότι δεν παρουσιάζονται τα αποτελέσματα του τρεξίματος αυτού στην εφαρμογή, όμως αποθηκεύονται και αυτά σε πίνακα μέσω της `getSignatures`. Στη συνέχεια, υπολογίζονται και σχεδιάζονται οι ποσοστιαίες μεταβολές των μεταβλητών της υπογραφής ανάμεσα στο τρέξιμο με και χωρίς παραμέτρους υγείας στην επιπλέον καρτέλα του `PlotTabGroup`. Ακόμη, στην περίπτωση που δεν υπάρχουν μη μηδενικές τιμές παραμέτρων υγείας, διαγράφεται η καρτέλα του γραφήματος της υπογραφής και πραγματοποιείται η προσομοίωση μέσω της `simulate_point`.

Επιπλέον, αν η εφαρμογή έχει τεθεί σε λειτουργία προσομοίωσης πολλαπλών σημείων, αφότου διαγραφεί, αν υπάρχει, η καρτέλα του γραφήματος της υπογραφής, καλείται η συνάρτηση `simulate_range`. Αρχικά, μέσω αυτής, βρίσκεται ποια μεταβλητή εισόδου έχει επιλεγεί από τον χρήστη ως παράμετρος της οποίας η τιμή μεταβάλλεται από μία ελάχιστη σε μία μέγιστη τιμή. Έπειτα, αφότου ληφθούν από την διεπαφή χρήστη ο αριθμός των σημείων και τα όρια της επιλεγμένης μεταβλητής, υπολογίζονται κάνοντας χρήση αυτών οι συντελεστές μίας γραμμικής σχέσης ( $var = m \cdot i + b$ ) ανάμεσα σε έναν δείκτη  $i$  που λαμβάνει τιμές από 1 μέχρι τον αριθμό των σημείων και της τιμής της επιλεγμένης μεταβλητής  $var$ . Δηλαδή, όταν ο δείκτης είναι ίσος με 1 η μεταβλητή λαμβάνει την ελάχιστη της τιμή και όταν είναι ίσος με τον αριθμό των σημείων την μέγιστη. Συνεχόμενα, για κάθε σημείο πραγματοποιούνται τα εξής: Αφότου υπολογιστεί η τρέχουσα τιμή για την επιλεγμένη μεταβλητή, οι μεταβλητές εισόδου και οι παράμετροι υγείας (μέσω της `update_health`) εισάγονται στο μοντέλο Simulink και πραγματοποιείται τρέξιμο του. Τέλος, ενημερώνεται η διεπαφή χρήστη με τα αποτελέσματα της προσομοίωσης με παρόμοιο τρόπο με την περίπτωση προσομοίωσης ενός σημείου, με τη διαφορά ότι στα πεδία του `Perf Panel` και του σχηματικού ως αποτελέσματα παρουσιάζονται το πρώτο και το τελευταίο σημείο με άνω και κάτω τελεία ανάμεσά τους. Τελευταίο βήμα της συνάρτησης `run_model` και

για τις δύο περιπτώσεις προσομοίωσης είναι η ενημέρωση των τιμών των πεδίων του σχηματικού με τα αποτελέσματά της μέσω της συνάρτησης `update_schematic`.

Για την περίπτωση της λειτουργίας διάγνωσης, αρχικά η συνάρτηση `run_model` θέτει στο `analysis_flag` block του μοντέλου Simulink την τιμή 1, ώστε αυτό να τεθεί σε λειτουργία διάγνωσης, και στα blocks των μετρήσεων την κατάλληλη τιμή, η οποία προέρχεται από το αντίστοιχο πεδίο στο Perf Panel. Ακολούθως, ρυθμίζονται οι διάφορες παράμετροι του μοντέλου για την πραγματοποίηση της προσομοίωσης μέσω της συνάρτησης `setup_diagnosis_parameters`. Αυτή πραγματοποιεί τα εξής: Πρώτα βρίσκει τους δείκτες των επιλεγμένων μετρήσεων και θέτει στα αντίστοιχα σήματα ελέγχου στο μοντέλο Simulink την τιμή 1, ενώ στα υπόλοιπα την τιμή 0. Στη συνέχεια, αφού βρει τους δείκτες και των παραμέτρων υγείας, θέτει την κατάλληλη τιμή στα blocks που χρησιμεύουν για την επιλογή των εξαρτημένων μεταβλητών (`dep_ind`) και των παραμέτρων υγείας (`hp_ind`). Έπειτα, θέτει τις μεταβλητές εισόδου και τις παραμέτρους του επιλύτη στο μοντέλο. Τέλος, ελέγχει τα ακόλουθα:

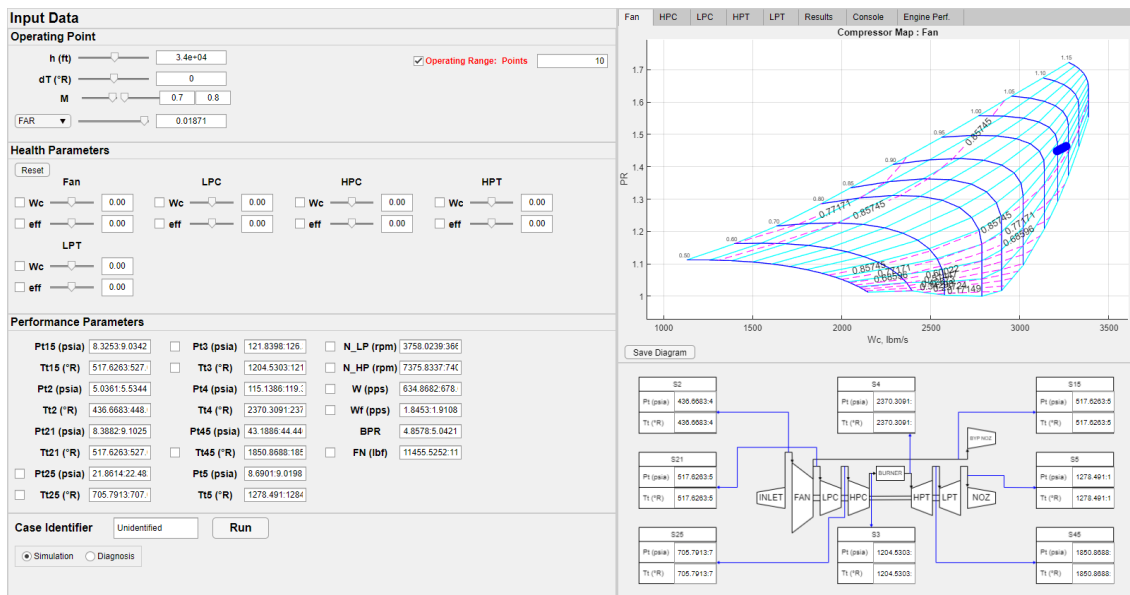
1. Αν έχει επιλεγεί ίσος αριθμός μετρήσεων και παραμέτρων υγείας, καθώς ο επιλύτης μπορεί να χειριστεί μόνο τετραγωνικά προβλήματα
2. Αν η επιλεγμένη παράμετρος ελέγχου περιλαμβάνεται στις επιλεγμένες μετρήσεις. Αυτό δεν πρέπει να συμβαίνει για λόγους σύγκλισης του επιλύτη

Η συνάρτηση `setup_diagnosis_parameters` επιστρέφει την μεταβλητή `isValid`, η οποία έχει την τιμή:

- "no\_equal", αν δεν έχει επιλεγεί ίσος αριθμός μετρήσεων και παραμέτρων υγείας
- "no\_control", αν η επιλεγμένη παράμετρος ελέγχου περιλαμβάνεται στις επιλεγμένες μετρήσεις
- "yes", σε κάθε άλλη περίπτωση

Στην περίπτωση που η μεταβλητή `isValid` δεν έχει την τιμή "yes", η διάγνωση δεν πραγματοποιείται και εμφανίζεται στην διεπαφή χρήστη ένα μήνυμα σφάλματος (`uialert`) που προειδοποιεί κατάλληλα τον χρήστη. Σε αντίθετη περίπτωση, η διάγνωση πραγματοποιείται μέσω της `sim_and_send_to_workspace` και αν οι τιμές των εκτιμήσεων των παραμέτρων υγείας που υπολογίστηκαν είναι μέσα στα όρια ( $\pm 20\%$ ), τότε παρουσιάζονται στα κατάλληλα πεδία του Health Panel, αλλιώς ο χρήστης ενημερώνεται μέσω της κονσόλας ότι η διάγνωση ήταν ανεπιτυχής.

Η συνολική διεπαφή χρήστη παρουσιάζεται στο **Σχήμα 44**, όπου παρουσιάζονται ενδεικτικά τα αποτελέσματα μίας προσομοίωσης 10 σημείων σε υψόμετρο 34000 ft και αριθμό Mach από 0.7 έως 0.8.



Σχήμα 44: Συνολική διεπαφή χρήστη της εφαρμογής

### 3.3 Παραδείγματα χρήσης της εφαρμογής

#### 3.3.1 Προσομοίωση ενός σημείου

Αρχικά, ο χρήστης καλείται να καθορίσει το σημείο λειτουργίας και την μεταβλητή ελέγχου που επιθυμεί μέσω του πλαισίου εισόδων (βλ. **Σχήμα 45**). Στη συνέχεια, αν είναι επιθυμητό, πρέπει να ρυθμιστούν οι τιμές των παραμέτρων υγείας μέσω του αντίστοιχου πλαισίου. Στο παράδειγμα αυτό χρησιμοποιείται -1% βαθμός απόδοσης στο Fan, όπως φαίνεται στο **Σχήμα 46**. Τέλος, η προσομοίωση ξεκινάει μέσω του κουμπιού "Run" στο πλαίσιο ελέγχου.

The screenshot shows the input data panel with sliders for  $h$  (ft),  $dT$  (°R),  $M$ , and FAR. The values are  $3.4e+04$ , 0, 0.8, and 0.01871 respectively.

Σχήμα 45: Επιλογή εισόδων για προσομοίωση ενός σημείου

**Health Parameters**

Reset

Fan		LPC		HPC		HPT	
<input type="checkbox"/> Wc	0.00	<input type="checkbox"/> Wc	0.00	<input type="checkbox"/> Wc	0.00	<input type="checkbox"/> Wc	0.00
<input type="checkbox"/> eff	-1.00	<input type="checkbox"/> eff	0.00	<input type="checkbox"/> eff	0.00	<input type="checkbox"/> eff	0.00

LPT

<input type="checkbox"/> Wc	0.00
<input type="checkbox"/> eff	0.00

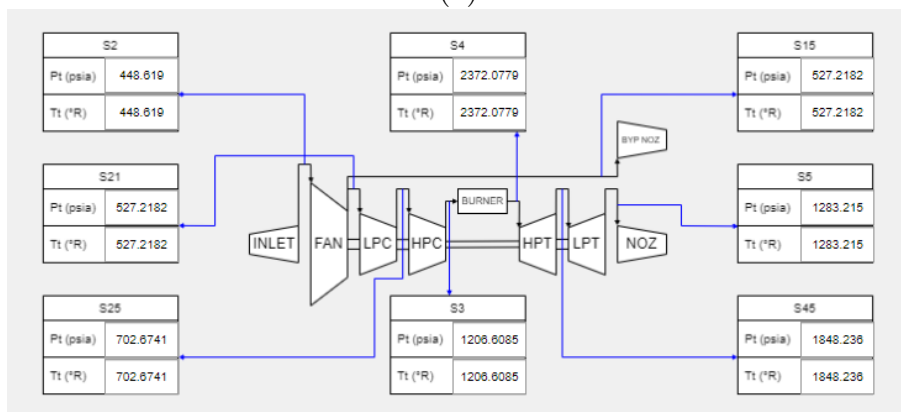
Σχήμα 46: Επιλογή παραμέτρων υγείας για προσομοίωση

Τα αποτελέσματα της προσομοίωσης φαίνονται τόσο στο πλαίσιο των παραμέτρων απόδοσης, όσο και στο σχηματικό του κινητήρα, όπως παρουσιάζονται στο Σχήμα 47.

**Performance Parameters**

Pt15 (psia)	8.9962	<input type="checkbox"/> Pt3 (psia)	124.6175	<input type="checkbox"/> N_LP (rpm)	3622.9442
Tt15 (°R)	527.2182	<input type="checkbox"/> Tt3 (°R)	1206.6085	<input type="checkbox"/> N_HP (rpm)	7390.7897
Pt2 (psia)	5.5344	Pt4 (psia)	117.7635	<input type="checkbox"/> W (pps)	674.5041
Tt2 (°R)	448.619	Tt4 (°R)	2372.0779	<input type="checkbox"/> Wf (pps)	1.8866
Pt21 (psia)	9.0642	Pt45 (psia)	43.7212	BPR	5.0873
Tt21 (°R)	527.2182	<input type="checkbox"/> Tt45 (°R)	1848.236	<input type="checkbox"/> FN (lbf)	11091.0146
<input type="checkbox"/> Pt25 (psia)	21.9592	Pt5 (psia)	8.9011		
<input type="checkbox"/> Tt25 (°R)	702.6741	Tt5 (°R)	1283.215		

(α)

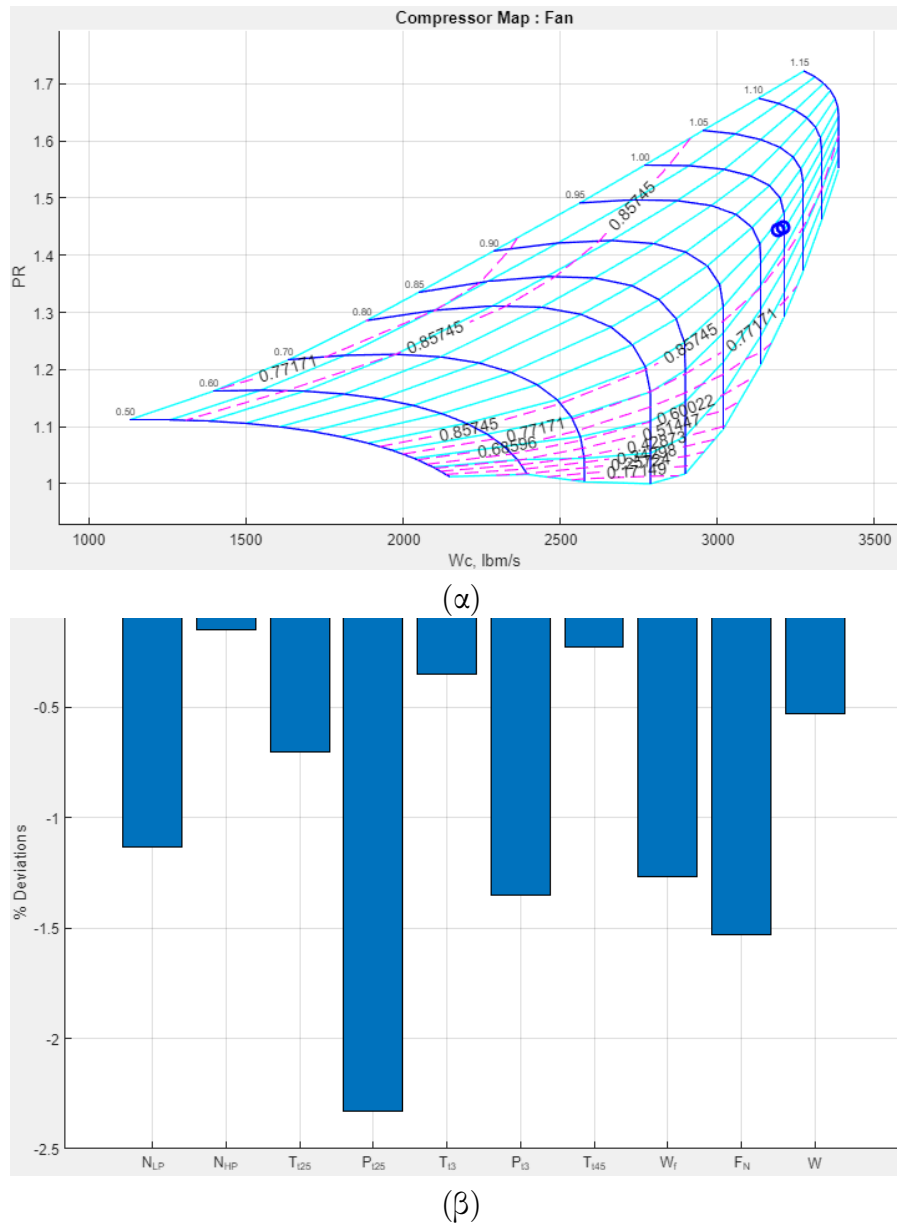


(β)

Σχήμα 47: Αποτελέσματα προσομοίωσης ενός σημείου στο (α) πλαίσιο των παραμέτρων απόδοσης και (β) στο σχηματικό του κινητήρα

Ακόμη, το σημείο που τρέχτηκε, καθώς και το ονομαστικό σημείο λειτουργίας σχεδιάζονται στους χάρτες λειτουργίας των στροβιλοσυριστών (βλ. πχ. για τον Fan

το Σχήμα 48α). Τέλος, οι ποσοστιαίες μεταβολές των μεταβλητών της υπογραφής του κινητήρα σε σχέση με την υγιή κατάστασή του φαίνονται στην καρτέλα της υπογραφής (βλ. Σχήμα 48β).



Σχήμα 48: Αποτελέσματα προσομοίωσης ενός σημείου (α) στον χάρτη του Fan και (β) στο διάγραμμα της υπογραφής του κινητήρα

### 3.3.2 Προσομοίωση πολλαπλών σημείων

Αρχικά, ο χρήστης πρέπει να τσεκάρει το πλαίσιο ελέγχου με ετικέτα "Operating Range" και να θέσει στο πεδίο που εμφανίζεται τον αριθμό σημείων που επιθυμεί. Στη συνέχεια, καλείται να επιλέξει ποια μεταβλητή επιθυμεί να μεταβάλλεται από

σημείο σε σημείο κατά τη διάρκεια της προσομοίωσης, είτε μία από τις μεταβλητές εισόδου, είτε μία από τις μεταβλητές ελέγχου, και να θέσει τις υπόλοιπες όπως στην περίπτωση της προσομοίωσης ενός σημείου. Στο παρόν παράδειγμα επιλέχθηκε να γίνει προσομοίωση για αριθμό Mach από 0.7 έως 0.8, όπως φαίνεται στο **Σχήμα 49**. Επίσης, μπορούν να τεθούν και τιμές για τις παραμέτρους υγείας με παρόμοιο τρόπο με την περίπτωση της προσομοίωσης ενός σημείου. Τέλος, η προσομοίωση ξεκινάει μέσω του κουμπιού "Run" στο πλαίσιο ελέγχου.

h (ft)  ☒ Operating Range: Points

dT (°R)

M

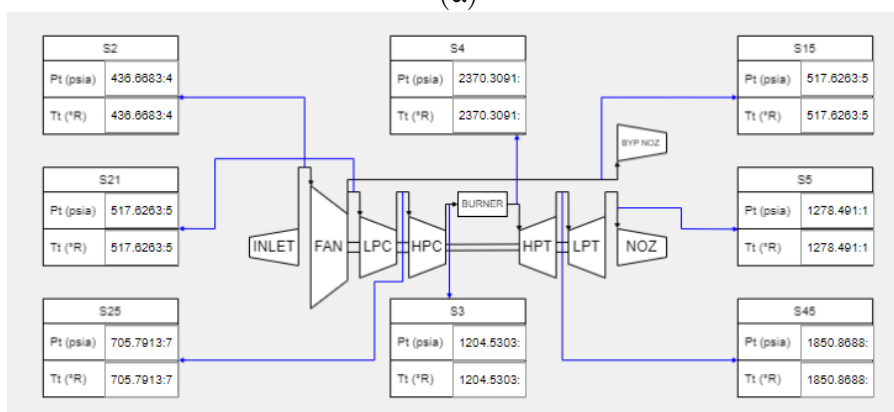
FAR

Σχήμα 49: Επιλογή εισόδων για προσομοίωση πολλαπλών σημείων

Τα αποτελέσματα της προσομοίωσης φαίνονται τόσο στο πλαίσιο των παραμέτρων απόδοσης, όσο και στο σχηματικό του κινητήρα, όπως παρουσιάζονται στο **Σχήμα 50**.

Performance Parameters					
<input type="checkbox"/> Pt15 (psia)	8.3253:9.0342	<input type="checkbox"/> Pt3 (psia)	121.8398:126.	<input type="checkbox"/> N_LP (rpm)	3758.0239:366
<input type="checkbox"/> Tt15 (°R)	517.6263:527.	<input type="checkbox"/> Tt3 (°R)	1204.5303:121	<input type="checkbox"/> N_HP (rpm)	7375.8337:740
<input type="checkbox"/> Pt2 (psia)	5.0361:5.5344	<input type="checkbox"/> Pt4 (psia)	115.1386:119.	<input type="checkbox"/> W (pps)	634.8682:678.
<input type="checkbox"/> Tt2 (°R)	436.6683:448.	<input type="checkbox"/> Tt4 (°R)	2370.3091:237	<input type="checkbox"/> Wf (pps)	1.8453:1.9108
<input type="checkbox"/> Pt21 (psia)	8.3882:9.1025	<input type="checkbox"/> Pt45 (psia)	43.1886:44.44	<input type="checkbox"/> BPR	4.8578:5.0421
<input type="checkbox"/> Tt21 (°R)	517.6263:527.	<input type="checkbox"/> Tt45 (°R)	1850.8688:185	<input type="checkbox"/> FN (lbf)	11455.5252:11
<input type="checkbox"/> Pt25 (psia)	21.8614:22.48	<input type="checkbox"/> Pt5 (psia)	8.6901:9.0198		
<input type="checkbox"/> Tt25 (°R)	705.7913:707.	<input type="checkbox"/> Tt5 (°R)	1278.491:1284		

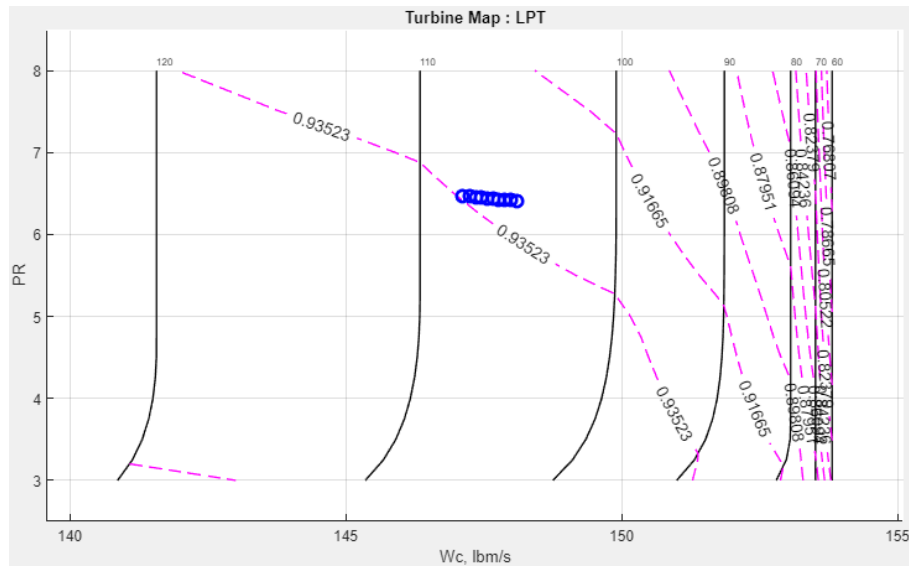
(α)



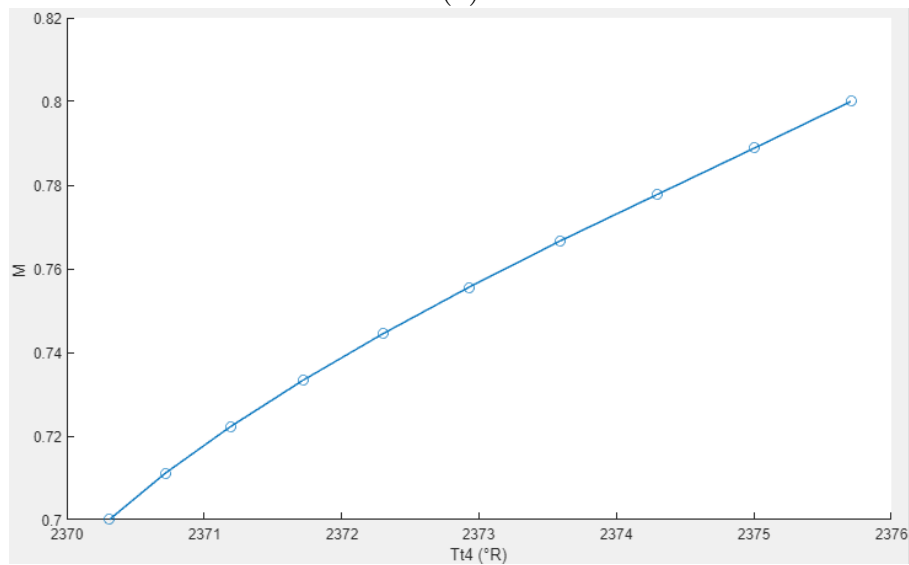
(β)

Σχήμα 50: Αποτελέσματα προσομοίωσης πολλαπλών σημείων στο (α) πλαίσιο των παραμέτρων απόδοσης και (β) στο σχηματικό του κινητήρα

Ακόμη, τα σημεία που τρέχτηκαν σχεδιάζονται στους χάρτες λειτουργίας των στροβιλοσυνιστωσών (βλ. πχ. για τον LPT το **Σχήμα 51α**). Τέλος, οι τιμές μίας μεταβλητής μπορούν να σχεδιαστούν συναρτήσει μίας οποιασδήποτε άλλης στην καρτέλα του γραφήματος των αποτελεσμάτων (βλ. πχ. το διάγραμμα  $Tt4 - Mach$  στο **Σχήμα 51β**).



(α)



(β)

Σχήμα 51: Αποτελέσματα προσομοίωσης πολλαπλών σημείων (α) στον χάρτη του LPT και (β) στο γράφημα των αποτελεσμάτων  $Tt4 - Mach$

### 3.3.3 Διάγνωση

Αρχικά, η εφαρμογή πρέπει να τεθεί σε λειτουργία διάγνωσης επιλέγοντας την κατάλληλη επιλογή στο πλαίσιο ελέγχου. Στο παρόν παράδειγμα παρουσιάζεται η



διαδικασία πραγματοποίησης διάγνωσης στον Fan, δηλαδή θα εκτιμηθούν οι τιμές των παραμέτρων υγείας του Fan. Οπότε, επιλέγονται τα αντίστοιχα πλαίσια ελέγχου στο πλαίσιο των παραμέτρων υγείας, όπως φαίνεται στο **Σχήμα 52**. Στη συνέχεια, ο χρήστης καλείται να επιλέξει τις κατάλληλες μεταβλητές από την υπογραφή του κινητήρα για την εκτίμηση των παραμέτρων υγείας. Αυτό μπορεί να γίνει απευθείας επιλέγοντας τα επιθυμητά πλαίσια ελέγχου στο πλαίσιο των παραμέτρων απόδοσης ή εναλλακτικά, αφού πατηθεί το κουμπί με ετικέτα "find measurements", να επιλεγθεί μία από τις επιλογές που παρουσιάζονται. Ακόμη, στα πεδία των επιλεγμένων μεταβλητών πρέπει να τεθεί η τιμή τους στην κατάσταση του κινητήρα στην οποία είναι επιθυμητό να πραγματοποιηθεί διάγνωση. Στο παρόν παράδειγμα τέθηκαν οι τιμές που προέκυψαν από προσομοίωση στο ονομαστικό σημείο λειτουργίας για παραμέτρους υγείας Fan -1% και οι δύο, όπως φαίνεται στο **Σχήμα 53**. Τέλος, η διάγνωση πραγματοποιείται, όπως και οι προσομοιώσεις, πατώντας το κουμπί "Run". Τα αποτελέσματα της διάγνωσης παρουσιάζονται στα αντίστοιχα πεδία του πλαισίου των παραμέτρων υγείας και, όπως φαίνεται στο **Σχήμα 54**, η διάγνωση ήταν επιτυχής.

**Health Parameters**

Reset

**Fan**

☒ Wc

☒ eff

**LPC**

☐ Wc

☐ eff

**HPC**

☐ Wc

☐ eff

**HPT**

☐ Wc

☐ eff

**LPT**

☐ Wc

☐ eff

Σχήμα 52: Επιλογή παραμέτρων υγείας για πραγματοποίηση διάγνωσης

**Performance Parameters**

Pt15 (psia) 8.9565

Tt15 (°R) 527.8177

Pt2 (psia) 5.5344

Tt2 (°R) 448.619

Pt21 (psia) 9.0242

Tt21 (°R) 527.8177

☐ Pt25 (psia) 22.4666

☐ Tt25 (°R) 709.6932

☐ Pt3 (psia) 125.7156

☒ Tt3 (°R) 1212.6276

Pt4 (psia) 118.8013

Tt4 (°R) 2377.2023

Pt45 (psia) 44.2861

☐ Tt45 (°R) 1854.1634

Pt5 (psia) 8.9754

Tt5 (°R) 1284.9949

☐ N\_LP (rpm) 3681.5116

☐ N\_HP (rpm) 7406.517

☒ W (pps) 672.5432

☐ Wf (pps) 1.901

BPR 5.0236

FN (lbf) 11083.6275

**Case Identifier** Unidentified

**Run**

**Find measurements** Tt3, W - CN: 1.1807

☐ Simulation ☒ Diagnosis

**Find health params**

Σχήμα 53: Επιλογή μεταβλητών υπογραφής για πραγματοποίηση διάγνωσης

**Health Parameters**

Reset

Fan		LPC		HPC		HPT	
<input checked="" type="checkbox"/> Wc	-1.00	<input type="checkbox"/> Wc	0.00	<input type="checkbox"/> Wc	0.00	<input type="checkbox"/> Wc	0.00
<input checked="" type="checkbox"/> eff	-1.00	<input type="checkbox"/> eff	0.00	<input type="checkbox"/> eff	0.00	<input type="checkbox"/> eff	0.00

LPT	
<input type="checkbox"/> Wc	0.00
<input type="checkbox"/> eff	0.00

Σχήμα 54: Αποτελέσματα παραδείγματος διάγνωσης

### 3.4 Ιακωβιανός πίνακας και αριθμός κατάστασης

Ο Ιακωβιανός πίνακας ή αλλιώς Ιακωβιανή, είναι ένας πίνακας ο οποίος, για την περίπτωση της διάγνωσης ενός αεριοστροβίλου, αντιπροσωπεύει την ευαισθησία ενός συνόλου μετρούμενων μεγεθών σε μία αλλαγή ενός συνόλου παραμέτρων υγείας. Κάθε όρος του πίνακα αυτού προκύπτει από την Εξίσωση 1 και αντιπροσωπεύει πόσο ευαίσθητη είναι μία μέτρηση  $Y_i$  σε μοναδιαία μεταβολή της παραμέτρου υγείας  $f_j$  [12].

$$\Delta Y_i^j = \frac{Y_i^j - Y_i^{ref}}{Y_i^{ref}} \cdot 100 \quad (1)$$

όπου:

$i$ : δείκτης που αντιστοιχεί στην μέτρηση

$j$ : δείκτης που αντιστοιχεί στη παράμετρο υγείας

$Y_i^j$ : τιμή της μέτρησης  $i$  κατά την μεταβολή της παραμέτρου υγείας  $j$

$Y_i^{ref}$ : τιμή της μέτρησης  $i$  στην κατάσταση αναφοράς

Στην παρούσα εργασία, ο Ιακωβιανός πίνακας υπολογίστηκε μέσω του προγράμματος calcJacobian (βλ. **Παράρτημα Ι**). Μέσω αυτού, αρχικά, πραγματοποιήθηκε προσομοίωση με μηδενικές παραμέτρους υγείας και οι τιμές των μετρήσεων (μεταβλητές της υπογραφής του κινητήρα) αποθηκεύτηκαν σε ένα μονοδιάστατο πίνακα (array). Στη συνέχεια, για κάθε μία από τις παραμέτρους υγείας, υπολογίστηκαν οι αντίστοιχοι όροι του Ιακωβιανού πίνακα, τόσο για θετική, όσο και για αρνητική μοναδιαία μεταβολή της μέσω της Εξίσωσης 1. Τέλος, με σκοπό την αύξηση της ακρίβειας, προέκυψε ο "μέσος" Ιακωβιανός πίνακας αφαιρώντας από κάθε όρο του πίνακα που προέκυψε από τις θετικές μεταβολές τον αντίστοιχο όρο του πίνακα που προήλθε από τις αρνητικές μεταβολές των παραμέτρων υγείας και διαιρώντας με το 2.

Επιπλέον, με σκοπό την εύρεση του καταλληλότερου συνδυασμού μετρήσεων για ένα δεδομένο σύνολο παραμέτρων υγείας και το αντίθετο, για την πραγματοποίηση

διάγνωσης σε έναν αεριοστρόβιλο, είναι απαραίτητη η χρήση ενός κριτηρίου αποτελεσματικής εκτίμησης. Αφού κατά τη διάγνωση επιλύεται ένα γραμμικό σύστημα, ένα τέτοιο κριτήριο είναι ο αριθμός κατάστασης (condition number), καθώς έχει άμεση σχέση με τη ικανότητά του για ακριβείς εκτιμήσεις. Ο αριθμός κατάστασης ορίζεται ως η νόρμα 2 του εκάστοτε Ιακωβιανού πίνακα (βλ. Εξίσωση 2), δηλαδή είναι ο λόγος της μέγιστης προς την ελάχιστη ιδιομορφή του πίνακα αυτού. Στο σημείο αυτό αξίζει να σημειωθεί ότι ο αριθμός κατάστασης του Ιακωβιανού πίνακα ενός συστήματος είναι αντιστρόφως ανάλογος του βαθμού ενίσχυσης των διαταραχών κατά την διάγνωση, με συνέπεια για την επιτυχή πραγματοποίησή της είναι απαραίτητη η επιλογή συνδυασμού μετρήσεων και παραμέτρων υγείας που οδηγούν σε ένα Ιακωβιανό πίνακα με όσο το δυνατόν χαμηλότερο αριθμό κατάστασης [12].

$$\kappa = \frac{\max\{w_j\}}{\min\{w_j\}} \quad (2)$$

Τέλος, με σκοπό την επιλογή των κατάλληλων μετρήσεων για δεδομένες παραμέτρους υγείας, σχηματίζονται όλοι οι δυνατοί συνδυασμοί μετρήσεων και παραμέτρων υγείας διατηρώντας τις στήλες του Ιακωβιανού πίνακα και αλλάζοντας τις γραμμές. Ενώ, για την επιλογή των κατάλληλων παραμέτρων υγείας για δεδομένες μετρήσεις, σχηματίζονται και πάλι όλοι οι δυνατοί συνδυασμοί μετρήσεων και παραμέτρων υγείας διατηρώντας τις γραμμές του Ιακωβιανού πίνακα και αλλάζοντας τις στήλες. Ακόμη, και για τις δύο περιπτώσεις, υπολογίζεται ο αριθμός κατάστασης της Ιακωβιανής όλων των συνδυασμών και αυτοί κατατάσσονται από αυτόν με τον χαμηλότερο αριθμό κατάστασης προς το μεγαλύτερο [12].

### 3.5 Αρχικές συνθήκες επιλύτη

Με σκοπό την διευκόλυνση της σύγκλισης του επιλύτη είναι απαραίτητη η δημιουργία κατάλληλων πινάκων που να περιέχουν τις κατάλληλες αρχικές τιμές για τις ανεξάρτητες μεταβλητές του συστήματος συναρτήσει του αριθμού Mach και της διορθωμένης καθαράς ώσης.

Προς αυτή τη κατεύθυνση αρχικά μέσω του προγράμματος calcInitTable (βλ. **Παράρτημα Ι**) πραγματοποιήθηκε προσομοίωση σε μία κατάλληλη ακολουθία σημείων που καλύπτουν ικανοποιητικά τον φάκελο πτήσης (flight envelope) του κινητήρα του μοντέλου. Οι μεταβλητές που μεταβάλλονταν από σημείο σε σημείο ήταν οι εξής:

- Το υψόμετρο με τιμές από 0 m έως 10000 m με βήμα 1000 m και για 10668 m
- Ο αριθμός Mach με τιμές από 0 έως 0.8 με βήμα 0.1
- Ο λόγος ισχύος (throttle ratio - TR) με τιμές από 0.5 έως 1 με βήμα 0.125

Ο λόγος ισχύος ορίζεται ως ο λόγος της μέγιστης ολικής θερμοκρασίας εξόδου από τον θάλαμο καύσης προς την αντίστοιχη θερμοκρασία στο επίπεδο της θάλασσας σε τυπικές στατικές συνθήκες [10]. Το σύνολο των σημείων στα οποία πραγματοποιήθηκε προσομοίωση περιλαμβάνουν όλους τους συνδυασμούς των τριών παραπάνω μεταβλητών και είναι 540 στο σύνολο ( $12 \cdot 9 \cdot 5 = 540$ ). Ακόμη, με σκοπό την σύγκλιση όσο το δυνατόν περισσότερων σημείων έγινε χρήση της ακολουθίας που επεξηγείται μέσω του **Πίνακα 6**, όπου όταν αλλάζει μία παράμετρος οι άλλες δύο παραμένουν σταθερές ώστε η μετάβαση από σημείο σε σημείο να είναι όσο πιο ομαλή γίνεται, καθώς σε κάθε σημείο ως αρχικές συνθήκες στον επιλύτη τέθηκαν οι τελικές τιμές των ανεξάρτητων μεταβλητών στο προηγούμενο σημείο που συνέκλινε.

Πίνακας 6: Ακολουθία σημείων που καλύπτουν τον φάκελο πτήσης

Υψόμετρο	TR	Mach
10668	1	0.8
10668	1	:
10668	1	0
10668	0.875	0
10668	0.875	:
10668	0.875	0.8
10668	0.75	0.8
10668	:	:
10668	0.5	0
10000	0.5	0
10000	0.5	:
10000	0.5	0.8
10000	0.625	0.8
10000	0.625	:
10000	0.625	0
10000	0.75	0
10000	:	:
10000	1	0.8
9000	1	0.8
:	:	:
0	1	0.8

Για κάθε σημείο του **Πίνακα 6** ακολουθήθηκε η εξής διαδικασία: Αρχικά, ως αρχικές συνθήκες στον επιλύτη τέθηκαν οι τελικές τιμές των ανεξάρτητων μεταβλητών του τελευταίου σημείου που συνέκλινε εκτός του λόγου καυσίμου-αέρα, στον οποίο τέθηκε η τιμή 0.018 σε κάθε τρέξιμο (βλ. **υποκεφάλαιο 3.1.3**), καθώς οι τιμές του κυμαίνονται αρκετά κοντά σε αυτή την τιμή σε όλες σχεδόν τις περιπτώσεις. Για να είναι διαθέσιμες βέβαια οι τιμές αυτές στον χώρο εργασίας της Matlab, προστέθηκε στο μοντέλο Simulink ένα ToWorkspace block στην κατάλληλη θέση. Στη συνέχεια, υπολογίστηκαν ο αδιάστατος λόγος της στατικής θερμοκρασίας περιβάλλοντος προς την στατική θερμοκρασία στο επίπεδο της θάλασσας για την τυπική

ατμόσφαιρα  $\theta$  και ο αντίστοιχος λόγος για την πίεση  $\delta$ , μέσω των Εξισώσεων 3 και 4 [7], όπου  $alt$ , το υψόμετρο. Έπειτα, μέσω των Εξισώσεων 5 και 6, υπολογίστηκαν ο αδιάστατος λόγος της ολικής θερμοκρασίας περιβάλλοντος προς την στατική θερμοκρασία στο επίπεδο της θάλασσας για την τυπική ατμόσφαιρα  $\theta_0$  και ο αντίστοιχος λόγος για την πίεση  $\delta_0$ . Ακολούθως, υπολογίστηκε ο λόγος της εγκατεστημένης ώσης προς την αντίστοιχη στο επίπεδο της θάλασσας (installed full throttle thrust lapse) μέσω της Εξίσωσης 7. Ακόμη, η καθαρή ώση υπολογίστηκε από την Εξίσωση 8 [10], όπου η καθαρή ώση απογείωσης  $FN_{TO}$  είναι ίση με 50000 lbf[1]. Εδώ, αξίζει να σημειωθεί ότι ως κάτω όριο της καθαρής ώσης επιλέχθηκε το 3% της καθαρής ώσης απογείωσης.

$$\theta = 1 - 0.022557696 \cdot \frac{alt}{1000} \quad (3)$$

$$\delta = \left( 1 - 0.022557696 \cdot \frac{alt}{1000} \right)^{5.2558798} \quad (4)$$

$$\theta_0 = \theta \cdot (1 + 0.2 \cdot Mach^2) \quad (5)$$

$$\delta_0 = \delta \cdot (1 + 0.2 \cdot Mach^2)^{1.4/0.4} \quad (6)$$

$$\alpha = \begin{cases} \delta_0 \cdot (1 - 0.49\sqrt{Mach}) & \theta_0 \leq TR \\ \delta_0 \cdot \left( 1 - 0.49\sqrt{Mach} - \frac{3(\theta_0 - TR)}{1.5 + Mach} \right) & \theta_0 > TR \end{cases} \quad (7)$$

$$FN = a \cdot FN_{TO} \quad (8)$$

Στη συνέχεια, τέθηκαν οι τιμές των εισόδων και της μεταβλητής ελέγχου (καθαρή ώση) στο μοντέλο Simulink και πραγματοποιήθηκε προσομοίωση σε αυτό. Τέλος, οι είσοδοι, τα διάφορα υπολογισμένα μεγέθη, οι τελικές τιμές των ανεξάρτητων μεταβλητών και των λοιπών μεταβλητών ελέγχου (λόγος αέρα-καυσίμου και μεταβλητές της υπογραφής του κινητήρα), καθώς και αν συνέκλινε ο επιλύτης, αποθηκεύτηκαν σε έναν πίνακα. Στο σημείο αυτό αξίζει να σημειωθεί ότι τα μεγέθη που αφορούν παροχές, στροφές ή ώση αποθηκεύτηκαν στην διορθωμένη τους μορφή με σκοπό τη γενικότητα των αποθηκευμένων δεδομένων (βλ. Εξισώσεις 9, 10 και 11) [13].

$$W_{cor} = \frac{W \cdot \sqrt{\theta}}{\delta} \quad (9)$$

$$N_{cor} = \frac{N}{\sqrt{\theta}} \quad (10)$$

$$FN_{cor} = \frac{FN}{\delta} \quad (11)$$

Επόμενο βήμα ήταν η δημιουργία των απαραίτητων πινάκων ώστε να μπορούν να εκτιμηθούν οι αρχικές τιμές των ανεξάρτητων μεταβλητών του επιλύτη πριν το τρέξιμο του μοντέλου, το οποίο πραγματοποιήθηκε μέσω του προγράμματος

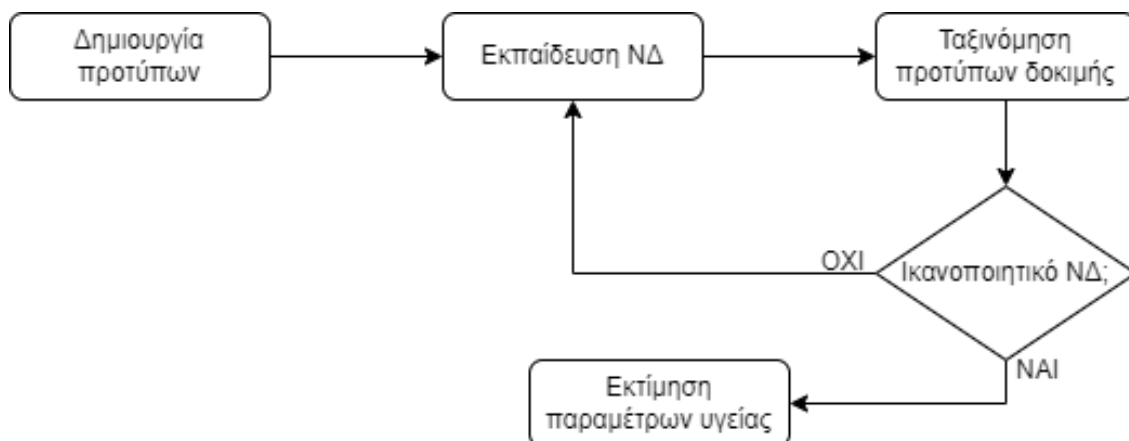
generateTables (βλ. **Παράρτημα Ι**). Αρχικά, αφότου αφαιρέθηκαν τα σημεία που δεν συνέκλιναν, τα δεδομένα χωρίστηκαν σε υποπίνακες ανάλογα με το αριθμό Mach. Στην συνέχεια, βρέθηκε η ελάχιστη και η μέγιστη διορθωμένη καθαρή ώση όλων των σημείων και το διάστημα ανάμεσα τους χωρίστηκε σε 150 υποδιαστήματα. Έπειτα, για κάθε υποπίνακα και για όλες τις ανεξάρτητες μεταβλητές και λοιπές μεταβλητές ελέγχου πραγματοποιήθηκε γραμμική παρεμβολή ανάμεσα στην εκάστοτε μεταβλητή και την διορθωμένη καθαρή ώση για κάθε μία από τις 151 τιμές της διορθωμένης καθαρής ώσης και τα αποτελέσματα αποθηκεύτηκαν στην κατάλληλη στήλη του αντίστοιχου πίνακα που θα χρησιμοποιηθεί για την εκτίμηση των αρχικών συνθηκών. Όμως, καθώς οι τιμές της διορθωμένης καθαρής ώσης στις οποίες πραγματοποιήθηκε γραμμική παρεμβολή αφορούν όλα τα δεδομένα, κάποιες τιμές ήταν εκτός του διαστήματος της ελάχιστης και μέγιστης διορθωμένης ώσης για ένα εκάστοτε αριθμό Mach με αποτέλεσμα σε κάποια σημεία η γραμμική παρεμβολή να αποτύχει. Γι' αυτό τον λόγο ήταν απαραίτητη η πραγματοποίηση γραμμικής προεκβολής (extrapolation). Αυτή πραγματοποιήθηκε για κάθε υποπίνακα και για κάθε μεταβλητή μέσω της συνάρτησης `extr`, η οποία περιλαμβάνεται στο πρόγραμμα `generateTables`, ενώ στις περιπτώσεις που προέκυψαν αρνητικές τιμές μέσω της προεκβολής, αυτές τέθηκαν ίσες με μία πολύ μικρή τιμή. Έτσι, δημιουργήθηκε ένας πίνακας  $151 \times 9$  για κάθε μεταβλητή.

Τέλος, οι πίνακες που δημιουργήθηκαν μέσω του προγράμματος `generateTables` χρησιμοποιούνται από την συνάρτηση `getNRIC` με τον ακόλουθο τρόπο: Αν η μεταβλητή ελέγχου είναι η καθαρή ώση, τότε χρησιμοποιώντας την διορθωμένη της τιμή και τον αριθμό Mach πραγματοποιείται δισδιάστατη γραμμική παρεμβολή στους πίνακες των ανεξάρτητων μεταβλητών. Σε αντίθετη περίπτωση, χρησιμοποιείται η τιμή της μεταβλητής ελέγχου, η οποία, αν είναι απαραίτητο, μετατρέπεται στη διορθωμένη της μορφή, για να βρεθεί η αντίστοιχη τιμή της διορθωμένης καθαρής ώσης και στη συνέχεια αυτή χρησιμοποιείται για την εύρεση των αρχικών συνθηκών του επιλύτη όπως στην προηγούμενη περίπτωση. Εδώ αξίζει να σημειωθεί ότι στις αρχικές συνθήκες χρησιμοποιείται πάντα η τιμή 0.02 για τον λόγο καυσίμου-αέρα γιατί οι τιμές που λαμβάνει είναι πάντα κοντά στην τιμή αυτή.

## 4 Διερεύνηση μεθόδων διάγνωσης μέσω Νευρωνικών Δικτύων

### 4.1 Αλγόριθμος διάγνωσης

Πρώτο βήμα για την πραγματοποίηση διάγνωσης μέσω ενός Νευρωνικού Δικτύου (ΝΔ) είναι η δημιουργία κατάλληλων προτύπων εκπαίδευσης και δοκιμής. Είναι σημαντικό να παραχθεί ένας ικανοποιητικός αριθμός προτύπων τόσο για κάθε περίπτωση βλάβης που είναι επιθυμητό να μπορεί το ΝΔ να αναγνωρίσει, όσο και για την υγιή κατάσταση του κινητήρα, καθώς με αυτόν τον τρόπο, μετά την εκπαίδευσή του, το ΝΔ τείνει να πραγματοποιεί ακριβέστερη ταξινόμηση των προτύπων δοκιμής. Στο σημείο αυτό αξίζει να σημειωθεί ότι στην παρούσα εργασία όλα τα πρότυπα προέκυψαν μέσω του μοντέλου Simulink που αναπτύχθηκε για τα πλαίσια της εφαρμογής που περιγράφηκε στο προηγούμενο κεφάλαιο. Όμως, τα πρότυπα που προέρχονται από έναν πραγματικό κινητήρα εμπεριέχουν θόρυβο. Οπότε, είναι απαραίτητη η εισαγωγή θορύβου στα πρότυπα που προέρχονται από το μοντέλο ώστε να εξομοιωθεί καλύτερα η πραγματικότητα. Στη συνέχεια, το ΝΔ εκπαιδεύεται μέσω των προτύπων εκπαίδευσης και αξιολογείται η ακρίβειά του. Αν το ΝΔ κριθεί ικανοποιητικό, τότε, αν είναι επιθυμητό, εκτιμώνται οι παράμετροι υγείας του κινητήρα για ένα δεδομένο πρότυπο μέσω του μοντέλου Simulink, ενώ σε αντίθετη περίπτωση το ΝΔ επανεκπαιδεύεται. Ο αλγόριθμος διάγνωσης που περιγράφηκε παραπάνω παρουσιάζεται σχηματικά στο **Σχήμα 55**. Επιπλέον, η επιλογή του κατάλληλου τύπου ΝΔ που είναι επιθυμητό να εκπαιδευτεί και να χρησιμοποιηθεί για ταξινόμηση μπορεί να πραγματοποιηθεί μέσω της συνάρτησης `diagnoseUsingNN` (βλ. **Παράρτημα II**).



Σχήμα 55: Διάγραμμα ροής του αλγορίθμου διάγνωσης

### 4.2 Δημιουργία προτύπων

Αρχικά, επιλέχθηκαν οι κλάσεις στις οποίες ανήκουν τα πρότυπα. Στην παρούσα εργασία οι κλάσεις που χρησιμοποιήθηκαν είναι:

1. Βλάβη στον Fan
2. Βλάβη στον LPC
3. Βλάβη στον HPC
4. Βλάβη στον HPT
5. Βλάβη στον LPT
6. Υγιής κατάσταση κινητήρα

Στη συνέχεια, ακολουθώντας την μεθοδολογία που προτείνεται από τον [9], υπολογίστηκαν οι παράμετροι υγείας για την περίπτωση βλάβης σε κάθε συνιστώσα ξεχωριστά ως εξής: Κατ' αρχάς, επιλέχθηκε μέγεθος βλάβης από 0.5% έως 7% με βήμα 0.5% για την περίπτωση των προτύπων εκπαίδευσης και από 0.625% έως 6.75% με βήμα 0.5% για την περίπτωση των προτύπων δοκιμής. Το μέγεθος βλάβης ορίζεται μέσω της Εξίσωσης 12 και θεωρείται πως η τιμή του είναι πάντα αρνητική.

$$|FM| = \sqrt{SW^2 + SE^2} \quad (12)$$

όπου:

$SW$ : Παράμετρος υγείας διορθωμένης παροχής

$SE$ : Παράμετρος υγείας βαθμού απόδοσης

Συνακόλουθα, με σκοπό τον προσδιορισμό των παραμέτρων υγείας επιλέχθηκαν προκαθορισμένοι λόγοι ανάμεσα σε  $SW$  και  $SE$ . Συγκεκριμένα, για τους συμπιεστές χρησιμοποιήθηκαν οι λόγοι 1, 1.5 και 2, ενώ για τους στροβίλους -0.5, -0.75 και -1. Οπότε, οι παράμετροι υγείας δίνονται από τις Εξισώσεις 13 και 14.

$$SE = \frac{FM}{\sqrt{(SW/SE)^2 + 1}} \quad (13)$$

$$SW = (SW/SE) \cdot SE \quad (14)$$

Τέλος, για κάθε ζεύγος παραμέτρων υγείας και για κάθε συνιστώσα πραγματοποιήθηκε προσομοίωση μέσω του μοντέλου Simulink, αποθηκεύοντας τις μεταβολές των μεταβλητών του προτύπου, οι οποίες στην παρούσα εργασία είναι οι εξής:  $N_{HP}$ ,  $Tt25$ ,  $Pt25$ ,  $Tt3$ ,  $Pt3$ ,  $Tt45$  και  $Wf$ . Ως μεταβλητή ελέγχου επιλέχθηκαν οι στροφές χαμηλής πίεσης. Οι μεταβολές των μεγεθών ορίζονται μέσω της Εξίσωσης 15.

$$\Delta Y = \left( \frac{Y_{given}}{Y_{ref}} - 1 \right) \cdot 100 \quad (15)$$

όπου:

$Y_{given}$ : Η τρέχουσα τιμή του μεγέθους

$Y_{ref}$ : Η τιμή του μεγέθους στην υγιή κατάσταση του κινητήρα



Στην παρούσα εργασία, η παραπάνω διαδικασία πραγματοποιήθηκε μέσω του προγράμματος `getPatterns` (βλ. **Παράρτημα II**). Για την περίπτωση της υγιούς κατάστασης του κινητήρα οι παράμετροι υγείας προέκυψαν θέτοντας σε κάθε μία από αυτές μια τυχαία τιμή με κανονική κατανομή και τυπική απόκλιση 0.05% μέσω του προγράμματος `getHealthyPatterns` (βλ. **Παράρτημα II**).

### 4.3 Θόρυβος

Η εισαγωγή θορύβου στα πρότυπα που δημιουργήθηκαν πραγματοποιήθηκε μέσω της συνάρτησης `getNoisyPatterns` (βλ. **Παράρτημα II**). Πιο συγκεκριμένα, σε κάθε μεταβλητή ενός προτύπου προστίθεται τυχαία μία τιμή με κανονική κατανομή, τυπική απόκλιση που προτείνεται από τον [9] (βλ. **Πίνακα 7**) και μέση τιμή την τιμή της χωρίς θόρυβο (βλ. Εξίσωση 16).

$$\Delta Y_{noisy} = \Delta Y_{noise-free} + \sigma \cdot n \quad (16)$$

όπου:

$\Delta Y_{noisy}$ : Η τιμή της μεταβλητής με θόρυβο

$\Delta Y_{noise-free}$ : Η τιμή της μεταβλητής χωρίς θόρυβο

$\sigma$ : Η τυπική απόκλιση της μεταβλητής

$n$ : Τυχαίος αριθμός με κανονική κατανομή, ο οποίος έχει 99.73% πιθανότητα να έχει τιμή από -3 έως 3

Πίνακας 7: Τυπική απόκλιση των μεταβλητών της υπογραφής

Μεταβλητή	$\sigma$ [%]
N_HP	0.17
Tt25	0.16
Pt25	0.2
Tt3	0.16
Pt3	0.2
Tt45	0.16
Wf	0.6

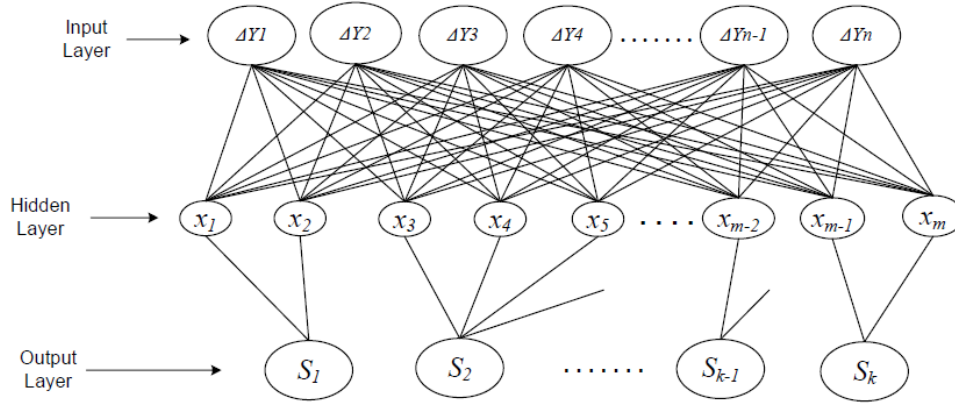
### 4.4 Τύποι Νευρωνικών Δικτύων

#### 4.4.1 Πιθανολογικό Νευρωνικό Δίκτυο

Το Πιθανολογικό Νευρωνικό Δίκτυο (Probabilistic Neural Network - PNN) είναι ένα ΝΔ πρόσθιας τροφοδότησης πολλαπλών επιπέδων, το οποίο ταξινομεί πρότυπα εισόδου σε προκαθορισμένες κλάσεις εκτιμώντας την πιθανότητα να ανήκουν

σε κάθε μία από αυτές [9]. Μια σχηματική αναπαράσταση του PNN παρουσιάζεται στο **Σχήμα 56**.

Το επίπεδο εισόδου έχει ένα κόμβο για κάθε μεταβλητή των προτύπων, το κρυφό επίπεδο διαθέτει τόσους κόμβους όσα είναι τα πρότυπα εκπαίδευσης και ο αριθμός των κόμβων του επιπέδου εξόδου είναι ίσος με τον αριθμό των κλάσεων. Εδώ αξίζει να σημειωθεί ότι κάθε πρότυπο εκπαίδευσης ανήκει σε μία και μόνο κλάση. Οπότε, κάθε κόμβος του κρυφού επιπέδου συνδέεται με ένα μόνο κόμβο εξόδου [9].



Σχήμα 56: Σχηματική αναπαράσταση του PNN [9]

Για ένα δεδομένο πρότυπο εισόδου το PNN υπολογίζει την συνάρτηση πυκνότητας πιθανότητας τύπου Gauss (βλ. Εξίσωση 17) για κάθε κλάση  $S_i$  [9].

$$P(S_i|x) = \frac{P(S_i)}{P(x) \cdot 2\pi^{\frac{m}{2}} \cdot \sigma_i^m \cdot |S_i|} \cdot \sum_{j=1}^{q_i} \exp \left[ \frac{-\left(x - x_j^{(i)}\right)^T \left(x - x_j^{(i)}\right)}{2\sigma_i^2} \right] \quad (17)$$

όπου:

$x$ : Το πρότυπο εισόδου

$x_j^{(i)}$ : Το  $j$ -οστό πρότυπο εκπαίδευσης που ανήκει στην κλάση  $S_i$

$q_i = |S_i|$ : Ο αριθμός των προτύπων εκπαίδευσης που ανήκουν στην κλάση  $S_i$

$\sigma_i$ : Η παράμετρος εξομάλυνσης της κλάσης  $S_i$

$m$ : Ο συνολικός αριθμός των προτύπων εκπαίδευσης

$P(S_i)$ : Η εκ των προτέρων (a-priori) πιθανότητα της κλάσης  $S_i$

$P(x)$ : Παράγοντας κανονικοποίησης

Αν υποτεθούν ίσες εκ των προτέρων πιθανότητες για όλες τις κλάσεις τότε:  $P(S_i) = \frac{1}{k}$ , όπου  $k$  ο αριθμός των κλάσεων. Ακόμη, αξίζει να σημειωθεί ότι ο παράγοντας κανονικοποίησης  $P(x)$  αντιπροσωπεύει την εκ των προτέρων πιθανότητα του προτύπου εισόδου  $x$ , ο οποίος είναι σταθερός αν υποτεθούν εξαντλητικές και αλληλοαποκλειόμενες μεταξύ τους κλάσεις και υπολογίζεται αν το άθροισμα των συναρτήσεων πυκνότητας πιθανότητας τεθεί ίσο με την μονάδα. Τέλος, στην παρούσα εργασία διάγνωση μέσω PNN πραγματοποιήθηκε μέσω της συνάρτησης PNN (βλ. **Παράρτημα II**).

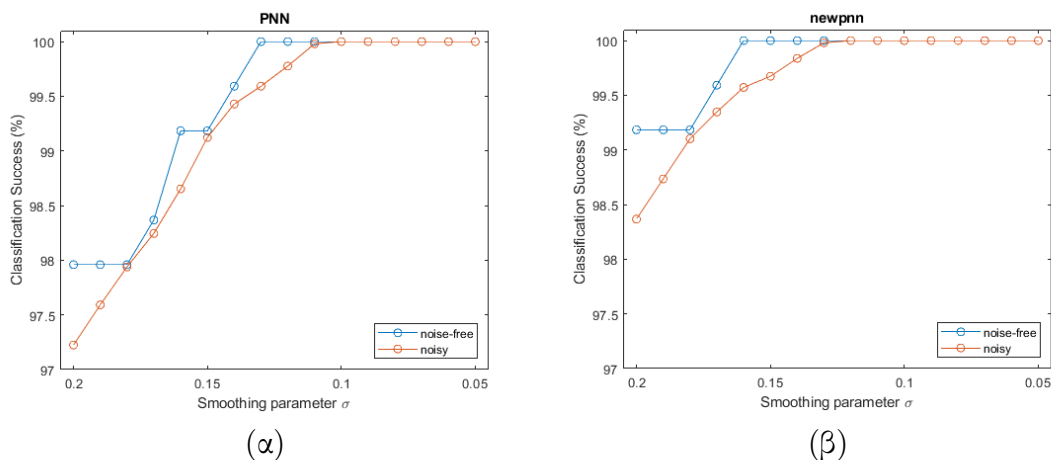
Παράλληλα, για την δημιουργία ενός PNN χρησιμοποιήθηκε και η συνάρτηση της Matlab `newpnn`, η οποία δημιουργεί ένα PNN δύο επιπέδων που χρησιμοποιεί συναρτήσεις ακτινικής βάσης. Η διάγνωση μέσω της `newpnn` πραγματοποιήθηκε μέσω της συνάρτησης `newpnnFcn` (βλ. Παράρτημα II).

#### 4.4.2 Λοιπά ΝΔ πρόσθιας τροφοδότησης

Δύο επιπλέον μέθοδοι διάγνωσης μέσω ΝΔ είναι η χρήση των συναρτήσεων `feedforwardnet` και `patternnet` της Matlab. Η πρώτη δημιουργεί ένα ΝΔ πρόσθιας τροφοδότησης «γενικής χρήσεως», ενώ, η δεύτερη, ένα το οποίο είναι εξειδικευμένο για εφαρμογές αναγνώρισης προτύπων και ταξινόμησης. Στην παρούσα εργασία χρησιμοποιήθηκε η προκαθορισμένη αρχιτεκτονική και για τις δύο περιπτώσεις. Δηλαδή, τα ΝΔ που δημιουργήθηκαν έχουν ένα κρυφό επίπεδο με 10 νευρώνες. Τέλος, η διάγνωση κάνοντας χρήση αυτών των ΝΔ έγινε μέσω των συναρτήσεων `diagFeedforwardnet` και `diagPatternnet` (βλ. Παράρτημα II).

### 4.5 Εκπαίδευση των ΝΔ

Η εκπαίδευση ενός PNN πραγματοποιείται μέσω της εύρεσης της τιμής της παραμέτρου εξομάλυνσης για την οποία όλα τα πρότυπα εκπαίδευσης ταξινομούνται στη σωστή κλάση [9], η οποία στην παρούσα εργασία θεωρείται ίδια για κάθε κλάση. Τόσο για την περίπτωση του PNN, όσο και για εκείνη του `newpnn` πραγματοποιήθηκε εκπαίδευση με και χωρίς θόρυβο στα πρότυπα μέσω του προγράμματος `trainPNN` (βλ. Παράρτημα II). Όπως φαίνεται στο Σχήμα 57, καθώς μειώνεται η τιμή της παραμέτρου εξομάλυνσης  $\sigma$ , το ποσοστό επιτυχούς ταξινόμησης αυξάνεται σε όλες τις περιπτώσεις. Για λόγους απλότητας στην παρούσα εργασία επιλέχθηκε  $\sigma = 0.1$  καθώς είναι η υψηλότερη τιμή της που δίνει 100% επιτυχία.



Σχήμα 57: Ποσοστό επιτυχούς ταξινόμησης συναρτήσει της παραμέτρου εξομάλυνσης για πρότυπα χωρίς θόρυβο (μπλε) και με θόρυβο (πορτοκαλί) για το (α) PNN και (β) `newpnn`

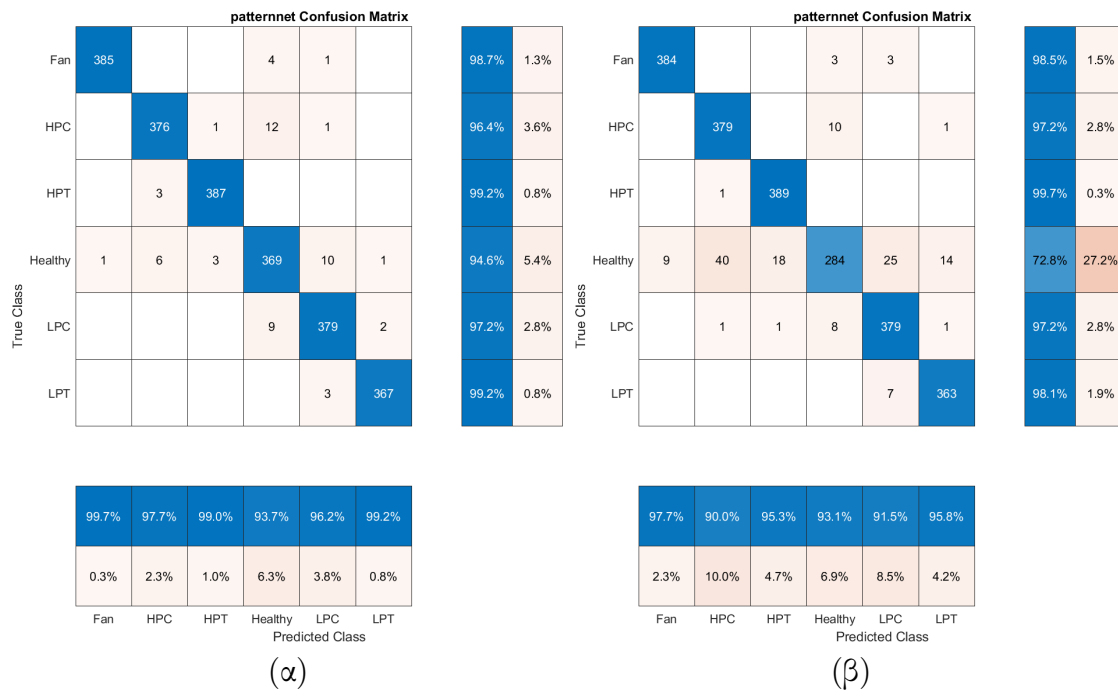
Η εκπαίδευση των λοιπών ΝΔ πραγματοποιήθηκε μέσω της συνάρτησης `train` της Matlab χρησιμοποιώντας όλες τις διαθέσιμες συναρτήσεις εκπαίδευσης τόσο με χαμηλό, όσο και με υψηλό θόρυβο. Τα αποτελέσματα συνοψίζονται στους **Πίνακες 8** και **9**. Σε αυτούς παρατηρείται ότι πέραν των `traingdm` και `traingd` τα ποσοστά επιτυχίας είναι παρόμοια μεταξύ τους και επιπλέον, όπως ήταν αναμενόμενο, είναι χαμηλότερα στην περίπτωση του υψηλότερου θορύβου. Για λόγους απλότητας στην παρούσα εργασία επιλέχθηκαν οι προεπιλεγμένες συναρτήσεις εκπαίδευσης, δηλαδή η `trainscg` για την περίπτωση του `patternnet` και η `trainlm` για εκείνη του `feedforwardnet`. Επιπλέον, στο **Σχήμα 58** παρουσιάζονται οι πίνακες σύγχυσης για εκπαίδευση με και χωρίς θόρυβο και όπως ήταν αναμενόμενο στην περίπτωση εκπαίδευσης με θόρυβο τα ποσοστά επιτυχούς ταξινόμησης είναι υψηλότερα.

Πίνακας 8: Ποσοστά επιτυχούς ταξινόμησης για χαμηλό θόρυβο

Συνάρτηση	Fan	LPC	HPC	HPT	LPT	Υγιής	Συνολικά
<code>trainlm</code>	98.7	96.9	96.2	99.7	99.5	95.1	97.7
<code>trainbr</code>	99.2	97.2	95.6	99.5	99.2	95.4	97.7
<code>trainbfg</code>	98.7	97.7	96.7	99.5	98.9	94.1	97.6
<code>trainrp</code>	98.7	96.9	96.2	99.5	99.5	94.4	97.5
<code>trainscg</code>	98.7	97.2	96.4	99.5	99.2	94.1	97.5
<code>traincgb</code>	98.7	96.9	96.2	99.5	99.5	94.4	97.5
<code>traincgf</code>	98.7	97.2	96.7	99.5	99.2	94.1	97.6
<code>traincgp</code>	98.7	96.9	97.4	99.7	99.2	94.4	97.7
<code>trainoss</code>	98.7	97.4	95.6	99.0	98.9	94.1	97.3
<code>traingdx</code>	98.7	96.2	94.9	99.2	98.4	95.9	97.2
<code>traingdm</code>	100.0	0.0	0.0	0.0	31.4	0.0	21.9
<code>traingd</code>	100.0	0.0	0.0	0.0	31.9	0.0	22.0

Πίνακας 9: Ποσοστά επιτυχούς ταξινόμησης για υψηλό θόρυβο

Συνάρτηση	Fan	LPC	HPC	HPT	LPT	Υγιής	Συνολικά
<code>trainlm</code>	93.6	89.0	89.7	96.4	95.7	88.7	92.2
<code>trainbr</code>	94.1	89.2	89.2	95.4	94.9	90.8	92.3
<code>trainbfg</code>	94.1	91.0	90.0	96.2	95.7	86.2	92.2
<code>trainrp</code>	94.1	90.3	90.8	95.1	95.9	86.9	92.2
<code>trainscg</code>	94.1	90.3	91.3	96.9	95.1	87.7	92.6
<code>traincgb</code>	94.9	89.7	91.3	96.7	95.4	87.2	92.5
<code>traincgf</code>	94.6	90.3	91.3	96.9	95.1	87.2	92.6
<code>traincgp</code>	94.1	89.7	91.3	96.9	95.4	87.7	92.5
<code>trainoss</code>	94.9	89.7	90.8	95.9	95.9	86.4	92.3
<code>traingdx</code>	95.1	92.1	90.8	96.4	95.7	84.6	92.4
<code>traingdm</code>	99.2	0.0	0.0	0.0	27.8	0.0	21.2
<code>traingd</code>	99.2	0.0	0.0	0.0	28.4	0.0	21.3

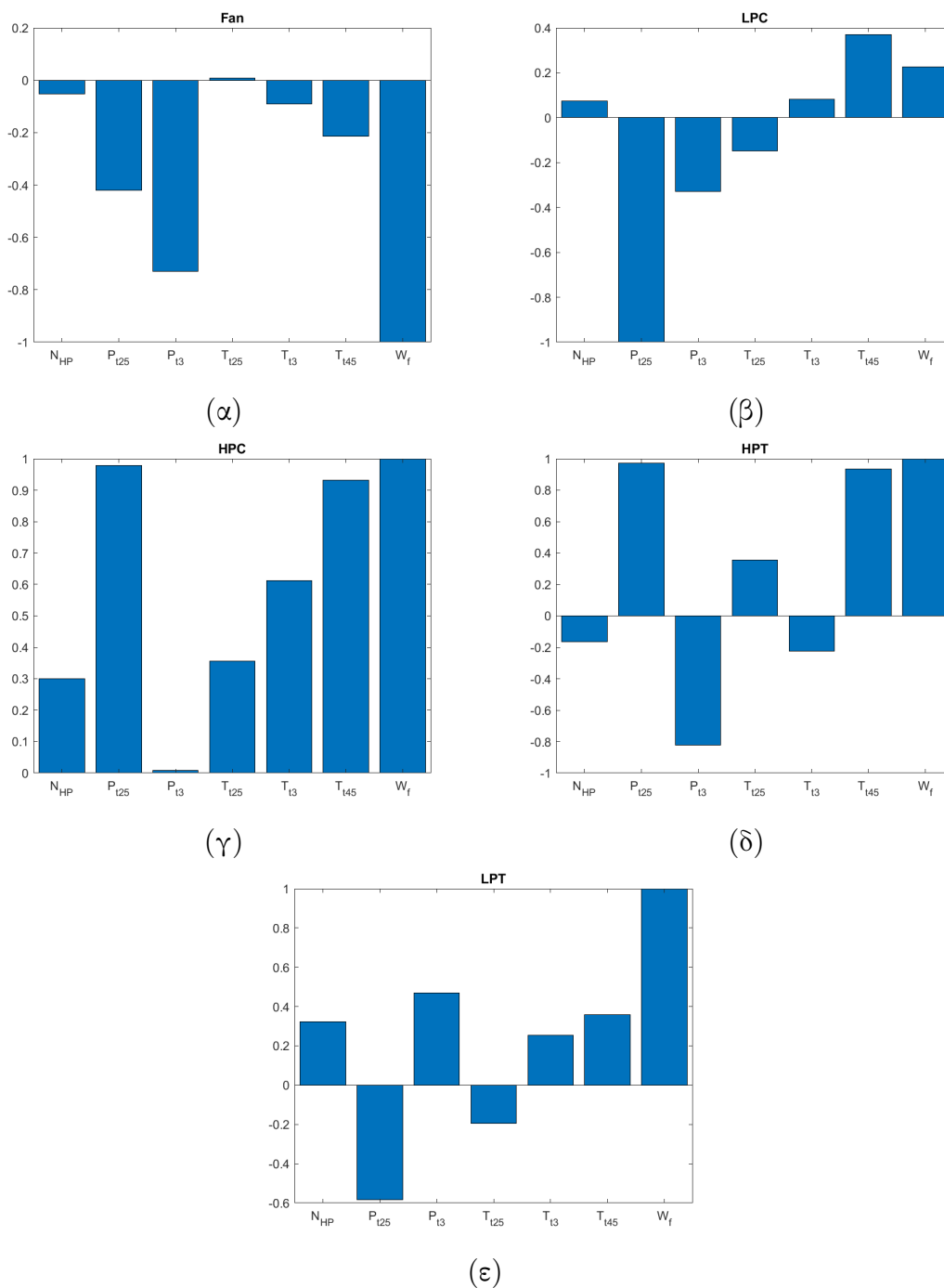


Σχήμα 58: Πίνακες σύγκρισης για εκπαίδευση (α) με και (β) χωρίς θόρυβο

Στο σημείο αυτό αξίζει να εξηγηθεί πώς ερμηνεύονται τα αποτελέσματα που παρουσιάζονται σε έναν οποιοδήποτε πίνακα σύγκρισης. Αρχικά, ο οριζόντιος άξονας αντιπροσωπεύει την κλάση στην οποία το Νευρωνικό Δίκτυο ταξινόμησε ένα συγκεκριμένο πρότυπο και ο κάθετος αντιπροσωπεύει την πραγματική κλάση που αυτό ανήκει. Οπότε, οι αριθμοί στην κύρια διαγώνιο του πίνακα είναι τα πρότυπα που ταξινομήθηκαν σωστά από το ΝΔ. Επιπλέον, σε κάθε γραμμή του πίνακα σύγκρισης οι αριθμοί εκτός της κύριας διαγωνίου δείχνουν πόσα πρότυπα που ανήκουν σε μία συγκεκριμένη κλάση ταξινομήθηκαν στις υπόλοιπες. Για παράδειγμα, στο **Σχήμα 58α** για την κλάση της υγιούς κατάστασης του κινητήρα: 1 πρότυπο ταξινομήθηκε λανθασμένα στην κλάση Fan, 6 στην HPC, 3 στην HPT, 10 στην LPC και 1 στην LPT. Ακόμη, τα ποσοστά δεξιά του πίνακα είναι τα ποσοστά επιτυχούς και ανεπιτυχούς ταξινόμησης αντίστοιχα για κάθε κλάση. Τέλος, με παρόμοιο τρόπο μπορούν να ερμηνευθούν τα αποτελέσματα ενός πίνακα σύγκρισης και τα αντίστοιχα ποσοστά ανά στήλη.

Τα υψηλά ποσοστά επιτυχούς ταξινόμησης που παρουσιάζονται στο **Σχήμα 58** μπορούν να ερμηνευθούν μέσω των μέσων κανονικοποιημένων υπογραφών (βλ. **Σχήμα 59**), όπου τα πρότυπα εκπαίδευσης για κάθε κλάση κανονικοποιούνται μέσω της υψηλότερης τιμής των μεταβολών των μεταβλητών της υπογραφής και οι μέσες τιμές ανά μεταβλητή σχεδιάζονται στο αντίστοιχο γράφημα. Είναι εμφανές ότι οι υπογραφές είναι διακριτές μεταξύ τους, γεγονός που διευκολύνει ένα Νευρωνικό Δίκτυο να τα αναγνωρίσει. Ακόμη, επειδή η μέση κανονικοποιημένη υπογραφή της υγιούς κατάστασης του κινητήρα είναι 0 για κάθε μεταβλητή και λόγω της τυχαιότητας του θορύβου που προστίθεται, είναι λογικό στην περίπτωση της εκπαίδευσης χωρίς θόρυβο να παρατηρείται σημαντική μείωση στο ποσοστό επιτυχούς ταξινόμησης για

την κλάση αυτή όταν τα πρότυπα δοκιμής περιέχουν θόρυβο.



Σχήμα 59: Υπογραφή βλάβης (α)Fan, (β)LPC, (γ)HPC, (δ)HPT, και (ε)LPT

## 4.6 Εκτίμηση παραμέτρων υγείας

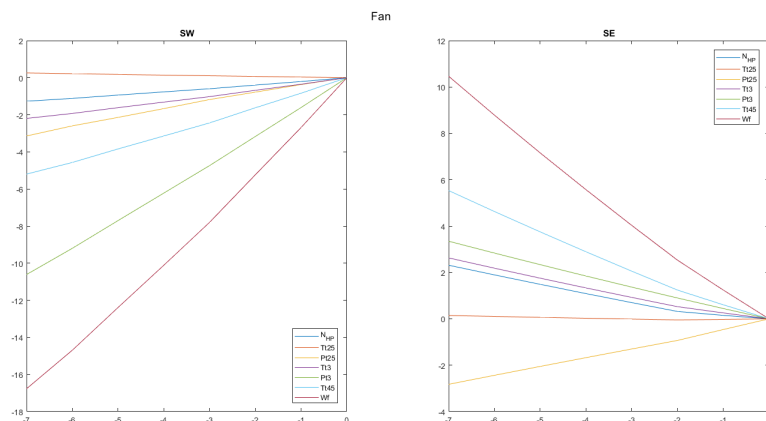
Η εκτίμηση των παραμέτρων υγείας πραγματοποιήθηκε χρησιμοποιώντας το μοντέλο Simulink που αναπτύχθηκε σε λειτουργία διάγνωσης μέσω του προγράμματος

NNDiagnosis (βλ. **Παράρτημα II**). Πιο συγκεκριμένα, αφότου ένα πρότυπο ταξινομηθεί σε μία κλάση μέσω ενός από τα προαναφερθέντα ΝΔ, το μοντέλο εκτιμά τις παραμέτρους υγείας της συνιστώσας στην οποία το ΝΔ προέβλεψε ότι υπάρχει η συγκεκριμένη βλάβη.

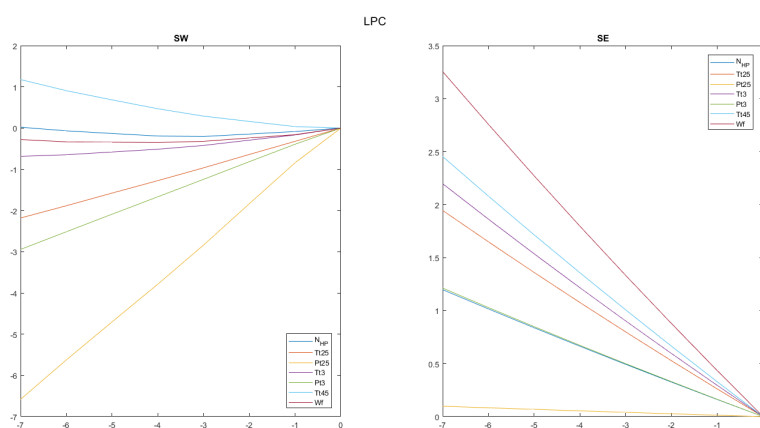
Οι αντίστοιχες μετρήσεις που χρησιμοποιήθηκαν για την εκτίμηση κάθε ζεύγους παραμέτρων υγείας παρατίθενται στον **Πίνακα 10**, οι οποίες προέκυψαν μέσω των μεταβολών ( $\Delta Y$ ) και της υγιούς κατάστασης του κινητήρα. Η επιλογή των κατάλληλων μετρήσεων έγινε λαμβάνοντας υπόψη τον αριθμό κατάστασης κάθε πιθανού συνδυασμού και τα αποτελέσματα μίας ανάλυσης ευαισθησίας κάθε παραμέτρου υγείας έναντι όλων των μεταβλητών. Πιο συγκεκριμένα, όσον αφορά την ανάλυση ευαισθησίας, μέσω προσομοιώσεων, καταγράφηκε η ποσοστιαία μεταβολή όλων των μεταβλητών για SE από -1% έως -7% και για SW από -1% έως -7% (συμπίεστές) ή 1% έως 7% (στρόβιλοι) και τα αποτελέσματα παρουσιάζονται στα **Σχήματα 60** έως **64**. Για την επιλογή των μετρήσεων πρώτο κριτήριο ήταν ο χαμηλός αριθμός κατάστασης και σε δεύτερο στάδιο έγινε έλεγχος της ευαισθησίας των δύο μεταβλητών. Αν η ευαισθησία μίας από τις δύο ήταν μεγάλη ή πολύ μικρή, τότε ο εκάστοτε συνδυασμός απορρίφθηκε και επιλέχθηκε ο επόμενος κ.ο.κ.

Πίνακας 10: Μετρήσεις για την εκτίμηση των παραμέτρων υγείας κάθε συνιστώσας

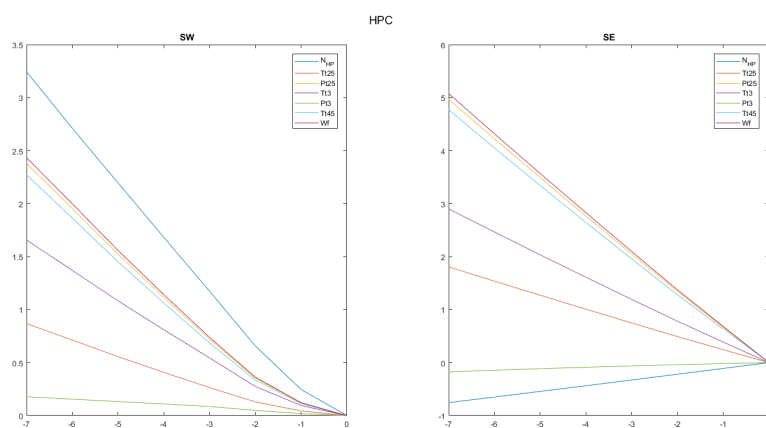
Κλάση	Μεταβλητές
Fan	Pt25 Tt3
LPC	Tt25 Pt3
HPC	N_HP Tt25
HPT	Pt3 Wf
LPT	N_HP Tt45
Υγιής	N/A



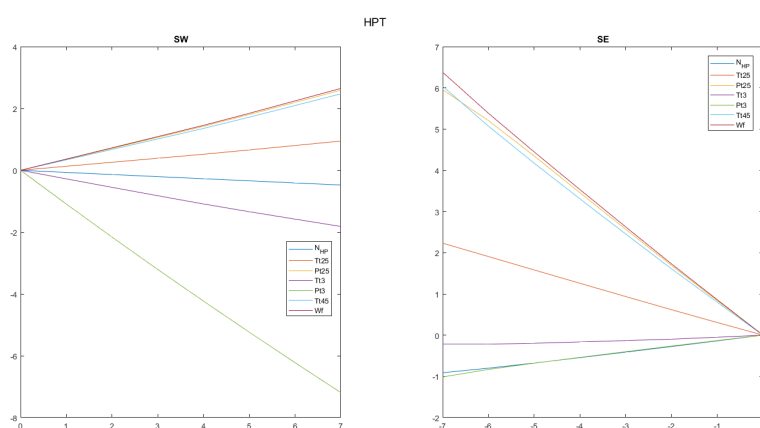
Σχήμα 60: Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον Fan



Σχήμα 61: Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον LPC

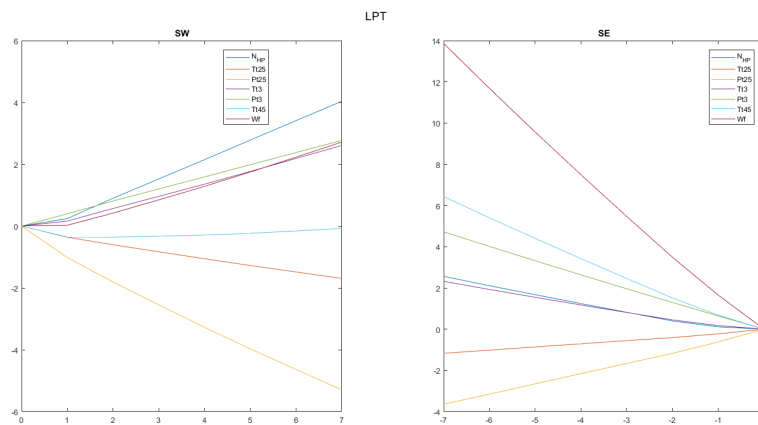


Σχήμα 62: Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον HPC



Σχήμα 63: Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον HPT





Σχήμα 64: Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον LPT

## 4.7 Διερεύνηση

Αρχικά, αξίζει να σημειωθεί ότι όλα τα αποτελέσματα που παρουσιάζονται στην παρούσα εργασία προέκυψαν χρησιμοποιώντας την ίδια αρχικοποίηση της γεννήτριας τυχαίων αριθμών (rng seed) με σκοπό τόσο την δίκαιη σύγκρισή τους, όσο και την δυνατότητα αναπαραγωγής τους, ενώ όλα τα ΝΔ έχουν εκπαιδευτεί με πρότυπα που περιέχουν θόρυβο. Συγκεκριμένα πραγματοποιήθηκαν τα ακόλουθα:

1. Σύγκριση των διαφόρων τύπων Νευρωνικών Δικτύων μεταξύ τους
2. Έλεγχος απόδοσης σε διαφορετικά σημεία λειτουργίας
3. Έλεγχος απόδοσης όταν υπάρχει γενική χειροτέρευση (deterioration)
4. Έλεγχος απόδοσης για συνδυασμό γενικής χειροτέρευσης με βλάβη μίας συνιστώσας
5. Έλεγχος απόδοσης για ταυτόχρονη βλάβη δύο συνιστωσών

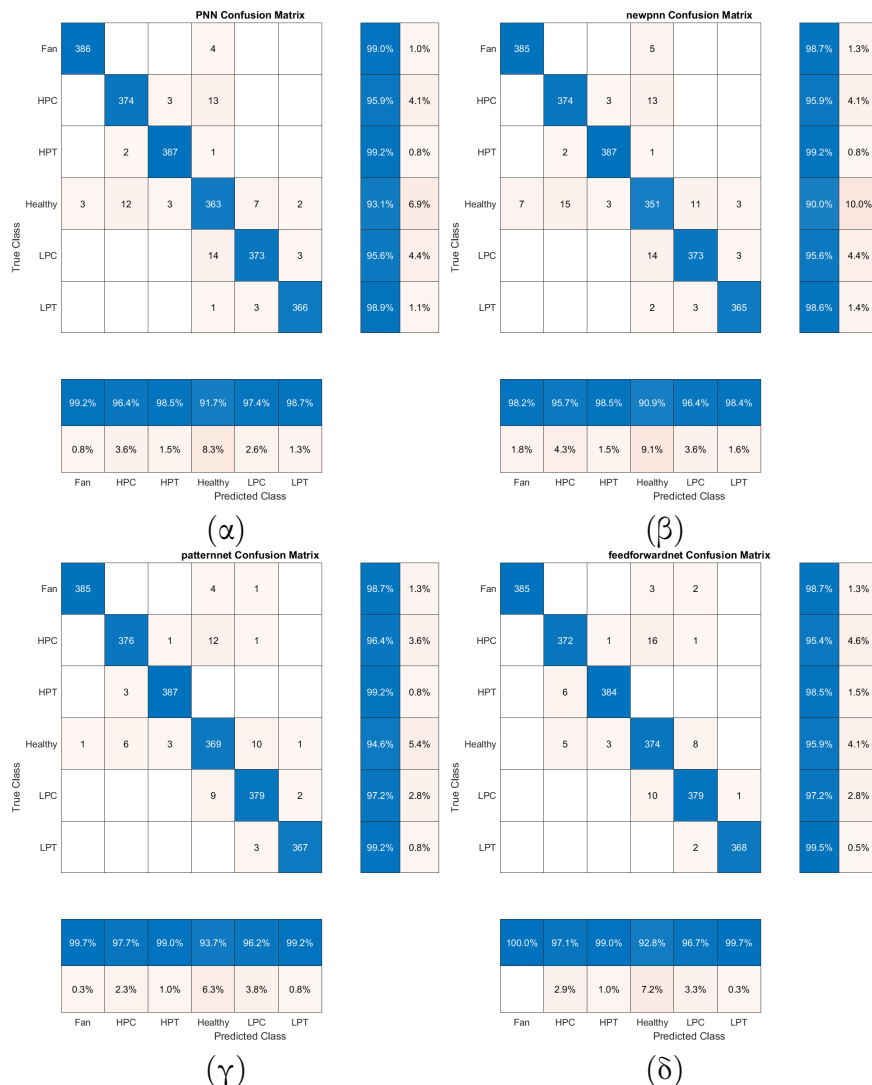
### 4.7.1 Σύγκριση ΝΔ

Όλα τα ΝΔ που αναπτύχθηκαν χρησιμοποιήθηκαν για την ταξινόμηση των προτύπων δοκιμής σε κάθε ένα από τα οποία εισήχθηκε θόρυβος 10 φορές. Οι πίνακες σύγκρισης που προέκυψαν παρουσιάζονται στο **Σχήμα 65**. Το συνολικό ποσοστό επιτυχούς ταξινόμησης κάθε μεθόδου ήταν:

- PNN: 96.7%

- newpnn: 96.3%
- patternnet: 97.6%
- feedforwardnet: 97.5%

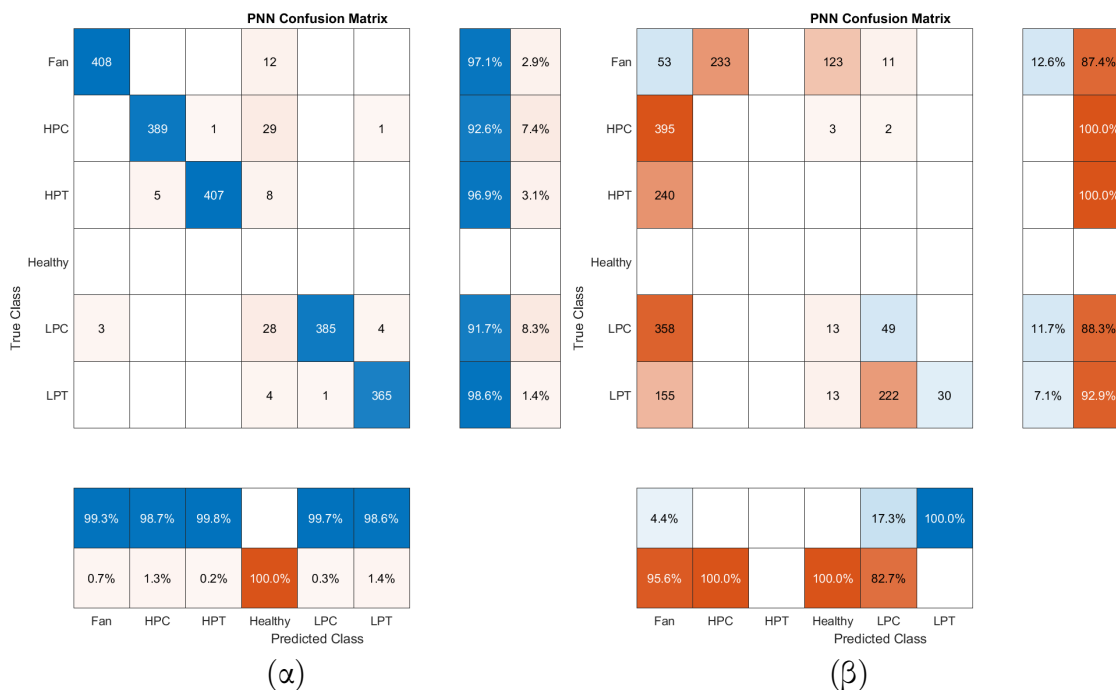
Εξετάζοντας τα αποτελέσματα αυτά, αρχικά, παρατηρήθηκε ότι το ποσοστό επιτυχίας για όλες τις μεθόδους ήταν μεγαλύτερο του 95%, γεγονός ιδιαίτερα ικανοποιητικό. Ακόμη, το ποσοστό επιτυχούς ταξινόμησης ήταν παρόμοιο μεταξύ των μεθόδων, με τα PNN να είναι ελάχιστα χειρότερα (περίπου 1%). Τέλος, η κλάση με το χαμηλότερο ποσοστό επιτυχίας είναι εκείνη της υγιούς κατάστασης του κινητήρα. Αυτό είναι λογικό καθώς στις περιπτώσεις ενός μικρού μεγέθους βλάβης μια συνιστώσας, ο τυχαίος θόρυβος που προστίθεται οδηγεί κάποια πρότυπα πιο κοντά στην υγιή κατάσταση ή αντίστοιχα πιο μακριά από αυτή με αποτέλεσμα τη σύγχυση τους από τα ΝΔ.



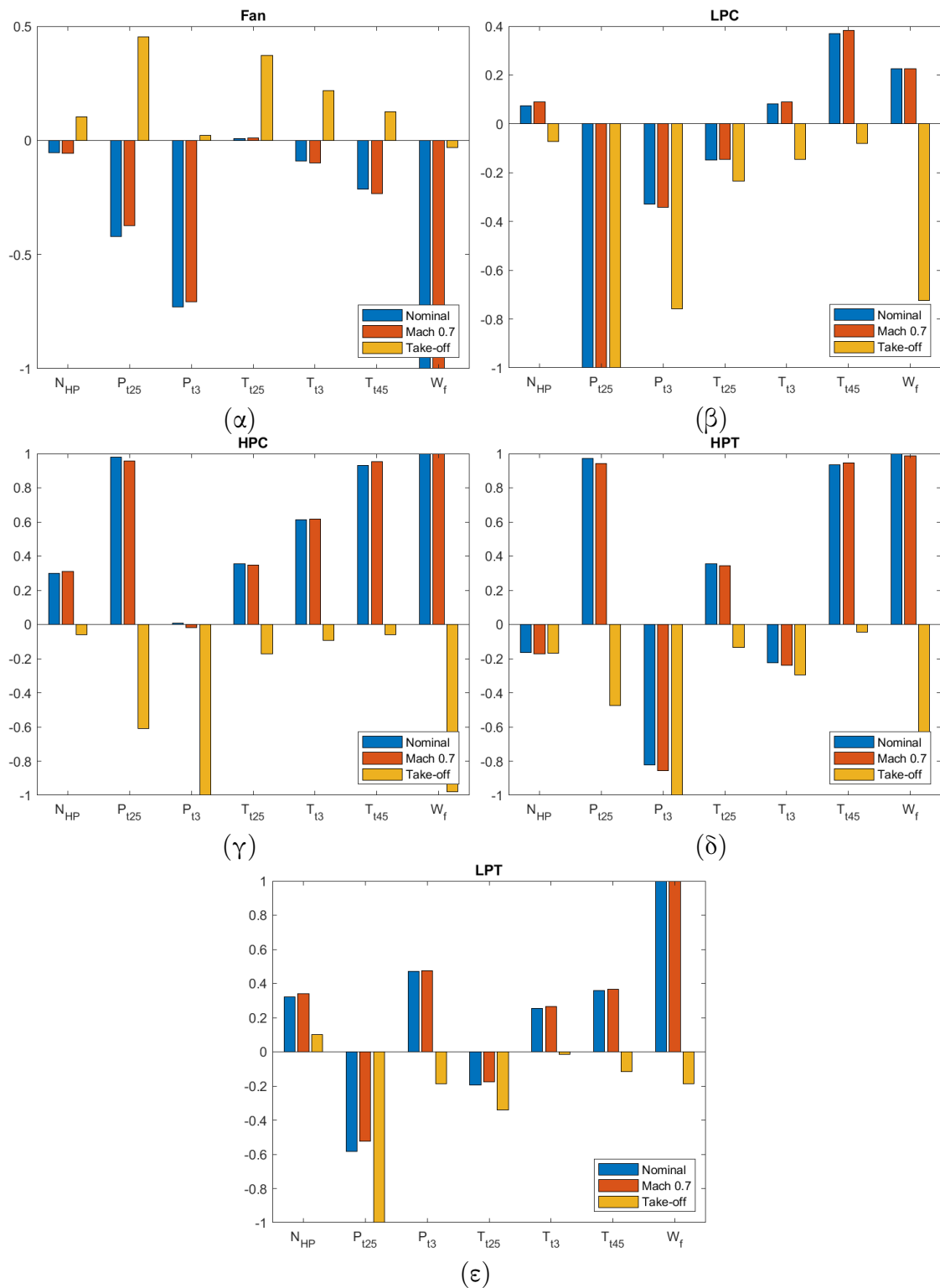
Σχήμα 65: Πίνακες σύγχυσης για (α) PNN, (β) newpnn, (γ) patternnet και (δ) feedforwardnet

#### 4.7.2 Διαφορετικά σημεία λειτουργίας

Αρχικά, μέσω του προγράμματος getTestingPatsOP (βλ. **Παράρτημα II**), δημιουργήθηκαν πρότυπα δοκιμής για δύο επιπλέον σημεία λειτουργίας. Το πρώτο στο ίδιο ύψος αλλά για Mach ίσο με 0.7 και το δεύτερο στις συνθήκες απογείωσης [1]. Για λόγους συντομίας τα αποτελέσματα παρουσιάζονται μόνο για την περίπτωση του PNN. Ακόμη, ως πρότυπα εκπαίδευσης χρησιμοποιήθηκαν εκείνα στο αρχικό σημείο λειτουργίας. Τα αποτελέσματα παρουσιάζονται στο **Σχήμα 66**. Όπως ήταν αναμενόμενο, στο πρώτο σημείο, επειδή η μέση κανονικοποιημένη υπογραφή των προτύπων αυτών είναι κοντά σε εκείνη των προτύπων στα οποία έχει εκπαιδευτεί το ΝΔ, τα αποτελέσματα είναι ικανοποιητικά, ενώ, στο δεύτερο, επειδή τα αντίστοιχα πρότυπα είναι πολύ διαφορετικά, τα αποτελέσματα είναι απολύτως μη ικανοποιητικά (βλ. **Σχήμα 67**).



Σχήμα 66: Πίνακες σύγχυσης για (α) Mach 0.7 και (β) συνθήκες απογείωσης



Σχήμα 67: Υπογραφή βλάβης (α) Fan, (β) LPC, (γ) HPC, (δ) HPT, και (ε) LPT για το σημείο εκπαίδευσης (μπλε), Mach 0.7 (πορτοκαλί) και συνθήκες απογείωσης (κίτρινο)

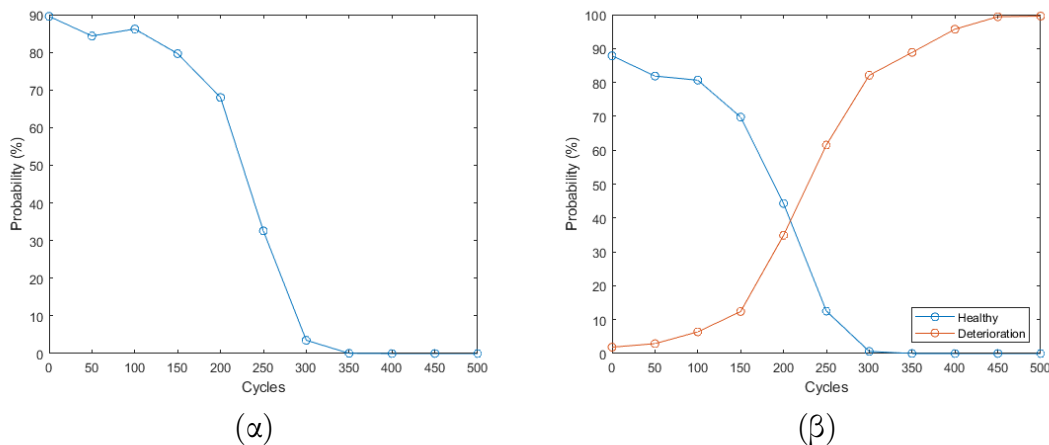
#### 4.7.3 Γενική χειροτέρευση (Deterioration)

Αρχικά, τα δεδομένα του ρυθμού γενικής χειροτέρευσης για κάθε συνιστώσα του κινητήρα JT9D συνοψίζονται στον **Πίνακα 11**, όπου κάθε ζευγάρι παραμέτρων υγείας αναφέρεται στους αντίστοιχους κύκλους. Στην παρούσα εργασία ο ρυθμός γενικής χειροτέρευσης κάθε συνιστώσας θεωρήθηκε σταθερός, δηλαδή η σχέση των παραμέτρων υγείας με τους κύκλους του κινητήρα θεωρήθηκε γραμμική με αρχή το 0. Τα αντίστοιχα πρότυπα δοκιμής δημιουργήθηκαν για 0 έως 6000 κύκλους ανά 50 μέσω του προγράμματος Deterioration (βλ. **Παράρτημα II**).

Πίνακας 11: Δεδομένα γενικής χειροτέρευσης [11]

Συνιστώσα	SW [%]	SE [%]	Κύκλοι
Fan	-4	-3	6000
LPC	-3	-2.2	5000
HPC	-10	-6	5000
HPT	1.2	-2	2000
LPT	0.15	-0.25	1000

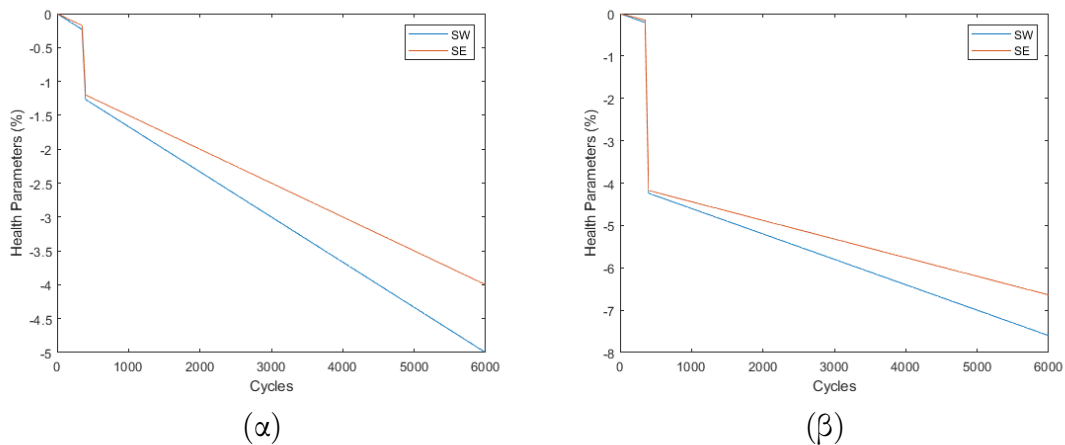
Στην παρούσα εργασία εξετάστηκε η συμπεριφορά του PNN με και χωρίς εκπαίδευση με τα πρότυπα της γενικής χειροτέρευσης. Στην περίπτωση της εκπαίδευσης με τα πρότυπα της γενικής χειροτέρευσης προστέθηκε μία επιπλέον κλάση, ενώ και στις δύο τα πρότυπα δοκιμής δεν περιείχαν θόρυβο για λόγους ευκρίνειας των αποτελεσμάτων. Όπως φαίνεται στο **Σχήμα 68**, μετά τους 200 περίπου κύκλους η επίδραση της γενικής χειροτέρευσης κυριαρχεί.



Σχήμα 68: Πιθανότητα συναρτήσεως των κύκλων για την περίπτωση (α) χωρίς και (β) με τα πρότυπα γενικής χειροτέρευσης

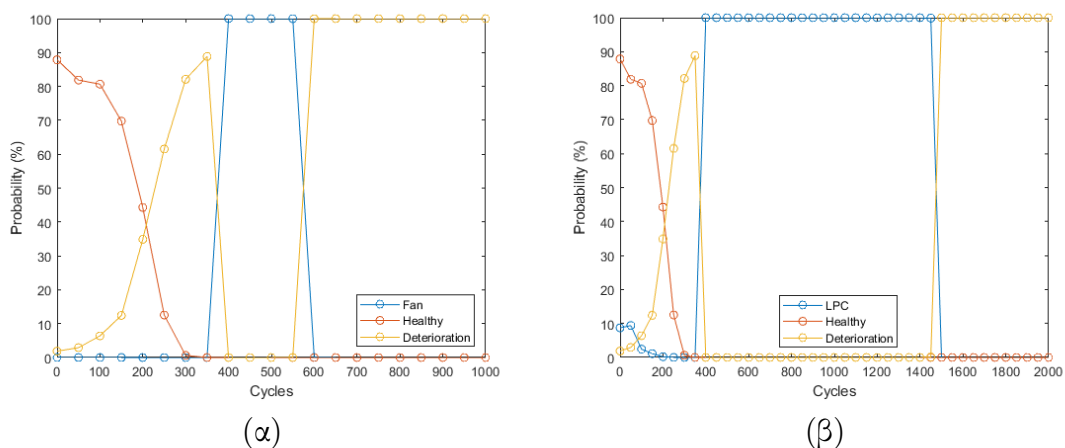
#### 4.7.4 Συνδυασμός γενικής χειροτέρευσης με βλάβη συνιστώσας

Στην παρούσα εργασία εξετάστηκαν δύο περιπτώσεις βλάβης: η πρώτη αφορά μία μικρού βαθμού βλάβη του Fan ( $SW = -1\%$ ,  $SE = -1\%$ ) και η δεύτερη μία μεγαλύτερου βαθμού βλάβη του LPC ( $SW = -4\%$ ,  $SE = -4\%$ ). Αμφότερες οι βλάβες εμφανίζονται στους 400 κύκλους (βλ. **Σχήμα 69**).



Σχήμα 69: Παράμετροι υγείας συναρτήσει των κύκλων για βλάβη (α) Fan και (β) LPC

Τα πρότυπα δοκιμής και για τις δύο περιπτώσεις δημιουργήθηκαν μέσω του προγράμματος combinedFault (βλ. **Παράρτημα II**), όπου και πάλι τα πρότυπα δοκιμής δεν περιείχαν θόρυβο για λόγους ευκρίνειας των αποτελεσμάτων. Όπως φαίνεται στο **Σχήμα 70**, πριν την εμφάνιση της βλάβης η εικόνα είναι παρόμοια με την προηγούμενη υποενότητα. Όταν εμφανίζεται η βλάβη, αυτή κυριαρχεί για ένα διάστημα ανάλογα με το μέγεθος της βλάβης και στη συνέχεια κυριαρχεί η γενική χειροτέρευση, καθώς η μείωση στη απόδοση που οφείλεται σε αυτή είναι μεγαλύτερη, με τις μεταβολές ανάμεσα στις καταστάσεις αυτές να είναι απότομες.



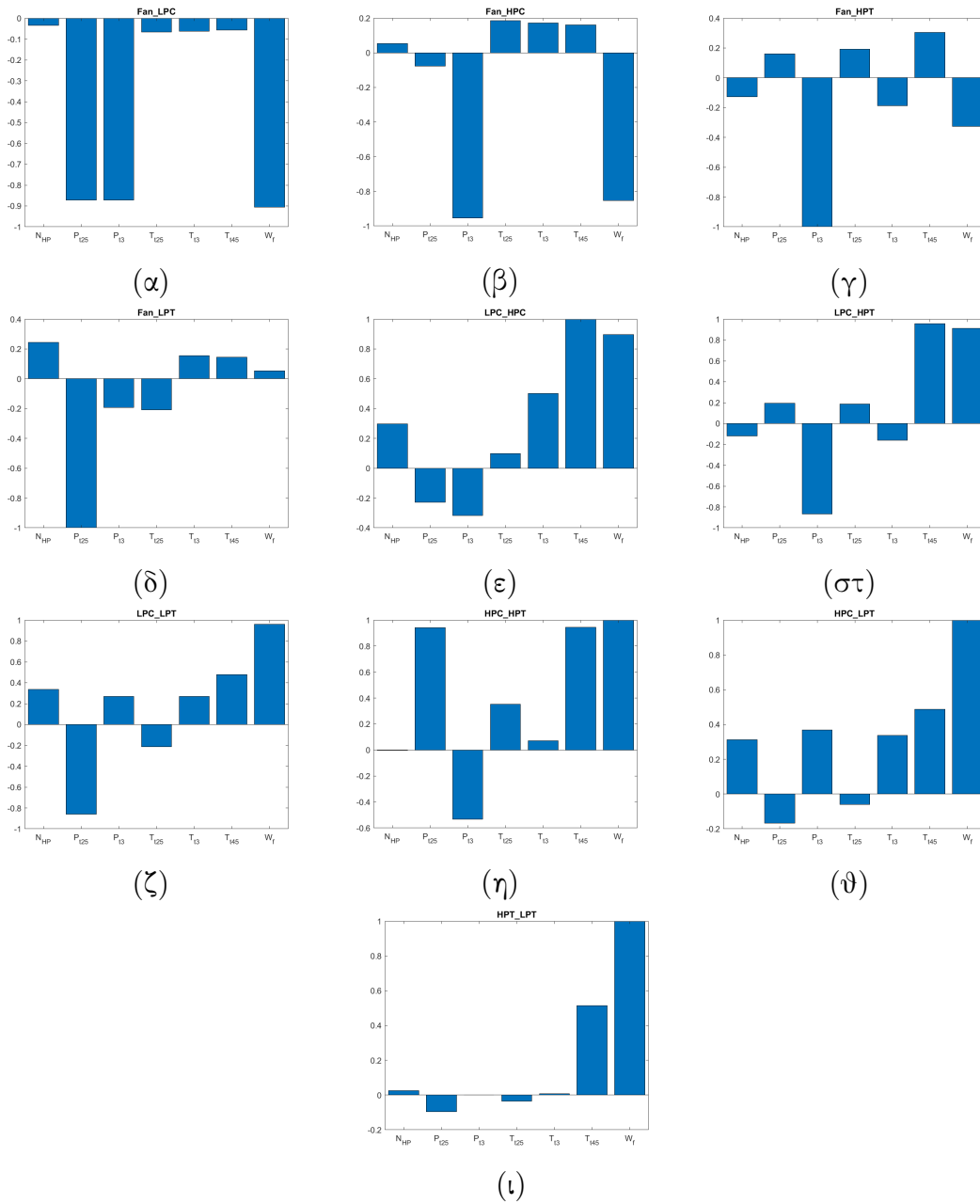
Σχήμα 70: Πιθανότητα συναρτήσει των κύκλων για βλάβη (α) Fan και (β) LPC

#### 4.7.5 Βλάβη 2 συνιστωσών

Αρχικά, δημιουργήθηκαν πρότυπα εκπαίδευσης και δοκιμής για όλους τους συνδυασμούς βλάβης 2 συνιστωσών (10 στον αριθμό) μέσω του προγράμματος combinedFaultComp (βλ. **Παράρτημα II**). Στη συνέχεια, προστέθηκαν 10 επιπλέον κλάσεις, μία για κάθε συνδυασμό. Για εκπαίδευση και δοκιμή χρησιμοποιήθηκαν τόσο τα πρότυπα για βλάβη δύο συνιστωσών, αλλά και μίας μεμονωμένης συνιστώσας, ενώ σε όλα προστέθηκε θόρυβος. Όπως φαίνεται στο **Σχήμα 71**, τα ποσοστά επιτυχούς ταξινόμησης είναι ικανοποιητικά.

PNN Confusion Matrix																			
True Class	Fan	354	4		28						4						90.8%	9.2%	
	Fan_HPC	5	372	4							8				1		95.4%	4.6%	
	Fan_HPT		3	384						1					2		98.5%	1.5%	
	Fan_LPC	27	1		358	1					1	2					91.8%	8.2%	
	Fan_LPT				2	374						9	1			4	95.9%	4.1%	
	HPC						371	2			1	15		1			95.1%	4.9%	
	HPC_HPT						2	373		14					1		95.6%	4.4%	
	HPC_LPT						1		328		2			1		4	4	96.5%	3.5%
	HPT			1			1	15		372					1			95.4%	4.6%
	HPT_LPT						1		4		380			3	1		1	97.4%	2.6%
	Healthy	4	6		1		6			2	6	346	9	7		1	2	88.7%	11.3%
	LPC	3			2	9			1			16	353	3		1	2	90.5%	9.5%
	LPC_HPC		2		1		6		2		4	3	3	368	1			94.4%	5.6%
	LPC_HPT							1		5		1		1	382			97.9%	2.1%
	LPC_LPT					4			1				3	1		260	11	92.9%	7.1%
	LPT					1			4							13	352	95.1%	4.9%

Σχήμα 71: Πίνακας σύγχυσης για βλάβη μίας ή δύο συνιστωσών



Σχήμα 72: Υπογραφή συνδυασμού βλάβης (α) Fan + LPC, (β) Fan + HPC, (γ) Fan + HPT, (δ) Fan + LPT, (ε) LPC + HPC, (στ) LPC + HPT, (ζ) LPC + LPT, (η) HPC + HPT, (θ) HPC + LPT και (ι) HPT + LPT

Όπως φαίνεται στο **Σχήμα 72**, οι υπογραφές των συνδυασμών βλαβών δύο συνιστωσών είναι αρκετά διακριτές μεταξύ τους, γεγονός που αντανακλάται στα υψηλά ποσοστά επιτυχούς ταξινόμησης του ΝΔ. Ο μόνος συνδυασμός που φαίνεται πως συγχέεται περισσότερο από το ΝΔ είναι ο συνδυασμός Fan και Fan+LPC, καθώς οι υπογραφές τους είναι περισσότερο όμοιες από άλλους συνδυασμούς.



## 5 Ανακεφαλαίωση/Συμπεράσματα

Κύριοι στόχοι της παρούσας διπλωματικής εργασίας ήταν η ανάπτυξη μιας εφαρμογής στο περιβάλλον Matlab/Simulink για την πραγματοποίηση προσομοίωσης και διάγνωσης ενός αεριοστροβίλου, καθώς και η διερεύνηση των Νευρωνικών Δικτύων ως μέθοδο διάγνωσης.

Προς αυτή τη κατεύθυνση, αρχικά, αναπτύχθηκε ένα μοντέλο Simulink, το οποίο βασίζεται στον κινητήρα JT9D της εταιρείας Pratt & Whitney, χρησιμοποιώντας την βιβλιοθήκη συνιστωσών αεριοστροβίλων T-MATS της NASA. Στις σημαντικότερες δυνατότητές του συγκαταλέγονται η επιλογή του σημείου λειτουργίας, ο έλεγχος των παραμέτρων υγείας και η δημιουργία της υπογραφής του κινητήρα, οι οποίες είναι απαραίτητες για την πραγματοποίηση προσομοιώσεων και διάγνωσης σε όλες τις περιοχές λειτουργίας του κινητήρα. Στη συνέχεια, δημιουργήθηκε η διεπαφή χρήστη (user interface) της εφαρμογής μέσω του App Designer της Matlab, η οποία αποτελείται από τα ακόλουθα βασικά στοιχεία:

- Πλαίσιο για την επιλογή του σημείου λειτουργίας
- Πλαίσιο για τον έλεγχο των παραμέτρων υγείας του κινητήρα
- Πλαίσιο όπου παρουσιάζονται οι βασικές παράμετροι απόδοσης του κινητήρα
- Πλαίσιο ελέγχου της εφαρμογής
- Πλαίσιο χαρτών συνιστωσών και αποτελεσμάτων
- Πλαίσιο σχηματικού του κινητήρα

Όσον αφορά τη διάγνωση μέσω Νευρωνικών Δικτύων ακολουθήθηκε η εξής διαδικασία: Αρχικά δημιουργήθηκαν κατάλληλα πρότυπα εκπαίδευσης και δοκιμής. Στη συνέχεια, τα ΝΔ εκπαιδεύτηκαν και χρησιμοποιήθηκαν για την ταξινόμηση των προτύπων δοκιμής. Τέλος, εφόσον η απόδοσή τους ήταν ικανοποιητική, πραγματοποιήθηκε εκτίμηση των παραμέτρων υγείας του κινητήρα για κάθε βλάβη.

Κατά τη διερεύνηση των ΝΔ, εξετάστηκε αρχικά η απόδοσή τους σε διαφορετικά σημεία λειτουργίας από εκείνο στο οποίο είχαν εκπαιδευτεί, από όπου προέκυψε ότι τα ποσοστά επιτυχίας είναι ικανοποιητικά εφόσον το σημείο λειτουργίας κατά τη δοκιμή είναι κοντά σε εκείνο κατά την εκπαίδευση. Ακολούθως, ελέγχθηκε η συμπεριφορά τους σε πρότυπα γενικής χειροτέρευσης και εντοπίστηκε ότι η κατάσταση αυτή αναγνωρίζεται από το ΝΔ από τους 200 κύκλους λειτουργίας περίπου και μετά. Στη συνέχεια, εξετάστηκε ο συνδυασμός της γενικής χειροτέρευσης με βλάβη μίας συνιστώσας, όπου παρατηρήθηκε πως η βλάβη εντοπίζεται για ένα διάστημα κύκλων από τα ΝΔ προτού κυριαρχήσει ξανά η γενική χειροτέρευση. Τέλος, διερευνήθηκε η απόδοση των ΝΔ κατά την εκπαίδευση και δοκιμή με πρότυπα ταυτόχρονης βλάβης δύο συνιστωσών, όπου τα ποσοστά επιτυχούς ταξινόμησης ήταν υψηλά. Άρα,

συνολικά τα Νευρωνικά Δίκτυα βρέθηκαν ότι αποτελούν μια αρκετά ικανοποιητική μέθοδο διάγνωσης αεριοστροβίλων.

Σε επόμενο στάδιο προτείνεται η δημιουργία μοντέλων Simulink επιπλέον τύπων κινητήρων και η προσαρμογή της εφαρμογής σε αυτούς ώστε να είναι ένα εργαλείο προσομοίωσης και διάγνωσης αεριοστροβίλων με πιο γενική εφαρμογή. Επιπλέον, σε μελλοντική εργασία θα μπορούσε να γίνει χρήση πραγματικών μετρήσεων από αεροπορικούς κινητήρες για την εκπαίδευση και την δοκιμή Νευρωνικών Δικτύων με σκοπό την εξέταση της βιωσιμότητάς της μεθόδου διάγνωσης αυτής εκτός του ακαδημαϊκού περιβάλλοντος. Το γεγονός αυτό είναι πιθανό να οδηγήσει στην ανάγχη βελτιστοποίησης των ΝΔ, όπως για παράδειγμα της αρχιτεκτονικής τους, της μεθόδου εκπαίδευσης τους κλπ. Ακόμη, θα μπορούσε να διερευνηθεί μία επέκταση των κλάσεων στις οποίες ταξινομούνται τα πρότυπα ώστε ένα ΝΔ να μπορεί να αναγνωρίζει συγκεκριμένες βλάβες και όχι μόνο τη συνιστώσα στην οποία εντοπίζονται, καθώς και την αναγνώριση σφαλμάτων σε αισθητήρες. Τέλος, σε μελλοντική έρευνα θα μπορούσαν να εξεταστούν περαιτέρω μέθοδοι διάγνωσης, όπως η Γεωμετρική Αναγνώριση Προτύπων και η χρήση Νευρωνικών Δικτύων Βαθιάς Μάθησης (Deep Learning Neural Networks) για την πραγματοποίηση διάγνωσης μέσω εικόνας.

## Κατάλογος σχημάτων

1	Constant και Terminator blocks . . . . .	11
2	Sum, Subtract και Product blocks . . . . .	11
3	From και Goto blocks . . . . .	12
4	Mux, Demux και BusSelector blocks . . . . .	12
5	Subsystem, Inport και Outport blocks . . . . .	12
6	Function block . . . . .	13
7	Switch και Selector blocks . . . . .	13
8	Display και ToWorkspace blocks . . . . .	13
9	Ambient block . . . . .	14
10	SS NR Solver w Jacobian Calc block . . . . .	14
11	Inlet block . . . . .	15
12	Compressor block . . . . .	15
13	Splitter block . . . . .	15
14	Burner block . . . . .	16
15	Turbine NPSS block . . . . .	16
16	Duct block . . . . .	16
17	Nozzle block . . . . .	16
18	Shaft block . . . . .	17
19	Διάγραμμα επικοινωνίας ανάμεσα στην εφαρμογή και στο μοντέλο Simulink . . . . .	18
20	Σχηματικό διάγραμμα του επιλύτη στο αρχικό μοντέλο Simulink . . . . .	19
21	Σχηματικό διάγραμμα του υπο-μοντέλου του κινητήρα στο αρχικό μοντέλο Simulink . . . . .	19
22	Σχηματικό διάγραμμα του εμπρός μέρους του κινητήρα στο αρχικό μοντέλο Simulink . . . . .	20
23	Σχηματικό διάγραμμα του πυρήνα του κινητήρα στο αρχικό μοντέλο Simulink . . . . .	20
24	Σχηματικό διάγραμμα του πίσω μέρους του κινητήρα στο αρχικό μοντέλο Simulink . . . . .	20
25	Σχηματικό διάγραμμα του άξονα χαμηλής πίεσης του κινητήρα στο αρχικό μοντέλο Simulink . . . . .	21
26	Σχηματικό διάγραμμα του άξονα υψηλής πίεσης του κινητήρα στο αρχικό μοντέλο Simulink . . . . .	21
27	Σχηματικό διάγραμμα της εφαρμογής των μεταβλητών σημείου λειτουργίας στο Simulink . . . . .	22
28	Παράδειγμα λήψης μεταβλητών υπογραφής από μία θέση του κινητήρα . . . . .	23
29	Σχηματικό διάγραμμα του υπολογισμού της καθαρής ώσης στο Simulink . . . . .	24
30	Σχηματικό διάγραμμα της εφαρμογής της μεταβλητής ελέγχου στο Simulink . . . . .	25
31	Υλοποίηση επιλογής σφαλμάτων στο Simulink . . . . .	26
32	Υλοποίηση των σφαλμάτων των μετρήσεων στο Simulink . . . . .	26
33	Σχηματικό διάγραμμα της υλοποίησης της επίλυσης του συστήματος στο Simulink . . . . .	28

34	Σχηματικό διάγραμμα της υλοποίησης των παραμέτρων υγείας στο Simulink . . . . .	31
35	Πλαίσιο εισόδων . . . . .	34
36	Πλαίσιο παραμέτρων υγείας . . . . .	35
37	Πλαίσιο παραμέτρων απόδοσης . . . . .	36
38	Καρτέλα χάρτη Fan . . . . .	37
39	Καρτέλα πίνακα αποτελεσμάτων . . . . .	38
40	Καρτέλα κονσόλας . . . . .	39
41	Καρτέλα γραφήματος αποτελεσμάτων . . . . .	39
42	Πλαίσιο σχηματικού . . . . .	40
43	Πλαίσιο ελέγχου . . . . .	40
44	Συνολική διεπαφή χρήστη της εφαρμογής . . . . .	44
45	Επιλογή εισόδων για προσομοίωση ενός σημείου . . . . .	44
46	Επιλογή παραμέτρων υγείας για προσομοίωση . . . . .	45
47	Αποτελέσματα προσομοίωσης ενός σημείου στο (α) πλαίσιο των παραμέτρων απόδοσης και (β) στο σχηματικό του κινητήρα . . . . .	45
48	Αποτελέσματα προσομοίωσης ενός σημείου (α) στον χάρτη του Fan και (β) στο διάγραμμα της υπογραφής του κινητήρα . . . . .	46
49	Επιλογή εισόδων για προσομοίωση πολλαπλών σημείων . . . . .	47
50	Αποτελέσματα προσομοίωσης πολλαπλών σημείων στο (α) πλαίσιο των παραμέτρων απόδοσης και (β) στο σχηματικό του κινητήρα . . . . .	47
51	Αποτελέσματα προσομοίωσης πολλαπλών σημείων (α) στον χάρτη του LPT και (β) στο γράφημα των αποτελεσμάτων Tt4 - Mach . . . . .	48
52	Επιλογή παραμέτρων υγείας για πραγματοποίηση διάγνωσης . . . . .	49
53	Επιλογή μεταβλητών υπογραφής για πραγματοποίηση διάγνωσης . . . . .	49
54	Αποτελέσματα παραδείγματος διάγνωσης . . . . .	50
55	Διάγραμμα ροής του αλγορίθμου διάγνωσης . . . . .	55
56	Σχηματική αναπαράσταση του PNN [9] . . . . .	58
57	Ποσοστό επιτυχούς ταξινόμησης συναρτήσει της παραμέτρου εξομάλυνσης για πρότυπα χωρίς θόρυβο (μπλε) και με θόρυβο (πορτοκαλί) για το (α) PNN και (β) newpnn . . . . .	59
58	Πίνακες σύγχυσης για εκπαίδευση (α) με και (β) χωρίς θόρυβο . . . . .	61
59	Υπογραφή βλάβης (α)Fan, (β)LPC, (γ)HPC, (δ)HPT, και (ε)LPT . . . . .	62
60	Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον Fan . . . . .	63
61	Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον LPC . . . . .	64
62	Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον HPC . . . . .	64
63	Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον HPT . . . . .	64
64	Ποσοστιαία μεταβολή των μεταβλητών συναρτήσει των παραμέτρων υγείας για τον LPT . . . . .	65
65	Πίνακες σύγχυσης για (α) PNN, (β) newpnn, (γ) patternnet και (δ) feedforwardnet . . . . .	66
66	Πίνακες σύγχυσης για (α) Mach 0.7 και (β) συνθήκες απογείωσης . . . . .	67

67	Υπογραφή βλάβης (α) Fan, (β) LPC, (γ) HPC, (δ) HPT, και (ε) LPT για το σημείο εκπαίδευσης (μπλε), Mach 0.7 (πορτοκαλί) και συνθήκες απογείωσης (κίτρινο) . . . . .	68
68	Πιθανότητα συναρτήσει των κύκλων για την περίπτωση (α) χωρίς και (β) με τα πρότυπα γενικής χειροτέρευσης . . . . .	69
69	Παράμετροι υγείας συναρτήσει των κύκλων για βλάβη (α) Fan και (β) LPC . . . . .	70
70	Πιθανότητα συναρτήσει των κύκλων για βλάβη (α) Fan και (β) LPC . . . . .	70
71	Πίνακας σύγχυσης για βλάβη μίας ή δύο συνιστωσών . . . . .	71
72	Υπογραφή συνδυασμού βλάβης (α) Fan + LPC, (β) Fan + HPC, (γ) Fan + HPT, (δ) Fan + LPT, (ε) LPC + HPC, (στ) LPC + HPT, (ζ) LPC + LPT, (η) HPC + HPT, (θ) HPC + LPT και (ι) HPT + LPT . . . . .	72

## Κατάλογος πινάκων

1	Όνομα block και τιμή στο σημείο σχεδιασμού για τις μεταβλητές του σημείου λειτουργίας . . . . .	22
2	Διαθέσιμες μετρήσεις του κινητήρα . . . . .	24
3	Παράμετροι υγείας του κινητήρα . . . . .	28
4	Λοιπές παράμετροι επιλύτη . . . . .	29
5	Ιδιότητες της εφαρμογής . . . . .	32
6	Ακολουθία σημείων που καλύπτουν τον φάκελο πτήσης . . . . .	52
7	Τυπική απόκλιση των μεταβλητών της υπογραφής . . . . .	57
8	Ποσοστά επιτυχούς ταξινόμησης για χαμηλό θόρυβο . . . . .	60
9	Ποσοστά επιτυχούς ταξινόμησης για υψηλό θόρυβο . . . . .	60
10	Μετρήσεις για την εκτίμηση των παραμέτρων υγείας κάθε συνιστώσας . . . . .	63
11	Δεδομένα γενικής χειροτέρευσης [11] . . . . .	69

## Κατάλογος Αναφορών

- [1] J.W. Chapman et al. “A Process for the Creation of T-MATS Propulsion System Models from NPSS data”. In: *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. 2014. DOI: [10.2514/6.2014-3931](https://doi.org/10.2514/6.2014-3931).
- [2] J.W. Chapman et al. *T-MATS*. 2020. URL: <https://github.com/nasa/T-MATS>.
- [3] J.W. Chapman et al. *Toolbox for the Modeling and Analysis of Thermodynamic Systems (T-MATS) User’s Guide*. English. NASA. 2020.
- [4] Amare Fentaye, Syed Gilani, and Aklilu Baheta. “Gas turbine gas path diagnostics: A review”. In: *MATEC Web of Conferences* 74 (Jan. 2016). DOI: [10.1051/mateconf/20167400005](https://doi.org/10.1051/mateconf/20167400005).
- [5] Simon Haykin. *Neural Networks and Learning Machines Third Edition*. New Jersey: Pearson, 2009.
- [6] MathWorks Inc. *Simulink*. 2022. URL: <https://mathworks.com/products/simulink.html>.
- [7] ISO Standard 2533:1975. *Standard Atmosphere*.
- [8] Anastasios O. Koskoletos et al. “Evaluation of Aircraft Engine Gas Path Diagnostic Methods Through ProDiMES”. In: *Journal of Engineering for Gas Turbines and Power* 140.12 (Nov. 2018). DOI: [10.1115/1.4040909](https://doi.org/10.1115/1.4040909).
- [9] Orestis A. Koskoletos. “Aircraft Engine Diagnostic Methods Benchmarking Using Propulsion Diagnostic Method Evaluation Strategy (ProDiMES)”. Diploma Thesis. National Technical University of Athens, 2017.
- [10] Jack D. Mattingly, William H. Heiser, and David T. Pratt. *Aircraft Engine Design 2nd Edition*. Virginia, USA: American Institute of Aeronautics and Astronautics, Inc., 2002.
- [11] G.P. Sallee. *Performance Deterioration Based On Existing (Historical) Data: JT9D Jet Engine Diagnostics Program*. Tech. rep. NASA, 1978.
- [12] Κωνσταντίνος Μαθιουδάκης. *Διαγνωστική αεριοστροβίλων*. Αθήνα: Εκδόσεις ΕΜΠ, 2016.
- [13] Κωνσταντίνος Μαθιουδάκης. *Λειτουργία Αεροπορικών Κινητήρων: Επιδόσεις, Εκπομπές Καυσερίων και Ήχου*. Αθήνα: Εκδόσεις ΕΜΠ, 2007.

## Παραρτήματα

### I Κώδικας εφαρμογής

#### add\_perf\_item

Προσθέτει ένα στοιχείο στο πλαίσιο των παραμέτρων απόδοσης

```
% Adds an item to the "Performance Parameters" Panel
% Inputs:
%   - app: app handle
%   - x: x position for checkbox
%   - y: y position of the item
%   - w: width
%   - h: height
%   - label_name: name of performance parameter
%   - value: initial value of field
%   - units: units of performance parameter
```

---

#### calcInitTable

Τρέχει προσομοιώσεις σε ένα μοντέλο σε διάφορα σημεία στον φάκελο πτήσης και αποθηκεύει τα αποτελέσματα

```
% Script that runs simulations on a model at various
%   points in the flight
% envelope and saves the results
```

---

#### calcJacobian

Υπολογίζει την Ιακωβιανή ενός δεδομένου μοντέλου αεριοστροβίλου

```
% Script that calculates the Jacobian matrix of a given
%   gas turbine model
```

---

#### checkHealthValues

Ελέγχει αν υπάρχουν μη μηδενικές τιμές στο πλαίσιο των παραμέτρων υγείας

```
% Checks the "Health Parameters" Panel for non-zero values
% Input:
%   - app: app handle
% Output:
%   - areAllZero: boolean that is true if all values are
%     zero
```

-----  
**checkValid**

Ελέγχει αν ο χρήστης έχει θέσει έγκυρο αριθμό σημείων για προσομοίωση

```
% Checks if the the user has inputted a negative number of
points
% Inputs:
%   - points_input: points input field handle
%   - event: event handle
```

-----  
**clearTable**

Πραγματοποιεί εκκαθάριση στον πίνακα των αποτελεσμάτων

```
% Clears the "Results" table
% Inputs:
%   - app: app handle
%   - results_table: "Results" table handle
%   - empty_table: An empty table with the appropriate
headers
```

-----  
**generateTables**

Δημιουργεί τους πίνακες που χρησιμοποιούνται για την εκτίμηση των αρχικών συνθηκών του επιλύτη Newton-Raphson

```
% Script that generates the tables used to estimate the
Newton-Raphson
% solver initial conditions
```

-----  
**getHPcases**

Υπολογίζει τον αριθμό κατάστασης για κάθε συνδυασμό ενός συγκεκριμένου αριθμού μετρήσεων και τους κατατάσσει από τον μικρότερο στον μεγαλύτερο

```
% Calculates the Condition Number for all combinations of
measurements of
% specified length and sorts them from smallest to largest
% Inputs:
%   - J: Jacobian matrix
%   - numHP_m: number of measurements
% Output:
%   - cases_HP: matrix that contains all combinations and
their
%               corresponding condition number
```



-----  
**getMcases**

Υπολογίζει τον αριθμό κατάστασης για κάθε συνδυασμό ενός συγκεκριμένου αριθμού παραμέτρων υγείας και τους κατατάσσει από τον μικρότερο στον μεγαλύτερο

```
% Calculates the Condition Number for all combinations of
% health parameters
% of specified length and sorts them from smallest to
% largest
% Inputs:
%   - J: Jacobian matrix
%   - numMeas_h: number of health parameters
% Output:
%   - cases_M: matrix that contains all combinations and
%   their
%               corresponding condition number
```

-----

**getNRIC**

Υπολογίζει τις αρχικές συνθήκες του επιλύτη Newton-Raphson μέσω έτοιμων πινάκων για την διευκόλυνση της σύγκλισης του επιλύτη

```
% Returns the Initial Conditions for the Newton-Raphson
% solver based on
% pre-generated tables to ease the convergence of the
% solver
% Inputs:
%   - alt: altitude
%   - Mach: Mach number
%   - sel_var: string indicating the user-selected control
%   variable
%   - sel_var_value: value of the user-selected control
%   variable
% Output:
%   - NRIC: Initial Conditions for the Newton-Raphson
%   solver
```

-----

**getSignatures**

Επιστρέφει τις τιμές των μεταβλητών της υπογραφής

```
% Returns the values for the variables contained in the
% Signature
% Output:
%   - sign_values: array containing the Signature values
```

---

### get\_misc\_perfs

Επιστρέφει τις τιμές των παραμέτρων απόδοσης που δεν αφορούν μία συγκεκριμένη θέση του κινητήρα

```
% Returns the values for the performance parameters that
% are unrelated to
% stations
% Output:
% - misc_values: array of miscellaneous performance
% parameters
```

---

### lock\_param

Καθιστά την επιλεγμένη μεταβλητή για παραμετρική ανάλυση την μοναδική με δύο ολισθητές και πεδία

```
% Makes the selected variable for parametric analysis the
% only one with a
% min and max sliders and input fields
% Inputs:
% - app: app handle
% - slider: slider handle for which the callback is
% triggered
```

---

### mapAndJoin

Βοηθητική συνάρτηση για την δημιουργία ετικετών

```
% Maps and joins array values
% Inputs:
% - x: input array
% - name_map: map container
```

---

### plot\_results\_SS

Σχεδιάζει τα αποτελέσματα μόνιμης κατάστασης

```
% T-MATS steady state data plotter.
% TMATS.SSplot(filename, StationVarVec) - Search for
% station data and performance
% maps, then plot them.
% TMATS.SSplot(_,Name,Value,...,Value) - plot steady state
% T-MATS data with alternate formating
% Name Definition
```

```
% 'TSR' - Don't plot T-s data, (_, 'TSR')
% 'SR' - Don't plot station data, (_, 'SR')
% 'NR' - Don't plot speed figures, (_, 'NR')
% 'CompR' - Don't plot compressor map figures, (_, 'CompR')
% 'TurbR' - Don't plot turbine map figures, (_, 'TurbR')
% 'MappntsR' - Don't plot data points on map figures, (_, 'MappntsR')
```

### PlotCMap\_mod

Σχεδιάζει τα αποτελέσματα στους χάρτες των συμπιεστών

```
% Plot Compressor Map
% TMATS.PlotCMap(NcVec, WcArray, PRArray, EffArray) - plot
    default compressor map
% TMATS.PlotCMap(_,Name,Value,...,Value) - plot compressor
    map with alternate formatting
% Name Definition
% 'linesize' - Define linesize (Default=1), (_, 'linesize',Num)
% 'defsurge' - Define surge line, (_, 'defsurge',Wc_SM,PR_SM)
% 'efflines' - Define number of efficiency lines (Default=10), (_, 'efflines',Num)
% 'slformat' - Define speed line marker value format (Default='%2.2f'), (_, 'slformat',Str)
% 'offset' - Define speed line marker location offset (Default=0.02), (_, 'offset',Num)
% 'scalar' - Define compressor map values updated with scalars, (_, 'scalar', s_Wc, s_PR, s_Eff)
% 'mapname' - Define the plot title (Default='Compressor Map'), (_, 'mapname',Str)
% 'plotlim' - Define the plot sizing limit by fraction of total size (Default=0.1), (_, 'plotlim',Num)
% 'NcR' - Remove Nc lines from plot, (_, 'NcR')
% 'RlineR' - Remove Rlines from plot, (_, 'RlineR')
% 'EffR' - Remove Efficiency from plot, (_, 'EffR')

% TMATS.PlotCMaps.m
% Jeffyres Chapman based on work by Jeffrey Csank and George Thomas
% NASA Glenn Research Center
% Jan 12, 2016
%
```

% This file plots compressor maps

---

### PlotTMap\_mod

Σχεδιάζει τα αποτελέσματα στους χάρτες των στροβίλων

```
% Plot Turbine Map
% TMATS.PlotTMap(NcVec, PRVec, WcArray, EffArray) - plot
  default compressor map
% TMATS.PlotTMap(_,Name,Value,...,Value) - plot compressor
  map with alternate formatting
% Name          Definition
% 'linesize'    -   Define linesize (Default=1), (_, '
  linesize',Num)
% 'efflines'     -   Define number of efficiency lines (
  Default=10), (_, 'efflines',Num)
% 'slformat'     -   Define speed line marker value format
  (Default='%2.0f'), (_, 'slformat',Str)
% 'offset'       -   Define speed line marker location
  offset (Default=0.02), (_, 'offset',Num)
% 'scalar'       -   Define compressor map values updated
  with scalars, (_, 'scalar', s_Wc, s_PR, s_Eff)
% 'mapname'      -   Define the plot title (Default='
  Turbine Map'), (_, 'mapname',Str)
% 'plotlim'      -   Define the plot sizing limit by
  fraction of total size (Default=0.1), (_, 'plotlim',Num)
% 'EffR'         -   Remove Efficiency from plot , (_, 'EffR
  ')

% TMATS.PlotTMaps.m
% Jeffyres Chapman based on work by Jeffrey Csank and
  George Thomas
% NASA Glenn Research Center
% Jan 12, 2016
%
% This file plots turbine maps
```

---

### refreshConsole

Ανανεώνει την κονσόλα

```
% Sets the value of the console text area to the value
  contained in the
% diary file
% Input:
%   - app: app handle
```

---

### reset\_health

Θέτει τις τιμές των παραμέτρων υγείας στην διεπαφή χρήστη ίση με 0

```
% Resets all sliders and input fields in the "Health  
Parameters" panel to 0  
% Input:  
%   - HealthPanel: "Health Parameters" panel handle
```

---

### run\_model

Πραγματοποιεί προσομοίωση ενός ή πολλαπλών σημείων και διάγνωση σε ένα μοντέλο

```
% Performs single or multiple point simulation and  
diagnosis on a model  
% Inputs:  
%   - btn_group: Button group handle for mode selection  
%   - app: app handle  
%   - range_cb: Checkbox handle for simulation type  
selection
```

---

### saveTable

Αποθηκεύει τον πίνακα των αποτελεσμάτων σε ένα υπολογιστικό φύλλο

```
% Saves the "Results" table to an Excel spreadsheet  
% Input:  
%   - results_table: "Results" table handle
```

---

### SelectControlVar

Υπολογίζει το σφάλμα της μεταβλητής ελέγχου εισόδου

```
% Calculates and outputs the error of the input control  
variable  
% Inputs:  
%   - iControl: index of the desired control variable  
%   - Meas: signature signal  
%   - ControlValue: value of the control variable  
% Output:  
%   - y: control variable error
```

---

### selectCBs

Επιλέγει τα πλαίσια ελέγχου που υποδεικνύονται από την τιμή ενός μενού

```
% Selects checkboxes according the selected value of a  
dropdown menu  
% Inputs:  
%   - app: app handle  
%   - dd: dropdown menu handle
```

---

### select\_hp

Επιστρέφει τα κατάλληλα σήματα παραμέτρων υγείας

```
% Outputs the correct health parameter signals  
% Inputs:  
%   - u: solver output signal  
%   - hp_indices: indices of health parameters to select  
% Output:  
%   - y: signal containing the desired health parameter  
signals
```

---

### setup\_cp\_dd

Εισάγει στο κατάλληλο μενού τις επιλογές για την πραγματοποίηση διάγνωσης

```
% Populates the appropriate dropdown menu with the options  
for diagnosis  
% Inputs:  
%   - app: app handle  
%   - btn: button handle that triggered the callback
```

---

### setup\_diag\_btns

Εισάγει τα κουμπιά που χρησιμεύουν για την εύρεση μετρήσεων ή παραμέτρων υγείας

```
% Sets up the buttons used to find the measurements or  
health parameters  
% and their dropdown menus  
% Input:  
%   - app: app handle
```

---

### setup\_diagnosis\_parameters

Θέτει όλες τις παραμέτρους του μοντέλου που χρειάζονται για την πραγματοποίηση διάγνωσης και ελέγχει αν ο αριθμός των επιλεγμένων μετρήσεων είναι ίσος με τον αριθμό των επιλεγμένων παραμέτρων υγείας

```
% Sets all model parameters that are needed to perform a
    diagnosis and
% checks if the number of selected measurements is equal
    to the number of
% selected health parameters
% Input:
%   - app: app handle
% Output:
%   - isValid: string indicating if the number of selected
        measurements is
%               equal to the number of selected health
        parameters
```

---

### setup\_double\_input

Εισάγει ένα στοιχείο εισόδου το οποίο έχει 2 ολισθητές και 2 πεδία

```
% Sets up an input item that has 2 sliders and 2 input
    fields
% Inputs:
%   - app: app handle
%   - ind: index of input item to setup
%   - x_label: x coordinate for label or dropdown menu
%   - y: y coordinate of item
%   - value: slider and input field default value
%   - x_input: x coordinate of the first input field
%   - height: item height
%   - flag: boolean indicating whether a label or dropdown
        menu should be
%               used
```

---

### setup\_health

Δημιουργεί το πλαίσιο των παραμέτρων υγείας (βλ. Σχήμα 36)

```
% Sets up the "Health Parameters" panel UI
% Input:
%   - app: app handle
% Output:
%   - health_info: cell array storing the health block
        name and the component
%               name and type
```

---

### setup\_health\_ui

Εισάγει ένα στοιχείο στο πλαίσιο των παραμέτρων υγείας

```
% Sets up a health parameter item
% Inputs:
%   - app: app handle
%   - name: component name
%   - health_params: default value for health parameters
%   - x: reference x coordinate
%   - y: reference y coordinate
%   - w: reference width value
%   - h: item height
```

---

### setup\_inputs

Δημιουργεί το πλαίσιο των εισόδων (βλ. Σχήμα 35)

```
% Sets up the "Operating Point" panel
% Input:
%   - app: app handle
```

---

### setup\_perf

Δημιουργεί το πλαίσιο των παραμέτρων απόδοσης (βλ. Σχήμα 37)

```
% Sets up the "Performance Parameters" panel
% Input:
%   - app: app handle
```

---

### setup\_schematic\_data

Δημιουργεί το πλαίσιο του σχηματικού (βλ. Σχήμα 42)

```
% Sets up the model Schematic
% Input:
%   - app: app handle
```

---

### setup\_single\_input

Εισάγει ένα στοιχείο εισόδου το οποίο έχει 1 ολισθητή και 1 πεδίο

```
% Sets up an input item that has 1 slider and 1 input
  field
% Inputs:
%   - app: app handle
%   - value: slider and input field default value
%   - ind: index of input item to setup
%   - x_input: x coordinate of the first input field
```



```
% - y: y coordinate of item
% - height: item height
% - x_label: x coordinate for label or dropdown menu
% - flag: boolean indicating whether a label or dropdown
%         menu should be
%         used
```

---

### sim\_and\_send\_to\_workspace

Τρέχει μία προσομοίωση ενός μοντέλου και επιστρέφει το αποτέλεσμα στο χώρο εργασίας (workspace)

```
% Runs a model simulation and sends output variables to
%   the workspace while
% displaying if the current operating point has converged
% Input:
%   - app: app handle
```

---

### simulate\_point

Θέτει τις εισόδους και τις παραμέτρους υγείας στο μοντέλο, τρέχει μία προσομοίωση ενός σημείου, σχεδιάζει τα αποτελέσματα και ενημερώνει τον πίνακα αποτελεσμάτων και τα πεδία του πλαισίου των παραμέτρων απόδοσης

```
% Assigns the input values and health parameters in the
%   model, runs a
% single point simulation, plots the results and updates
%   the "Results"
% table and all "Performance Parameters" fields
% Input:
%   - app: app handle
```

---

### simulate\_point\_zero\_hp

Θέτει τις εισόδους στο μοντέλο και τρέχει μία προσομοίωση ενός σημείου, θέτοντας μηδενική τιμή σε όλες τις παραμέτρους υγείας

```
% Assigns the input values in the model and runs a single
%   point simulation
% setting all health parameters to zero
% Input:
%   - app: app handle
```

---

### simulate\_range

Θέτει τις εισόδους και τις παραμέτρους υγείας στο μοντέλο, τρέχει μία προσομοίωση πολλαπλών σημείων, σχεδιάζει τα αποτελέσματα και ενημερώνει τον πίνακα αποτελεσμάτων και τα πεδία του πλαισίου των παραμέτρων απόδοσης

```
% Assigns the input values and health parameters in the
    model, runs a
% multi-point simulation, plots the results and updates
    the "Results"
% table and all "Performance Parameters" fields
% Input:
%   - app: app handle
```

---

### startupFcn

Συνάρτηση που καλείται κατά την εκκίνηση της εφαρμογής και μέσω της οποίας δημιουργείται η διεπαφή χρήστη

---

### toggle\_mode

Αλλάζει την εφαρμογή από λειτουργία προσομοίωσης σε λειτουργία διάγνωσης και αντιστρόφως

```
% Toggles between "Simulation" and "Diagnosis" mode
% Inputs:
%   - app: app handle
%   - btn_group: button group that controls the mode
```

---

### toggle\_range

Αλλάζει την εφαρμογή από λειτουργία προσομοίωσης ενός σημείου σε λειτουργία προσομοίωσης πολλαπλών σημείων και αντιστρόφως

```
% Toggles between single point and multi-point simulation
    modifying the UI
% Inputs:
%   - app: app handle
%   - range_cb: checkbox that controls the type of
    simulation
```

---

### toggle\_sign

Επιλέγει ή από-επιλέγει μια μεταβλητή της υπογραφής στο μοντέλο

```
% Sets the value of the input signature variable's flag to
    1 if it's
```

```
% selected and 0 if not, in the model
% Inputs:
%   - app: app handle
%   - cb: signature checkbox
```

---

### UIFigureCloseRequest

Συνάρτηση που καλείται όταν ο χρήστης κλείσει την εφαρμογή

```
% Close request function
```

---

### updateLimits

Ενημερώνει τα όρια ενός ζευγαριού ολισθητών για δυναμική επιλογή εύρους προσομοίωσης

```
% Update the limits of a slider pair for dynamic range
% selection
% Inputs:
%   - event: event handle
%   - app: app handle
```

---

### update\_health

Θέτει τις τιμές των παραμέτρων υγείας στο μοντέλο

```
% Assigns the health parameter values to the model
% Inputs:
%   - filename: path to the model
%   - health_info: cell array storing the health block
%                   name and the
%                   component name and type
```

---

### update\_input\_dd

Ενημερώνει τη διεπαφή χρήστη και το μοντέλο σύμφωνα με την επιλεγμένη μεταβλητή ελέγχου

```
% Updates the UI and the model with the newly selected
% dropdown menu
% control variable
% Inputs:
%   - app: app handle
%   - dd: dropdown menu handle
%   - input: input field handle
%   - slider: slider handle
```

---

### update\_input\_pair

Θέτει την καινούργια τιμή ενός πεδίου εισόδου στον αντίστοιχο ολισθητή και αντίστροφως

```
% Sets the new value of the input field to the slider and  
    vice versa  
% Inputs:  
%   - event: event handle  
%   - pair: handle of the component to the update the  
    value of
```

---

### update\_perf\_plot

Ενημερώνει το γράφημα αποτελεσμάτων σύμφωνα με τις επιλεγμένες μεταβλητές

```
% Updates the "Engine Perf." plot with newly selected  
    variables  
% Inputs:  
%   - xdrop: dropdown menu handle to select x-axis  
    variable  
%   - ydrop: dropdown menu handle to select y-axis  
    variable  
%   - perf_plot: "Engine Perf." plot handle
```

---

### update\_schematic

Ενημερώνει τα πεδία του σχηματικού με τα αποτελέσματα της τελευταίας προσομοίωσης

```
% Update the "Schematic" fields with the results of the  
    latest simulation  
% Input:  
%   - app: app handle
```

---

## II Κώδικας Νευρωνικών Δικτύων

### combinedFault

Δημιουργεί πρότυπα και πραγματοποιεί διάγνωση για την περίπτωση συνδυασμού γενικής χειροτέρευσης και βλάβης μίας συνιστώσας

```
% Script that generates patterns and performs diagnosis  
    for the combination  
% of deterioration and single component fault
```

---

### combinedFaultComp

Δημιουργεί πρότυπα και πραγματοποιεί διάγνωση για την περίπτωση συνδυασμένης βλάβης 2 συνιστωσών

```
% Script that generates patterns and performs diagnosis  
    for the 2 component  
% combined fault
```

---

### Deterioration

Δημιουργεί πρότυπα δοκιμής για την περίπτωση της γενικής χειροτέρευσης

```
% Script that generates testing patterns for the  
    deterioration case
```

---

### diagFeedforwardnet

Πραγματοποιεί διάγνωση χρησιμοποιώντας ένα Νευρωνικό Δίκτυο Πρόσθιας Τροφοδότησης (Feedforward Neural Network)

```
% Performs diagnosis on a set of testing patterns using a  
    Feedforward  
% Neural Network (either already trained or new), outputs  
    how many patterns  
% were successfully classified and plots the Confusion  
    Matrix and the ROC  
% plot  
% Input:  
%   - tr_pats: S x P matrix containing the training  
        patterns, where S is  
%           the number of pattern variables and P the  
        number of patterns  
%   - te_pats: S x Q matrix containing the testing  
        patterns where Q is the  
%           number of patterns
```

```
% - tr_classes: String vector of length P containing the
corresponding
%               class for every training pattern
% - te_classes: String vector of length Q containing the
corresponding
%               class for every testing pattern
% - classes: Vector containing the classes to be used in
the
%               classification
% - flag: Boolean for plotting the Confusion Matrix and
the ROC plot
% - trainFcn: Training function name
% - net: Trained Neural Network (optional)
% Output:
% - results: cell containing how many and what
percentage of the training
%               patterns where classified correctly or not
% - predictedClass: Vector of length Q containing the
predicted class for
%               every testing pattern
```

---

### diagnoseUsingNN

Πραγματοποιεί διάγνωση σε ένα σύνολο προτύπων δοκιμής χρησιμοποιώντας την επιλεγμένη μέθοδο και σχεδιάζει τα αποτελέσματα

```
% Performs diagnosis on a set of testing patterns using
the selected method
% and plots the Confusion Matrix and the ROC plot
% Inputs:
% - tr_pats: S x P matrix containing the training
patterns, where S is
%               the number of pattern variables and P the
number of patterns
% - te_pats: S x Q matrix containing the testing
patterns where Q is the
%               number of patterns
% - tr_classes: String vector of length P containing the
corresponding
%               class for every training pattern
% - te_classes: String vector of length P containing the
corresponding
%               class for every testing pattern
% - std_scale: Scaling factor to be applied to the
standard deviation of
%               every pattern variable
% - method: String containing the name of the desired
```

```

method
%   - n_tr: Number of noisy patterns per noise-free
      training pattern
%   - n_te: Number of noisy patterns per noise-free
      testing pattern
%   - s: Smoothing factor(PNN) or spread of radial basis
      functions(newpnn)
%   - classes: Vector containing the classes to be used in
      the
%               classification
%   - flag: Boolean for plotting the Confusion Matrix and
      the ROC plot
%   - trainFcn: Training function name
%   - net: Trained Neural Network (optional)
% Outputs:
%   - results: Cell containing how many and what
      percentage of the training
%               patterns were classified correctly or not
%   - predictions: Vector of length Q * n_te containing
      the predicted class
%               for every testing pattern
%   - te_pats: S x (Q * n_te) matrix containing the noisy
      testing patterns
%   - prob: S x (Q * n_te) matrix containing the
      probability each testing
%               pattern belongs to each class (only for PNN)

```

---

### diagPatternnet

Πραγματοποιεί διάγνωση χρησιμοποιώντας ένα Νευρωνικό Δίκτυο Αναγνώρισης Προτύπων (Pattern Recognition Neural Network)

```

% Performs diagnosis on a set of testing patterns using a
      Pattern
% Recognition Neural Network (either already trained or
      new), outputs how
% many patterns were successfully classified and plots the
      Confusion Matrix
% and the ROC plot
% Input:
%   - tr_pats: S x P matrix containing the training
      patterns, where S is
%               the number of pattern variables and P the
      number of patterns
%   - te_pats: S x Q matrix containing the testing
      patterns where Q is the
%               number of patterns

```

```
% - tr_classes: String vector of length P containing the
corresponding
%               class for every training pattern
% - te_classes: String vector of length Q containing the
corresponding
%               class for every testing pattern
% - classes: Vector containing the classes to be used in
the
%               classification
% - flag: Boolean for plotting the Confusion Matrix and
the ROC plot
% - trainFcn: Training function name
% - net: Trained Neural Network (optional)
% Output:
% - results: cell containing how many and what
percentage of the training
%               patterns where classified correctly or not
% - predictedClass: Vector of length Q containing the
predicted class for
%               every testing pattern
```

---

### getHealthyPatterns

Δημιουργεί πρότυπα εκπαίδευσης για την υγιή κατάσταση του κινητήρα

```
% Script that generates training patterns for the healthy
state of the
% engine
```

---

### getNoisyPatterns

Προσθέτει θόρυβο σε ένα σύνολο προτύπων

```
% Adds noise to a set of patterns
% Inputs:
% - n: Number of noisy patterns per noise-free testing
pattern
% - stdVec: Vector containing the standard deviation for
every pattern
%               variable
% - noiseFreePats: S x P matrix containing the noise-
free patterns, where
%               S is the number of pattern variables
and P the number
%               of patterns
% - noiseFreeClasses: Vector of strings containing the
corresponding
```



```
%                               class for every noise-free pattern
% Outputs:
%   - noisyPats: S x (n*P) matrix containing the noisy
%     patterns
%   - noisyClasses: Vector containing the correspondings
%     class for every
%                               noisy pattern
```

---

### getPatterns

Δημιουργεί πρότυπα εκπαίδευσης και δοκιμής

```
% Script that generates training and testing patterns
```

---

### getTestingPatsOP

Δημιουργεί πρότυπα δοκιμής για διαφορετικά σημεία λειτουργίας

```
% Script that generates testing patterns for different
operating points
```

---

### newpnnFcn

Πραγματοποιεί διάγνωση χρησιμοποιώντας την συνάρτηση newpnn

```
% Performs diagnosis on a set of testing patterns using
% the newpnn function,
% outputs how many patterns were successfully classified
% and plots the
% Confusion Matrix and the ROC plot
% Input:
%   - tr_pats: S x P matrix containing the training
%     patterns, where S is
%               the number of pattern variables and P the
%     number of patterns
%   - te_pats: S x Q matrix containing the testing
%     patterns where Q is the
%               number of patterns
%   - tr_classes: String vector of length P containing the
%     corresponding
%                 class for every training pattern
%   - te_classes: String vector of length Q containing the
%     corresponding
%                 class for every testing pattern
%   - classes: Vector containing the classes to be used in
%     the
```

```
%           classification
%   - s: Spread of radial basis functions
%   - flag: Boolean for plotting the Confusion Matrix and
%         the ROC plot
% Output:
%   - results: cell containing how many and what
%             percentage of the training
%             patterns where classified correctly or not
%   - predictedClass: Vector of length Q containing the
%             predicted class for
%             every testing pattern
```

---

### NNDiagnosis

Πραγματοποιεί διάγνωση εκτιμώντας και τις παραμέτρους υγείας

```
% Script that performs diagnosis that estimates the health
% parameters
```

---

### plotCMAndROC

Σχεδιάζει τον πίνακα σύγχυσης και το διάγραμμα ROC

```
% Plots the Confusion Matrix and the ROC plot
% Inputs:
%   - te_classes: String vector of length Q containing the
%             corresponding
%             class for every testing pattern, where Q
%             is the number of
%             testing patterns
%   - predictedClass: Vector of length Q containing the
%             predicted class for
%             every testing pattern
%   - classMap: Map container that maps every class to a
%             number
%   - method: String containing the method name to be used
%             in the plot
%             title
```

---

### PNN

Πραγματοποιεί διάγνωση χρησιμοποιώντας ένα Πιθανολογικό Νευρωνικό Δίκτυο (Probabilistic Neural Network)

```
% Performs diagnosis on a set of testing patterns using a
% Probabilistic
```

```
% Neural Network, outputs how many patterns were
% successfully classified
% and plots the Confusion Matrix and the ROC plot
% Input:
%   - tr_pats: S x P matrix containing the training
%     patterns, where S is
%       the number of pattern variables and P the
%     number of patterns
%   - te_pats: S x Q matrix containing the testing
%     patterns where Q is the
%       number of patterns
%   - tr_classes: String vector of length P containing the
%     corresponding
%       class for every training pattern
%   - te_classes: String vector of length Q containing the
%     corresponding
%       class for every testing pattern
%   - classes: vector containing the classes to be used in
%     the
%       classification
%   - s: Smoothing factor
%   - flag: Boolean for plotting the Confusion Matrix and
%     the ROC plot
% Output:
%   - results: Cell containing how many and what
%     percentage of the training
%       patterns were classified correctly or not
%   - predictedClass: Vector of length Q containing the
%     predicted class for
%       every testing pattern
%   - P_te: Matrix containing the probability each testing
%     pattern belongs
%       to each class
```

---

### TestSensitivity

Δημιουργεί τα δεδομένα για τα διαγράμματα ευαισθησίας

```
% Script that generates the data for the sensitivity plots
```

---

### TestSensitivityPlots

Δημιουργεί τα διαγράμματα ευαισθησίας

```
% Script that creates the sensitivity plots
```

— — — — —  
**trainPNN**

Έκπαιδεύει ένα Πιθανολογικό Νευρωνικό Δίκτυο (Probabilistic Neural Network)

% Script that performs training on a PNN

— — — — —