



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

**Αξιολόγηση της Ποιότητας Συστημάτων Μέτρησης  
Υψηλών Κρουστικών Τάσεων σύμφωνα με το  
Ευρωπαϊκό Πρότυπο EN 60060-2**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

Ασημίνα Μ. Σαρλή

**Επιβλέπων :** Φραγκίσκος Β. Τοπαλής  
Καθηγητής ΕΜΠ

Αθήνα, Μάιος 2008





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΗΣ ΙΣΧΥΟΣ

## Αξιολόγηση της Ποιότητας Συστημάτων Μέτρησης Υψηλών Κρουστικών Τάσεων σύμφωνα με το Ευρωπαϊκό Πρότυπο EN 60060-2

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ασημίνα Μ. Σαρλή

Επιβλέπων : Φραγκίσκος Β. Τοπαλής  
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Μαΐου 2008 .

.....  
Φραγκίσκος Τοπαλής  
Καθηγητής ΕΜΠ

.....  
Ιωάννης Σταθόπουλος  
Καθηγητής ΕΜΠ

.....  
Περικλής Μπούρκας  
Καθηγητής ΕΜΠ

Αθήνα, Μάιος 2008

.....  
Ασημίνα Μ. Σαρλή

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Ασημίνα Σαρλή, 2008.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## ΠΕΡΙΛΗΨΗ

Μία κατηγορία δοκιμών στις οποίες υπόκειται ο εξοπλισμός υψηλής τάσης πριν από τη σύνδεσή του σε μια ηλεκτρική εγκατάσταση, είναι οι διηλεκτρικές (δοκιμές μονώσεως). Οι διηλεκτρικές δοκιμές γίνονται κυρίως για να διαπιστωθεί η αντοχή μιας διάταξης στις ατμοσφαιρικές και εσωτερικές υπερτάσεις. Οι ατμοσφαιρικές υπερτάσεις οφείλονται στους κεραυνούς και προσομοιώνονται στο εργαστήριο με τις διπλοεκθετικές κρουστικές τάσεις της μορφής 1,2/50  $\mu$ s, ενώ οι εσωτερικές υπερτάσεις οφείλονται σε εσωτερικές διαταραχές του συστήματος όπως η απομάκρυνση μιας γραμμής μεταφοράς και προσομοιώνονται με τις διπλοεκθετικές κρουστικές τάσεις μεγάλης διάρκειας της μορφής 250/2500  $\mu$ s. Μία γεννήτρια παράγει την κρουστική τάση η οποία εφαρμόζεται στο υπό εξέταση δοκίμιο. Η υψηλή αυτή τάση πρέπει να καταγραφεί από ένα σύστημα μέτρησης προκειμένου να είναι γνωστή η τιμή και η μορφή της. Άρα το σύστημα μέτρησης πρέπει να είναι ακριβές και να μην εισάγει σημαντικά σφάλματα που μπορεί να οδηγήσουν στην εξαγωγή λανθασμένων συμπερασμάτων όσον αφορά τον εξοπλισμό. Το ευρωπαϊκό πρότυπο EN 60060-2 διατυπώνει τις προϋποθέσεις που πρέπει να πληρεί ένα σύστημα μέτρησης προκειμένου να χαρακτηριστεί ως εγκεκριμένο. Στην εργασία αυτή περιγράφονται διάφορες μεθοδολογίες υπολογισμού των παραμέτρων απόκρισης ενός συστήματος μέτρησης όπως ορίζονται στο ευρωπαϊκό πρότυπο EN 60060-2. Παράλληλα αναπτύχθηκε μια σχετική εφαρμογή σε γλώσσα προγραμματισμού Java μέσω της οποίας υπολογίζονται με αυτοματοποιημένο τρόπο οι παράμετροι απόκρισης ενός συστήματος μέτρησης όπως η υπερύψωση  $\beta$ , ο μερικός χρόνος απόκρισης  $T_w$ , ο πειραματικός χρόνος απόκρισης  $T_N$ , ο χρόνος ισορρόπησης  $T_s$  και ο αρχικός χρόνος παραμόρφωσης  $T_0$ . Τέλος, χρησιμοποιώντας την εφαρμογή σε δύο περιπτώσεις, για δύο διαφορετικά συστήματα μέτρησης, υπολογίζονται και συγκρίνονται οι παράμετροι απόκρισης σε καθένα από αυτά, για διάφορες επιλογές εξομάλυνσης της καμπύλης απόκρισης.

**Λέξεις κλειδιά:** IEC 60060-2, συστήματα μέτρησης υψηλών τάσεων, κρουστική τάση, παράμετροι απόκρισης, βηματική απόκριση, εξομάλυνση, χρόνος απόκρισης, υπερύψωση



## ABSTRACT

High voltage devices are subject to various tests before their use in an electrical installation, one of which is the category of dielectric insulation tests. The dielectric tests are usually performed in order to guarantee the proper functioning of the device even when facing atmospheric or internal overshoots. Atmospheric overshoots are caused by lightnings and are modelled in the lab by double exponential impulse voltages of the type 1,2/50  $\mu$ s. Internal overshoots derive from internal instabilities of the system such as the removal of a transmission line and are modelled as double exponential long duration impulse voltages of the type 250/2500  $\mu$ s. One generator produces the impulse that is applied to the device under examination. The value and the shape of this high voltage has to be recorded by a measuring system, which consequently has to be accurate enough so as not to introduce significant errors that can lead to wrong conclusions about the device. The European Standard EN 60060-2 states the requirements which the measuring systems shall meet in order to be considered as approved. This thesis describes various methodologies through which the response parameters of a measuring system are estimated, according to the european standard EN 60060-2. In parallel a Java application was developed, which allows the automatic estimation of the response parameters such as the overshoot  $\beta$ , the partial response time  $T_{\alpha}$ , the experimental response time  $T_N$ , the settling time  $T_s$  and the initial distortion time  $T_0$ . Finally, using the application in two use cases for two different measuring systems, the response parameters of each of them are estimated and compared, for various smoothing options of the response curve.

**Keywords:** IEC 60600-2, high voltage measuring systems, impulse voltage, response parameters, step response, smoothing, response time, overshoot





## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Φραγκίσκο Τοπαλή που μου εμπιστεύθηκε το θέμα της διπλωματικής αυτής εργασίας, καθώς και για την καθοδήγηση και τις συμβουλές που μου παρείχε καθ' όλη την διάρκεια της εκπόνησής της.

Επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου για την αμέριστη συμπαράσταση και υποστήριξή τους όλα τα χρόνια των σπουδών μου.



## Πίνακας Περιεχομένων

1.	Εισαγωγή .....	17
1.1.	Γενικά στοιχεία .....	17
1.2.	Βασικά στοιχεία για τους κεραυνούς.....	19
1.2.1.	Είδη νεφών που γεννούν κεραυνούς.....	19
1.2.2.	Δημιουργία ενός κεραυνού .....	20
1.2.3.	Κατηγορίες κεραυνικών εκκενώσεων.....	21
1.2.4.	Παράμετροι ενός κεραυνού .....	22
1.3.	Βασικά στοιχεία για τις γραμμές μεταφοράς.....	22
1.4.	Βασικά στοιχεία για τους μονωτήρες .....	23
2.	Δοκιμές με υψηλές τάσεις.....	25
2.1.	Συσκευές υψηλής τάσης .....	25
2.2.	Παραδείγματα δοκιμών με υψηλές τάσεις.....	25
2.2.1.	Μονωτήρες.....	28
2.2.2.	Καλώδια .....	28
3.	Συστήματα μέτρησης υψηλών τάσεων .....	29
3.1.	Μέτρηση υψηλών τάσεων .....	30
3.1.1.	Ηλεκτροστατικό βολτόμετρο .....	31
3.1.2.	Διάκενο σφαιρών .....	32
3.1.3.	Αμπερόμετρο με αυτεπαγωγή για τη μέτρηση εναλλασσόμενης τάσης.....	34
3.1.4.	Αμπερόμετρο με χωρητικότητα για τη μέτρηση εναλλασσόμενης τάσης.....	34
3.1.5.	Μετασχηματιστές μετρήσεως .....	34
3.1.6.	Καταμεριστές τάσεως .....	36
3.1.7.	Καταμεριστές για συνεχή τάση.....	37
3.1.8.	Καταμεριστές για ημιτονοειδή τάση.....	38
3.1.9.	Μέτρηση ημιτονοειδούς τάσης με χωρητικούς καταμεριστές.....	39
3.1.10.	Καταμεριστές για μέτρηση κρουστικών τάσεων .....	39
3.1.11.	Ωμικός καταμεριστής για μέτρηση κρουστικών τάσεων.....	40
3.1.12.	Χωρητικός καταμεριστής .....	41
3.1.13.	Χωρητικός καταμεριστής με απόσβεση .....	42
3.1.14.	Οι βραχίονες χαμηλής τάσεως των καταμεριστών .....	43
4.	Συγκριτική μελέτη μεθόδων αξιολόγησης των παραμέτρων των κεραυνικών παλμών ..	45
4.1.	Γενικά στοιχεία .....	45
4.2.	Σχετικές εργασίες.....	46
4.3.	Τεχνική MSD .....	47
4.4.	Τεχνική προσαρμογής καμπύλης.....	48
5.	Το πρότυπο EN 60060-2.....	51
5.1.	Γενικά στοιχεία .....	51
5.2.	Ορισμοί βασικών εννοιών του προτύπου .....	51
5.2.1.	Συστήματα μέτρησης .....	51
5.2.2.	Δοκιμές .....	52
5.2.3.	Συντελεστές αναλογίας.....	52
5.2.4.	Δυναμική συμπεριφορά συστήματος μέτρησης.....	52
5.2.5.	Παράμετροι απόκρισης.....	53
5.3.	Διαδικασίες για τον χαρακτηρισμό των συστημάτων μέτρησης.....	55
5.3.1.	Γενικές αρχές .....	55
5.4.	Μέτρηση συνεχούς τάσης.....	56
5.4.1.	Απαιτήσεις για ένα εγκεκριμένο σύστημα μέτρησης .....	56
5.4.2.	Ευστάθεια του συντελεστή αναλογίας.....	56

5.4.3.	Δυναμική συμπεριφορά για μέτρηση της τάσης ανόδου.....	56
5.5.	Μέτρηση κεραυνικής κρουστικής τάσης.....	56
5.5.1.	Απαιτήσεις ενός εγκεκριμένου συστήματος μέτρησης .....	56
5.5.2.	Δυναμική συμπεριφορά .....	57
5.6.	Μέτρηση τυπικής κρουστικής τάσης.....	57
5.6.1.	Απαιτήσεις ενός εγκεκριμένου συστήματος μέτρησης .....	57
5.6.2.	Δυναμική συμπεριφορά .....	57
5.7.	Συστήματα αναφοράς .....	58
5.7.1.	Απαιτήσεις για συστήματα αναφοράς συνεχούς τάσης.....	58
5.7.2.	Απαιτήσεις για συστήματα αναφοράς εναλλασσόμενης τάσης .....	58
5.7.3.	Απαιτήσεις για συστήματα αναφοράς για πλήρεις και αποκομμένους στην ουρά κεραυνικούς και τυπικούς κρουστικούς παλμούς .....	58
5.7.4.	Απαιτήσεις για συστήματα αναφοράς για κεραυνικές τάσεις αποκομμένες στο μέτωπο .....	58
6.	Διαδικασία υπολογισμού των παραμέτρων απόκρισης συστήματος μέτρησης .....	59
6.1.1.	Εξομάλυνση βηματικής απόκρισης .....	59
6.1.2.	Εύρεση ουσιαστικής αρχής $O_1$ .....	61
6.1.3.	Εύρεση χρονικών παραμέτρων απόκρισης.....	61
7.	Εφαρμογή λογισμικού για τον υπολογισμό των παραμέτρων απόκρισης συστημάτων μέτρησης.....	63
7.1.	Γενικά στοιχεία.....	63
7.2.	Δυνατότητες της εφαρμογής.....	63
7.3.	Χρήση της εφαρμογής .....	65
7.3.1.	Εκκίνηση.....	65
7.3.2.	Επιλογή αρχείου δεδομένων .....	65
7.3.3.	Φόρτωση δεδομένων .....	66
7.3.4.	Ρύθμιση επιλογών εξομάλυνσης .....	66
7.3.5.	Εφαρμογή διαδικασίας – υπολογισμός παραμέτρων.....	67
7.3.6.	Αποθήκευση αποτελεσμάτων .....	68
7.3.7.	Καθαρισμός Υπολογισμών .....	70
7.3.8.	Αυτοματοποιημένη εκτέλεση σειράς υπολογισμών .....	70
7.4.	Τα κυριότερα σημεία του κώδικα της εφαρμογής.....	71
7.4.1.	Οι κλάσεις της εφαρμογής.....	71
7.4.2.	Η κλάση FooterPanel.....	72
7.4.3.	Η κλάση InfoDialog.....	72
7.4.4.	Η κλάση HVLchart.....	72
7.4.5.	Η κλάση MassiveMeasurementsTask.....	73
7.4.6.	Η κλάση HVLgui.....	74
8.	Ανάλυση δεδομένων και αποτελέσματα .....	79
8.1.	Απόκριση χωρίς αντιστάθμιση .....	80
8.2.	Απόκριση με αντιστάθμιση .....	82
8.3.	Συμπεράσματα.....	84
	Πηγές – Αναφορές.....	89
	Παράρτημα Α1 .....	91
	Ανάλυση παραμέτρων συστήματος μέτρησης χωρίς αντιστάθμιση.....	91
	Παράρτημα Α2 .....	121
	Ανάλυση παραμέτρων συστήματος μέτρησης με αντιστάθμιση.....	121
	Παράρτημα Β.....	151
	Ο πλήρης κώδικας της εφαρμογής .....	151
	Η κλάση HVLgui.....	151

Η κλάση HVLchart .....	166
Η κλάση MassiveMeasurementsTask .....	168
Η κλάση SwingWorker .....	169
Η κλάση InfoDialog .....	171
Η κλάση FooterPanel .....	173
Η κλάση ExampleFileFilter .....	174



## Πίνακας Εικόνων

Εικόνα 1: Στάδια δημιουργίας ενός κεραυνού.....	21
Εικόνα 2: Διάταξη δοκιμών υψηλών κρουστικών τάσεων.....	27
Εικόνα 3: Ηλεκτροστατικό βολτόμετρο .....	32
Εικόνα 4: Κατακόρυφο διάκενο σφαιρών .....	33
Εικόνα 5: Οριζόντιο διάκενο σφαιρών .....	33
Εικόνα 6: Καταμεριστής τάσεως .....	36
Εικόνα 7: Γραμμή μεγάλου μήκους.....	41
Εικόνα 8: Χωρητικός καταμεριστής.....	41
Εικόνα 9: Μικτός χωρητικός καταμεριστής .....	42
Εικόνα 10: Χωρητικός καταμεριστής με απόσβεση.....	44
Εικόνα 11: Γραφική απεικόνιση παραμέτρων απόκρισης.....	55
Εικόνα 12: Το γραφικό περιβάλλον της εφαρμογής.....	65
Εικόνα 13: Παράθυρο επιλογής αρχείου .....	66
Εικόνα 14: Το GUI μετά την φόρτωση της αρχικής καμπύλης.....	67
Εικόνα 15: Παρουσίαση των παραμέτρων απόκρισης.....	68
Εικόνα 16: Παρουσίαση παραμέτρων απόκρισης για δύο διαφορετικά αρχεία δεδομένων εισόδου.....	69
Εικόνα 17: Σχηματική απεικόνιση της διάταξης των πάνελ στο GUI.....	76
Εικόνα 18: Σύστημα μέτρησης χωρίς αντιστάθμιση .....	80
Εικόνα 19: Σύστημα μέτρησης με αντιστάθμιση.....	80
Εικόνα 20: Γραφική παράσταση απόκρισης χωρίς αντιστάθμιση.....	80
Εικόνα 21: Η αρχική (κόκκινη) και η εξομαλυσμένη (μπλε) καμπύλη απόκρισης, καθώς και η ευθεία $O_1 S_{max}$ (πράσινη) .....	81
Εικόνα 22: Χρόνος απόκρισης $T(t)$ .....	81
Εικόνα 23: Γραφική παράσταση απόκρισης με αντιστάθμιση.....	82
Εικόνα 24: Η αρχική (κόκκινη) και η εξομαλυσμένη (μπλε) καμπύλη απόκρισης, καθώς και η ευθεία $O_1 S_{max}$ (πράσινη) .....	83
Εικόνα 25: Χρόνος απόκρισης $T(t)$ .....	83
Εικόνα 26: Εξομάλυνση με 6 γειτονικά σημεία και 4 επαναλήψεις για απόκριση συστήματος χωρίς αντιστάθμιση.....	86
Εικόνα 27: Χρόνος απόκρισης συστήματος χωρίς αντιστάθμιση έπειτα από εξομάλυνση με 6 γειτονικά σημεία και 4 επαναλήψεις .....	86
Εικόνα 28: Εξομάλυνση με 8 γειτονικά σημεία και 4 επαναλήψεις για απόκριση συστήματος με αντιστάθμιση.....	87
Εικόνα 29: Χρόνος απόκρισης συστήματος με αντιστάθμιση έπειτα από εξομάλυνση με 8 γειτονικά σημεία και 4 επαναλήψεις .....	87





# 1. Εισαγωγή

## 1.1. Γενικά στοιχεία

Η ηλεκτρική ενέργεια είναι ένα αγαθό αναπόσπαστο από τον σημερινό τεχνολογικά ανεπτυγμένο κόσμο μας, καθώς για τις περισσότερες εφαρμογές του ηλεκτρισμού δεν υπάρχει άλλο υποκατάστατο. Γι' αυτό και η απρόσκοπτη παροχή του είναι πολύ σημαντική για κάθε τομέα, όπως η οικονομία, η δημόσια υγεία ή η εθνική ασφάλεια. Ακόμα και σύντομες διακοπές του ηλεκτρικού ρεύματος μπορούν να έχουν μεγάλο κόστος και να επιφέρουν ευρεία αναστάτωση, καθώς υπάρχει διαρκώς ζήτηση, της οποίας βέβαια το μέγεθός μπορεί να μεταβάλλεται κατά πολύ ακόμα και μέσα στην ίδια ημέρα. Είναι συνεπώς πρωταρχικής σημασίας η διασφάλιση της ικανότητας του συστήματος αφενός μεν να παράγει την απαιτούμενη ισχύ, αφετέρου δε να τη μεταφέρει χωρίς προβλήματα στους καταναλωτές.

Σχετικά με το πρώτο και με δεδομένο το γεγονός ότι δεν υπάρχει οικονομικά βιώσιμη λύση στο θέμα της αποθήκευσης μεγάλων ποσοτήτων ηλεκτρισμού, είναι απαραίτητη η τήρηση σημαντικών περιθωρίων εφεδρείας ισχύος προκειμένου να υπάρχει πάντοτε δυνατότητα κάλυψης του κυμαινόμενου φορτίου. Για παράδειγμα το σημερινό διασυνδεδεμένο σύστημα της ΔΕΗ έχει ένα μέσο φορτίο της τάξεως των 6000MW, αλλά διατηρεί μία εγκατεστημένη ισχύ 10700MW περίπου, προκειμένου να μπορεί να αντιμετωπίζει την καλοκαιρινή, λόγω κλιματιστικών, αιχμή των 9100MW που διαρκεί ελάχιστες, λιγότερες από 30, ώρες το χρόνο.

Όσον αφορά το δεύτερο, έχουν αναπτυχθεί και συνεχίζουν να ερευνώνται διάφοροι μηχανισμοί προστασίας των γραμμών μεταφοράς από διαταραχές, οι οποίες πολύ συχνά οφείλονται σε φυσικά φαινόμενα όπως ο κεραυνός, αλλά και σε εσωτερικές αιτίες όπως αναφέρεται στις επόμενες παραγράφους.

Ειδικά για την περίπτωση γραμμών υψηλής τάσης, η κεραυνοπληξία είναι ένα από τα πιο τακτικά προβλήματα, γι' αυτό και καταβάλλονται ιδιαίτερες προσπάθειες να μελετηθεί η επίδραση των κεραυνικών παλμών (ΚΠ) τόσο στις γραμμές μεταφοράς όσο και σε άλλες συνιστώσες ενός συστήματος μεταφοράς ηλεκτρικής ενέργειας, όπως οι πυλώνες ή οι μονωτήρες. Είναι συνεπώς αυτονόητο, ότι εργαζόμενοι για την

επίτευξη του στόχου της κατανόησης και της πρόβλεψης της συμπεριφοράς αυτών των στοιχείων όταν πληγούν από έναν ΚΠ, είναι απαραίτητο να μπορούμε να παράγουμε έναν τέτοιο παλμό. Και πηγαίνοντας ένα βήμα παραπέρα, θα πρέπει να είμαστε σε θέση να ελέγξουμε ότι ο ΚΠ που δημιουργήσαμε και ετοιμαζόμαστε να εφαρμόσουμε έχει τα χαρακτηριστικά που πρέπει και που θέλουμε, έτσι ώστε η δοκιμή που θα κάνουμε να έχει βαρύτητα και σημασία.

Εκτός όμως από τις ατμοσφαιρικές υπερτάσεις που οφείλονται στους κεραυνούς, τα στοιχεία του δικτύου κινδυνεύουν και από τις εσωτερικές υπερτάσεις που δημιουργούνται από εσωτερικές διαταραχές της ομαλής λειτουργίας του συστήματος. Οι εσωτερικές υπερτάσεις διακρίνονται στις υπερτάσεις με κανονική συχνότητα και μέσης συχνότητας (μεταβατικές). Οι υπερτάσεις μέσης συχνότητας εμφανίζονται κατά την απομόνωση μέσω διακόπτη μιας γραμμής μεγάλου μήκους και φτάνουν μέχρι και 500% της τάσης ομαλής λειτουργίας. Οι υπερτάσεις με κανονική συχνότητα εμφανίζονται είτε λόγω της γείωσης μίας φάσης σε τριφασικό δίκτυο με αποτέλεσμα να φτάνουν, στις άλλες δύο φάσεις, έως και το 40% της κανονικής τιμής, είτε λόγω της απότομης απώλειας φορτίου και φτάνουν μέχρι και 50% παραπάνω από την κανονική τάση. Κατά κανόνα οι υπερτάσεις (εσωτερικές, εξωτερικές) παρόλο που διαρκούν λίγο ασκούν στις μονώσεις των στοιχείων σοβαρότερες καταπονήσεις από τις τάσεις ομαλής λειτουργίας. Υπάρχουν όμως και περιπτώσεις όπου λόγω της συνεχούς εφαρμογής της, η τάση ομαλής λειτουργίας, αποτελεί πιο σοβαρή καταπόνηση από μία παροδική υπέρταση.

Η διασφάλιση της ποιότητας ενός συστήματος μέτρησης είναι συνεπώς πολύ σημαντικό κομμάτι για τις δοκιμές υψηλής τάσης, γι' αυτό και η Διεθνής Ηλεκτροτεχνική Επιτροπή (International Electrotechnical Commission, IEC) όρισε με το πρότυπο EN 60060-2 διάφορες διαδικασίες και νόρμες μέσω των οποίων μπορεί να αποδειχθεί αν ένα σύστημα μέτρησης πληροί τις απαραίτητες προϋποθέσεις ώστε να χαρακτηριστεί ως εγκεκριμένο. Περισσότερες λεπτομέρειες για το εν λόγω πρότυπο δίνονται στα επόμενα κεφάλαια της διπλωματικής αυτής εργασίας, ενώ ευθύς αμέσως ακολουθούν κάποιες γενικές πληροφορίες για τους κεραυνούς αλλά και για τον εξοπλισμό υψηλής τάσης που θέλουμε να προστατεύσουμε.

## **1.2. Βασικά στοιχεία για τους κεραυνούς**

Ο κεραυνός πήρε την θέση του ανάμεσα στα φυσικά φαινόμενα μόλις το 1752, καθώς μέχρι τότε αποτελούσε κάτι το ανεξήγητο, ένα σύμβολο μιας πανίσχυρης δύναμης. Πριν 250 περίπου χρόνια λοιπόν, κατά τη διάρκεια μιας καταιγίδας στην Πενσιλβάνια των ΗΠΑ, ο Βενιαμίν Φραγκλίνος πέταξε έναν αετό από μεταξωτό ύφασμα. Στο σπάγκο είχε δέσει ένα μεταλλικό κλειδί, το οποίο προσέλκυσε το αντίθετο προς το δικό του φορτίο των νεφών και έτσι δημιουργήθηκε ο πρώτος τεχνητός κεραυνός στην ιστορία. Με το πείραμα αυτό ο Φραγκλίνος απέδειξε ότι πίσω από τα καταιγιδοφόρα νέφη δεν κρύβεται η μήνις των θεών αλλά κάτι πολύ πιο απλό: ένα ηλεκτρικό φορτίο αντίθετο από εκείνο της Γης.

### **1.2.1. Είδη νεφών που γεννούν κεραυνούς**

Σύμφωνα με την διεθνή κατάταξη των νεφών [1] συνολικά διακρίνονται 10 είδη που είναι αμοιβαίως αποκλειόμενα και δεν μπορούν να συνυπάρχουν, τα οποία κατηγοριοποιούνται σε τρεις ομάδες, ανάλογα με το κανονικό ύψος της βάσης του νέφους πάνω από το έδαφος, όπως αναφέρει και ο Πίνακας 1. Τα ονόματά τους προέρχονται από τους λατινικούς όρους *altum* (ύψος), *cumulus* (σωρός) για νέφη κατακόρυφης ανάπτυξης, *stratus* (στρώμα) για στρωματικά νέφη, *cirrus* (ίνα) για ινώδη νέφη και *nimbus* για βροχοφόρα νέφη, μαζί με τα σύνθετά τους (π.χ. *cirrocumulus*, *cirrostratus*). Κεραυνούς γεννούν συνήθως τα χαμηλά νέφη *cumulonimbus* (σωρειτομελανίων) και *nimbostratus* (μελανοστρωμάτων). Τα πρώτα είναι “βαριά” και πυκνά νέφη με σημαντική κατακόρυφη ανάπτυξη με μορφή πελώριου πύργου. Το ανώτερο τμήμα τους συχνά εξαπλώνεται σε σχήμα άκμονα μέχρι το επίπεδο της τροπόπαυσης. Οι σωρειτομελανίες αναγνωρίζονται εύκολα καθώς η βάση τους είναι συνήθως πολύ σκούρα, αλλά και γιατί συνδέονται με βροχές και καταιγίδες. Επιπλέον, είναι τα μόνα σύννεφα που δημιουργούν χαλάζι, πράγμα που σημαίνει ότι μπορεί να είναι πολύ καταστροφικά. Ευτυχώς, συνήθως έχουν μικρή διάρκεια ζωής (γύρω στα 30 λεπτά). Από την άλλη, το μελανοστρώμα είναι σκούρο γκριζο νεφικό στρώμα που καλύπτει το σύνολο του ουρανού και είναι αρκετά παχύ ώστε να αποκρύπτει τον ήλιο. Ταυτόχρονα συνοδεύεται από εκτεταμένη βροχόπτωση η οποία περιορίζει την οριζόντια ορατότητα. Επιπλέον, το νέφος αυτό μπορεί να επιφέρει χιονόπτωση.

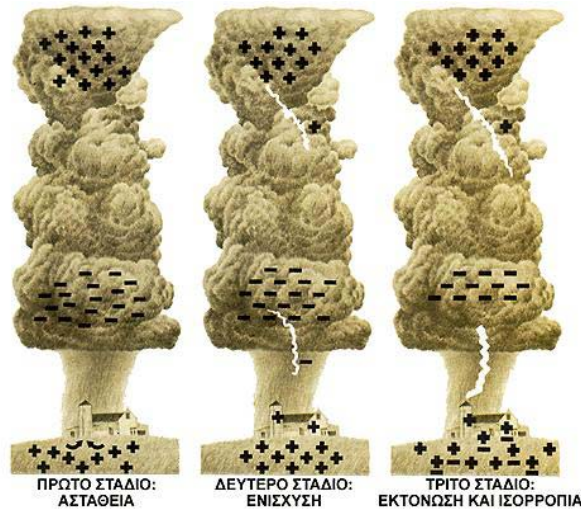
		Ύψος Βάσης Νέφους		
Είδος	Στρώμα	Πολικές περιοχές	Εύκρατες περιοχές	Τροπικές περιοχές
Cumulus	Χαμηλό	Κάτω των 2 km	Κάτω των 2 km	Κάτω των 2 km
Cumulonimbus				
Stratus				
Stratocumulus				
Nimbostratus				
Altostratus	Μεσαίο	2-4 km	2-7 km	2-8 km
Alto cumulus				
Cirrus	Υψηλό	3-8 km	5-13 km	6-18 km
Cirrostratus				
Cirrocumulus				

Πίνακας 1: Είδη νεφών

### 1.2.2. Δημιουργία ενός κεραυνού

Σύμφωνα με την Θεωρία του Ατμοσφαιρικού Ηλεκτρισμού την οποία διατύπωσε το 1920 ο Αμερικανός φυσικός Τσαρλς Ουίλσον, η Γη και η ηλεκτόσφαιρα, που είναι το κάτω τμήμα της ιονόσφαιρας, σε ύψος 50-60 χιλιομέτρων, αποτελούν τους δύο αντίθετους πόλους ενός σφαιρικού πυκνωτή τους οποίους χωρίζει η ατμόσφαιρα. Το σύστημα ισορροπεί καθώς ο αέρας λειτουργεί ως μονωτής, η διηλεκτρική αντοχή του οποίου εμποδίζει την ηλεκτρική εκφόρτιση και την εκδήλωση κεραυνών με αίθριο καιρό. Αν όμως δημιουργηθούν καταγιδοφόρα νέφη σωρειτομελανίων (cumulonimbus) ή σπανιότερα μελανοστρωμάτων (nimbostratus), η κατάσταση φορτίζεται επικίνδυνα, καθώς μέσα στα νέφη αυτά υπάρχει πολύπλοκη ηλεκτρική αστάθεια. Οι σωρειτομελανίες είναι κατ' ουσίαν τεράστιες μηχανές παραγωγής ηλεκτρικού φορτίου στο εσωτερικό των οποίων επικρατούν βίαια ρεύματα που μεταφέρουν φορτισμένες σταγόνες νερού και παγοκρυστάλλους. Στην κορυφή σχηματίζονται θετικά φορτία, ενώ στο κάτω μέρος συγκεντρώνονται τα αρνητικά και δημιουργούν μια έλξη με τα θετικά φορτία του εδάφους. Μέσα σε ελάχιστο χρόνο η διαφορά δυναμικού μεταξύ της βάσης του νέφους και του εδάφους φτάνει σε εκατομμύρια βολτ και ξεπερνά τη διηλεκτρική

αντοχή της ατμόσφαιρας. Ακριβώς εκείνη τη στιγμή ξεσπά ο κεραυνός, όπως συμβαίνει με όλες τις ηλεκτρικές εκκενώσεις μεταξύ αντίθετων ηλεκτρικών πεδίων, ώστε να αποκατασταθεί και πάλι η ηλεκτρική ισορροπία. Λόγω των υψηλών θερμοκρασιών που αναπτύσσονται κατά μήκος της διαδρομής του κεραυνού, ο αέρας θερμαίνεται και "εκρήγνυται", δημιουργώντας έτσι τη βροντή που τον συνοδεύει.



Εικόνα 1: Στάδια δημιουργίας ενός κεραυνού

### 1.2.3. Κατηγορίες κεραυνικών εκκενώσεων

Ανάλογα με τον τρόπο εκδήλωσης ενός κεραυνού, διακρίνουμε τις ακόλουθες τρεις κατηγορίες:

- i) Μεταξύ νέφους και γης ή αντίστροφα: οι εκκενώσεις αυτής της κατηγορίας είναι κατερχόμενες όταν το ηλεκτρικό πεδίο πάρει την κρίσιμη τιμή κοντά στο νέφος, ή ανερχόμενες αν αυτό συμβεί κοντά στη γη. Σαν πολικότητα του κεραυνού θεωρούμε αυτή του φορτίου του κάτω μέρους του νέφους που τον προκάλεσε, π.χ. ένας αρνητικός κεραυνός εκδηλώνεται μεταξύ γης και κάτω μέρους του νέφους και είναι η πιο συνηθισμένη περίπτωση για την Ελλάδα.
- ii) Εντός του ίδιου νέφους: αυτή είναι η πιο συνηθισμένη κατηγορία εκκενώσεων, η οποία εκδηλώνεται μέσα στο νέφος, μεταξύ του ανώτερου (θετικού) και του κατώτερου (αρνητικού) μέρους του και φαίνεται έξω από αυτό ως μια διάχυτη λάμψη που τρεμοσβήνει. Η διάρκειά της είναι μεγάλη και το ρεύμα που δημιουργείται έχει τιμές που φτάνουν τα εκατοντάδες Ampere.

- iii) Μεταξύ νεφών: οι κεραυνοί αυτής της κατηγορίας εκδηλώνονται σε μεγάλα ύψη, από 1 έως 12 χλμ και διακρίνονται από το μεγάλο μήκος κεραυνικού τόξου, το οποίο μπορεί να φτάσει τα 40 χλμ.

#### **1.2.4. Παράμετροι ενός κεραυνού**

Παρακάτω αναφέρονται οι τέσσερις βασικές παράμετροι που μας ενδιαφέρουν κατά την μελέτη των επιδράσεων ενός κεραυνού σαν ηλεκτρικό φαινόμενο:

- i) Η μέγιστη τιμή ρεύματος  $i_{\max}$  στην οποία οφείλονται οι υπερπηδήσεις μονωτήρων και οι καταστροφές στα μονωτικά υλικά, λόγω της ανύψωσης του δυναμικού στο σημείο που έπεσε ο κεραυνός.
- ii) Η μέγιστη κλίση του ρεύματος  $\left(\frac{di}{dt}\right)_{\max}$  από την οποία εξαρτώνται οι τάσεις που επάγονται σε βρόγχους κυκλωμάτων, σε λογικά κυκλώματα ή άλλα κυκλώματα που σχετίζονται π.χ. με συστήματα πλοήγησης ή τηλεπικοινωνιών.
- iii) Το μεταφερόμενο φορτίο  $\int i(t)dt$  στο οποίο οφείλεται η πιθανή τοπική τήξη και η διάτρηση μεταλλικών επιφανειών με μικρό πάχος
- iv) Το ολοκλήρωμα του τετραγώνου του ρεύματος  $\int i^2(t)dt$  το οποίο είναι ανάλογο της ενέργειας που εκλύεται από την κεραυνική εκκένωση και είναι υπεύθυνη για τυχόν ανεπιθύμητα θερμικά φαινόμενα όπως η έναυση εύφλεκτων ατμών ή αερίων, η τήξη μετάλλων κ.α.

### **1.3. Βασικά στοιχεία για τις γραμμές μεταφοράς**

Για τη μεταφορά της ηλεκτρικής ενέργειας από τους σταθμούς παραγωγής στα σημεία κατανάλωσης χρησιμοποιούνται οι ηλεκτρικές γραμμές μεταφοράς, το μήκος των οποίων μπορεί να φτάσει μέχρι μερικές εκατοντάδες χιλιόμετρα. Οι γραμμές μεταφοράς λειτουργούν υπό υψηλές τάσεις και είναι ιδιαίτερα επικίνδυνες για τον άνθρωπο. Για το λόγο αυτό είναι απαραίτητο να εγκαθίστανται κατά τρόπο ώστε να αποφεύγεται η προσέγγιση από τους ανθρώπους πλην των εξουσιοδοτημένων προσώπων τα οποία λαμβάνουν κατάλληλες προφυλάξεις.

Οι γραμμές μεταφοράς είναι είτε υπόγειες, είτε εναέριες. Οι υπόγειες γραμμές είναι πολύ δαπανηρές και για το λόγο αυτό χρησιμοποιούνται μόνο μέσα σε πόλεις δηλαδή για σχετικά μικρές αποστάσεις και χαμηλότερες τάσεις. Οι εναέριες γραμμές είναι αυτές που χρησιμοποιούνται για μεγάλες αποστάσεις και για τη μεταφορά πολύ υψηλών τάσεων. Οι αγωγοί των εναέριων γραμμών στηρίζονται σε στύλους ή πυλώνες οι οποίοι μπορεί να είναι ξύλινοι (γραμμές διανομής χαμηλής τάσης), από σκυρόδεμα (δίκτυα μέσης και χαμηλής τάσης) ή μεταλλικοί (για υψηλές τάσεις). Οι διαστάσεις των στύλων διαφέρουν ανάλογα με τη χρήση τους. Έτσι, οι πυλώνες πρέπει να είναι ψηλοί ώστε οι αγωγοί να απέχουν πολύ από το έδαφος αλλά και μεταξύ τους ώστε να αποφεύγεται η δημιουργία ηλεκτρικού τόξου και συνεπώς βραχυκυκλώματος. Οι στύλοι διακρίνονται ανάλογα με τη θέση τους στους κανονικούς που τοποθετούνται ενδιάμεσα σε ευθύγραμμα τμήματα, στους γωνιακούς που τοποθετούνται στα σημεία όπου αλλάζει κατεύθυνση η γραμμή και στους τερματικούς που μπαίνουν στα άκρα της γραμμής.

#### **1.4. Βασικά στοιχεία για τους μονωτήρες**

Οι αγωγοί των γραμμών μεταφοράς αναρτώνται στους πυλώνες μέσω των μονωτήρων αναρτήσεως, οι οποίοι παρέχουν και την απαιτούμενη μόνωση των αγωγών έναντι των μεταλλικών μερών των πυλώνων. Οι μονωτήρες πρέπει να έχουν μεγάλη ηλεκτρική αντοχή ώστε να εξασφαλίζουν τέλεια μόνωση μεταξύ αγωγού και πυλώνα και μηχανική αντοχή ώστε να αντέχουν το βάρος του αγωγού.

Στις γραμμές ηλεκτρικής ενέργειας χρησιμοποιούνται δύο τύποι μονωτήρων οι μονωτήρες στηρίξεως ή τύπου στελέχους και οι μονωτήρες αναρτήσεως ή δισκοειδείς. Οι πρώτοι χρησιμοποιούνται συνήθως στις γραμμές διανομής μέσης τάσης (6,6-15-20 KV) και φέρουν μεταλλικό στέλεχος στηρίξεως του μονωτήρα στο στύλο. Οι μονωτήρες αναρτήσεως χρησιμοποιούνται κυρίως σε γραμμές τάσεων άνω των 69000 V. Αποτελούνται από αριθμό ομοίων δισκοειδών μονωτήρων συνδεδεμένων εν σειρά. Ο αριθμός των δίσκων εξαρτάται από την τάση της γραμμής και την επιθυμητή στάθμη μόνωσης αυτής, δηλαδή την επιθυμητή κρουστική αντοχή αυτής.

Οι μονωτήρες κατασκευάζονται από πορσελάνη σε μεγέθη που σχετίζονται με την τάση της γραμμής και σχήμα που εμποδίζει την επικάθηση υγρασίας και σκόνης.

Σαν ηλεκτρική αντοχή του μονωτήρα ονομάζεται η ελάχιστη τάση που είναι απαραίτητη για να δημιουργηθεί ηλεκτρικός σπινθήρας με αποτέλεσμα να προκαλείται υπερπήδηση και τοπικό βραχυκύκλωμα της γραμμής. Η ηλεκτρική αντοχή του μονωτήρα μειώνεται σημαντικά από οποιαδήποτε ανωμαλία στην επιφάνειά του όπως ακαθαρσία, ράγισμα, υγρασία.



## **2. Δοκιμές με υψηλές τάσεις**

### **2.1. Συσκευές υψηλής τάσης**

Ως υψηλές τάσεις χαρακτηρίζονται, όλες οι τάσεις με ενεργό (ενδεικνύμενη) τιμή  $V_{en} > 1\text{KV}$  [2]. Οι συσκευές υψηλής τάσης είναι συσκευές σχεδιασμένες έτσι ώστε να λειτουργούν υπό υψηλή τάση χωρίς να καταστρέφονται ή να χάνουν μέρος των λειτουργιών τους. Προκειμένου να εξασφαλιστεί η αξιοπιστία μιας ηλεκτροτεχνικής κατασκευής υψηλής τάσης, πρέπει, πριν αυτή τεθεί σε χρήση να δοκιμαστεί σε λειτουργικές συνθήκες που αντιστοιχούν στις μελλοντικές μέγιστες καταπονήσεις της στο δίκτυο.

Οι δοκιμές αυτές που προβλέπονται από τους κανονισμούς (VDE, IEC, κ.λ.π.) διακρίνονται στις δοκιμές τύπου και σειράς. Οι δοκιμές τύπου γίνονται σε περιορισμένο αριθμό των παραχθέντων και έτοιμων προς διάθεση προϊόντων ενώ οι δοκιμές σειράς πραγματοποιούνται σε όλα ανεξαιρέτως τα τεμάχια. Όταν μία συσκευή υψηλής τάσης δεν ανταπεξέλθει επιτυχώς σε κάποια από τις προβλεπόμενες γι' αυτή δοκιμές τύπου και σειράς τότε κρίνεται ακατάλληλη και η ζημιά για τον κατασκευαστή μπορεί να είναι μεγάλη.

### **2.2. Παραδείγματα δοκιμών με υψηλές τάσεις**

Για να διαπιστωθεί η αντοχή μίας ηλεκτρικής εγκατάστασης σε υψηλές τάσεις, ορισμένης μορφής και τιμής, πρέπει να δοκιμασθεί στις τάσεις αυτές. Οι συνήθεις μορφές των υψηλών τάσεων δοκιμής είναι οι ακόλουθες:

- i) Εναλλασσόμενη ημιτονοειδής τάση, τόσο βιομηχανικής συχνότητας (50Hz), όσο και υψηλής συχνότητας, απαλλαγμένη όσο το δυνατόν περισσότερο από αρμονικές.
- ii) Συνεχής τάση, συνήθως ως προς γη, δηλαδή με προσγειωμένο τον έναν πόλο της γεννήτριας συνεχούς τάσης.
- iii) Κρουστική τάση, πλήρης ή και αποκομμένη, θετικής και αρνητικής πολικότητας ως προς γη.

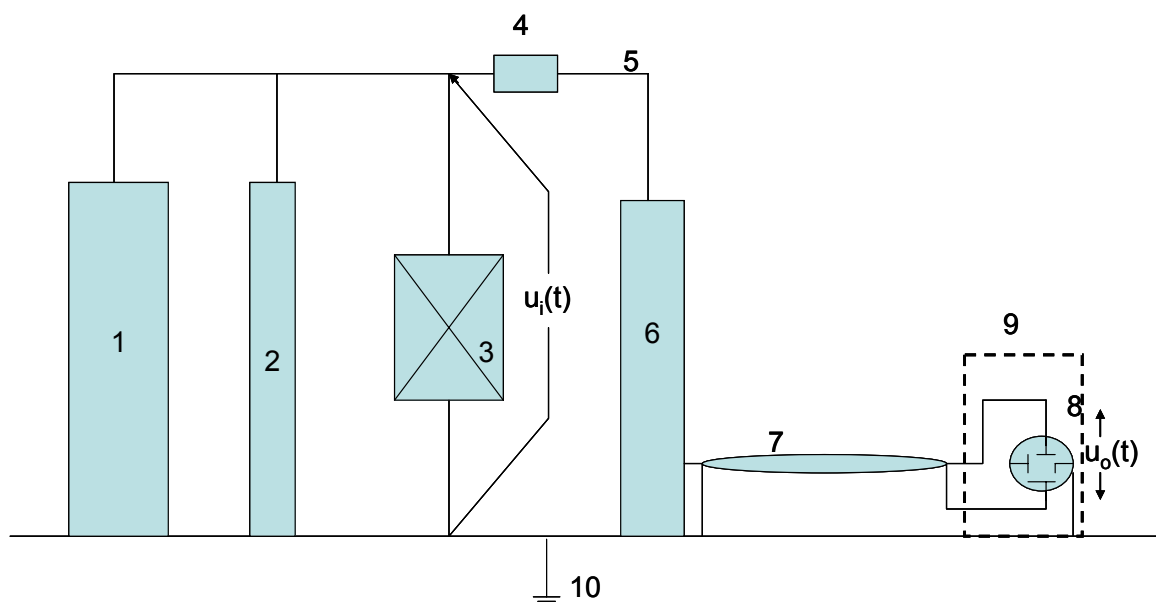
Οι ηλεκτρικές καταπονήσεις που εμφανίζονται στα δίκτυα από ατμοσφαιρικά ή εσωτερικά αίτια έχουν, κατά κανόνα, ανώμαλη μορφή ενώ ο έλεγχος των μονώσεων στο εργαστήριο γίνεται με τις ομαλές διπλοεκθετικές τάσεις που παράγονται από τις κρουστικές γεννήτριες. Επειδή σκοπός των δοκιμών είναι ο έλεγχος της ικανότητας της μονώσεως να αντέχει τις πραγματικές καταπονήσεις, είναι αναγκαία η γνώση ισοδυναμίας μεταξύ πραγματικών και εργαστηριακών καταπονήσεων. Η ουσία της ισοδυναμίας αυτής είναι πως οι εργαστηριακές και πραγματικές καταπονήσεις δημιουργούν για την εξεταζόμενη μόνωση τον ίδιο κίνδυνο διασπάσεως. Για τις ατμοσφαιρικές υπερτάσεις (κεραυνοί) έχει οριστεί σαν τάση δοκιμής η κρούση 1,2/50 μς γιατί η μέση τιμή πολλών καταγραφέντων ρευμάτων κεραυνού κατέληξε σ' αυτή περίπου τη μορφή. Είναι φανερό όμως πως, επειδή ο κάθε κεραυνός έχει, κατά κανόνα, διαφορετική μορφή από την τυποποιημένη τάση, η καταπόνηση που αντιπροσωπεύει θα διαφέρει από αυτή της τυποποιημένης κρούσεως.

Για τις υπερτάσεις χειρισμών η ισοδυναμία καθίσταται ακόμα πιο πολύπλοκη διότι εκτός από την ανωμαλία της μορφής των, απαιτείται μεγαλύτερη ακρίβεια στη σχεδίαση των μονώσεων. Με εμπειριστατωμένες δοκιμές έχουν, εν πάση περιπτώσει καθοριστεί ορισμένες βασικές αρχές της ισοδυναμίας αυτής. Συνήθως χρησιμοποιούνται κρουστικές τάσεις με διάρκεια μετώπου 250 μς και διάρκεια ουράς 2500μς, δηλαδή κρουστικές τάσεις της μορφής 250/2500 μς.

Για τη διεξαγωγή των εργαστηριακών δοκιμών σε συστήματα υψηλής τάσης χρησιμοποιείται μια διάταξη για την παραγωγή της υψηλής τάσης που εφαρμόζεται στο δοκίμιο και μία διάταξη για τον υποβιβασμό της υψηλής τάσης και την καταγραφή της (σύστημα μέτρησης υψηλής τάσης). Το σύστημα μέτρησης υψηλής τάσης αποτελείται συνήθως από έναν καταμεριστή τάσης (που υποβιβάζει την υψηλή τάση σε χαμηλή) και από ένα καταγραφικό όργανο χαμηλής τάσης (βολτόμετρο, παλμογράφος). Η διάταξη για την παραγωγή της υψηλής τάσης διαφέρει ανάλογα με τη μορφή της τάσης που θέλουμε να παράγουμε. Χρησιμοποιούνται πολυβάθμιοι μετασχηματιστές για την παραγωγή εναλλασσόμενων τάσεων, κρουστικές γεννήτριες για την παραγωγή κρουστικών τάσεων, πολυβάθμιοι ανορθωτές και ηλεκτροστατικές γεννήτριες για την παραγωγή συνεχούς τάσης.

Έτσι για δοκιμές με κρουστικές τάσεις χρησιμοποιείται η παρακάτω διάταξη που περιλαμβάνει

- Καταμεριστή υψηλής τάσης
- Καταγραφικό όργανο (παλμογράφο ή ψηφιοποιητή αναλογικών κυμάτων)
- Αγωγό υψηλής τάσης που συνδέει τον ακροδέκτη υψηλής τάσης του δοκιμίου με την είσοδο του καταμεριστή
- Ομοαξονικό καλώδιο που συνδέει την έξοδο του καταμεριστή με το καταγραφικό όργανο
- Αντίσταση απόσβεσης που τοποθετείται σε ένα από τα άκρα του αγωγού υψηλής τάσης



- |                               |                        |
|-------------------------------|------------------------|
| 1 Γεννήτρια κρουστικών τάσεων | 6 Καταμεριστής τάσης   |
| 2 Πυκνωτής φόρτισης           | 7 Ομοαξονικό καλώδιο   |
| 3 Δοκίμιο                     | 8 Καταγραφικό όργανο   |
| 4 Αντίσταση απόσβεσης         | 9 Θωρακισμένος θάλαμος |
| 5 Αγωγός υψηλής τάσης         | 10 Γείωση εργαστηρίου  |

**Εικόνα 2: Διάταξη δοκιμών υψηλών κρουστικών τάσεων**

Ακολουθούν παραδείγματα συσκευών υψηλής τάσης που υποβάλλονται σε διηλεκτρικές δοκιμές.

### **2.2.1. Μονωτήρες**

Οι μονωτήρες είναι υλικά που χρησιμοποιούνται για την απομόνωση των ηλεκτροφόρων στοιχείων μιας ηλεκτρικής εγκατάστασης από τα μη ηλεκτροφόρα. Οι εναέριες γραμμές μεταφοράς και διανομής ηλεκτρικής ενέργειας πρέπει να μονώνονται από τις κατασκευές στηριξής τους για λόγους ασφαλείας και αποφυγής βραχυκυκλωμάτων. Η μόνωση αυτή εξασφαλίζεται με τους μονωτήρες.

Οι μονωτήρες είναι δυνατόν να αστοχήσουν στον προορισμό τους εξαιτίας της διάτρησης (υπερνίκηση της διηλεκτρικής αντοχής τους), της υπερπήδησης (δημιουργία ηλεκτρικού τόξου που έρπει στην επιφάνεια του μονωτήρα), της διάσπασης του αέρα στη γειτονιά του μονωτήρα, της ρύπανσης που μειώνει κατά πολύ τη διηλεκτρική αντοχή στην επιφάνεια του μονωτήρα.

Για τον έλεγχο της διηλεκτρικής αντοχής των μονωτήρων, αυτοί υποβάλλονται σε δοκιμές με υψηλές τάσεις. Οι δοκιμές των μονωτήρων υπό κρουστικές τάσεις είναι δοκιμές τύπου (υποβάλλεται σε αυτές ένα μικρό τυχαίο δείγμα) και συνίστανται στον προσδιορισμό της κρουστικής χαρακτηριστικής τους (χαρακτηριστική τάσης-χρόνου). Κατά προτίμηση οι δοκιμές αυτές διεξάγονται υπό κρουστική τάση 1,2/50. Επιπλέον προσδιορίζονται η κρουστική τάση αντοχής (μέγιστη τιμή της κρουστικής τάσης κατά την οποία διακόπτονται οι υπερπηδήσεις στον μονωτήρα), η κρουστική τάση υπερπήδησης 50% (μέγιστη τιμή της κρουστικής τάσης κάτω από την οποία το 50% των κρούσεων στο μονωτήρα καταλήγουν σε υπερπήδηση), η κρουστική τάση υπερπήδησης 100% (μέγιστη τιμή της πιο χαμηλής κρουστικής τάσης κάτω από την οποία όλες οι κρούσεις καταλήγουν σε υπερπήδηση).

### **2.2.2. Καλώδια**

Τα καλώδια υψηλής τάσης δοκιμάζονται υπό υψηλές τάσεις για να διαπιστωθεί η αντοχή της μονώσεώς τους στις τάσεις αυτές. Ένα καλώδιο 19/33 KV δοκιμάζεται υπό κρουστική τάση ως εξής: σε κάθε μία από τις τρεις φάσεις επιβάλλεται κρουστική τάση  $V_k = 194 \text{ KV}$  (10 αρνητικές και 10 θετικές κρούσεις). Αν δεν σημειωθεί αστοχία, δηλαδή διάσπαση της μονώσεως, το καλώδιο κρίνεται ότι διαθέτει ικανοποιητική αντοχή σε κρουστική τάση.

### 3. Συστήματα μέτρησης υψηλών τάσεων

Κατά τη διεξαγωγή δοκιμών με υψηλές τάσεις είναι αναγκαία η ακριβής καταγραφή των μετρούμενων μεγεθών για να διαπιστωθεί η τήρηση των διεθνών κανονισμών, που καθορίζουν τη μορφή των ανωτέρω τάσεων, αλλά και να προσδιοριστεί με ακρίβεια η τιμή τους.

Εάν το σύστημα που διατίθεται για τη μέτρηση των υψηλών τάσεων εισάγει σημαντικά σφάλματα, είναι πιθανό οι δοκιμές να οδηγήσουν στην εξαγωγή λανθασμένων συμπερασμάτων όσον αφορά την ποιότητα των δοκιμών με σοβαρές επιπτώσεις στα δίκτυα στα οποία θα εγκατασταθούν. Ακόμα, μία υποτίμηση των χαρακτηριστικών ενός υλικού λόγω εσφαλμένων μετρήσεων, μπορεί να αποτελέσει κριτήριο ακαταλληλότητάς του, με σοβαρές οικονομικές συνέπειες για τον κατασκευαστή.

Ένα σύστημα μέτρησης υψηλών τάσεων θεωρείται απόλυτα ακριβές όταν μπορεί να υποβιβάσει την τάση δοκιμής  $V_i(t)$  κατά μέτρο χωρίς να την παραμορφώσει:

$$V_0(t) = V_i(t) / N$$

όπου  $V_0(t)$  είναι η μετρούμενη τάση στην έξοδο του συστήματος και  $N$  είναι ένας αριθμός που εκφράζει τη σχέση υποβιβασμού της υψηλής τάσης δοκιμής ώστε αυτή να είναι δυνατόν να καταγραφεί (λόγος καταμερισμού).

Για την εξακρίβωση της ικανότητας ενός συστήματος να καταγράφει όσο το δυνατόν πιστότερα το μετρούμενο μέγεθος, πρέπει αυτό να δοκιμαστεί υπό τους πιο δυσμενείς όρους. Ένα διεθνώς καθιερωμένο κριτήριο για μια τέτοια δοκιμασία είναι αυτό της βηματικής απόκρισης. Εξετάζεται δηλαδή η συμπεριφορά του συστήματος κατά την καταπόνησή του από μία βηματική τάση  $s(t)$ :

$$s(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

Η βηματική τάση είναι ένα εξαιρετικά γρήγορο φαινόμενο που περιέχει αρκετά υψίσυχνες συνιστώσες. Η απόκριση του συστήματος σε μία τάση αυτής της μορφής ονομάζεται βηματική απόκριση  $G(t)$ .

Σε ένα ιδανικό σύστημα (απόλυτα ακριβές) η βηματική απόκριση  $G(t)$  πρέπει να είναι ίδια με την βηματική τάση  $s(t)$  που εφαρμόζεται στην είσοδο, υποβιβασμένη κατά μέτρο σύμφωνα με το λόγο καταμερισμού  $N$ .

Επειδή το πλάτος της βηματικής απόκρισης  $G(t)$  είναι  $N$  φορές μικρότερο από αυτό της διέγερσης  $s(t)$ , είναι δυσδιάκριτη η απεικόνιση της  $G(t)$  σε σύγκριση με της  $s(t)$ . Για το λόγο αυτό χρησιμοποιείται το ανηγμένο μέγεθος της κανονικοποιημένης βηματικής απόκρισης  $g(t)$ :

$$g(t) = N \cdot G(t)$$

Και άρα ισχύει στη μόνιμη κατάσταση  $\lim_{t \rightarrow \infty} g(t) = s(t) = 1$ .

Το ιδανικό σύστημα υποβιβάζει την τάση χωρίς να την παραμορφώνει, επομένως η κανονικοποιημένη βηματική απόκριση  $g(t)$  ισούται με τη βηματική διέγερση  $s(t)$ :  $s(t) = g(t)$

Με δεδομένες τις  $s(t)$  και  $g(t)$  είναι εύκολο να εκτιμηθεί η ποιότητα του συστήματος μέτρησης. Όσο περισσότερο αποκλίνει η απόκριση  $g(t)$  από τη διέγερση  $s(t)$ , τόσο λιγότερο ικανό είναι το σύστημα να καταγράψει πιστά το μετρούμενο μέγεθος.

### **3.1. Μέτρηση υψηλών τάσεων**

Η μέτρηση υψηλών τάσεων οποιασδήποτε μορφής γίνεται σήμερα με τη βοήθεια ενός καταμεριστή τάσεως και ενός αισθητήριου οργάνου χαμηλής τάσης που μπορεί να είναι βολτόμετρο, παλμογράφος ή κάτι άλλο. Ένας καταμεριστής τάσεως είναι μία διάταξη που όταν συνδεθεί με το ένα άκρο της στην υπό μέτρηση τάση προκύπτει στο άλλο άκρο της μια χαμηλή τάση που έχει μία γνωστή και σταθερή αναλογία με τη μετρούμενη και που μπορεί να μετρηθεί με ένα συνηθισμένο όργανο χαμηλής τάσης.

Πριν γενικευθεί η χρήση των καταμεριστών, χρησιμοποιήθηκαν για τη μέτρηση των υψηλών τάσεων διάφορες μέθοδοι πολλές από τις οποίες χρησιμοποιούνται σποραδικά μέχρι και σήμερα. Αυτές είναι το ηλεκτροστατικό βολτόμετρο, διάκενο σφαιρών, αμπερόμετρο με αυτεπαγωγή, αμπερόμετρο με χωρητικότητα, μετασχηματιστές μετρήσεως.

### 3.1.1. Ηλεκτροστατικό βολτόμετρο

Είναι γνωστό πως όταν ένας αγωγός βρίσκεται σε ένα ηλεκτρικό πεδίο ασκείται πάνω του μια δύναμη. Τη δύναμη αυτή μπορούμε να εκτιμήσουμε από τη μεταβολή που υφίσταται η ηλεκτροστατική ενέργεια,  $W$ , του πεδίου για μια μικρή μετακίνηση  $dS$  του αγωγού και δίνεται από τη σχέση:

$$F = -\frac{\partial W}{\partial S}$$

Εάν το ηλεκτρικό πεδίο σχηματίζεται ανάμεσα στους δύο παράλληλους οπλισμούς ενός επίπεδου πυκνωτή ισχύει:

$$W = \frac{1}{2} \cdot C \cdot V^2$$

όπου  $C$  η χωρητικότητα του πυκνωτή και  $V$  η τάση ανάμεσα στους οπλισμούς του.

$$\text{Οπότε προκύπτει } F = -\frac{d}{dS} \left( \frac{1}{2} \cdot C \cdot V^2 \right) = -\frac{1}{2} V^2 \frac{dC}{dS}.$$

Και επειδή για έναν επίπεδο πυκνωτή με επιφάνεια οπλισμών  $A$  και διάκενο  $S$  είναι  $C = \epsilon_0 \cdot \frac{A}{S}$  και  $\frac{dC}{dS} = -\epsilon_0 \cdot \frac{A}{S^2}$ , προκύπτει

$$F = \frac{1}{2} \epsilon_0 V^2 \frac{A}{S^2}.$$

Αν είναι γνωστά τα  $A$  και  $S$  τότε μπορεί από τη μέτρηση της δύναμης  $F$  που ασκείται στους οπλισμούς του πυκνωτή να υπολογιστεί η τάση  $V$ .

Τα ηλεκτροστατικά βολτόμετρα μπορούν να χρησιμοποιηθούν μόνο για τη μέτρηση συνεχούς ή εναλλασσόμενης τάσης.

Το ηλεκτροστατικό βολτόμετρο στην Εικόνα 3 αποτελείται από δύο παράλληλα δισκοειδή ηλεκτρόδια. Στο ένα από τα δύο ηλεκτρόδια εφαρμόζεται η προς μέτρηση τάση ενώ το άλλο γειώνεται. Στο μέσο του γειωμένου ηλεκτροδίου είναι τοποθετημένος ένας μικρός δίσκος. Σύμφωνα με τα πιο πάνω η δύναμη που ασκείται πάνω στο μικρό δίσκο είναι ανάλογη του τετραγώνου της τάσης. Η δύναμη αυτή μετριέται από τη μετακίνηση του μικρού δίσκου που χρειάζεται για να εξισορροπηθούν η ηλεκτροστατική δύναμη και η δύναμη αντίδρασης που κρατά το δίσκο στη θέση του. Η μετακίνηση αυτή μετριέται από την απόκλιση μιας φωτεινής δέσμης πάνω σε ένα κάτοπτρο στερεωμένο στο κινητό σύστημα του μικρού δίσκου. Η περιοχή μετρήσεως του βολτομέτρου μεταβάλλεται με την απόσταση των δύο παράλληλων δίσκων. Η ακρίβεια του βολτομέτρου χειροτερεύει όσο μεγαλώνει η απόσταση ανάμεσα στους δύο δίσκους και αυτό είναι που περιορίζει τη μέγιστη τάση που μπορεί να μετρήσει ένα βολτόμετρο του τύπου αυτού.



**Εικόνα 3: Ηλεκτροστατικό βολτόμετρο**

### **3.1.2. Διάκενο σφαιρών**

Η μέτρηση μιας υψηλής τάσεως με τη βοήθεια ενός διακένου βασίζεται στο ότι ένα διάκενο με ορισμένο μήκος διασπάται σε ορισμένη τάση. Η τάση διασπάσεως όμως ενός διακένου δεν έχει απόλυτα καθορισμένη τιμή, παρουσιάζει στατιστική διασπορά που εξαρτάται από τη γεωμετρική μορφή του διακένου, τη μορφή της τάσεως κ.ά. Γι' αυτό για τη μέτρηση των τάσεων χρησιμοποιούνται διάκενα που



παρουσιάζουν τη μικρότερη δυνατή διασπορά. Έχει παρατηρηθεί πως το διάκενο που παρουσιάζει τη μικρότερη διασπορά στην τάση διασπάσεως για εναλλασσόμενη και κρουστική τάση είναι δύο σφαίρες της ίδιας διαμέτρου που η απόστασή τους είναι σχετικά μικρή σε σύγκριση με τη διάμετρό τους. Για συνεχή τάση έχει παρατηρηθεί πως μικρότερη διασπορά από τις σφαίρες παρουσιάζει το διάκενο ανάμεσα σε δύο ράβδους αλλά δεν έχει γίνει συστηματική μελέτη.

Η χρησιμοποίηση του διακένου δυο ίσων σφαιρών για μέτρηση υψηλών τάσεων έχει μελετηθεί εμπειριστατωμένα. Η ακρίβεια που παρουσιάζουν μετρήσεις με σφαίρες είναι της τάξεως του  $+3\%$  υπό την προϋπόθεση πως η απόστασή τους δεν υπερβαίνει την ακτίνα τους. Σφαιρικά διάκενα μετρήσεως μπορούν να κατασκευαστούν με τον άξονά τους οριζόντιο ή κατακόρυφο όπως φαίνεται στην Εικόνα 4 και την Εικόνα 5.



**Εικόνα 4: Κατακόρυφο διάκενο σφαιρών**



**Εικόνα 5: Οριζόντιο διάκενο σφαιρών**

Η διάσπαση ανάμεσα στις σφαίρες επιτυγχάνεται με ιονισμό από κρούσεις. Για να αρχίσει όμως ιονισμός, εκτός από το ηλεκτρικό πεδίο που προκύπτει από την

επιβαλλόμενη τάση, χρειάζονται μερικά αρχικά ηλεκτρόνια. Από φυσικό ιονισμό υπάρχουν στον αέρα πάντα ελεύθερα ηλεκτρόνια. Αυτό μπορεί να αυξήσει τη διασπορά της τάσεως διασπάσεως. Για να προληφθεί αυτό το μειονέκτημα συνιστάται για τα μικρά διάκενα, τεχνητή δημιουργία ελευθέρων ηλεκτρονίων με ακτινοβολία. Σαν πηγή ακτινοβολίας μπορεί να χρησιμοποιηθεί είτε λυχνία υπεριωδών ακτινών ή μια παστίλια ραδιενεργού ουσίας εντάσεως 0,2-06 millicurie.

Η κατάσταση της επιφάνειας των σφαιρών επηρεάζει επίσης την τάση διασπάσεως. Γι' αυτό η επιφάνεια των σφαιρών, κυρίως στην περιοχή που πλησιάζουν περισσότερο και στην οποία συμβαίνουν οι διασπάσεις πρέπει να διατηρείται καθαρή. Και κόκκοι σκόνης ακόμα επηρεάζουν την τάση διασπάσεως.

Μια άλλη παράμετρος που επηρεάζει την τάση διασπάσεως του διακένου είναι η πυκνότητα του αέρα και η ποσότητα υδρατμών που περιέχει. Η τάση διασπάσεως γίνεται μεγαλύτερη όσο μεγαλύτερη είναι η πυκνότητα του αέρα και όσο περισσότερους υδρατμούς περιέχει.

### **3.1.3. Αμπερόμετρο με αυτεπαγωγή για τη μέτρηση εναλλασσόμενης τάσης**

Ένα αμπερόμετρο με μια μεγάλη αυτεπαγωγή στη σειρά μπορεί να χρησιμοποιηθεί για την μέτρηση υψηλών εναλλασσόμενων τάσεων. Η δυσκολία που παρουσιάζει η μέθοδος είναι πως για να γίνει η αυτεπαγωγή αρκετά μεγάλη, ιδίως για σχετικά χαμηλές συχνότητες, πρέπει να έχει πυρήνα και αυτό κάνει την κατασκευή της δαπανηρή. Μια άλλη δυσκολία προκύπτει από την παράσιτη χωρητικότητα που παρουσιάζει το τύλιγμα της αυτεπαγωγής. Γι' αυτό η μέθοδος αυτή σπάνια χρησιμοποιείται.

### **3.1.4. Αμπερόμετρο με χωρητικότητα για τη μέτρηση εναλλασσόμενης τάσης**

Εάν ένας πυκνωτής χωρητικότητας  $C$  συνδεθεί στην υπό μέτρηση ημιτονοειδή τάση με κυκλική συχνότητα  $\omega$ , το ρεύμα που τον διαρρέει είναι:  $I = j\omega CV$ . Μετρώντας τη μέση τιμή του ρεύματος που ρέει κατά τη διάρκεια μιας ημιπεριόδου και γνωρίζοντας την τιμή  $C$  είναι δυνατόν να καθοριστεί η τιμή του  $V$ .

### **3.1.5. Μετασχηματιστές μετρήσεως**

Για τη μέτρηση υψηλών εναλλασσόμενων τάσεων μπορούν να χρησιμοποιηθούν μετασχηματιστές τάσεως όμοιοι με αυτούς που χρησιμοποιούνται

στα δίκτυα για μετρήσεις. Η μόνη διαφορά ανάμεσα σε ένα μετασχηματιστή ισχύος και ένα μετασχηματιστή μετρήσεως έγκειται στην έμφαση που δίνεται στα ιδιαίτερα χαρακτηριστικά του καθενός εξ αιτίας του ειδικού προορισμού τους. Σε ένα μετασχηματιστή τάσεως, η σχέση ανάμεσα στις τάσεις πρωτεύοντος και δευτερεύοντος πρέπει να είναι όσο γίνεται καλύτερα καθορισμένη και σταθερή. Σε ένα ιδανικό μετασχηματιστή μετρήσεως της τάσεως οι τάσεις στο πρωτεύον και το δευτερεύον πρέπει να έχουν ακριβώς την ονομαστική σχέση που αναφέρεται στην πινακίδα του μετασχηματιστή και να έχουν διαφορά φάσεως ακριβώς 180°. Εξ' αιτίας των απωλειών σιδήρου και χαλκού και τη ροή σκέδασης ανάμεσα στο πρωτεύον και το δευτερεύον, η σχέση ανάμεσα στις τάσεις πρωτεύοντος και δευτερεύοντος αποκλίνει από την ιδανική κατά ένα βαθμό που εξαρτάται από το φορτίο του μετασχηματιστή.

Το φορτίο ενός μετασχηματιστή μετρήσεως ονομάζεται «burden» για διάκριση από τον όρο «load» που χρησιμοποιείται για ένα μετασχηματιστή ισχύος.

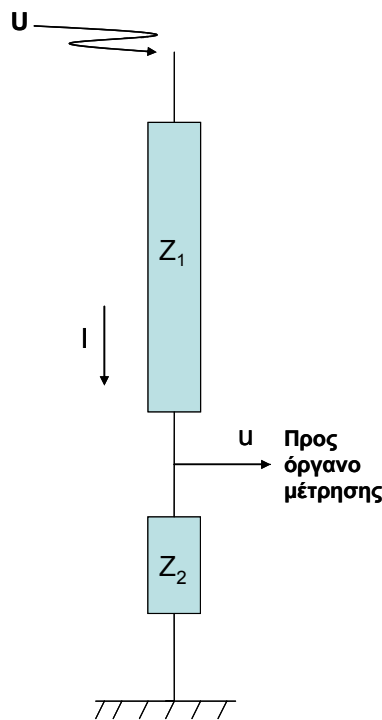
Για την μέτρηση τάσεων, εκείνο που ενδιαφέρει είναι η ακρίβεια της σχέσεως μεταφοράς μόνο. Το σφάλμα γωνίας ανάμεσα στις τάσεις πρωτεύοντος και δευτερεύοντος έχει σημασία όταν ο μετασχηματιστής χρησιμοποιείται για την τροφοδότηση του πηνίου τάσεως ενός βολτόμετρου για μέτρηση ισχύος.

Οι αιτίες για τις οποίες η σχέση μεταφοράς και η σχέση γωνίας ενός μετασχηματιστή μετρήσεως αποκλίνουν από τις ιδανικές γίνονται φανερές αν εξεταστεί το διανυσματικό διάγραμμα λειτουργίας του μετασχηματιστή. Η σχέση ανάμεσα στο εύρος και την γωνία των τάσεων στο πρωτεύον και το δευτερεύον μεταβάλλεται όταν το φορτίο του μετασχηματιστή αλλάξει. Για να μειωθεί η διακύμανση της σχέσεως τάσεως και γωνίας, το φορτίο των μετασχηματιστών μετρήσεως προβλέπεται από δύο μέχρι τριάντα φορές μικρότερο από αυτό που επιβάλλουν λόγοι υπερθερμάνσεως του μετασχηματιστή. Η σχέση μεγέθους και γωνίας ανάμεσα στις τάσεις πρωτεύοντος και δευτερεύοντος εξαρτάται όχι μόνο από την ένταση ρεύματος που απορροφά το φορτίο, αλλά και από τον συντελεστή ισχύος.

### 3.1.6. Καταμεριστές τάσεως

Οι καταμεριστές τάσεως αποτελούν σήμερα την σχεδόν γενικά χρησιμοποιούμενη μέθοδο για την μέτρηση υψηλών τάσεων. Βασικά ένας καταμεριστής είναι μια πολύ απλή συσκευή. Αποτελείται, όπως δείχνει η Εικόνα 6, από δύο αντιδράσεις  $Z_1$ ,  $Z_2$ , στη σειρά από τις οποίες η  $Z_1$  είναι πολύ μεγαλύτερη από την  $Z_2$  και ονομάζεται «βραχίονας υψηλής τάσεως» ενώ η  $Z_2$  ονομάζεται «βραχίονας χαμηλής τάσεως». Η υψηλή τάση,  $U$ , εφαρμόζεται στο πάνω άκρο της  $Z_1$  ενώ το κάτω άκρο της  $Z_2$  συνδέεται με τη γη. Το ρεύμα  $I$ , που διαρρέει τις  $Z_1$  και  $Z_2$  από την επίδραση της  $U$  είναι  $I=U/(Z_1+Z_2)$ . Αν  $u$  είναι η τάση που αναπτύσσεται στο μη γειωμένο άκρο της  $Z_2$  θα είναι

$$u = I \cdot Z_2 = U \frac{Z_2}{Z_1 + Z_2}$$



Εικόνα 6: Καταμεριστής τάσεως

Ο λόγος  $H=Z_2/(Z_1+Z_2)$  ονομάζεται «σχέση μεταφοράς» του καταμεριστή και επιλέγεται ώστε η τάση  $u$  να είναι αρκετά χαμηλή για να μπορεί να μετρηθεί από ένα όργανο χαμηλής τάσεως. Επειδή συνήθως  $Z_1 \gg Z_2$ , η σχέση μεταφοράς του καταμεριστή είναι με μεγάλη προσέγγιση ίση με το λόγο  $Z_2/Z_1$ . Γνωρίζοντας το  $H$  και μετρώντας το  $u$  θα είναι  $U=(1/H)u$ . Τα όργανα χαμηλής τάσεως που χρησιμοποιούνται

για τη μέτρηση εξαρτώνται από τη μορφή της τάσεως. Για συνεχή ή εναλλασσόμενη τάση αρκεί ένα κοινό βολτόμετρο αναλογικό ή ψηφιακό. Για κρουστικές τάσεις χρησιμοποιούνται παλμογράφοι ή κρουστικά βολτόμετρα. Τα κρουστικά βολτόμετρα δίνουν μόνο το εύρος και όχι ολόκληρο το σχήμα της τάσεως όπως οι παλμογράφοι αλλά είναι πολύ πιο εύχρηστα και συνήθως μεγαλύτερης ακρίβειας.

Παρ' όλη τους την απλότητα οι καταμεριστές τάσεως παρουσιάζουν σοβαρά προβλήματα ιδιαίτερα όταν χρησιμοποιούνται για την μέτρηση εναλλασσόμενων τάσεων μεγάλης συχνότητας ή κρουστικών τάσεων.

### **3.1.7. Καταμεριστές για συνεχή τάση**

Για τη μέτρηση συνεχών τάσεων οι αντιδράσεις  $Z_1$  και  $Z_2$  είναι απλές αντιστάσεις  $R_1$  και  $R_2$ . Οι πηγές συνεχούς τάσεως έχουν συνήθως περιορισμένη ισχύ και κατά συνέπεια η τάση τους επηρεάζεται από το ρεύμα  $I$  που τροφοδοτούν. Γι' αυτό η αντίσταση του καταμεριστή πρέπει να είναι αρκετά μεγαλύτερη ώστε το ρεύμα  $I$  να μην ξεπερνάει το 1 mA. Αν πρόκειται π.χ. να μετρηθεί μια τάση 1000 KV, που είναι ένα συνηθισμένο μέγεθος, η αντίσταση του καταμεριστή πρέπει να είναι

$$R_1 > (1000KV/1mA) = 1000M\Omega$$

Κατά την κατασκευή της αντιστάσεως  $R_1$  του βραχίονα υψηλής τάσης πρέπει ακόμα να λαμβάνεται πρόνοια ώστε η διαμήκης πτώση τάσεως να μη ξεπερνάει μια ορισμένη τιμή γιατί αλλιώς μπορεί να συμβούν βραχυκυκλώματα τμημάτων ή και ολόκληρου του καταμεριστή. Αν η αντίσταση βρίσκεται εκτεθειμένη στον αέρα, η διαμήκης πτώση τάσεως δεν πρέπει να υπερβαίνει τα 4-5 KV/cm. Αν είναι εμβαπτισμένη στο λάδι, μεγαλύτερη διαμήκης πτώση τάσεως είναι ανεκτή. Το μήκος της αντιστάσεως γίνεται κατά κανόνα σημαντικά μεγαλύτερο από αυτό που υπαγορεύει ο πιο πάνω περιορισμός γιατί η αντίσταση πρέπει να είναι απόλυτα απαλλαγμένη από τοπικές εκκενώσεις κορόνα.

Το είδος της χρησιμοποιούμενης αντιστάσεως πρέπει να εξασφαλίζει σταθερότητα τιμής ανεξάρτητης από το μέγεθος της μετρούμενης τάσεως (γραμμικότητα) και σταθερότητα με το χρόνο. Επίσης η τιμή της πρέπει να επηρεάζεται όσο γίνεται λιγότερο από τη θερμοκρασία (μικρός συντελεστής θερμοκρασίας).

Για την κατασκευή αντιστάσεων μεγάλης σταθερότητας χρησιμοποιήθηκαν αρχικά λεπτά σύρματα από κράματα μεγάλης ειδικής αντιστάσεως. Για το σκοπό αυτό χρησιμοποιούνται σύρματα με διάμετρο 0,1 mm ή ακόμα λεπτότερα. Είναι φανερό πως η κατασκευή μιας τέτοιας αντίστασης είναι μια αρκετά λεπτή δουλειά.

Επειδή οι ωμικοί καταμεριστές δεν είναι απαλλαγμένοι από αυτεπαγωγή, απότομη μεταβολή του ρεύματος που διαρρέει τον καταμεριστή, όπως συμβαίνει στην περίπτωση ενός βραχυκυκλώματος, μπορεί να δημιουργήσει μεγάλες υπερτάσεις  $L di/dt$ . Αυτός ο κίνδυνος προλαμβάνεται με ένα σχετικά μεγάλο ηλεκτρόδιο στο ανώτερο μέρος του καταμεριστή. Η χωρητικότητα του ηλεκτροδίου αυτού με τη γη απορροφάει στιγμιαία ένα μέρος από την ενέργεια  $\frac{1}{2} (LI^2)$  που είναι αποθηκευμένη στην αυτεπαγωγή και έτσι η υπέρταση μετριάζεται.

### 3.1.8. Καταμεριστές για ημιτονοειδή τάση

Αν οι μιγαδικές αντιδράσεις  $Z_1$  και  $Z_2$  στην Εικόνα 6 έχουν τις τιμές

$$Z_1 = R_1 + jX_1$$

$$Z_2 = R_2 + jX_2$$

η σχέση μεταφοράς του καταμεριστή που ορίστηκε σαν το πηλίκο

$$H = Z_2 / (Z_1 + Z_2)$$

παίρνει για εναλλασσόμενη τάση την τιμή

$$H = \frac{\sqrt{R_2^2 + X_2^2}}{\sqrt{(R_1 + R_2)^2 + (X_1 + X_2)^2}} \left| \arctan \frac{X_2}{R_2} - \arctan \frac{X_1 + X_2}{R_1 + R_2} \right|$$

Για να μετράει ο καταμεριστής πιστά τόσο το μέτρο όσο και τη γωνία της υπό μέτρηση τάσεως  $U$  θα πρέπει η σχέση μεταφοράς να είναι πραγματικός αριθμός που συμβαίνει μόνο όταν,  $(X_1 + X_2) / (R_1 + R_2) = X_2 / R_2$  ή  $X_1 / R_1 = X_2 / R_2$

Δηλαδή, για πιστή μέτρηση της εναλλασσόμενης τάσεως πρέπει ο λόγος της αντίδρασης προς την αντίσταση να είναι και για τους δύο βραχίονες του καταμεριστή ο ίδιος.

### 3.1.9. Μέτρηση ημιτονοειδούς τάσης με χωρητικούς καταμεριστές

Οι ωμικοί καταμεριστές, εξ αιτίας της παράσιτης χωρητικότητας, περιορίζονται σε σχετικά μικρές συχνότητες. Αν αντί για αντιστάσεις οι δύο βραχίονες του καταμεριστού είναι πυκνωτές ο περιορισμός αυτός δεν υπάρχει. Χωρητικοί καταμεριστές μπορούν να χρησιμοποιηθούν για πολύ μεγάλες συχνότητες και τα όρια τους περιορίζονται μόνο από την παράσιτη αυτεπαγωγή των πυκνωτών και τις απώλειες υστερήσεως μέσα στα διηλεκτρικά τους.

Ο πυκνωτής  $C_2$  που αποτελεί το βραχίονα χαμηλής τάσεως πρέπει να έχει μεγάλη σταθερότητα και μικρή γωνία απωλειών. Συνήθως χρησιμοποιούνται πυκνωτές αέρος, μίκας ή συνθετικού διηλεκτρικού.

Πυκνωτές χαρτιού-λαδιού παρουσιάζουν συνήθως μεγάλη γωνία απωλειών και μπορούν να χρησιμοποιηθούν μόνο για μικρές σχετικά συχνότητες. Για να αποφεύγεται η μόνιμη εναπόθεση φορτίου στους οπλισμούς του  $C_2$ , συνδέεται παράλληλα μ' αυτόν μια πολύ μεγάλη αντίσταση  $R_2$  η οποία μπορεί να είναι ακόμα και αυτή η εσωτερική απώλεια του πυκνωτή ή του οργάνου μετρήσεως.

### 3.1.10. Καταμεριστές για μέτρηση κρουστικών τάσεων

Μία κρουστική τάση είναι μία μη περιοδική συνάρτηση του χρόνου  $U(t)$  που μπορεί να εκφραστεί με το ζεύγος ολοκληρωμάτων Fourier

$$U(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} U(\omega) e^{i\omega t} d\omega$$

$$U(\omega) = \int_{-\infty}^{\infty} U(t) e^{-i\omega t} dt$$

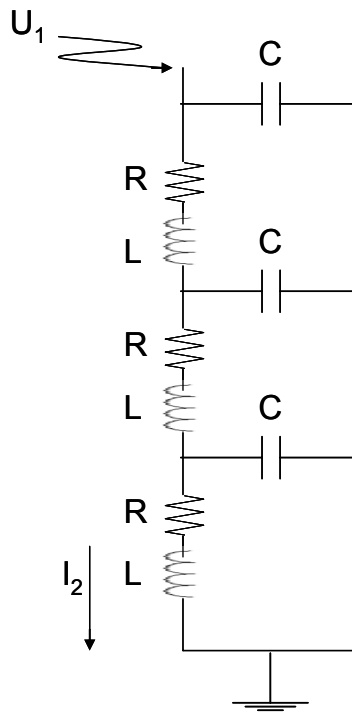
όπου  $U(\omega)$  παριστάνει το εύρος της συνιστώσας με συχνότητα  $\omega$ . Επειδή το φάσμα των συχνοτήτων  $\omega$  εκτείνεται μέχρι τιμές άπειρα μεγάλες η τάση  $U(t)$  θα περιέχει γενικά όλες αυτές τις συχνότητες. Για την πιστή μέτρηση μιας τέτοιας τάσεως θα πρέπει λοιπόν ο καταμεριστής που θα χρησιμοποιηθεί να είναι ικανός να μεταφέρει όσο γίνεται μεγαλύτερο μέρος αυτού του φάσματος χωρίς σημαντική αλλαγή εύρους και φάσεως.

Στην πραγματικότητα, η προς μέτρηση τάση δεν συνδέεται απ' ευθείας στην κορυφή του καταμεριστή αλλά μεσολαβούν συνδετικοί αγωγοί. Επίσης, για τη σύνδεση του βραχίονα χαμηλής τάσης με το όργανο μέτρησης χρησιμοποιείται ένα συγκεντρικό καλώδιο. Τα πρόσθετα αυτά εξαρτήματα της διατάξεως μετρήσεως προσθέτουν στη δυσκολία της μεταφοράς πολύ υψηλών συχνοτήτων από τη θέση μετρήσεως στο όργανο μετρήσεως. Για το λόγο αυτό όταν εξετάζονται τα χαρακτηριστικά μεταφοράς ενός καταμεριστή, πρέπει να θεωρούνται οι αγωγοί συνδέσεως ενσωματωμένοι σ' αυτόν. Ολόκληρη η διάταξη μετρήσεως θεωρείται έτσι σαν ένα τετράπολο. Το σημείο Α στο οποίο εφαρμόζεται η υψηλή τάση, μαζί με το σημείο Β της γης αποτελούν την είσοδο του τετραπόλου στην οποία εφαρμόζεται η προς μέτρηση τάση  $U$ . Τα σημεία Γ και Δ που συνδέονται στο όργανο μετρήσεως, από τα οποία το Δ είναι επίσης η γη, αποτελούν την έξοδο του τετραπόλου. Μελετώντας τα χαρακτηριστικά μεταφοράς του τετραπόλου αυτού μπορούμε ακόμα να θεωρήσουμε πως η θεωρητική σχέση μεταφοράς του καταμεριστή (που μετριέται με μια συνεχή ή εναλλασσόμενη τάση χαμηλής συχνότητας) είναι ίση με «ένα», όπως γίνεται συχνά και για την μελέτη του ηλεκτρικού ισοδυνάμου ενός μετασχηματιστή

### **3.1.11. Ωμικός καταμεριστής για μέτρηση κρουστικών τάσεων**

Ένας ωμικός καταμεριστής παρουσιάζει, εκτός από την αντίστασή του, παράσιτες χωρητικότητες και αυτεπαγωγές. Για μια, κατά προσέγγιση, θεωρητική διερεύνηση της ικανότητας ενός τέτοιου καταμεριστή να μεταφέρει τις υψηλές συχνότητες που περιέχονται σε μια κρουστική τάση, μπορεί αυτός να εξομοιωθεί με μια ομογενή γραμμή με το ένα άκρο της γειωμένο.

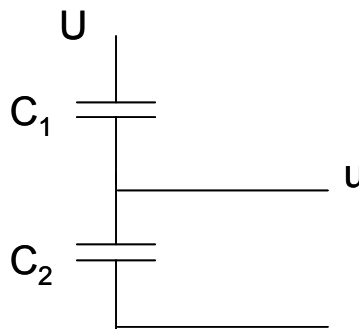




Εικόνα 7: Γραμμή μεγάλου μήκους

### 3.1.12. Χωρητικός καταμεριστής

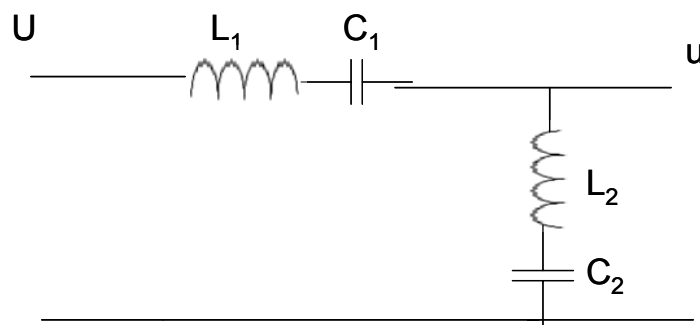
Ένας χωρητικός καταμεριστής αποτελείται από δύο χωρητικότητες στη σειρά  $C_1$  και  $C_2$  που αποτελούν αντίστοιχα τους βραχίονες υψηλής και χαμηλής τάσεως. Θεωρητικά, η σχέση μεταφοράς  $H=U/u$  ενός χωρητικού καταμεριστή είναι ανεξάρτητη της συχνότητας και ίση προς  $H= C_1/(C_1+C_2)$ .



Εικόνα 8: Χωρητικός καταμεριστής

Τόσο όμως αυτοί οι ίδιοι πυκνωτές όσο και οι αγωγοί συνδέσεως του ανώτερου αλλά και του κατώτερου άκρου του καταμεριστή περιέχουν αυτεπαγωγές που μαζί με την χωρητικότητα του καταμεριστή αποτελούν για τις υψηλότερες συχνότητες ταλαντούμενα κυκλώματα. Εκτός απ' αυτό, σε καταμεριστές πολύ υψηλής τάσεως, ο πυκνωτής υψηλής τάσεως θα πρέπει εξ αιτίας του μεγέθους του, να θεωρείται μάλλον

σαν μια γραμμή μεταφοράς με σχεδόν μηδενική απόσβεση παρά σαν μια συγκεντρωμένη χωρητικότητα. Αυτό έχει σαν συνέπεια πολλαπλές ανακλάσεις στα δυο άκρα του πυκνωτή που εμφανίζονται επίσης σαν ταλαντώσεις και δεν μπορούν να αποσβεστούν με μια συγκεντρωμένη αντίσταση αποσβέσεως συνδεδεμένη στο άκρο υψηλής τάσεως του καταμεριστή. Για την απόσβεση των ταλαντώσεων του τελευταίου τύπου η αντίσταση αποσβέσεως πρέπει να κατανέμεται κατά το δυνατόν ομοιόμορφα μέσα στον πυκνωτή υψηλής τάσεως και πάνω σ' αυτή την ιδέα στηρίζεται η κατασκευή του μικτού χωρητικού καταμεριστή με απόσβεση σειράς που φαίνεται στην Εικόνα 9.



Εικόνα 9: Μικτός χωρητικός καταμεριστής

### 3.1.13. Χωρητικός καταμεριστής με απόσβεση

Ο βραχίονας υψηλής τάσεως ενός χωρητικού καταμεριστή συμπεριφέρεται σαν ένα τμήμα γραμμής μεταφοράς με βραχυκυκλωμένο το πέρας της και πολύ μικρή απόσβεση. Αν στο άνω άκρο του καταμεριστή επιβληθεί βαθμίδα τάσεως, το μέτωπό της, με ανεστραμμένη πολικότητα, θα επιστρέψει μετά από ανάκλαση στο βραχυκυκλωμένο άκρο μετά χρόνο  $2\tau$  όπου  $\tau$  ο χρόνος απλής διαδρομής του καταμεριστή. Έτσι η βασική συχνότητα ταλαντώσεων που θα προκύψει εξ αιτίας της ανακλάσεως αυτής θα είναι  $F=1/2\tau$ .

Όπως είναι γνωστό, η ταχύτητα μεταδόσεως ενός κύματος πάνω σε μια γραμμή χωρίς απόσβεση με διαμήκη αυτεπαγωγή  $L'$  ανά μονάδα μήκους και εγκάρσια χωρητικότητα  $C'$  ανά μονάδα μήκους είναι  $u = \frac{1}{\sqrt{L'C'}}$ . Αν  $L$  είναι η συνολική αυτεπαγωγή του καταμεριστή,  $C_g$  η παράσιτη χωρητικότητα προς γη και  $l$  το μήκος του, θα είναι  $L' = L/l$  και  $C' = C_g/l$  οπότε  $u = \frac{l}{\sqrt{LC_g}}$ . Ο χρόνος απλής διαδρομής του

καταμεριστή θα είναι λοιπόν  $\tau = \frac{l}{u} = \frac{l}{\frac{l}{\sqrt{LC_g}}} = \sqrt{LC_g}$ , και η θεμελιώδης συχνότητα των

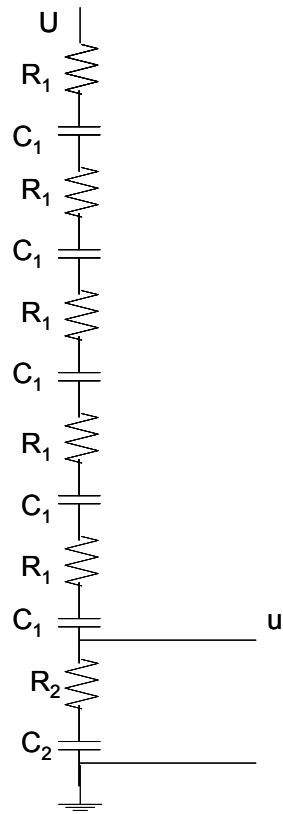
ταλαντώσεων από ανακλάσεις θα είναι  $f = \frac{1}{2\sqrt{LC_g}}$ .

Για την απόσβεση των ταλαντώσεων αυτών εισάγονται σε σειρά με τις χωρητικότητες του βραχίονα υψηλής τάσεως του καταμεριστή αντιστάσεις όπως δείχνεται στην Εικόνα 10. Οι αντιστάσεις αυτές πρέπει να έχουν αρκετά μεγάλη τιμή ώστε ένα κύμα τάσεως επιβαλλόμενο στην κορυφή του καταμεριστού να έχει σχεδόν αποσβεσθεί όταν φτάσει στο προς τη γη κατώτερο άκρο του. Αποδεικνύεται θεωρητικά και πειραματικά πως αυτό συμβαίνει για  $R = (3 \div 4) \sqrt{\frac{L}{C_g}}$  όπου  $R = \Sigma R_i$  η ολική τιμή της αντιστάσεως,  $L$  η ολική διαμήκης αυτεπαγωγή και  $C_g$  η ολική εγκάρσια (παράσιτη) χωρητικότητα. Με άλλα λόγια, επειδή  $\sqrt{\frac{L}{C_g}}$  παριστάνει την κυματική αντίσταση του βραχίονα υψηλής τάσεως, η συνολική τιμή της εισαγόμενης αντιστάσεως πρέπει να είναι ίση με 3 έως 4 φορές την κυματική αντίσταση.

Ο αποσβεσμένος χωρητικός καταμεριστής θεωρείται σήμερα ο πιο κατάλληλος για την μέτρηση τάσεων που μεταβάλλονται πολύ απότομα όπως είναι μια τάση με απότομο μέτωπο ή αποκεκομένη.

### **3.1.14. Οι βραχίονες χαμηλής τάσεως των καταμεριστών**

Οι καταμεριστές που εξετάστηκαν στα προηγούμενα ανήκουν βασικά σε δύο τύπους: τους ωμικούς (με διάφορες παραλλαγές) και τους χωρητικούς. Ο αποσβεσμένος χωρητικός ανήκει κι αυτούς στην κατηγορία των χωρητικών καταμεριστών. Οι βραχίονες χαμηλής τάσεως για τους δύο τύπους καταμεριστών παρουσιάζουν επίσης διαφορές. Στους ωμικούς καταμεριστές το κύριο στοιχείο του βραχίονα χαμηλής τάσεως είναι μια αντίσταση, στους χωρητικούς ένας πυκνωτής και στους χωρητικούς αποσβεσμένους ένας πυκνωτής με μια αντίσταση στη σειρά.



**Εικόνα 10: Χωρητικός καταμεριστής με απόσβεση**

## **4. Συγκριτική μελέτη μεθόδων αξιολόγησης των παραμέτρων των κεραυνικών παλμών**

### **4.1. Γενικά στοιχεία**

Οι παράμετροι των κεραυνικών παλμών (ΚΠ) μπορούν να προσδιοριστούν σύμφωνα με το πρότυπο IEC-60, στην περίπτωση που το σχήμα του παλμού είναι ομαλό. Παρόλα αυτά, εμφανίζονται δυσκολίες εάν στην κυματομορφή έχουν υπερτεθεί ταλαντώσεις ή υπερτάσεις. Αν η συχνότητα μιας τέτοιας ταλάντωσης είναι μεγαλύτερη από 0.5MHz ή η διάρκεια της υπέρτασης είναι μικρότερη από 1μsec, είναι σημαντικό να προσδιοριστεί μια μέση καμπύλη για την αξιολόγηση των παραμέτρων του ΚΠ. Στα παρακάτω, αναφέρονται δύο τεχνικές προσέγγισης αυτού του προβλήματος. Η πρώτη βασίζεται στη μέθοδο MSD (Multiresolution Signal Decomposition), ενώ η δεύτερη στην μαθηματική προσαρμογή της καμπύλης σε μια μη γραμμική εξίσωση κυματομορφής, αποτελούμενης από μια διπλά εκθετική συνάρτηση με ημιτονικούς και συνημιτονικούς όρους.

Κάθε τεχνική παρουσιάζει κάποια δυνατά και αδύναμα σημεία, ανάλογα και με την φύση του ΚΠ που μελετάται. Ακολούθως δίνονται μετρήσεις από συγκριτικά πειράματα για διάφορες περιπτώσεις ΚΠ, όπως ο κανονικός SLI (Standard Lightning Impulse), με εμπρόσθιες ταλαντώσεις LIFO (Lightning Impulse Front Oscillations), με ταλαντώσεις στην κορυφή LIPO (Lightning Impulse Peak Oscillations), με μακρά περίοδο υπέρτασης LILDO (Lightning Impulse Long Duration Overshoot), με εμπρόσθιες ταλαντώσεις και ταλαντώσεις κορυφής LIFPO (Lightning Impulse Front Peak Oscillations) ή με θόρυβο LIN (Lightning Impulse Noise).

Οι συσκευές υψηλής τάσης που χρησιμοποιούνται στα συστήματα ισχύος είναι σχεδιασμένες και κατασκευασμένες έτσι ώστε να μπορούν να αντέχουν σε διάφορες διηλεκτρικές πιέσεις όπως είναι οι πολύ γρήγορες μεταβάσεις (κεραυνικοί και μεταβατικοί παλμοί). Πρότυπα όπως το IEC 60060-1 / IEEE-4 περιγράφουν μερικές μορφές δοκιμών που έχουν σαν σκοπό τον έλεγχο του σχεδιασμού των συσκευών υψηλής τάσης. Μια από τις σημαντικές αυτές δοκιμές είναι αυτή της εφαρμογής τάσης κεραυνικού παλμού, που βασίζεται σε έναν πρότυπο παλμό 1.2/50μs και αποσκοπεί

στον έλεγχο της διηλεκτρικής μόνωσης. Συνεπώς είναι απολύτως απαραίτητο να ελεγχθούν το σχήμα και οι παράμετροι του ΚΠ, πριν γίνει η εφαρμογή του στο υπό εξέταση αντικείμενο. Οι ομαλές παράμετροι του παλμού χαρακτηρίζονται από την τάση κορυφής  $V_p$  (peak voltage), τον χρόνο μετώπου  $T_1$  (front time) και τον ουραίο χρόνο  $T_2$  (tail time). Ένας πρότυπος ΚΠ έχει τιμές  $T_1 = 1.2\mu s \pm 30\%$  και  $T_2 = 50\mu s \pm 20\%$ . Θα πρέπει να αναφερθεί πως οι μετρήσεις παλμών υψηλών τάσεων, πραγματοποιούνται πλέον με ψηφιακούς παλμογράφους, καθώς παρέχουν τα ακόλουθα πλεονεκτήματα:

- i) Δίνουν την δυνατότητα αποθήκευσης της κυματομορφής και των πειραματικών δεδομένων για περαιτέρω ανάλυση
- ii) Διευκολύνουν τον προσδιορισμό των παραμέτρων των παλμών με την χρήση μαθηματικών εργαλείων
- iii) Παρέχουν πολύ μεγάλη ακρίβεια

Οι κυματομορφές που έχουν καταγραφεί ψηφιακά, περιέχουν συνήθως θόρυβο και ταλαντώσεις που προέρχονται από διάφορες πηγές, όπως εσωτερικός θόρυβος του ψηφιοποιητή ή ταλαντώσεις από το σύστημα γέννησης και μέτρησης των παλμών. Σε αυτές τις περιπτώσεις προκύπτει ένα πρόβλημα για την αυτοματοποιημένη αξιολόγηση των παραμέτρων των παλμών και σύμφωνα με το πρότυπο, απαιτείται ο προσδιορισμός μιας μέσης καμπύλης, η οποία θα αντιπροσωπεύει έναν εξομαλυσμένο ΚΠ χωρίς ταλαντώσεις ή υπερτάσεις, ώστε να είναι δυνατός ο προσδιορισμός των παραμέτρων του. Ο τρόπος με τον οποίο θα προκύψει μια τέτοια μέση καμπύλη δεν καθορίζεται από το πρότυπο, με συνέπεια τα πειραματικά αποτελέσματα να διαφέρουν, ανάλογα με την εκάστοτε επιλογή και χάραξη της μέσης καμπύλης.

#### **4.2. Σχετικές εργασίες**

Οι T.R.McComb και J.E.Lagnese εισήγαγαν πρώτοι την τεχνική υπολογισμού των παραμέτρων των ΚΠ με βάση το ταίριασμα της καμπύλης του παλμού με διπλά και τριπλά εκθετικές συναρτήσεις. Καθώς δεν υπάρχει μια εξίσωση που να ικανοποιεί όλους τους τύπους κυματομορφών των παλμών, είναι σημαντικό να βρεθεί μια έκφραση για κάθε κυματομορφή [8]. Ο Y.M.Li et al, περιέγραψαν διάφορους κατάλληλους αλγόριθμους χρησιμοποιώντας γενικές πολυωνυμικές και εκθετικές

συναρτήσεις [9]. Πολλές ερευνητικές προσπάθειες έχουν επίσης συνεισφέρει προς την ίδια κατεύθυνση και έχουν ενσωματωθεί στην ομάδα εργασίας 33.03 του CIGRE, στην οποία συζητούνται διάφορες μέθοδοι (προσαρμογής καμπύλης και ψηφιακού φιλτραρίσματος) για την απομάκρυνση των ταλαντώσεων ή των υπερτάσεων που έχουν υπερτεθεί σε έναν καταγεγραμμένο παλμό [10]. Οι J.Perez και J.Martinez πρότειναν μια επέκταση της μεθόδου φιλτραρίσματος Kalman προκειμένου να βρουν την μέση καμπύλη για όλα τα είδη παλμών με υπερτάσεις [11]. Οι Yi Li και Juris Rugnis ανέλυσαν την απόδοση του φίλτρου απόκρισης πεπερασμένου παλμού (Finite Impulse Response, FIR) στην περίπτωση ΚΠ [12]. Ο A.P.Brede et al, πρότειναν τον προσδιορισμό της μέσης καμπύλης με διπλά εκθετικές συναρτήσεις με ημιτονικούς και συνημιτονικούς όρους, όπως η εξής:

$$V(t) = A * (e^{-\alpha t} - e^{-\beta t}) + B * (1 - \cos(\omega t)) e^{-\gamma t} + C * \sin(\omega t) e^{-\delta t}$$

Ο συνδυασμός διπλά εκθετικής συνάρτησης με ημιτονικούς και συνημιτονικούς όρους σε μία συνάρτηση, δίνει πιο ακριβές ταίριασμα και για κανονικούς ΚΠ αλλά και για ΚΠ με υπερτάσεις ή ταλαντώσεις [13]. Οι L.Satish και B.I. Gurajah περιγράφουν μια νέα μέθοδο για την εύρεση της μέσης καμπύλης που την ονομάζουν MSD (Multiresolution Signal Decomposition), σύμφωνα δε με αυτούς, η μέθοδος μπορεί να εφαρμοστεί σε κάθε τύπο σήματος, αφού δεν απαιτεί κάποιο μοντέλο προσαρμογής καμπύλης, ούτε εισάγει παραμορφώσεις λόγω της απομάκρυνσης του θορύβου.

Παρακάτω αναφέρονται αναλυτικά δύο επικρατούσες τεχνικές επίλυσης αυτού του προβλήματος, δηλαδή η προσαρμογή σε μια μαθηματική καμπύλη και η MSD.

### **4.3. Τεχνική MSD**

Ο διακριτός μετασχηματισμός κυματίων (discrete wavelet transform) μετατρέπει κάθε διακριτό σήμα από το πεδίο του χρόνου στο αντίστοιχο σήμα στο πεδίο των κυματίων. Για το σκοπό αυτό χρησιμοποιείται μια δυαδική ορθοκανονική βασική συνάρτηση και μέσω της τεχνικής MSD εξάγεται η σχετική πληροφορία, καθώς το σήμα αναλύεται σε κάποιο αριθμό από λεπτομερή και εξομαλυσμένα σήματα. Για μεγαλύτερη ακρίβεια, πρέπει να αυξήσουμε τον βαθμό ανάλυσης του

αρχικού σήματος. Οι υπερτιθέμενες ταλαντώσεις και υπερτάσεις μπορούν να εντοπιστούν εύκολα πια ανάμεσα στις διάφορες συνιστώσες στις οποίες αναλύθηκε το αρχικό σήμα. Το κατάλληλο μητρικό κυμάτιο που χρησιμοποιείται στην περίπτωση που μας ενδιαφέρει είναι συμμετρικό (symlet), εφόσον αυτό προσφέρει τη μεγαλύτερη ευελιξία για την επεξεργασία ΚΠ με ταλαντώσεις ή υπερτάσεις ή άλλων σημάτων με θόρυβο. Επιλέγοντας το κατάλληλο κατώφλι είναι δυνατόν να ελαχιστοποιήσουμε ή και να εξαφανίσουμε τελείως τις ανεπιθύμητες συνιστώσες του σήματος, χωρίς κάποια σημαντική παραμόρφωση του αρχικού σήματος. Τελικά από τους συντελεστές που προκύπτουν από τον μετασχηματισμό και την προαναφερθείσα επιλογή μπορούμε να ανακατασκευάσουμε το αρχικό σήμα, χρησιμοποιώντας μια αντίστροφη διαδικασία που περιλαμβάνει υπέρ-δειγματοληψία και φιλτράρισμα, σε αντίθεση με την ανάλυση κυματιδίων που περιλαμβάνει φιλτράρισμα και υπό-δειγματοληψία.

Αποτελέσματα πειραματικών ερευνών με συμμετρικά κυματίδια όγδοης τάξης (symlet 8) [14] έδειξαν ότι η μέθοδος MSD είναι πολύ αποδοτική σε περιπτώσεις όπου το αρχικό σήμα ΚΠ χαρακτηρίζεται από θόρυβο και ταλαντώσεις υψηλής συχνότητας, καθώς είναι σε θέση να αναγνωρίσει και να απομακρύνει θόρυβο και ταλαντώσεις χωρίς να εισάγει παραμόρφωση. Από την άλλη, όταν ο ΚΠ εμφανίζει υπερτάσεις ή αργές ταλαντώσεις, η συγκεκριμένη μέθοδος δεν δίνει ικανοποιητικά αποτελέσματα και είναι απαραίτητη τουλάχιστον η χρήση κυματιδίων μεγαλύτερης τάξης. Οι λόγοι στους οποίους οφείλεται αυτή η αναποτελεσματικότητα είναι πρώτον το γεγονός ότι απαιτείται περισσότερος χρόνος για να αναγνωριστούν οι υπερτάσεις ή οι αργές ταλαντώσεις και δεύτερον το ότι προκαλείται παραμόρφωση στην κορυφή. Θα πρέπει βέβαια να επισημανθεί, ότι η μέθοδος MSD θα μπορούσε να δώσει καλύτερα αποτελέσματα αν στην προκειμένη έρευνα η επιλογή του κατωφλίου για την απομάκρυνση των ανεπιθύμητων συνιστωσών δεν γινόταν με το χέρι, αλλά προσδιοριζόταν δυναμικά μέσω κάποιας κατάλληλης υπολογιστικής διαδικασίας.

#### **4.4. Τεχνική προσαρμογής καμπύλης**

Η μέση καμπύλη που θα επιλεγεί πρέπει να ταιριάζει με τον παλμό που καταγράφηκε και ιδανικά αυτό πρέπει να προκύπτει μέσω κάποιας μαθηματικής διαδικασίας. Μια καλή μέθοδος είναι αυτή των Ελαχίστων Μέσων Τετραγώνων (Least



Mean Square, LMS) η οποία προτάθηκε από τους Widrow και Hoff το 1959. Με δεδομένα σημεία  $(t_i, x_i)$  για  $i=1 \dots n$ , ζητείται να βρεθεί ένα μοντέλο  $V_{calc}(t,y)$  που να ταιριάζει σε αυτά τα δεδομένα, όπου  $y$  είναι ένα διάνυσμα από άγνωστες παραμέτρους. Το πρόβλημα είναι να υπολογιστούν οι εκτιμώμενες τιμές αυτών των παραμέτρων, έτσι ώστε να ελαχιστοποιείται μια αντικειμενική συνάρτηση (objective function)  $F$ , η οποία δίνεται από την σχέση:

$$F = \sum_{i=1}^n [V_{meas}(i) - V_{calc}(i)]^2 = \min$$

Σε αυτή την τεχνική το σφάλμα LMS χρησιμοποιείται σαν ποιοτικό κριτήριο και ο αλγόριθμος Levenberg – Marquardt (LM) για να λύσει την μη γραμμική εξίσωση της κυματομορφής του παλμού που αντιστοιχεί σε μια διπλά εκθετική συνάρτηση με ημιτονικούς και συνημιτονικούς όρους. Η ακόλουθη γενική συνάρτηση μπορεί να περιγράψει την μέση καμπύλη της κυματομορφής του ΚΠ:

$$V(t) = A \cdot (e^{-\alpha t} - e^{-\beta t}) + B \cdot (1 - \cos(\omega t)) \cdot e^{-\gamma t} + C \cdot \sin(\phi t) \cdot e^{-\delta t}$$

και η αντίστοιχη αντικειμενική συνάρτηση θα είναι:

$$F = \sum_{i=1}^n [V_{meas}(i) - A \cdot (e^{-\alpha t} - e^{-\beta t}) - B \cdot (1 - \cos(\omega t)) \cdot e^{-\gamma t} - C \cdot \sin(\phi t) \cdot e^{-\delta t}]^2$$

Η ελαχιστοποίηση της  $F(\alpha, \beta, \gamma, \delta, \omega, \phi, A, B, C)$  γίνεται βρίσκοντας τις τιμές των  $\alpha, \beta, \gamma, \delta, \omega, \phi, A, B, C$ . Υπολογίζονται οι μερικές παράγωγοι της  $F$  ως προς τις 9 αυτές παραμέτρους και εξισώνονται με το μηδέν. Αυτές οι μη γραμμικές εξισώσεις που προκύπτουν επιλύονται με την βοήθεια του επαναληπτικού αλγορίθμου Levenberg – Marquardt.

Η τεχνική προσαρμογής καμπύλης δοκιμάστηκε σε πειραματικά δεδομένα από διάφορα είδη ΚΠ (SLI, LIPO, LIFPO, LILDO) [14] και φάνηκε ότι η διπλά εκθετική συνάρτηση με ημιτονικούς και συνημιτονικούς όρους αναπαράγει το μέτωπο και την ουρά του ΚΠ σχεδόν τέλεια. Αυτό προέκυψε παίρνοντας την διαφορά της καμπύλης από τις μετρήσεις και της μέσης καμπύλης, οπότε ελέγχθηκε ότι η απόλυτη τιμή της διαφοράς είναι πάντα μικρότερη από ένα κατώφλι που τέθηκε. Στις περιπτώσεις που

υπήρχε πρόβλημα με αυτό το κριτήριο, έγινε ξανά υπολογισμός των παραμέτρων της αντικειμενικής συνάρτησης, ώστε να ταιριάξουν οι καμπύλες.

Γενικά από τα αποτελέσματα των δοκιμών φάνηκε ότι η επιλογή μιας συνάρτησης με αυτή την μορφή για την εύρεση της μέσης καμπύλης είναι καλύτερη από τις κλασικές διπλά εκθετικές συναρτήσεις χωρίς ημιτονικούς και συνημιτονικούς όρους. Οι μόνοι περιορισμοί είναι ότι ο αρχικός υπολογισμός των παραμέτρων της  $F$  έχει σημαντικό αντίκτυπο στα αποτελέσματα και ότι ο χρόνος υπολογισμού μεγαλώνει πολύ για μεγάλο αριθμό δειγμάτων.

Συνολικά και σε σύγκριση και με την τεχνική MSD που περιγράφηκε στην ενότητα 4.3 η τεχνική προσαρμογής καμπύλης μπορεί να είναι πολύ πιο αποδοτική με προσεκτική επιλογή των αρχικών παραμέτρων, αλλά η MSD θα έχει πάντα το πλεονέκτημα ότι δεν προϋποθέτει την ύπαρξη κάποιου μοντέλου και συνεπώς μπορεί να εφαρμοστεί σε όλα τα είδη σημάτων.

## **5. Το πρότυπο EN 60060-2**

### **5.1. Γενικά στοιχεία**

Το ευρωπαϊκό πρότυπο EN 60060-2 [15] δημιουργήθηκε από την ευρωπαϊκή επιτροπή ηλεκτροτεχνικής τυποποίησης (CENELEC) στις 6-12-1994 και αφορά τα συστήματα μέτρησης υψηλών τάσεων. Το EN 60060-2 διατυπώνει τις προϋποθέσεις που πρέπει να πληρεί ένα σύστημα μέτρησης και περιγράφει τη διαδικασία με την οποία ο χρήστης μπορεί να δείξει ότι ένα σύστημα μέτρησης υπακούει στις απαιτήσεις του προτύπου.

### **5.2. Ορισμοί βασικών εννοιών του προτύπου**

#### **5.2.1. Συστήματα μέτρησης**

- i) Σύστημα μέτρησης (Measuring System): Ένα ολοκληρωμένο σύνολο συσκευών κατάλληλο για τη μέτρηση μιας υψηλής τάσης ή κρουστικού ρεύματος.
- ii) Εγγραφή απόδοσης ενός συστήματος μέτρησης (Record of Performance of a Measuring System): Λεπτομερής εγγραφή του χρήστη, που περιλαμβάνει περιγραφή του συστήματος και απόδειξη ότι αυτό ακολουθεί τις απαιτήσεις του προτύπου. Η απόδειξη περιλαμβάνει τα αποτελέσματα του αρχικού τεστ λειτουργίας (Performance test) καθώς και των επόμενων τεστ λειτουργίας και ελέγχων λειτουργίας (Performance Checks).
- iii) Εγκεκριμένο σύστημα μέτρησης (Approved Measuring System): Σύστημα μέτρησης το οποίο υπακούει στις απαιτήσεις του προτύπου έχοντας ένα αρχικό τεστ λειτουργίας, επιτυχημένα τεστ λειτουργίας και ελέγχους λειτουργίας και συμπερίληψη των αποτελεσμάτων των τεστ αυτών στην καταγραφή της λειτουργίας (Record of Performance).
- iv) Σύστημα αναφοράς (Reference Measuring System): Σύστημα μέτρησης με επαρκή ακρίβεια και σταθερότητα ώστε να χρησιμοποιηθεί για την έγκριση άλλων συστημάτων μέτρησης κάνοντας ταυτόχρονες συγκρίσιμες

μετρήσεις με ειδικούς τύπους κυματομορφών και περιοχές τάσης ή ρεύματος.

### **5.2.2. Δοκιμές**

- i) Τεστ αποδοχής (acceptance test): Τεστ σε μία συσκευή ή σύστημα μέτρησης πριν εγκριθεί για χρήση. Περιλαμβάνει δοκιμές τύπου (εκτελούνται σε μία συσκευή της παρτίδας) και δοκιμές ρουτίνας (εκτελούνται ανεξαιρέτως σε κάθε συσκευή). Σκοπός είναι να εκτιμηθούν τα ειδικά χαρακτηριστικά κάθε συσκευής όπως για παράδειγμα ο συντελεστής θερμοκρασίας, η αντοχή κ.ά.
- ii) Τεστ λειτουργίας (Performance Test): Τεστ σε ένα ολοκληρωμένο σύστημα μέτρησης με σκοπό να χαρακτηριστεί κάτω από συνθήκες λειτουργίας.
- iii) Έλεγχος λειτουργίας (Performance Check): Είναι μια απλή διαδικασία που διαβεβαιώνει ότι το πιο πρόσφατο τεστ λειτουργίας είναι ακόμα σε ισχύ.

### **5.2.3. Συντελεστές αναλογίας**

- i) Συντελεστής αναλογίας συστήματος μέτρησης (scale factor of a Measuring System): Συντελεστής με τον οποίο πρέπει να πολλαπλασιαστεί η τιμή του οργάνου μέτρησης για να βρεθεί η τιμή της εισόδου.
- ii) Συντελεστής αναλογίας συστήματος μετάδοσης (scale factor of a transmission system): Συντελεστής με τον οποίο η έξοδος ενός συστήματος μετάδοσης πρέπει να πολλαπλασιαστεί για να δώσει την είσοδο.
- iii) Καθορισμένος συντελεστής αναλογίας (Assigned Scale Factor): Συντελεστής αναλογίας ενός συστήματος μέτρησης ορισμένος από το πιο πρόσφατο τεστ λειτουργίας.

### **5.2.4. Δυναμική συμπεριφορά συστήματος μέτρησης**

- i) Ονομαστική περίοδος (Nominal Epoch  $\tau_N$ ): Εύρος τιμών μεταξύ της ελάχιστης ( $t_{\min}$ ) και της μέγιστης ( $t_{\max}$ ) τιμής της χρονικής παραμέτρου

των παλμών για τους οποίους το σύστημα μέτρησης πρέπει να αποδειχθεί.

Οι χρονικές παράμετροι είναι:

- Χρόνος μετώπου  $T_1$ , για πλήρεις κεραυνικές και αποκομμένες στην ουρά κρουστικές τάσεις και κρουστικά ρεύματα
  - Χρόνος αποκοπής  $T_c$ , για αποκομμένες στο μέτωπο κρουστικές τάσεις
  - Χρόνος κορυφής  $T_p$ , για κρουστικές τάσεις
- ii) Απόκριση συστήματος μέτρησης (response of a Measuring System  $G$ ): Η έξοδος συναρτήσει του χρόνου ή της συχνότητας, όταν μία καθορισμένη τάση ή ρεύμα εφαρμόζεται στην είσοδο του συστήματος.
- iii) Βηματική απόκριση (Step response  $G(t)$ ): Η έξοδος του συστήματος μέτρησης συναρτήσει του χρόνου, όταν η είσοδος είναι βηματική συνάρτηση.

#### **5.2.5. Παράμετροι απόκρισης**

- i) Επίπεδο αναφοράς (reference level  $I_R$ ): Μέσος όρος της βηματικής απόκρισης κατά μία ονομαστική εποχή.
- ii) Ουσιαστική αρχή βηματικής απόκρισης (virtual origin of a step response  $O_1$ ): Τομή με τον άξονα του χρόνου μιας ευθείας γραμμής που φέρνουμε ως εφαπτομένη στο πιο κάθετο τμήμα του μπροστινού μέρους της βηματικής απόκρισης. Σε περίπτωση απόκρισης με ταλαντώσεις στην αρχή μία μέση καμπύλη σχεδιάζεται ανάμεσα από τις ταλαντώσεις και χρησιμοποιείται για να ορίσει την εφαπτομένη. Κάθε αρχική παραμόρφωση αμελείται όταν σχεδιάζεται η εφαπτομένη. Όλες οι χρονικές παράμετροι μετρούνται από την ουσιαστική αρχή  $O_1$ .
- iii) Ομαλοποιημένη βηματική απόκριση (normalized step response  $g(t)$ ): Βηματική απόκριση ομαλοποιημένη έτσι ώστε το επίπεδο αναφοράς να είναι μονάδα.
- iv) Χρόνος απόκρισης (step response integral  $T(t)$ ): Ολοκλήρωμα από το  $O_1$  έως  $t$  του  $1 - g(t)$ , με το αρχικό τμήμα της  $g(t)$  αντικατεστημένο από την εφαπτομένη που ορίζει το  $O_1$ .

$$T(t) = \int_{0_1}^t (1 - g(\tau)) d\tau$$

- v) Πειραματικός χρόνος απόκρισης (experimental response time  $T_N$ ): Η τιμή του χρόνου απόκρισης στο  $t_{\max}$ .

$$T_N = T(t_{\max})$$

- vi) Μερικός χρόνος απόκρισης (partial response time  $T_a$ ): Μέγιστη τιμή του χρόνου απόκρισης.

$$T_a = T(t_1) ,$$

όπου  $t_1$  είναι η χρονική στιγμή στην οποία η  $g(t)$  έχει για πρώτη φορά μοναδιαίο πλάτος.

- vii) Παραμένων χρόνος απόκρισης (residual response time  $T_R(t_i)$ ): Ο πειραματικός χρόνος απόκρισης μείων την τιμή του χρήσιμου χρόνου απόκρισης τη χρονική στιγμή  $t_i$  όπου  $t_i < t_{\max}$ .

$$T_R(t_i) = T_N - T(t_i)$$

- viii) Υπερύψωση (overshoot  $\beta$ ): Ποσότητα κατά την οποία η μέγιστη τιμή της ομαλοποιημένης βηματικής απόκρισης  $g(t)$  ξεπερνά την μονάδα.

- ix) Αρχικός χρόνος παραμόρφωσης ( $T_0$ ): Η περιοχή που ορίζεται από τον άξονα του χρόνου, την ομαλοποιημένη βηματική απόκριση  $g(t)$  και την εφαπτομένη που δίνει το  $O_1$ .

- x) Χρόνος ισορρόπησης (settling time  $t_s$ ): Ο συντομότερος χρόνος για τον οποίο ο παραμένων χρόνος απόκρισης  $T_R(t)$  γίνεται μικρότερος του 2% του  $t$ .

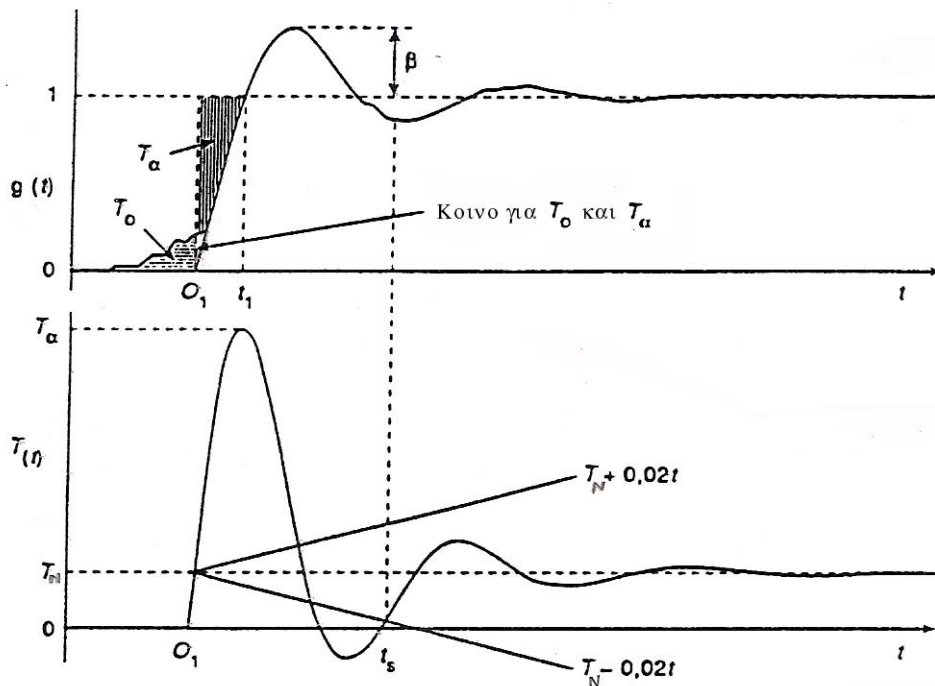
$$|T_N - T(t)| < 0,02 \cdot t_s ,$$

για όλες τις τιμές του  $t$  στην περίοδο από  $t_s$  ως  $t_{\max}$

- xi) Ισοδύναμος χρόνος (equivalent time  $T_E$ ): Χρονική παράμετρος ρύθμισης των κυματομορφών όπου

- ο για συνεχή τάση  $T_E = 100 \text{ ms}$  (αυθαίρετα)
- ο για εναλλασσόμενη τάση η  $T_E$  είναι ισοδύναμη με το ένα τέταρτο της περιόδου
- ο για κρουστικές τάσεις η  $T_E$  είναι ισοδύναμη με το χρόνο κορυφής  $T_p$

xii) Ολική αβεβαιότητα (overall uncertainty  $e$ ): Υπολογίζει το εύρος τιμών γύρω από το αποτέλεσμα μιας μέτρησης, εντός του οποίου μπορεί να βρίσκεται η πραγματική της τιμή. Είναι ο συνδυασμός πολλών μεμονωμένων αβεβαιοτήτων λόγω της παρουσίας πολλών μεγεθών που επηρεάζουν την μέτρηση.



Εικόνα 11: Γραφική απεικόνιση παραμέτρων απόκρισης

### 5.3. Διαδικασίες για τον χαρακτηρισμό των συστημάτων μέτρησης

#### 5.3.1. Γενικές αρχές

Τα εγκεκριμένα από το πρότυπο συστήματα μέτρησης υπόκεινται σε δοκιμές αποδοχής καθ' όλη τη διάρκεια της ζωής τους. Αυτές είναι οι ακόλουθες:

- Τεστ αποδοχής στα διάφορα μέρη που αποτελούν το σύστημα (μία φορά)

- ii) Τεστ λειτουργίας του συστήματος (περιοδικά)
- iii) Έλεγχος λειτουργίας του συστήματος (περιοδικά)

Το τεστ αποδοχής μίας συσκευής περιλαμβάνει δοκιμές τύπου (εκτελούνται σε μία συσκευή) και δοκιμές ρουτίνας (εκτελούνται ανεξαιρέτως σε κάθε συσκευή). Σκοπός είναι να εκτιμηθούν τα ειδικά χαρακτηριστικά κάθε συσκευής όπως για παράδειγμα ο συντελεστής θερμοκρασίας, η αντοχή κ.ά.

Το τεστ λειτουργίας γίνεται στο ολοκληρωμένο σύστημα μέτρησης με σκοπό να χαρακτηριστεί κάτω από συνθήκες λειτουργίας.

Ο έλεγχος λειτουργίας είναι μια απλή διαδικασία που διαβεβαιώνει ότι το πιο πρόσφατο τεστ λειτουργίας είναι ακόμα σε ισχύ.

## **5.4. Μέτρηση συνεχούς τάσης**

### **5.4.1. Απαιτήσεις για ένα εγκεκριμένο σύστημα μέτρησης**

Η γενική απαίτηση είναι να μετρηθεί η τιμή (αριθμητικός μέσος) της τάσης με μια ολική αβεβαιότητα  $\pm 3\%$ .

### **5.4.2. Ευστάθεια του συντελεστή αναλογίας**

Οι συντελεστές αναλογίας του μετατροπέα και του συστήματος μετάδοσης δεν θα πρέπει να διαφέρουν πάνω από  $\pm 1\%$  μέσα στο διάστημα θερμοκρασίας περιβάλλοντος και υγρασίας που δίνεται στην εγγραφή απόδοσης.

### **5.4.3. Δυναμική συμπεριφορά για μέτρηση της τάσης ανόδου**

Ο πειραματικός χρόνος απόκρισης  $T_N$  ενός εγκεκριμένου συστήματος μέτρησης δεν πρέπει να περνά τα 0,5s.

## **5.5. Μέτρηση κεραυνικής κρουστικής τάσης**

### **5.5.1. Απαιτήσεις ενός εγκεκριμένου συστήματος μέτρησης**

Οι γενικές απαιτήσεις είναι:

- i) να μετρηθεί η τιμή κορυφής μιας πλήρους κρουστικής τάσης με ολική αβεβαιότητα  $\pm 3\%$
- ii) να μετρηθεί η τιμή κορυφής της αποκομμένης κρουστικής τάσης με ολική αβεβαιότητα  $\epsilon$  που εξαρτάται από το χρόνο αποκοπής  $T_c$  ως ακολούθως:



- για αποκομμένες στο μέτωπο κρουστικές τάσεις ( $0,5\mu s \leq T_c < 2\mu s$ ) ,  
 $e \leq \pm 5\%$
  - για αποκομμένες στην ουρά κρουστικές τάσεις ( $T_c \geq 2\mu s$ ) ,  $e \leq \pm 3\%$
- iii) να μετρηθούν οι χρονικές παράμετροι που ορίζουν την κυματομορφή με ολική αβεβαιότητα  $\pm 10\%$

### **5.5.2. Δυναμική συμπεριφορά**

Η δυναμική συμπεριφορά ενός συστήματος μέτρησης είναι επαρκής για την μέτρηση της τάσης κορυφής και των χρονικών παραμέτρων όταν

- i) ο συντελεστής αναλογίας είναι σταθερός στα όρια :
  - $\pm 1\%$  για πλήρεις και αποκομμένες στην ουρά κρουστικές τάσεις
  - $\pm 3\%$  για αποκομμένες στο μέτωπο κρουστικές τάσεις
- ii) η αβεβαιότητα των χρονικών παραμέτρων είναι  $\pm 10\%$ .

## **5.6. Μέτρηση τυπικής κρουστικής τάσης**

### **5.6.1. Απαιτήσεις ενός εγκεκριμένου συστήματος μέτρησης**

Οι γενικές απαιτήσεις είναι:

- i) να μετρηθεί η τιμή κορυφής μιας τυπικής κρουστικής τάσης με ολική αβεβαιότητα  $\pm 3\%$
- ii) να μετρηθούν οι χρονικές παράμετροι που ορίζουν την κυματομορφή με ολική αβεβαιότητα  $\pm 10\%$

### **5.6.2. Δυναμική συμπεριφορά**

Η δυναμική συμπεριφορά ενός συστήματος μέτρησης είναι επαρκής όταν

- i) ο συντελεστής αναλογίας είναι σταθερός μέσα στο διάστημα  $\pm 1\%$  πάνω από το εύρος των κρουστικών κυματομορφών όπως ορίζεται στην εγγραφή της απόδοσης
- ii) η ολική αβεβαιότητα των μετρούμενων χρονικών παραμέτρων είναι μέσα στο διάστημα  $\pm 1\%$

## 5.7. Συστήματα αναφοράς

### 5.7.1. Απαιτήσεις για συστήματα αναφοράς συνεχούς τάσης

Ένα σύστημα αναφοράς για συνεχή τάση πρέπει να έχει μια ολική αβεβαιότητα  $\pm 1\%$ . Η ακρίβεια δε θα πρέπει να επηρεάζεται από έναν συντελεστή κυμάτωσης πάνω από 3%.

### 5.7.2. Απαιτήσεις για συστήματα αναφοράς εναλλασσόμενης τάσης

Ένα σύστημα αναφοράς για εναλλασσόμενη τάση πρέπει να έχει μια ολική αβεβαιότητα  $\pm 1\%$ .

### 5.7.3. Απαιτήσεις για συστήματα αναφοράς για πλήρεις και αποκομμένους στην ουρά κεραυνικούς και τυπικούς κρουστικούς παλμούς

Το σύστημα αναφοράς για αυτούς τους παλμούς πρέπει να έχει μια ολική αβεβαιότητα  $\pm 1\%$  για την τάση κορυφής των πλήρων παλμών,  $\pm 5\%$  για τις χρονικές παραμέτρους, ενώ ταλαντώσεις και υπερύψωση πρέπει να καταγράφονται επαρκώς.

### 5.7.4. Απαιτήσεις για συστήματα αναφοράς για κεραυνικές τάσεις αποκομμένες στο μέτωπο

Ένα τέτοιο σύστημα αναφοράς πρέπει να έχει μια ολική αβεβαιότητα  $\pm 3\%$  για τιμές κορυφής και  $\pm 5\%$  για τις χρονικές παραμέτρους.

Παράμετροι	Απαιτήσεις για		
	Πλήρεις και αποκομμένους στην ουρά κεραυνικούς παλμούς	Αποκομμένους στο μέτωπο κεραυνικούς παλμούς	Τυπικούς κρουστικούς παλμούς
Πειραματικός χρόνος απόκρισης $T_N$	$\leq 15$ ns	$\leq 10$ ns	
Χρόνος ισορρόπησης $T_s$	$\leq 200$ ns	$\leq 150$ ns	$\leq 10$ ns
Μερικός χρόνος απόκρισης $T_a$	$\leq 30$ ns	$\leq 20$ ns	
Αρχικός χρόνος παραμόρφωσης $T_o$		$\leq 2,5$ ns	

## 6. Διαδικασία υπολογισμού των παραμέτρων απόκρισης συστήματος μέτρησης

Για τον υπολογισμό των παραμέτρων απόκρισης ενός συστήματος μέτρησης εφαρμόζουμε στην είσοδό του τη βηματική συνάρτηση και παίρνουμε την κανονικοποιημένη βηματική απόκριση  $g(t)$ . Η απόκριση αυτή μπορεί να εμφανίζει ταλαντώσεις που οφείλονται στα RLC κυκλώματα απόσβεσης. Έτσι, απαιτείται η εξομάλυνση (smoothing) της  $g(t)$ .

Η βηματική απόκριση μπορεί να περιγραφεί από τις ακόλουθες παραμέτρους που είναι σχετικές με τη συμπεριφορά της στο πεδίο του χρόνου:

- Ουσιαστική αρχή βηματικής απόκρισης  $O_1$
- Χρόνος απόκρισης  $T(t)$
- Πειραματικός χρόνος απόκρισης  $T_N$
- Μερικός χρόνος απόκρισης  $T_a$
- Υπερύψωση  $\beta$
- Αρχικός χρόνος παραμόρφωσης  $T_0$
- Χρόνος ισορρόπησης  $t_s$

Αρχικά εξομαλύνεται η βηματική απόκριση  $g(t)$  και στη συνέχεια υπολογίζονται οι παράμετροι απόκρισης.

### 6.1.1. Εξομάλυνση βηματικής απόκρισης

Τα σημεία της βηματικής απόκρισης ομαλοποιούνται με μία επαναληπτική διαδικασία κατά την οποία υπολογίζεται ο μέσος όρος γειτονικών σημείων [16]. Δηλαδή για κάθε σημείο βρίσκεται το νέο (εξομαλυσμένο), από το μέσο όρο του σημείου με τα γειτονικά του. Ο αριθμός των γειτονικών σημείων που χρησιμοποιούνται για το μέσο όρο καθώς και ο αριθμός των επαναλήψεων ορίζονται πριν την εξομάλυνση.

Π.χ. για αριθμό γειτονικών σημείων 2, το νέο σημείο βρίσκεται από το μέσο όρο του σημείου με τα δύο γειτονικά του. Έστω  $(g_1, t_1)$ ,  $(g_2, t_2)$ ,  $(g_3, t_3)$ ,  $(g_4, t_4)$ , .....

διαδοχικά σημεία της  $g(t)$ . Για εξομάλυνση με χρήση 2 γειτονικών σημείων και 2 επαναλήψεις έχουμε:

$$g'_2 = \frac{g_1 + g_2 + g_3}{3}$$

$$g'_3 = \frac{g_2 + g_3 + g_4}{3}$$

$$g'_4 = \frac{g_3 + g_4 + g_5}{3}$$

κ.λ.π

όπου  $(g'_2, t_2)$ ,  $(g'_3, t_3)$ ,  $(g'_4, t_4)$ , ..... τα σημεία που προκύπτουν από την πρώτη επανάληψη. Ενώ  $(g''_2, t_2)$ ,  $(g''_3, t_3)$ ,  $(g''_4, t_4)$ , ..... είναι τα νέα εξομαλυσμένα σημεία όπου:

$$g''_2 = \frac{g'_1 + g'_2 + g'_3}{3}$$

$$g''_3 = \frac{g'_2 + g'_3 + g'_4}{3}$$

$$g''_4 = \frac{g'_3 + g'_4 + g'_5}{3}$$

κ.λ.π

Γενικεύοντας: Έστω  $(g_1, t_1)$ ,  $(g_2, t_2)$ ,  $(g_3, t_3)$ ,  $(g_4, t_4)$ , .....  $(g_{k-1}, t_{k-1})$ ,  $(g_k, t_k)$ ,  $(g_{k+1}, t_{k+1})$ , .....  $(g_N, t_N)$  N διαδοχικά σημεία της  $g(t)$ . Για εξομάλυνση με χρήση  $2 \cdot \mu$  γειτονικών σημείων και z επαναλήψεις, για το τυχαίο σημείο  $(g_k, t_k)$  έχουμε:

$$g'_k = \frac{g_{k-\mu} + g_{k-\mu+1} + \dots + g_{k-1} + g_k + g_{k+1} + \dots + g_{k+\mu-1} + g_{k+\mu}}{2 \cdot \mu + 1}$$

$$g''_k = \frac{g''_{k-\mu} + g''_{k-\mu+1} + \dots + g''_{k-1} + g''_k + g''_{k+1} + \dots + g''_{k+\mu-1} + g''_{k+\mu}}{2 \cdot \mu + 1}$$

.....

$$g^z_k = \frac{g^{z-1}_{k-\mu} + g^{z-1}_{k-\mu+1} + \dots + g^{z-1}_{k-1} + g^{z-1}_k + g^{z-1}_{k+1} + \dots + g^{z-1}_{k+\mu-1} + g^{z-1}_{k+\mu}}{2 \cdot \mu + 1}$$

Όπου  $(g'_k, t_k)$  το σημείο που προκύπτει από την πρώτη επανάληψη,  $(g''_k, t_k)$  το σημείο που προκύπτει από τη δεύτερη επανάληψη, ενώ  $(g^z_k, t_k)$  το εξομαλυσμένο σημείο (έπειτα από  $z$  επαναλήψεις).

### 6.1.2. Εύρεση ουσιαστικής αρχής $O_1$

Σύμφωνα με το ευρωπαϊκό πρότυπο EN 60060-2 η ουσιαστική αρχή  $O_1$  είναι το σημείο τομής με τον άξονα του χρόνου, μιας ευθείας γραμμής που φέρνουμε ως εφαπτομένη στο πιο απότομο τμήμα του μετώπου της βηματικής απόκρισης.

Για να βρούμε το πιο απότομο σημείο του μετώπου  $S_{max}$ , υπολογίζουμε την κλίση κάθε σημείου (μετά την εξομάλυνση) και επιλέγουμε αυτό με τη μεγαλύτερη ( $\lambda_{max}$ ). Η κλίση κάθε σημείου είναι ο μέσος όρος των κλίσεων των ευθειών που συνδέουν το σημείο αυτό με τα δύο γειτονικά του. Από το σημείο  $S_{max}$  φέρνουμε μία ευθεία με κλίση τη  $\lambda_{max}$  και εκεί που τέμνει τον άξονα  $t$  είναι το σημείο  $O_1$ .

Έστω για παράδειγμα τα διαδοχικά σημεία της απόκρισης (μετά την εξομάλυνση),  $(g_k, t_k)$ ,  $(g_l, t_l)$ ,  $(g_m, t_m)$ . Η κλίση του σημείου  $(g_l, t_l)$  βρίσκεται ως εξής:

Αρχικά υπολογίζουμε τις κλίσεις των ευθειών που συνδέουν το  $(g_l, t_l)$  με τα δύο γειτονικά του σημεία. Έστω  $\lambda_{kl} = \frac{g_l - g_k}{t_l - t_k}$  η κλίση που δίνουν τα σημεία  $(g_k, t_k)$ ,  $(g_l, t_l)$

και  $\lambda_{lm} = \frac{g_m - g_l}{t_m - t_l}$  η κλίση της ευθείας που συνδέει τα σημεία  $(g_l, t_l)$ ,  $(g_m, t_m)$ . Ο μέσος

όρος των κλίσεων που είναι και η κλίση του  $(g_l, t_l)$  είναι  $\lambda_l = \frac{\lambda_{kl} + \lambda_{lm}}{2}$ . Αν αυτή είναι

και η μέγιστη, τότε στο σημείο  $(g_l, t_l)$  φέρνουμε εφαπτομένη με κλίση  $\lambda_l$  και την προεκτείνουμε μέχρι τον άξονα  $t$ . Το σημείο όπου η εφαπτομένη τέμνει τον άξονα του χρόνου είναι η ουσιαστική αρχή  $O_1$ .

### 6.1.3. Εύρεση χρονικών παραμέτρων απόκρισης

Μετά την εξομάλυνση της βηματικής απόκρισης και την εύρεση του σημείου  $O_1$  υπολογίζονται οι χρονικές παράμετροι απόκρισης όπως ορίζονται από το πρότυπο.

Ο χρόνος απόκρισης  $T(t)$  είναι το ολοκλήρωμα από το  $O_1$  ως το  $t$  του  $1-g(t)$  αλλά με το αρχικό κομμάτι της  $g(t)$  αντικατεστημένο από την ευθεία  $O_1 S_{max}$ .

$$T(t) = \int_{0_1}^t (1 - g(\tau)) d\tau$$

Ενώ

$$T_1(t) = \int_0^{0_1} (1 - g(\tau)) d\tau = 0$$

Ο πειραματικός χρόνος απόκρισης  $T_N$  είναι η τιμή του χρόνου απόκρισης στο  $t_{\max}$ . Δηλαδή,  $T_N = T(t_{\max})$ .

Ο μερικός χρόνος απόκρισης  $T_\alpha$  είναι η μέγιστη τιμή του χρόνου απόκρισης και βρίσκεται από τη σχέση  $T_\alpha = T(t_1)$ , όπου  $t_1$  είναι η χρονική στιγμή για την οποία η  $g(t)$  γίνεται για πρώτη φορά 1,  $g(t_1) = 1$ .

Η υπερύψωση  $\beta$  είναι η ποσότητα κατά την οποία η μέγιστη τιμή της  $g(t)$  ξεπερνά τη μονάδα. Δηλαδή  $\beta = g_{\max} - 1$ .

Ο αρχικός χρόνος παραμόρφωσης  $T_0$  είναι το εμβαδόν της περιοχής που περικλείεται από τον άξονα  $t$ , την  $g(t)$  και την ευθεία  $O_1 S_{\max}$ .

Τέλος, ο χρόνος ισορρόπησης  $t_s$  βρίσκεται επιλύοντας τη σχέση:

$$|T_N - T(t)| < 0,02 \cdot t_s$$

## **7. Εφαρμογή λογισμικού για τον υπολογισμό των παραμέτρων απόκρισης συστημάτων μέτρησης**

### **7.1. Γενικά στοιχεία**

Στα πλαίσια της διπλωματικής αυτής εργασίας, αναπτύχθηκε μια εξειδικευμένη εφαρμογή λογισμικού μέσω της οποίας υπολογίζονται με αυτοματοποιημένο τρόπο οι παράμετροι απόκρισης ενός συστήματος μέτρησης. Η επιλογή της ανάπτυξης μιας εφαρμογής κρίθηκε σκόπιμη και προτιμότερη έναντι άλλων λύσεων, όπως η επεξεργασία των δεδομένων με το Microsoft Excel ή το υπολογιστικό περιβάλλον Matlab, καθώς παρέχει μεγαλύτερη ευελιξία και ταχύτητα, πλεονεκτήματα που αντισταθμίζουν και με το παραπάνω τον απαιτούμενο χρόνο για την υλοποίησή της.

Η εφαρμογή αναπτύχθηκε σε γλώσσα προγραμματισμού Java, η οποία είναι μια σύγχρονη αντικειμενοστραφής γλώσσα με πολύ καλή υποστήριξη για δημιουργία γραφικών διεπαφών χρήστη. Καθώς η εν λόγω εφαρμογή δεν έχει ιδιαίτερες υπολογιστικές απαιτήσεις, η Java είναι μια πολύ λογική επιλογή. Η εφαρμογή λαμβάνει ως είσοδο τις τιμές της τάσης του παλμού σε συγκεκριμένες χρονικές στιγμές, όπως μπορούν να προκύψουν από πραγματικές μετρήσεις ή μέσω κάποιας προσομοίωσης.

Στα παρακάτω περιγράφονται αναλυτικά οι δυνατότητες αυτής της εφαρμογής, δίνονται βήμα βήμα οι οδηγίες χρήσης της και επιπλέον, παρουσιάζονται τα κυριότερα σημεία του κώδικά της, ώστε να είναι εφικτή η τροποποίηση και αξιοποίηση αυτής από κάθε ενδιαφερόμενο. Με την χρήση της εφαρμογής γίνεται σημαντική εξοικονόμηση χρόνου κατά την επεξεργασία πειραματικών αποτελεσμάτων ή αποτελεσμάτων από προσομοιώσεις και δίνεται η δυνατότητα να εξαχθούν καλύτερα συμπεράσματα, βασιζόμενα σε μεγαλύτερο πλήθος μετρήσεων.

### **7.2. Δυνατότητες της εφαρμογής**

Η εφαρμογή διαβάζει κατ' αρχήν τα δεδομένα της κυματομορφής που θέλουμε να αναλύσουμε από ένα αρχείο με το κατάλληλο format και πιο συγκεκριμένα από ένα CSV (Comma Separated Values) αρχείο με δύο δεκαδικές τιμές σε κάθε γραμμή,

χωριζόμενες από το σύμβολο ;. Η πρώτη είναι η χρονική στιγμή, ενώ η δεύτερη η τιμή της τάσης της βηματικής απόκρισης. Παράλληλα, σχεδιάζει την γραφική παράσταση του παλμού καθιστώντας εφικτή την οπτική εποπτεία και επιβεβαίωση των δεδομένων εισόδου.

Στη συνέχεια είναι σε θέση να εξομαλύνει τα αρχικά δεδομένα, παίρνοντας τον μέσο όρο κάθε σημείου με κάποιον επιλέξιμο αριθμό γειτονικών σημείων. Η διαδικασία της εξομάλυνσης μπορεί να επαναληφθεί για όσες φορές επιθυμούμε. Με την ολοκλήρωση της διαδικασίας η νέα καμπύλη που προκύπτει από τα εξομαλυσμένα δεδομένα σχεδιάζεται στο ίδιο γράφημα με την αρχική καμπύλη αλλά με διαφορετικό χρώμα, έτσι ώστε να μπορεί και πάλι να ελεγχθεί ποιοτικά το αποτέλεσμα της εν λόγω διαδικασίας.

Τέλος γίνεται ο υπολογισμός των παραμέτρων που μας ενδιαφέρουν, οι τιμές των οποίων εμφανίζονται στην οθόνη του υπολογιστή. Πιο συγκεκριμένα, υπολογίζονται από τα δεδομένα εισόδου η ουσιαστική αρχή  $O_1$ , η υπέρταση  $b$ , ο μερικός χρόνος απόκρισης  $T_w$ , ο πειραματικός χρόνος απόκρισης  $T_N$ , ο χρόνος ισορροπίας  $T_s$ , καθώς και ο αρχικός χρόνος παραμόρφωσης  $T_0$ . Επίσης γίνεται σχεδιασμός της καμπύλης απόκρισης στο χρόνο, σε ξεχωριστό γράφημα από τις δύο προηγούμενες καμπύλες.

Η εφαρμογή υποστηρίζει την εμφάνιση οσονδήποτε καμπύλων στο χώρο των δύο γραφημάτων που περιλαμβάνει, ώστε να μπορεί να γίνει σύγκριση διαφόρων περιπτώσεων, αν και η δυνατότητα αυτή μπορεί να οδηγήσει και σε μη πρακτικές και αξιοποιήσιμες απεικονίσεις. Γι' αυτό και κρίνεται σκόπιμη η χρήση του κουμπιού καθαρισμού των γραφημάτων και των αποτελεσμάτων πριν από κάθε έναν ή δύο νέους υπολογισμούς, όπως περιγράφεται στη συνέχεια.

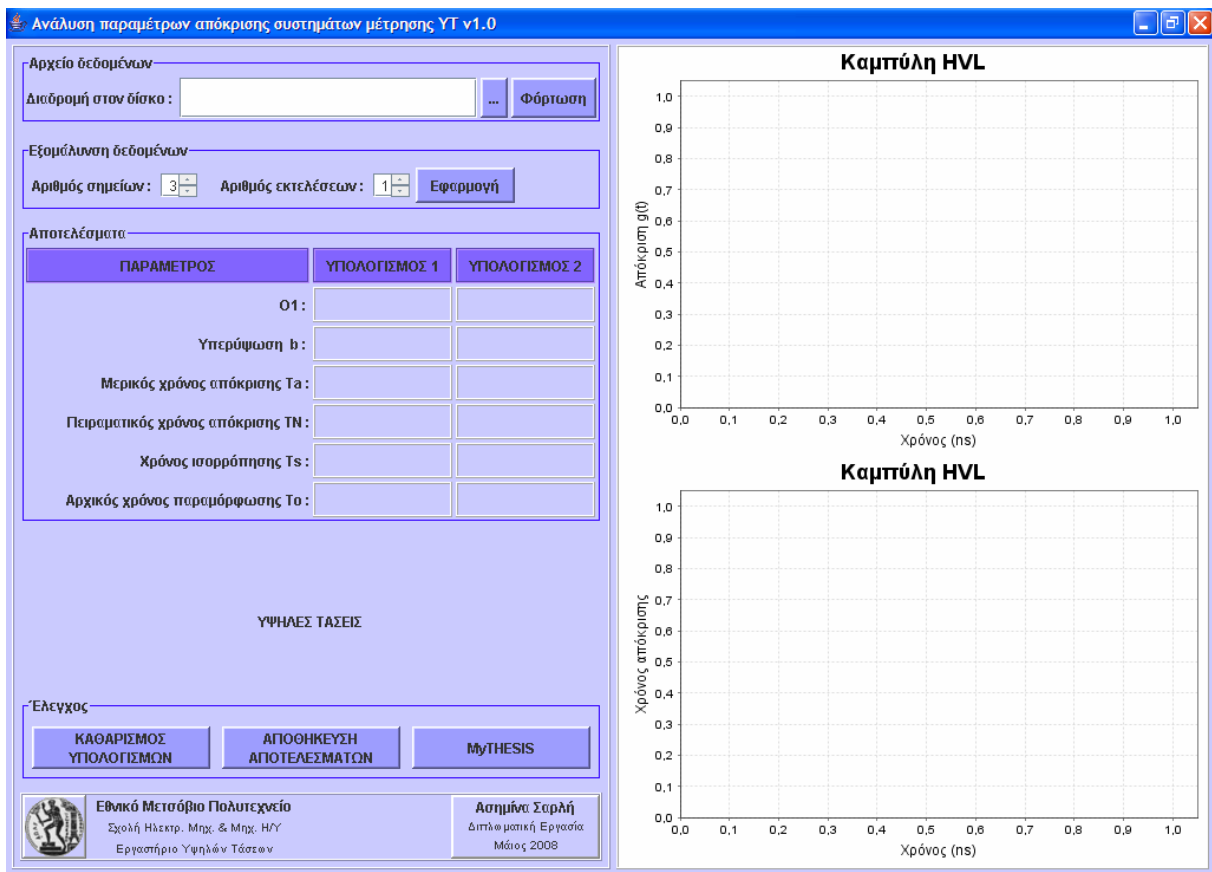
Θα πρέπει να σημειωθεί ότι εφόσον η εφαρμογή είναι γραμμένη σε Java, μπορεί να χρησιμοποιηθεί σε υπολογιστή με οποιοδήποτε λειτουργικό σύστημα υποστηρίζει και έχει εγκατεστημένο το περιβάλλον εκτέλεσης της Java (π.χ. MS Windows, Linux κ.α.). Στα πλαίσια της εργασίας χρησιμοποιήθηκε υπολογιστής με Windows XP, οπότε όλες οι οδηγίες εκτέλεσης αφορούν αυτό το λειτουργικό σύστημα.



## 7.3. Χρήση της εφαρμογής

### 7.3.1. Εκκίνηση

Για να ξεκινήσουμε την εφαρμογή εκτελούμε το αρχείο run.bat που βρίσκεται στον φάκελο HVL, κάνοντας διπλό κλικ πάνω του με το ποντίκι. Εμφανίζεται τότε ένα μαύρο παράθυρο εντολών, που χρησιμοποιείται κυρίως για debugging (εντοπισμό και διόρθωση λαθών στον κώδικα), καθώς και το γραφικό περιβάλλον της εφαρμογής, το οποίο φαίνεται στην Εικόνα 12.

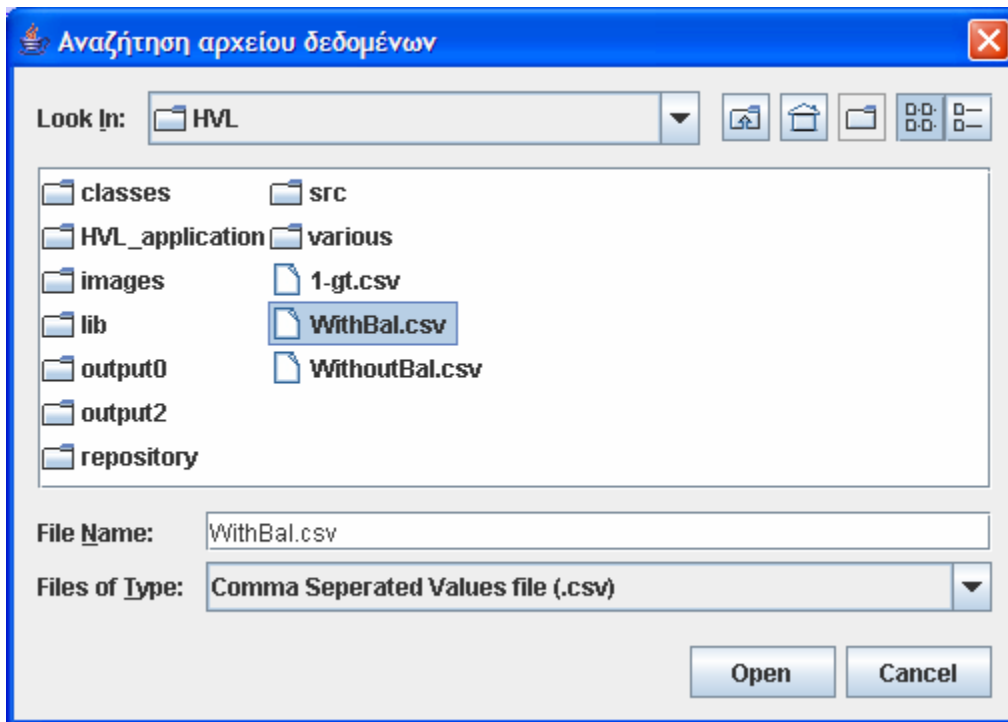


Εικόνα 12: Το γραφικό περιβάλλον της εφαρμογής

### 7.3.2. Επιλογή αρχείου δεδομένων

Το πρώτο πράγμα που έχουμε να κάνουμε στη συνέχεια, είναι να επιλέξουμε το αρχείο CSV στο οποίο υπάρχουν αποθηκευμένες οι μετρήσεις που θέλουμε να επεξεργαστούμε. Πατάμε το κουμπί με τις τρεις τελείες στο πάνω μέρος της γραφικής διεπαφής χρήστη (Graphical User Interface, GUI) της εφαρμογής, επιλέγουμε στο νέο παράθυρο που εμφανίζεται (Εικόνα 13) το αρχείο CSV και πατάμε το κουμπί *Open* κάτω δεξιά. Ως αποτέλεσμα εμφανίζεται στο αντίστοιχο πεδίο του GUI η πλήρης διαδρομή στο δίσκο που οδηγεί στο επιλεγμένο αρχείο. Εναλλακτικά θα μπορούσαμε

να καταχωρήσουμε αυτή την διαδρομή απευθείας στο πεδίο. Να σημειωθεί επίσης ότι αν το αρχείο βρίσκεται αποθηκευμένο στον ίδιο φάκελο με το αρχείο run.bat το οποίο τρέξαμε για να ξεκινήσουμε την εφαρμογή, αρκεί να γράψουμε στο πεδίο το όνομα του αρχείου, π.χ. WithBal.csv.



Εικόνα 13: Παράθυρο επιλογής αρχείου

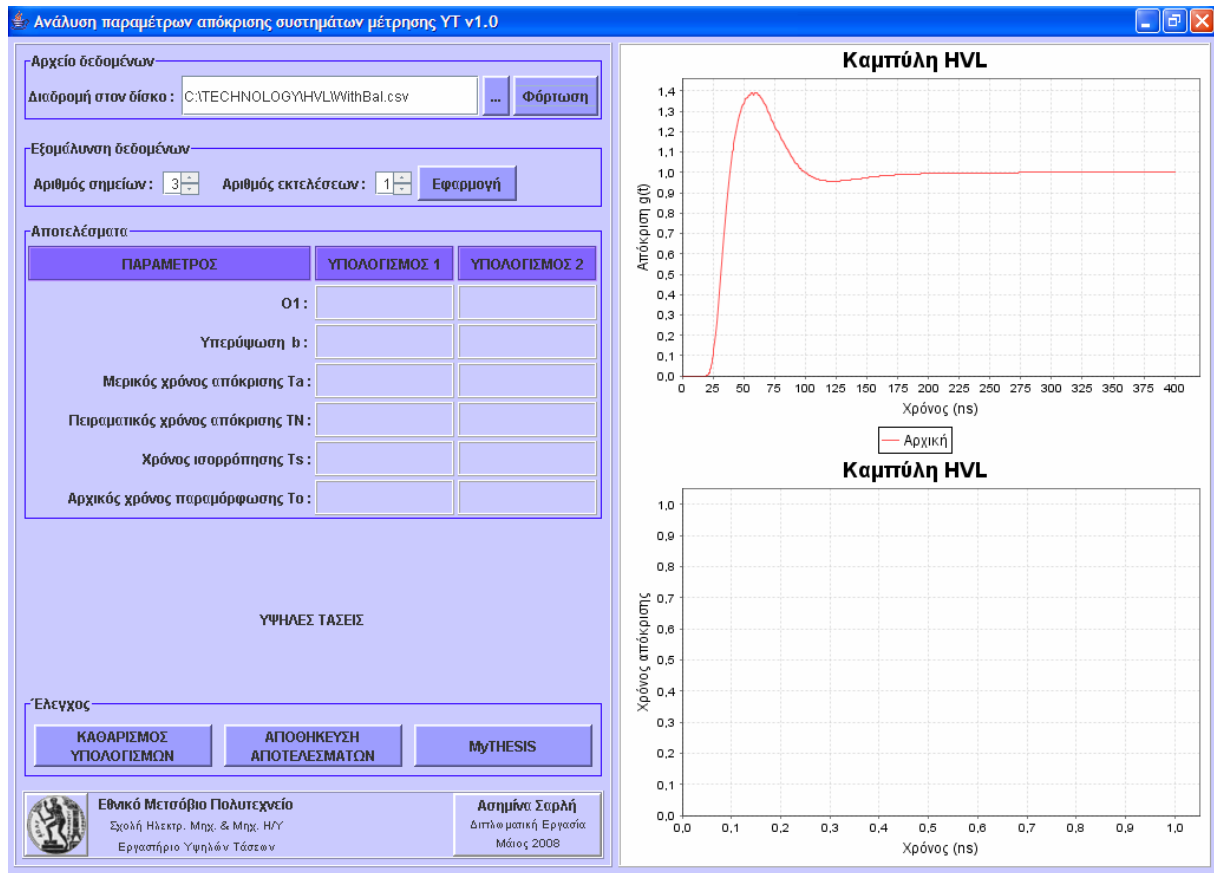
### 7.3.3. Φόρτωση δεδομένων

Ακολουθώς μπορούμε να πατήσουμε το κουμπί *Φόρτωση* οπότε και θα γίνει φόρτωση των δεδομένων του αρχείου στην εφαρμογή και παράλληλα θα γίνει η γραφική τους απεικόνιση στο πάνω δεξιά γράφημα του GUI. Η Εικόνα 14 δείχνει το GUI με την αρχική καμπύλη που θα επεξεργαστούμε.

### 7.3.4. Ρύθμιση επιλογών εξομάλυνσης

Το επόμενο βήμα είναι να ρυθμίσουμε τις επιλογές για την εξομάλυνση των δεδομένων, δηλαδή τον αριθμό των γειτονικών σημείων που θα χρησιμοποιηθούν κατά τον υπολογισμό της νέας τιμής καθώς και τον αριθμό των επαναλήψεων της διαδικασίας εξομάλυνσης. Ο αριθμός των σημείων κυμαίνεται από 0 έως 6, όπου 0 ουσιαστικά σημαίνει όχι εξομάλυνση ενώ με 6 η νέα τιμή για κάθε σημείο προκύπτει ως ο μέσος όρος τιμών των έξι προηγούμενων, των έξι επόμενων και του εν λόγω σημείου. Ο αριθμός εκτελέσεων μεταβάλλεται από 0 έως 5, όπου 0 σημαίνει και πάλι

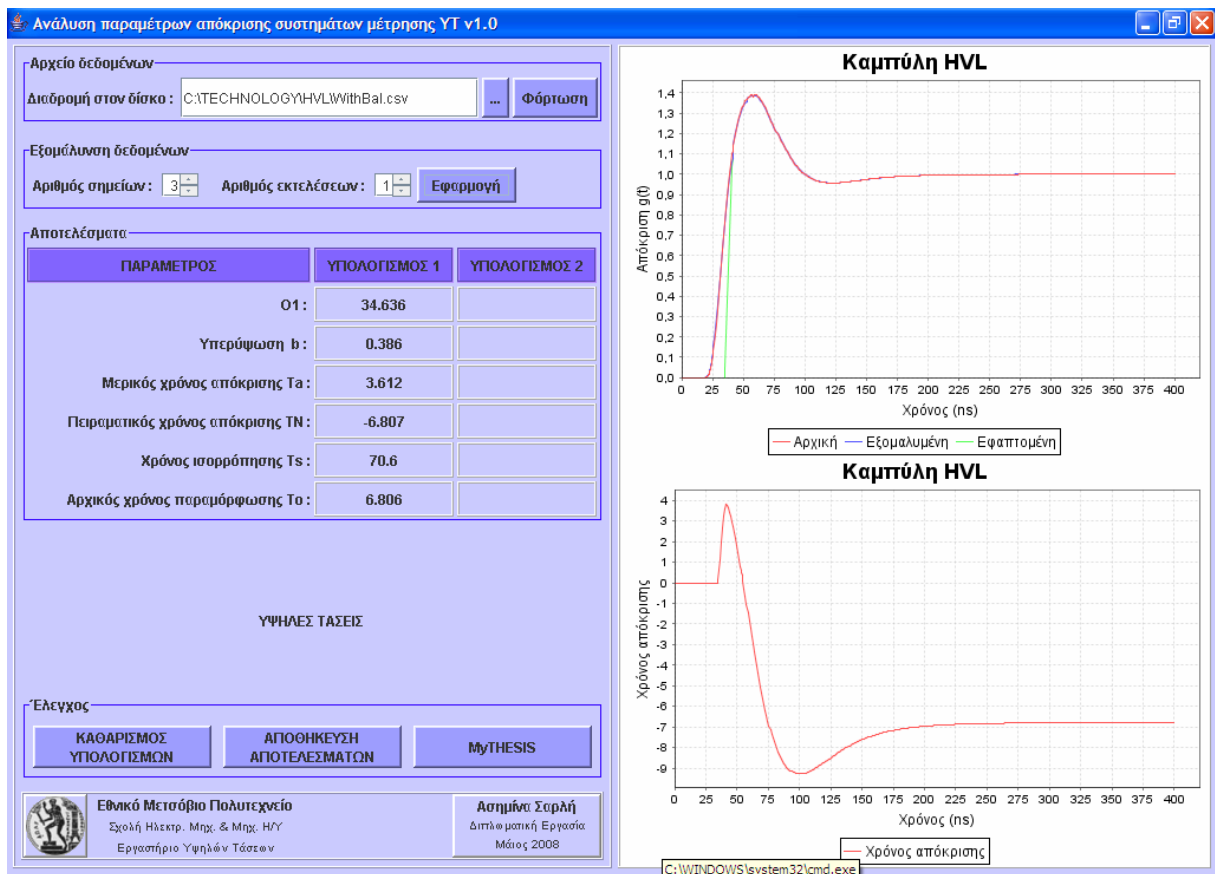
όχι εξομάλυνση ενώ αν επιλέξουμε 5 όλη η διαδικασία της εξομάλυνσης θα εκτελεστεί 5 διαδοχικές φορές.



Εικόνα 14: Το GUI μετά την φόρτωση της αρχικής καμπύλης

### 7.3.5. Εφαρμογή διαδικασίας – υπολογισμός παραμέτρων

Η εντολή για την εφαρμογή της διαδικασίας εξομάλυνσης και την εκτέλεση των υπολογισμών δίνεται πατώντας το κουμπί *Εφαρμογή*. Η νέα καμπύλη  $g(t)$  που προκύπτει από τα εξομαλυσμένα δεδομένα απεικονίζεται στο ίδιο γράφημα με την αρχική  $g(t)$  (πάνω δεξιά στο GUI) όπου σχεδιάζεται επίσης και η εφαπτομένη ευθεία στο σημείο μέγιστης κλίσης της νέας  $g(t)$ . Παράλληλα στο γράφημα κάτω δεξιά σχεδιάζεται η καμπύλη του χρόνου απόκρισης  $T(t)$ . Τέλος στον πίνακα αποτελεσμάτων στο κεντρικό αριστερό μέρος του GUI παρουσιάζονται οι τιμές των παραμέτρων απόκρισης του συστήματος μέτρησης και πιο συγκεκριμένα η ουσιαστική αρχή  $O_1$ , η υπέρταση  $b$ , ο μερικός χρόνος απόκρισης  $T_a$ , ο πειραματικός χρόνος απόκρισης  $T_N$ , ο χρόνος ισορρόπησης  $T_s$ , καθώς και ο αρχικός χρόνος παραμόρφωσης  $T_o$  όπως φαίνεται ενδεικτικά στην Εικόνα 15.



Εικόνα 15: Παρουσίαση των παραμέτρων απόκρισης

Τα αποτελέσματα του πρώτου υπολογισμού καταχωρούνται στην στήλη του πίνακα *Υπολογισμός 1* ενώ του δεύτερου στην στήλη *Υπολογισμός 2*. Από εκεί και πέρα και εφόσον δεν επιλέξουμε να γίνει *Καθαρισμός Υπολογισμών* πατώντας το αντίστοιχο κουμπί, τα αποτελέσματα του τρίτου υπολογισμού θα καταχωρηθούν και πάλι στην στήλη *Υπολογισμός 1* κ.ο.κ. Παράλληλα στα γραφήματα δεξιά θα σχεδιάζονται και όλες οι αντίστοιχες καμπύλες όπως δείχνει και η Εικόνα 16.

Τέλος θα πρέπει να σημειωθεί ότι αν πατήσουμε ξανά το κουμπί *Εφαρμογή* χωρίς να έχουμε μεταβάλλει την τιμή κάποιας από τις επιλογές εξομάλυνσης ή το αρχείο εισόδου των δεδομένων, δεν εκτελείται καμία ενέργεια.

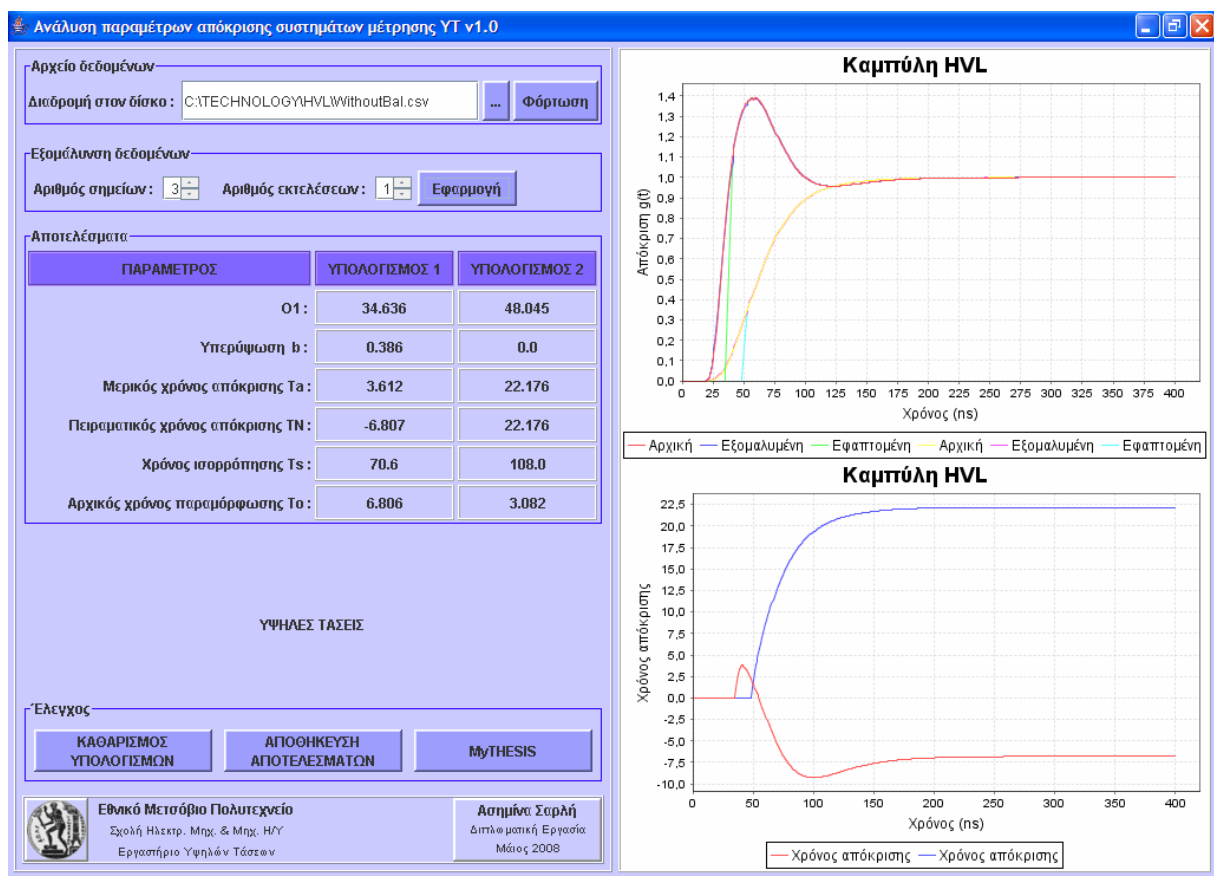
### 7.3.6. Αποθήκευση αποτελεσμάτων

Μπορούμε να σώσουμε τα αποτελέσματα της ανάλυσης πατώντας το κουμπί *Αποθήκευση Αποτελεσμάτων*. Τότε δημιουργείται, αν δεν υπάρχει ήδη, ένας φάκελος με το όνομα *output* στον ίδιο φάκελο με το αρχείο *run.bat* το οποίο τρέξαμε για να ξεκινήσουμε την εφαρμογή. Μέσα στον φάκελο *output* δημιουργούνται τα αρχεία με

όνομα xxxxxxxxxxx.txt, xxxxxxxxxxx\_a.jpg και xxxxxxxxxxx\_b.jpg όπου xxxxxxxxxxx ένα νούμερο που σχετίζεται με την ώρα του υπολογιστή και είναι μοναδικό κάθε φορά. Το πρώτο αρχείο είναι αρχείο κειμένου και αναφέρει το μονοπάτι του αρχείου δεδομένων εισόδου που χρησιμοποιήθηκε, τις επιλογές εξομάλυνσης και τις τιμές των παραμέτρων απόκρισης όπως παρακάτω:

```

Ανάλυση παραμέτρων απόκρισης συστημάτων μέτρησης ΥΤ
Δεδομένα από το αρχείο C:\TECHNOLOGY\HVL\WithBal.csv
Αριθμός σημείων : 0      Αριθμός εκτελέσεων : 0
O1      : 24.328
b       : 0.389
Ta      : 7.115
TN      : -3.735
Ts      : 70.6
To      : 0.227
    
```



Εικόνα 16: Παρουσίαση παραμέτρων απόκρισης για δύο διαφορετικά αρχεία δεδομένων εισόδου

Τα άλλα δύο αρχεία είναι εικόνες JPG οι οποίες περιέχουν τα δύο γραφήματα της εφαρμογής, το πάνω δεξιά με την απόκριση  $g(t)$  και το κάτω δεξιά με τον χρόνο απόκρισης  $T(t)$ , με όλες τις καμπύλες που αυτά απεικόνιζαν την ώρα της αποθήκευσης. Έτσι είναι πολύ απλή η περαιτέρω αξιοποίηση των αποτελεσμάτων της ανάλυσης μέσω π.χ. της ενσωμάτωσής τους σε κάποιο σχετικό έγγραφο κτλ.

Θα πρέπει επίσης να σημειωθεί ότι τα περιεχόμενα του αρχείου *xxxxxxxxxx.txt* αποθηκεύονται και στο αρχείο *ALL\_RESULTS.txt* χωρίς να διαγράφονται τα υπόλοιπα δεδομένα που τυχόν περιέχει αυτό. Έτσι μπορούμε να εκτελέσουμε μια σειρά μετρήσεων και να έχουμε σε ένα και μόνο αρχείο συγκεντρωμένα όλα τα αριθμητικά αποτελέσματα. Γενικά προτείνεται μετά από κάθε σειρά αναλύσεων που πραγματοποιούμε με την βοήθεια της εφαρμογής, να σώζουμε όλο τον φάκελο *output* με κάποιο διαφορετικό όνομα, οπότε την επόμενη φορά θα δημιουργηθεί πάλι ένας φάκελος *output* με τα νέα αποτελέσματα κ.ο.κ. επιτρέποντάς μας να τα αρχειοθετούμε τακτοποιημένα.

### **7.3.7. Καθαρισμός Υπολογισμών**

Καθώς η ταυτόχρονη απεικόνιση πολλών γραφικών παραστάσεων στο ίδιο γράφημα δεν είναι παραγωγική και αξιοποιήσιμη, χρειάζεται μετά από το πολύ τέσσερις κύκλους ανάλυσης αποτελεσμάτων να κάνουμε καθαρισμό των υπολογισμών πατώντας το αντίστοιχο κουμπί στο GUI. Τότε τα γραφήματα και ο πίνακας αποτελεσμάτων αρχικοποιούνται, όλες οι καμπύλες και οι τιμές διαγράφονται, και είναι έτοιμα για νέες μετρήσεις.

### **7.3.8. Αυτοματοποιημένη εκτέλεση σειράς υπολογισμών**

Ειδικά για τις ανάγκες της διπλωματικής αυτής εργασίας, υλοποιήθηκε μια διαδικασία η οποία επιτρέπει την αυτοματοποιημένη εκτέλεση μιας σειράς υπολογισμών και η οποία ενεργοποιείται με το πάτημα του κουμπιού *ΜΑΖΙΚΟΙ ΥΠΟΛΟΓΙΣΜΟΙ*. Πιο συγκεκριμένα, πατώντας το κουμπί αυτό γίνεται υπολογισμός των παραμέτρων απόκρισης για όλους τους συνδυασμούς επιλογών εξομάλυνσης με *Αριθμό Σημείων* από 1 έως 6 και *Αριθμό Εκτελέσεων* από 1 έως 5. Τα δεδομένα εισόδου είναι πάντα τα ίδια, από το αρχείο CSV που έχει επιλεγεί. Τα αποτελέσματα σώζονται αυτόματα στον φάκελο *output* με ονόματα αρχείων *filename-x-y.txt*,

*filename-x-y\_a.jpg* και *filename-x-y\_b.jpg*, όπου *filename* το όνομα του αρχείου CSV, *x* ο αριθμός των σημείων και *y* ο αριθμός των εκτελέσεων. Παράλληλα, όλες οι αριθμητικές τιμές αποθηκεύονται όπως πάντα και στο αρχείο *ALL\_RESULTS.txt*.

Τέλος θα πρέπει να αναφερθεί ότι προκειμένου να μπορεί να γίνεται παράλληλα και ένας ποιοτικός και οπτικός έλεγχος της εν λόγω διαδικασίας, έχει εισαχθεί μεταξύ των διαφόρων βημάτων της μια τεχνητή χρονική καθυστέρηση των 100ms.

## **7.4. Τα κυριότερα σημεία του κώδικα της εφαρμογής**

Στις επόμενες υποενότητες αναφέρονται αναλυτικά τα κυριότερα σημεία του κώδικα της εφαρμογής, έτσι ώστε να είναι δυνατή η τροποποίησή της και η πιθανή προσαρμογή της σε μελλοντικές ανάγκες. Κάποια σχετική εξοικείωση με την γλώσσα προγραμματισμού JAVA ή γενικότερα με τον αντικειμενοστραφή προγραμματισμό θεωρείται δεδομένη. Ο πλήρης κώδικας παρατίθεται στο Παράρτημα Β.

### **7.4.1. Οι κλάσεις της εφαρμογής**

Η εφαρμογή αποτελείται από επτά (7) κλάσεις, από τις οποίες δύο είναι βοηθητικές κλάσεις που προέρχονται από την Sun (την εταιρία που αναπτύσσει και διανέμει την Java). Πρόκειται για τις κλάσεις *ExampleFileFilter* και *SwingWorker*. Η πρώτη υλοποιεί το φιλτράρισμα, με βάση τον τύπο, των αρχείων που επιτρέπεται να επιλέξει ο χρήστης όταν εμφανίζεται το αντίστοιχο παράθυρο. Η δεύτερη αφορά την παράλληλη επεξεργασία σε νέο νήμα (thread) των συναρτήσεων που θέλουμε, κάτι που είναι απαραίτητο στην περίπτωση της εφαρμογής αυτής κατά την αυτοματοποιημένη εκτέλεση της σειράς υπολογισμών που περιγράφηκε στην ενότητα 7.3.8. Οι κλάσεις αυτές χρησιμοποιήθηκαν αυτούσιες και η μόνη αλλαγή που έγινε ήταν στο όνομα του πακέτου (package) της Java (hvl για την εν λόγω εφαρμογή).

Συνεπώς οι πέντε κλάσεις που αναπτύχθηκαν είναι ονομαστικά οι ακόλουθες: *HVLgui*, *HVLchart*, *FooterPanel*, *InfoDialog* και *MassiveMeasurementsTask*. Περισσότερες πληροφορίες για την καθεμία από αυτές δίνονται στα παρακάτω, ξεκινώντας από τις πιο απλές κλάσεις που αφορούν μόνο το GUI και συνεχίζοντας με τις πιο βασικές που περιέχουν και την λειτουργικότητα της εφαρμογής.

#### **7.4.2. Η κλάση FooterPanel**

Η κλάση *FooterPanel* κληρονομεί την κλάση *JPanel* της Java και αποτελεί το κάτω αριστερό μέρος του GUI της εφαρμογής, όπου φαίνεται το έμβλημα του ΕΜΠ, το όνομα της σχολής και του εργαστηρίου αλλά και το όνομα της συγγραφέως της διπλωματικής αυτής εργασίας και της εφαρμογής. Αποτελείται από μια σειρά *JPanels* και *JLabels* και γενικά δεν περιέχει κανένα ιδιόμορφο ή δυσνόητο σημείο που να χρήζει περαιτέρω σχολιασμό.

#### **7.4.3. Η κλάση InfoDialog**

Η κλάση *InfoDialog* κληρονομεί την κλάση *JDialog* της Java και είναι ένα απλό παράθυρο ενημέρωσης που εμφανίζεται για όσο χρόνο διαρκεί η διαδικασία της αυτοματοποιημένης εκτέλεσης σειράς υπολογισμών. Περιέχει ένα *JLabel* στο οποίο γράφεται η τρέχουσα κατάσταση εργασιών, καθώς και ένα *JProgressBar* το οποίο είναι μια μπάρα προόδου που αποδίδει γραφικά το ποσοστό των εργασιών που έχουν ολοκληρωθεί.

#### **7.4.4. Η κλάση HVLchart**

Η κλάση *HVLchart* είναι ένα πάνελ που περιέχει ένα γράφημα *JFreeChart*, το οποίο ορίζεται στο πακέτο *org.jfree.chart*. Η βιβλιοθήκη λογισμικού *JFreeChart* [18] έχει αναπτυχθεί 100% με Java και παρέχεται δωρεάν. Δίνει δε την δυνατότητα για την δημιουργία, εμφάνιση, αλλά και αποθήκευση μιας μεγάλης ποικιλίας όμορφων και πρακτικών γραφημάτων. Στο διαδίκτυο υπάρχει όλο το σχετικό API καθώς και πλήθος παραδειγμάτων που σίγουρα καλύπτουν τις βασικές ανάγκες. Υπάρχει όμως διαθέσιμο για αγορά και ένα αντίστοιχο βιβλίο με αναλυτικές οδηγίες χρήσης και συμβουλές για προχωρημένες χρήσεις της βιβλιοθήκης.

Για τις ανάγκες της εφαρμογής αυτής της διπλωματικής, χρησιμοποιήθηκε ένα *JFreeChart* (η μεταβλητή *chart* στην κλάση *HVLchart*) του οποίου τα δεδομένα (η μεταβλητή *dataset*) είναι ένα σύνολο σημείων *x, y*, γι' αυτό και για την κατασκευή του *chart* χρησιμοποιήθηκε η βοηθητική στατική συνάρτηση *createXYLineChart* της κλάσης *ChartFactory*. Ο τίτλος του γραφήματος καθώς και τα ονόματα των αξόνων *X* και *Y*, περνάνε από τον κατασκευαστή της κλάσης *HVLchart* στην κατασκευή του *chart*. Η αρχικοποίηση του *chart* περιλαμβάνει ακόμα τον καθορισμό του χρώματος φόντου του γραφήματος (με την εντολή *chart.setBackgroundPaint(Color.white);*) σε



λευκό. Τέλος να αναφερθεί ότι η κλάση *HVLchart* περιλαμβάνει και δύο ακόμα συναρτήσεις, την *addSeries* με την οποία προσθέτουμε στο ίδιο γράφημα μια νέα σειρά από σημεία *x*, *y*, και την *clearAll* η οποία όταν καλείται καθαρίζει το γράφημα από όλες τις καμπύλες που απεικονίζει (διαγράφοντας όλες τις σειρές δεδομένων από την μεταβλητή *dataset*).

#### **7.4.5. Η κλάση *MassiveMeasurementsTask***

Η κλάση αυτή χρησιμοποιείται σε συνδυασμό με την κλάση *SwingWorker* που αναφέρθηκε στην ενότητα 7.4.1 και υλοποιεί σε ένα νέο νήμα (thread) την διαδικασία αυτοματοποιημένης εκτέλεσης μιας σειράς από υπολογισμούς. Η μεταβλητή *statMessage* τύπου *String* που ορίζεται, διαβάζεται περιοδικά μέσω της μεθόδου *getMessage* και εμφανίζεται στο αντίστοιχο *JLabel* του παραθύρου *InfoDialog*. Ομοίως, το ποσοστό ολοκληρωμένων εργασιών που ανατίθεται στην ακέραια μεταβλητή *completionPercent*, διαβάζεται περιοδικά μέσω της μεθόδου *getCompletionPercent* και απεικονίζεται στην μπάρα προόδου του ίδιου παραθύρου. Οι τιμές των δύο αυτών μεταβλητών αλλάζουν καθώς εκτελούνται οι εργασίες που ορίζονται στην φωλιασμένη (nested) κλάση *ActualTask*, η οποία όπως δηλώνει και το όνομά της, περιέχει τις εργασίες που θέλουμε να εκτελεστούν στο νέο νήμα.

Πιο συγκεκριμένα, η *ActualTask* ελέγχει ότι έχει όντως επιλεγεί ένα αρχείο δεδομένων εισόδου που υπάρχει και προχωρά σε ένα διπλό βρόχο *for* εκτελώντας υπολογισμούς για όλους τους συνδυασμούς επιλογών εξομάλυνσης με *Αριθμό Σημείων* από 1 έως 6 και *Αριθμό Εκτελέσεων* από 1 έως 5. Κάθε υπολογισμός περιλαμβάνει 4 βήματα:

- φόρτωση των αρχικών δεδομένων από το αρχείο και της αντίστοιχης καμπύλης στο γράφημα πάνω δεξιά
- υπολογισμό των παραμέτρων απόκρισης και εμφάνιση των αντίστοιχων καμπύλων στα γραφήματα και τιμών στον πίνακα αποτελεσμάτων
- αποθήκευση των αποτελεσμάτων στον φάκελο *output* όπως περιγράφεται στην ενότητα 7.3.6
- καθαρισμό των υπολογισμών όπως περιγράφεται στην ενότητα 7.3.7.

Για τις παραπάνω ενέργειες καλούνται οι αντίστοιχες μέθοδοι της κλάσης *HVLgui* οι οποίες παρουσιάζονται αναλυτικά στην επόμενη ενότητα. Πριν από κάθε βήμα έχει εισαχθεί μια τεχνητή χρονική καθυστέρηση που καθορίζεται από την μεταβλητή *sleeptime* σε milliseconds (100ms στην εφαρμογή), έτσι ώστε να είναι δυνατή η παρακολούθηση της όλης διαδικασίας από τον χρήστη καθώς αυτή εξελίσσεται.

#### **7.4.6. Η κλάση *HVLgui***

Η κλάση *HVLgui* είναι η κύρια κλάση της εφαρμογής, η οποία περιέχει άλλωστε και την συνάρτηση *main*. Εδώ ορίζονται τα διάφορα στοιχεία της γραφικής διεπαφής χρήστη της εφαρμογής, όπως επίσης και οι μέθοδοι που σχετίζονται με τον υπολογισμό των παραμέτρων απόκρισης του συστήματος μέτρησης με βάση τα δεδομένα από το αρχείο εισόδου.

Οι μεταβλητές *backgrndColor*, *buttonColor* και *mobColor* καθορίζουν τα χρώματα φόντου, κουμπιών και επικεφαλίδων του πίνακα αποτελεσμάτων αντίστοιχα. Ακολουθώς δηλώνονται τα αντικείμενα που σχετίζονται με την επιλογή του αρχείου δεδομένων εισόδου, όπου *inputFile* είναι το αρχείο, ενώ *browseBut* και *loadBut* είναι τα κουμπιά που πατά ο χρήστης για την αναζήτηση και επιλογή ενός αρχείου από τον δίσκο το πρώτο και την φόρτωση των δεδομένων στην εφαρμογή το δεύτερο. Καθώς η κλάση *HVLgui* υλοποιεί έναν *ActionListener*, όλα τα κουμπιά χρησιμοποιούν αυτή για την λήψη γεγονότων όπως το πάτημα του κουμπιού.

Παρακάτω ορίζονται τα *JSpinners spinner1* και *spinner2* μέσω των οποίων ο χρήστης κάνει τις επιλογές για την διαδικασία εξομάλυνσης. Η βασική τους ιδιότητα σαν κλάσεις είναι ότι επιτρέπουν την επιλογή τιμών μέσα από ένα ευπροσδιόριστο σύνολο με βάση κάποιο μοντέλο. Π.χ. στην περίπτωση της εφαρμογής αυτής, χρησιμοποιήθηκε ένα αριθμητικό μοντέλο όπου ορίζει κανείς την αρχική τιμή που θα δείχνει το *JSpinner*, την ελάχιστη και την μέγιστη τιμή καθώς και το βήμα, οπότε ο χρήστης, χρησιμοποιώντας τα βελάκια που παρέχονται από το GUI, περιορίζεται στο να επιλέγει μόνο κάποια από αυτές τις αποδεκτές και προεπιλεγμένες τιμές.

Το κουμπί *computeBut* δίνει εντολή για την εκτέλεση των υπολογισμών καλώντας την συνάρτηση *computeAction*, περνώντας της ως ορίσματα τις τιμές των

δύο *JSpinner*, μόνον εφόσον δεν έχει γίνει ήδη ο ίδιος υπολογισμός. Για τον σκοπό αυτό χρησιμοποιούνται οι μεταβλητές *lastComputedSmoothRange*, *lastComputedTimes* και *lastInputFileName* στις οποίες αποθηκεύουμε τις τελευταίες τιμές των δύο *JSpinner* και μια αναφορά στο αρχείο δεδομένων αντίστοιχα.

Έπειτα δηλώνονται οι ετικέτες τύπου *JLabel* στις οποίες αναγράφονται τα αποτελέσματα (δύο σειρές αποτελεσμάτων όπως έχει αναφερθεί), καθώς και τα κουμπιά ελέγχου *clearChartsBut*, *saveChartsBut* και *massiveMeasBut*. Το πρώτο δίνει εντολή για τον καθαρισμό των υπολογισμών καλώντας την συνάρτηση *clearChartsAction*. Το δεύτερο αποθηκεύει τα αποτελέσματα με τη βοήθεια της συνάρτησης *saveChartsAction*. Το τρίτο δημιουργεί ένα νέο αντικείμενο τύπου *MassiveMeasurementsTask* το οποίο αναθέτει στην μεταβλητή *mmtask*, εμφανίζει το παράθυρο πληροφοριών *idg* τύπου *InfoDialog* και ξεκινά την διαδικασία αυτοματοποιημένης εκτέλεσης υπολογισμών. Επίσης δημιουργεί ένα χρονόμετρο *timer* τύπου *JTimer* έτσι ώστε οι πληροφορίες που φαίνονται στο παράθυρο *idg* να ανανεώνονται κάθε 300ms. Οι μεταβλητές *mmtask*, *idg* και *timer* είναι δηλωμένες *global* στην κλάση *HVLgui* ώστε να είναι δυνατή η πρόσβαση σε αυτές και από τις υπόλοιπες κλάσεις της εφαρμογής, όπως απαιτείται.

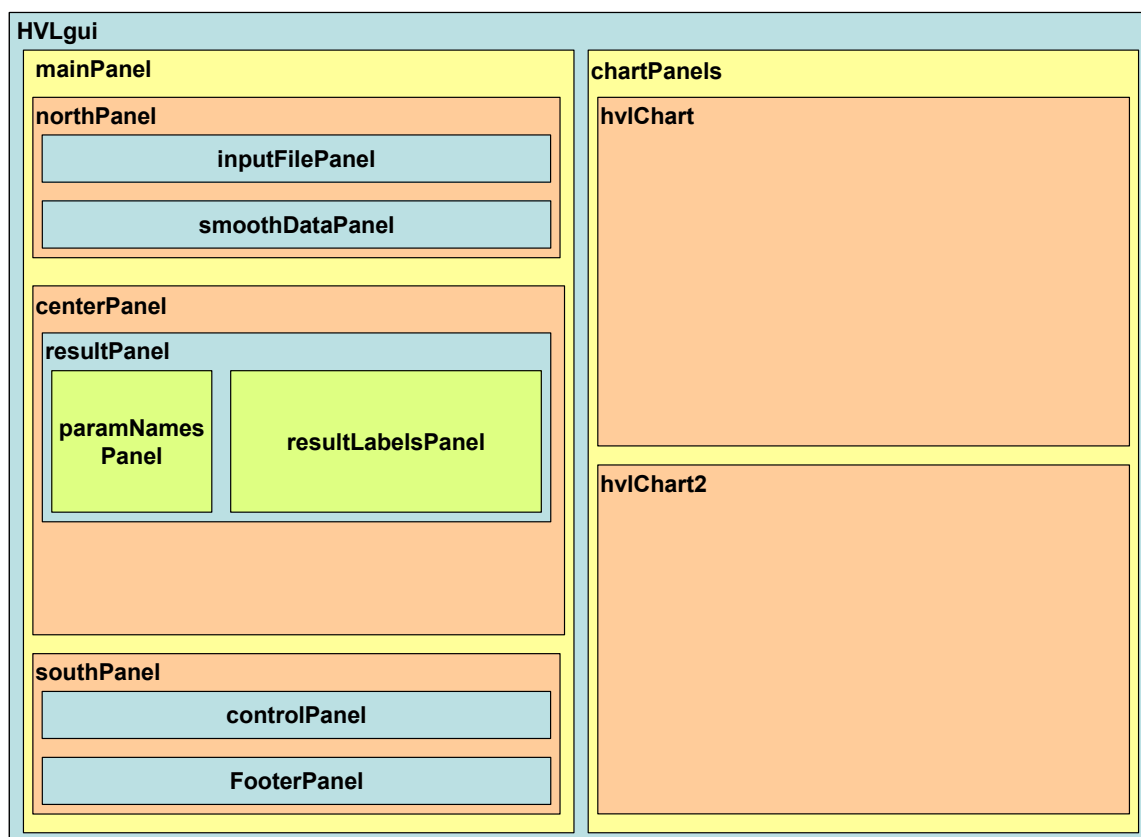
Οι μεταβλητές *hvlChart* και *hvlChart2* είναι τύπου *HVLchart* και αποτελούν τα δύο γραφήματα που βρίσκονται πάνω δεξιά και κάτω δεξιά στο GUI αντίστοιχα. Τα ζεύγη τιμών  $x$  και  $y$  των δεδομένων εισόδου, δηλαδή τα σημεία που αποτελούν την αρχική καμπύλη απόκρισης, αποθηκεύονται στα διανύσματα *xVector* και *yVector* τύπου *Vector* ως δεκαδικοί αριθμοί διπλής ακρίβειας. Ειδικά οι τιμές του χρόνου πολλαπλασιάζονται επί  $10^9$  καθώς δίνονται σε sec ενώ όλη η επεξεργασία και απεικόνιση από την εφαρμογή γίνεται σε msec.

Η ακέραια μεταβλητή *state* δηλώνει την φάση στην οποία βρίσκεται η εφαρμογή, κάτι που είναι χρήσιμο σε διάφορα σημεία της. Π.χ. αρχικά είναι *state = 0* ενώ μόλις φορτωθεί ένα αρχείο δεδομένων γίνεται *state = 1*. Αν εκτελεστεί κάποιος υπολογισμός το *state* παίρνει την τιμή 2. Έτσι ξέρουμε ότι αν δοθεί εντολή για υπολογισμό ενώ το *state* είναι 2, δεν έχει νόημα να εκτελεστεί, καθώς ο υπολογισμός έχει ήδη πραγματοποιηθεί. Αν αλλάξουν οι επιλογές εξομάλυνσης ή φορτωθεί άλλο

αρχείο δεδομένων το *state* γίνεται πάλι 1. Τέλος αν πατηθεί το κουμπί *Καθαρισμός Υπολογισμών* η μεταβλητή αυτή αρχικοποιείται πάλι στο 0.

Η ακέραια μεταβλητή *computationIndex* είναι μια ακόμα μεταβλητή ελέγχου που μετράει πόσους υπολογισμούς έχουμε κάνει από την τελευταία φορά που πατήθηκε το κουμπί *Καθαρισμός Υπολογισμών* έτσι ώστε τα αποτελέσματα να παρουσιάζονται στην κατάλληλη στήλη του αντίστοιχου πίνακα.

Από εκεί και πέρα ορίζονται αρκετά αντικείμενα τύπου *JPanel* των οποίων τα αντίστοιχα ονόματα μεταβλητών και η διάταξη στο GUI φαίνονται σχηματικά στην Εικόνα 17.



Εικόνα 17: Σχηματική απεικόνιση της διάταξης των πάνελ στο GUI

Σχετικά με την συνάρτηση *computeAction* η οποία παίρνει ως ορίσματα τις επιλογές για την εξομάλυνση (αριθμό σημείων και εκτελέσεων) και είναι υπεύθυνη για τον υπολογισμό των παραμέτρων απόκρισης, επισημαίνονται τα εξής:

- i) Για κάθε καμπύλη η συνάρτηση χρησιμοποιεί και ένα νέο διάνυσμα, π.χ. τα *xVectorSm* και *yVectorSm* περιέχουν τις τιμές *x* και *y* που προκύπτουν

από την διαδικασία εξομάλυνσης, ενώ τα  $xVectorTan$  και  $yVectorTan$  τα σημεία  $(x,y)$  της εφαπτομένης ευθείας στο σημείο μέγιστης κλίσης. Για καθεμία από τις καμπύλες  $1-g(t)$  και  $T(t)$  χρειάζεται από ένα μόνο διάνυσμα, το  $yVectorIminus$  και το  $yVectorTt$  αντίστοιχα, καθώς και οι δύο βασίζονται στις τιμές χρόνου από το διάνυσμα  $xVectorSm$ .

ii) Τα βήματα κάθε υπολογισμού είναι με τη σειρά τα ακόλουθα:

- Εξομάλυνση με ακτίνα  $smoothRange$  για  $times$  φορές.
- Αποθήκευση τιμών στα διανύσματα  $xVectorSm$  και  $yVectorSm$ .
- Εμφάνιση της αντίστοιχης καμπύλης στο  $hvlChart$ .
- Σάρωση των τιμών του διανύσματος  $yVectorSm$ , υπολογισμός της κλίσης σε κάθε σημείο ως τον μέσο όρο των κλίσεων με το προηγούμενο και το επόμενο γειτονικό του και αποθήκευση στην μεταβλητή  $maxSlope$  της τιμής που προκύπτει, εφόσον είναι μεγαλύτερη από την ήδη αποθηκευμένη. Παράλληλα, εύρεση της μέγιστης τιμής της  $g(t)$  και αποθήκευσή της στην μεταβλητή  $maxGt$ .
- Μετά το παραπάνω βήμα, έχουμε στην μεταβλητή  $maxSlope$  την μέγιστη κλίση, στις μεταβλητές  $x\_maxSlope$  και  $y\_maxSlope$  τις συντεταγμένες του σημείου μέγιστης κλίσης και στη μεταβλητή  $maxSlopeIndex$  τη θέση των αντίστοιχων τιμών  $x$  και  $y$  στα διανύσματα  $xVectorSm$  και  $yVectorSm$ .
- Έχοντας την κλίση και το σημείο μέγιστης κλίσης, μπορούμε να βρούμε το σημείο τομής της ευθείας με τον άξονα  $x$ , το οποίο είναι και το σημείο ουσιαστικής αρχής  $O_1$ . Η τιμή του αποθηκεύεται στην μεταβλητή  $xOI$ .
- Στη συνέχεια βρίσκουμε και μια τιμή  $yOI$  που θα αντιστοιχούσε σε αυτό το σημείο, ως τον μέσο όρο των τιμών  $y$  από τα δύο γειτονικά σημεία ως προς το  $xOI$ . Παρεμβάλλουμε το σημείο  $(xOI, yOI)$  στη θέση που πρέπει στα διανύσματα  $xVectorSm$  και  $yVectorSm$ .
- Εφόσον παρεμβάλλαμε ένα σημείο, ελέγχουμε μήπως πρέπει να αυξηθεί κατά ένα και το  $maxSlopeIndex$ .

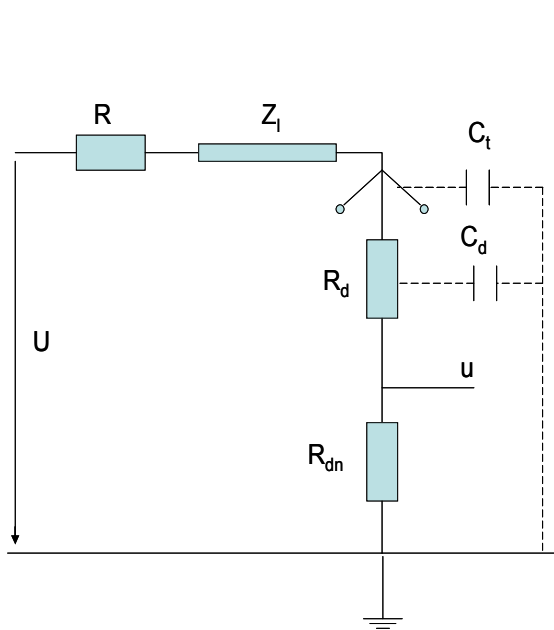
- Με βάση τον μαθηματικό τύπο της ευθείας  $y = \lambda x + \beta$  υπολογίζουμε την τιμή του  $y$  της εφαπτομένης ευθείας για κάθε τιμή του  $x$  από το διάνυσμα  $xVectorSm$  μέχρι και το σημείο μέγιστης κλίσης. Προσθέτουμε την εφαπτομένη στο *hvlChart*.
- Υπολογίζουμε την τιμή της  $1-g(t)$  η οποία είναι 0 μέχρι το σημείο ουσιαστικής αρχής  $O_1$ , 1 μείον την τιμή της εφαπτομένης μέχρι το σημείο  $xO1$  και  $1-g(t)$  αλλού.
- Υπολογίζουμε και εμφανίζουμε την υπέρταση  $\beta$ , η οποία αποθηκεύεται στην μεταβλητή  $o\_b$  και προκύπτει πολύ απλά ως  $maxGt - 1$ .
- Υπολογίζουμε τον χρόνο απόκρισης  $T(t)$  ως το ολοκλήρωμα της  $1-g(t)$ . χρησιμοποιώντας τον τύπο εμβαδού τραπεζίου. Οι τιμές αποθηκεύονται, όπως ήδη αναφέρθηκε, στο διάνυσμα  $yVectorTt$ . Παράλληλα αθροίζουμε και όλα τα εμβαδά και τα αποθηκεύουμε στην μεταβλητή  $tt$  η οποία αποτελεί και τον πειραματικό χρόνο απόκρισης  $T_N$ .
- Η αντίστοιχη καμπύλη εμφανίζεται στο *hvlChart2*.
- Σαρώνουμε τις τιμές του διανύσματος  $yVectorTt$  ψάχνοντας για ποια τιμή του  $x$ , η οποία αντιστοιχεί στην προσωρινή μεταβλητή  $x_{tt}$ , θα ισχύει η σχέση  $|experRespTime - tt| / 0.02 < x_{tt}$ , όπου  $experRespTime$  ο χρόνος  $T_N$ . Έτσι βρίσκουμε τον χρόνο ισορροπίας  $T_s$ .
- Τέλος υπολογίζουμε τον αρχικό χρόνο παραμόρφωσης  $T_0$  από τα αντίστοιχα εμβαδά των τμημάτων που ορίζονται από την  $g(t)$  και την εφαπτομένη.

Ο πλήρης κώδικας που παρατίθεται στο παράρτημα Β, περιέχει και αυτός διάφορα διευκρινιστικά σχόλια, έτσι ώστε να είναι όσο το δυνατόν πιο κατανοητός, ελέγξιμος και αξιοποιήσιμος.

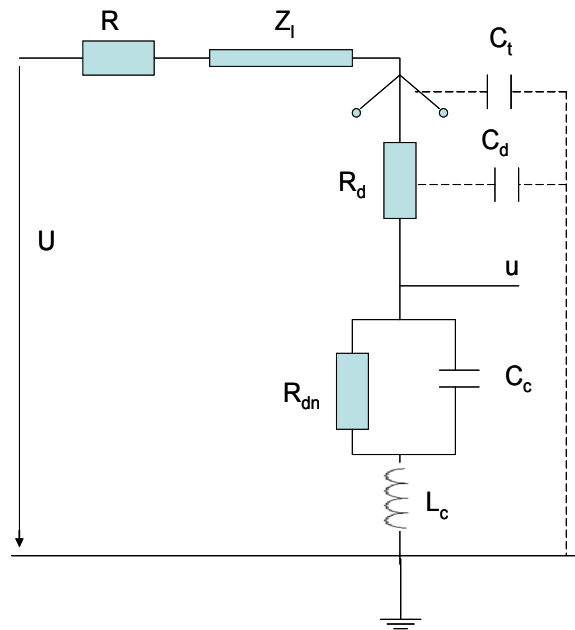
## 8. Ανάλυση δεδομένων και αποτελέσματα

Στα παρακάτω γίνεται επεξεργασία των δεδομένων που προέρχονται από δύο προσομοιώσεις μετρήσεων υψηλής τάσης με το πρόγραμμα SPICE [19], με σκοπό να υπολογιστούν οι παράμετροι απόκρισης του συστήματος μέτρησης. Το σύστημα μέτρησης που χρησιμοποιήθηκε αποτελείται από έναν ωμικό καταμεριστή με αντίσταση υψηλής τάσης  $R_d=10K\Omega$  με παράσιτη χωρητικότητα  $C_d=25pF$  ως προς γη. Η αντίσταση του κλάδου χαμηλής τάσης είναι  $R_{dn}=10\Omega$  ενώ η χωρητικότητα του ηλεκτροδίου κορυφής είναι  $C_f=30pF$ . Μία γραμμή κυματικής αντίστασης  $Z_l=500\Omega$  συνδέει τη γεννήτρια υψηλής τάσης με τον καταμεριστή και μία αντίσταση προσαρμογής  $R=500\Omega$  συνδέεται στην είσοδο της γραμμής με στόχο την προσαρμογή των ανακλώμενων κυμάτων που οδεύουν από τον καταμεριστή προς τη γεννήτρια. Το εν λόγω κύκλωμα (Εικόνα 18) αποτελεί το σύστημα μέτρησης χωρίς αντιστάθμιση. Με σκοπό την ελαχιστοποίηση των σφαλμάτων, έγινε παρέμβαση στον κλάδο χαμηλής τάσης του καταμεριστή. Μία χωρητικότητα  $C_c=1,19nF$  προστίθεται παράλληλα προς την αντίσταση  $R_{dn}$  και εν σειρά προς αυτά μια αυτεπαγωγή  $L_c=0,454\mu H$ . Το νέο σύστημα είναι αντισταθμισμένο (Εικόνα 19).

Η πρώτη σειρά δεδομένων που εξετάστηκε, αποτελεί την απόκριση σε περίπτωση συστήματος μέτρησης χωρίς αντιστάθμιση και βρίσκεται αποθηκευμένη στο αρχείο WithoutBalance.csv, τα περιεχόμενα του οποίου παρήχθησαν αυτόματα ως έξοδος του προσομοιωτή SPICE. Αντίστοιχα η δεύτερη σειρά δεδομένων είναι αποθηκευμένη στο αρχείο WithBalance.csv και αποτελεί την απόκριση του συστήματος μέτρησης με αντιστάθμιση. Ουσιαστικά σε κάθε γραμμή των δύο παραπάνω αρχείων, υπάρχει ένα ζευγάρι τιμών με τον χρόνο και την απόκριση, για όλη την διάρκεια της προσομοίωσης. Με την χρήση της εφαρμογής που αναπτύχθηκε στα πλαίσια της διπλωματικής αυτής εργασίας και παρουσιάστηκε στο προηγούμενο κεφάλαιο, ο υπολογισμός των ζητούμενων παραμέτρων είναι μια αυτοματοποιημένη και γρήγορη διαδικασία.



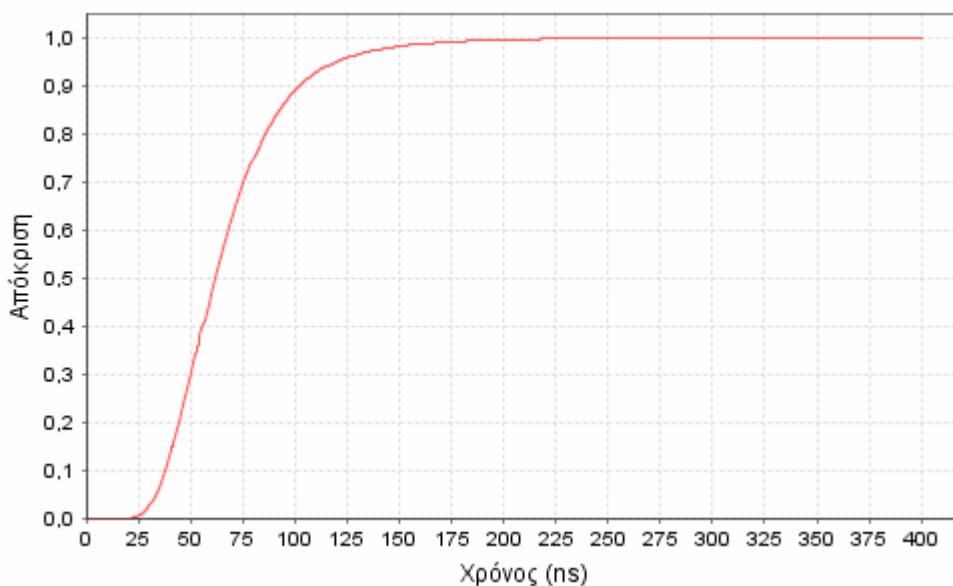
Εικόνα 18: Σύστημα μέτρησης χωρίς αντιστάθμιση



Εικόνα 19: Σύστημα μέτρησης με αντιστάθμιση

### 8.1. Απόκριση χωρίς αντιστάθμιση

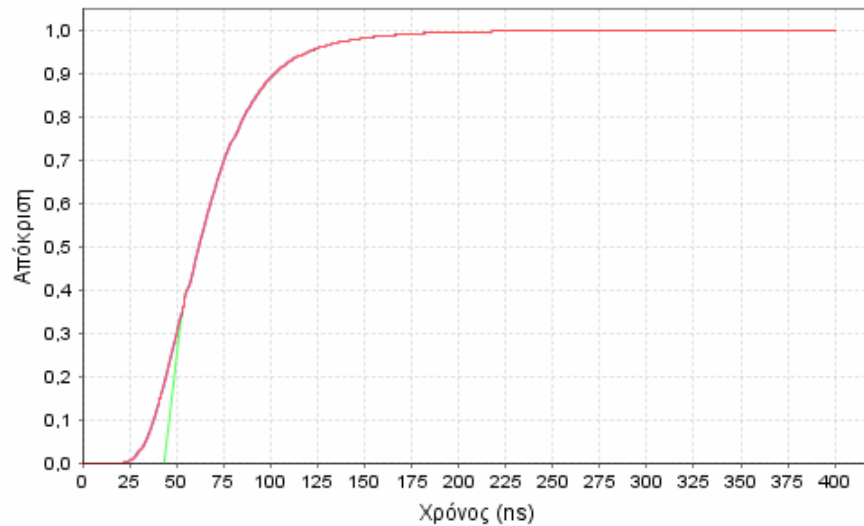
Στην Εικόνα 20 φαίνεται η γραφική παράσταση των δεδομένων εισόδου από το αρχείο WithoutBalance.csv, όπου στον οριζόντιο άξονα έχουμε τον χρόνο σε ns και στον κάθετο την κανονικοποιημένη βηματική απόκριση.



Εικόνα 20: Γραφική παράσταση απόκρισης χωρίς αντιστάθμιση

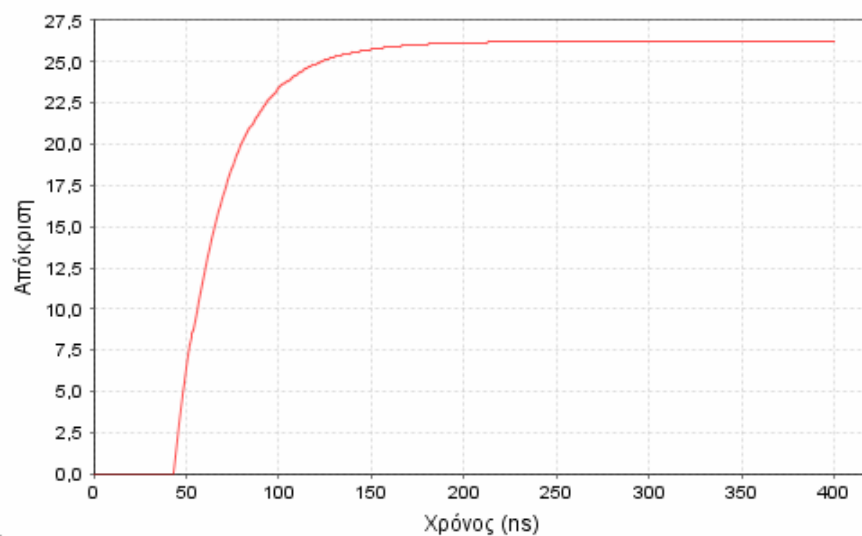


Μετά την διαδικασία εξομάλυνσης των δεδομένων εισόδου την οποία εκτελούμε 1 φορά με ακτίνα 1 σημείο (οπότε παίρνουμε τον μέσο όρο του σημείου που εξετάζουμε και των δύο γειτονικών του), προκύπτει η μπλε καμπύλη που φαίνεται στην Εικόνα 21. Θα πρέπει να σημειωθεί ότι στο ίδιο γράφημα απεικονίζεται με κόκκινη γραμμή και η αρχική καμπύλη και όπως είναι φανερό οι δυο τους ουσιαστικά συμπίπτουν.



**Εικόνα 21: Η αρχική (κόκκινη) και η εξομαλυσμένη (μπλε) καμπύλη απόκρισης, καθώς και η ευθεία  $O_1S_{max}$  (πράσινη)**

Στη συνέχεια υπολογίζεται ο χρόνος απόκρισης, του οποίου η γραφική παράσταση είναι αυτή της παρακάτω εικόνας.



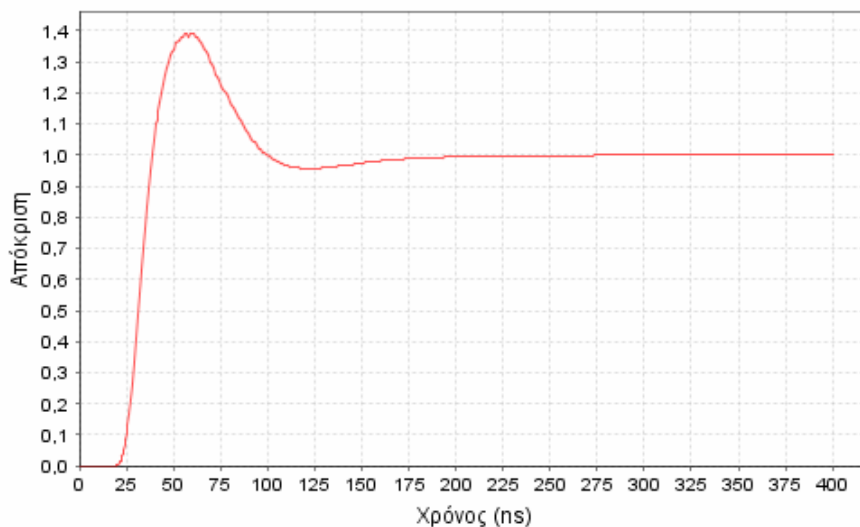
**Εικόνα 22: Χρόνος απόκρισης  $T(t)$**

Με βάση τις διαδικασίες που αναφέρονται αναλυτικά στο κεφάλαιο 6, υπολογίζονται από την εφαρμογή οι τιμές των παραμέτρων για την περίπτωση αυτή, οι οποίες παρουσιάζονται στον ακόλουθο πίνακα:

Παράμετρος	Τιμή
$O_1$	43.251
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	26.221
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	26.221
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	2.186

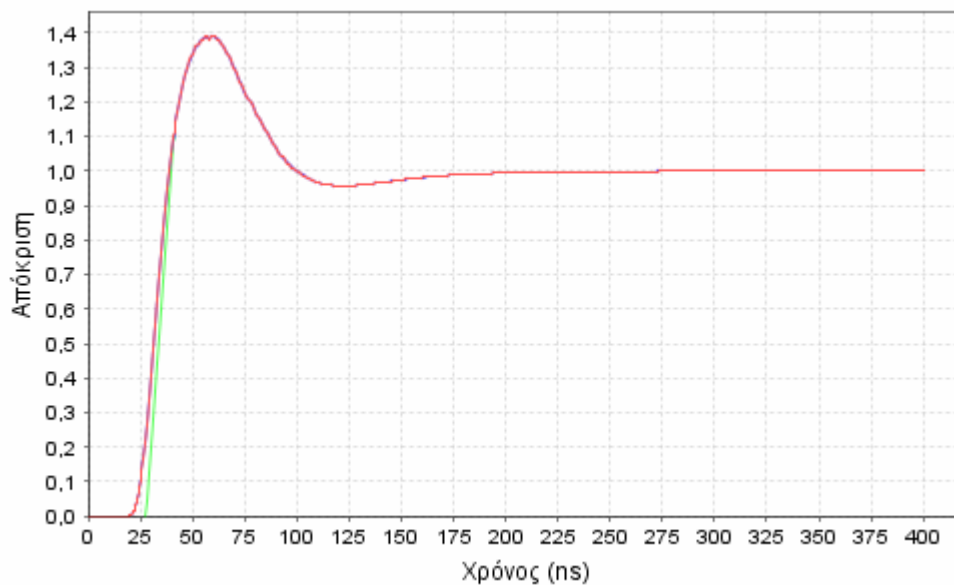
## 8.2. Απόκριση με αντιστάθμιση

Αντίστοιχα βήματα κάνουμε και στην περίπτωση της δεύτερης σειράς των δεδομένων που αναλύουμε και τα οποία αφορούν απόκριση του συστήματος με αντιστάθμιση. Η αρχική γραφική παράσταση που προκύπτει από τα ζεύγη σημείων του αρχείου WithBalance.csv φαίνεται στην Εικόνα 23.



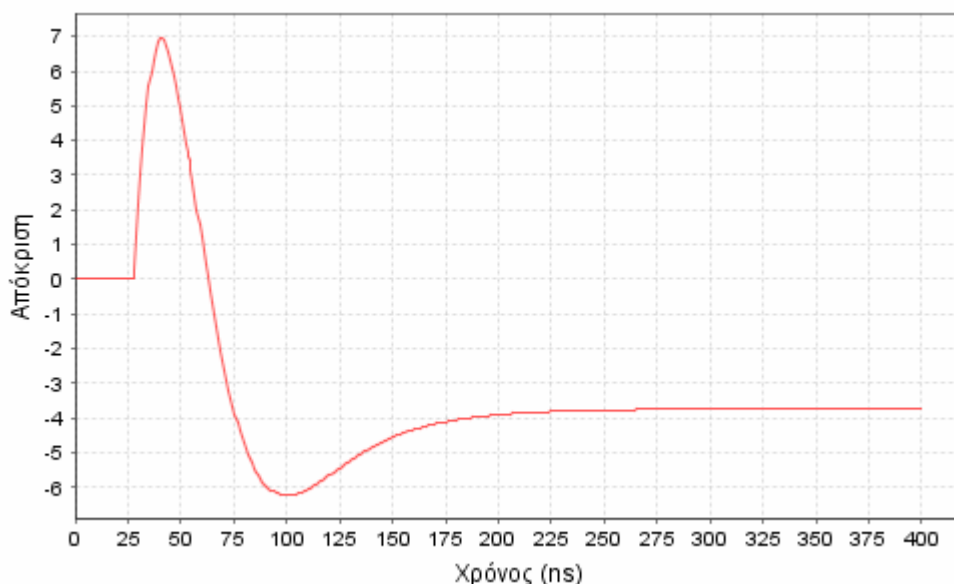
Εικόνα 23: Γραφική παράσταση απόκρισης με αντιστάθμιση

Μετά την εξομάλυνση και τους υπολογισμούς, έχουμε τις παραστάσεις που φαίνονται στην Εικόνα 24.



**Εικόνα 24:** Η αρχική (κόκκινη) και η εξομαλυσμένη (μπλε) καμπύλη απόκρισης, καθώς και η ευθεία  $O_1S_{max}$  (πράσινη)

Όπως μπορούμε εύκολα να παρατηρήσουμε, και στην περίπτωση αυτή η εξομαλυσμένη και η αρχική καμπύλη ουσιαστικά συμπίπτουν. Η γραφική παράσταση του χρόνου απόκρισης είναι η ακόλουθη:



**Εικόνα 25:** Χρόνος απόκρισης T(t)

Οι δε τιμές των παραμέτρων του συστήματος μέτρησης, όπως υπολογίζονται στην περίπτωση της απόκρισης του συστήματος με αντιστάθμιση, είναι αυτές που φαίνονται στον παρακάτω πίνακα:

Παράμετρος	Τιμή
$O_1$	27.353
Υπερύψωση $b$	0.389
Μερικός χρόνος απόκρισης $T_a$ (ns)	6.941
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-3.748
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	2.667

Θα πρέπει να σημειωθεί ότι η αρνητική τιμή που προκύπτει για τον πειραματικό χρόνο απόκρισης, οφείλεται στην μεγάλη υπερύψωση που παρουσιάζει η απόκριση, σύμφωνα με τα αποτελέσματα της προσομοίωσης.

Στα παραρτήματα A1 και A2 παρουσιάζονται αποτελέσματα από υπολογισμούς που έγιναν με βάση τα ίδια δεδομένα εισόδου, αλλά με διαφορετικές επιλογές εξομάλυνσης, δηλαδή με την λήψη περισσότερων γειτονικών σημείων για τον υπολογισμό της μέσης τιμής ή την διαδοχική εκτέλεση της διαδικασίας εξομάλυνσης περισσότερες φορές, έτσι ώστε να μελετηθεί η επίδρασή της στον υπολογισμό των τιμών των παραμέτρων απόκρισης.

### 8.3. Συμπεράσματα

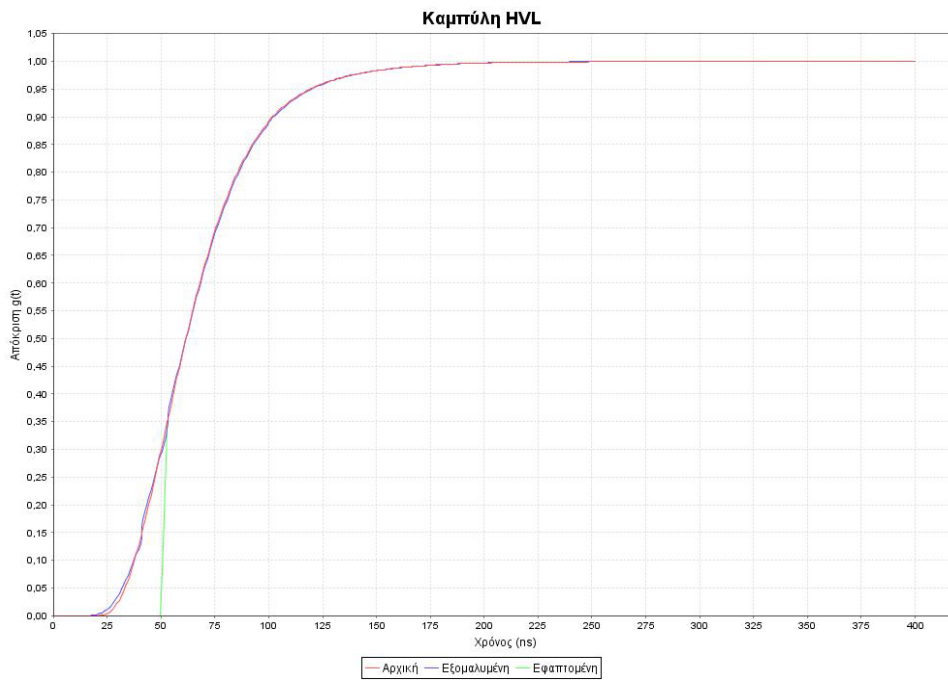
Όπως προκύπτει και από τα αποτελέσματα που παρατίθενται στα παραρτήματα A1 και A2, με την εξομάλυνση της καμπύλης για διάφορες τιμές γειτονικών σημείων και επαναλήψεων οδηγούμαστε σε βελτιωμένες τιμές των παραμέτρων απόκρισης. Χωρίς εξομάλυνση της καμπύλης, ο μερικός χρόνος απόκρισης  $T_a$  καθώς και η υπερύψωση  $\beta$  παρουσιάζουν και για τις δύο σειρές μετρήσεων (χωρίς ή με αντιστάθμιση) τη μέγιστη τιμή τους, συγκρινόμενα με τα υπόλοιπα δεδομένα που

προέκυψαν έπειτα από εξομάλυνση. Άρα η διαδικασία της εξομάλυνσης (smoothing) είναι απαραίτητη για πιο ακριβή αποτελέσματα. Ο μερικός χρόνος απόκρισης δείχνει το πόσο γρήγορα ανέρχεται η  $g(t)$  μέχρι τη μονάδα. Όσο μικρότερος είναι, τόσο πιο κάθετη και ταχεία είναι η άνοδος της απόκρισης προσεγγίζοντας έτσι την ιδανική περίπτωση της βηματικής απόκρισης όπου  $T_a=0$ . Η υπερύψωση  $\beta$  δείχνει την ποσότητα κατά την οποία η μέγιστη τιμή της  $g(t)$  ξεπερνά τη μονάδα. Όσο μικρότερη είναι η υπερύψωση τόσο καλύτερη είναι η  $g(t)$  αφού δεν ξεπερνά κατά πολύ τη μονάδα. Ο πειραματικός χρόνος απόκρισης δίνει μία γενικότερη εικόνα κατά πόσο η  $g(t)$  αποκλίνει από την ιδανική κατάσταση. Όσο πιο κοντά είναι στο μηδέν τόσο ακριβέστερο είναι το σύστημα. Όμως δεν αποτελεί καλό κριτήριο αφού και συστήματα με έντονη ταλάντωση και απόκλιση από το ιδανικό, μπορεί να εμφανίσουν μηδενικό πειραματικό χρόνο απόκρισης. Έτσι για την επιλογή των καλύτερων καμπυλών δεν λαμβάνεται ιδιαίτερα υπόψιν αυτή η παράμετρος.

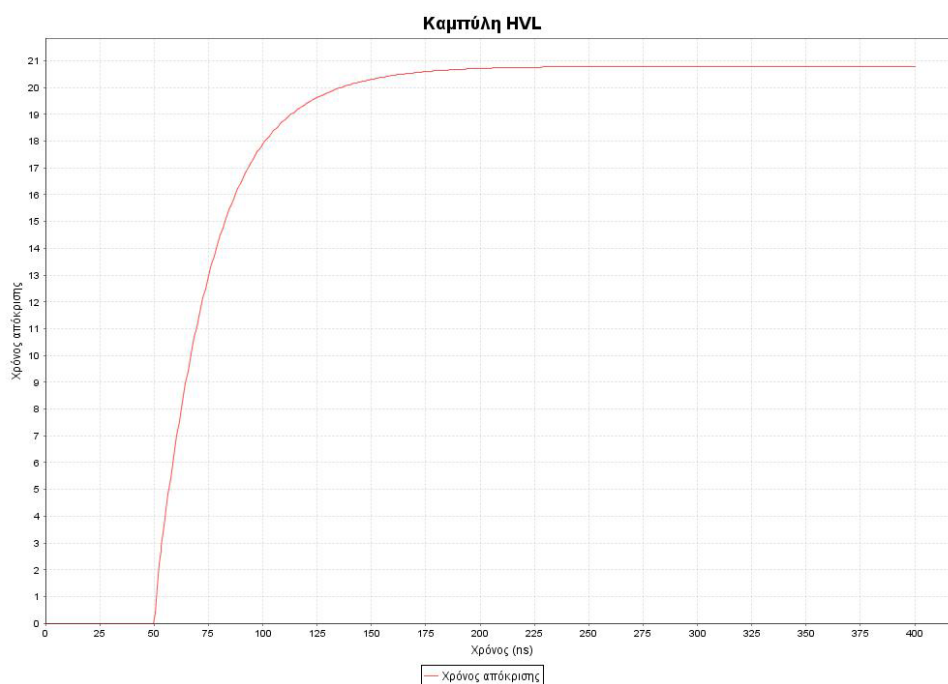
Για τις μετρήσεις που προέκυψαν από το σύστημα χωρίς αντιστάθμιση, η περίπτωση που έδωσε τα καλύτερα αποτελέσματα είναι αυτή με την εξομάλυνση της καμπύλης με χρήση 6 γειτονικών σημείων και 4 διαδοχικές επαναλήψεις (Εικόνα 26, Εικόνα 27). Στην περίπτωση αυτή έχουμε τον μικρότερο μερικό χρόνο απόκρισης  $T_a=20.781$  ns. Αυτό οφείλεται στο ότι η εφαπτομένη που μας δίνει το  $O_1$  και με την οποία αντικαθιστούμε το αρχικό κομμάτι της  $g(t)$  έχει σχεδόν κάθετη κλίση με αποτέλεσμα τη γρήγορη άνοδο της  $g(t)$ . Επίσης στην περίπτωση αυτή παρατηρείται και ο μικρότερος πειραματικός χρόνος απόκρισης  $T_N=20,781$  ns (Πίνακας 2).

Παράμετρος	Τιμή
$O_1$	49.599
Υπερύψωση $\beta$	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.781
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.781
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.468

**Πίνακας 2: Παράμετροι απόκρισης συστήματος χωρίς αντιστάθμιση έπειτα από εξομάλυνση με 6 γειτονικά σημεία και 4 επαναλήψεις**



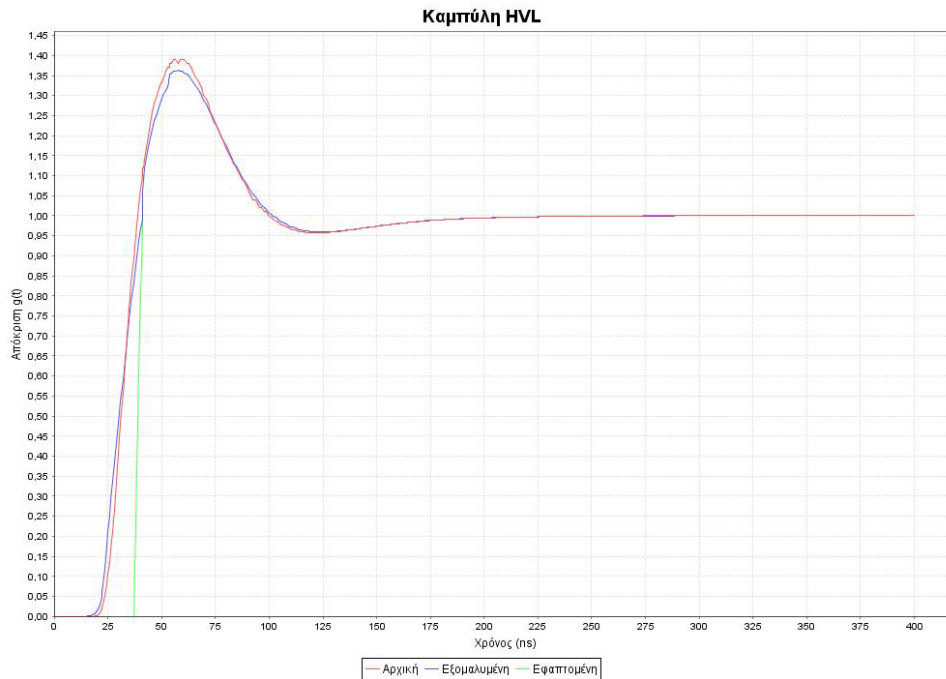
**Εικόνα 26:** Εξομάλυνση με 6 γειτονικά σημεία και 4 επαναλήψεις για απόκριση συστήματος χωρίς αντιστάθμιση



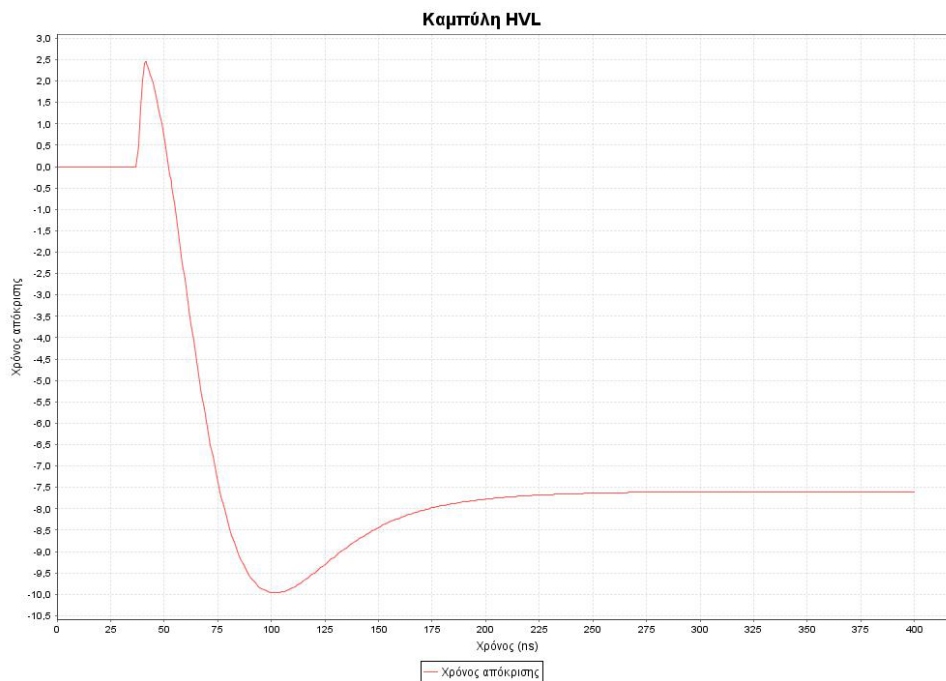
**Εικόνα 27:** Χρόνος απόκρισης συστήματος χωρίς αντιστάθμιση έπειτα από εξομάλυνση με 6 γειτονικά σημεία και 4 επαναλήψεις

Στην περίπτωση συστήματος μέτρησης με αντιστάθμιση τα καλύτερα αποτελέσματα προέκυψαν μετά από εξομάλυνση με χρήση 8 γειτονικών σημείων και 4 διαδοχικές επαναλήψεις της διαδικασίας (Εικόνα 28, Εικόνα 29). Ο μερικός χρόνος απόκρισης εμφανίζει την πολύ μικρή τιμή των 2,462 ns (η ελάχιστη από όλες τις

περιπτώσεις εξομάλυνσης) με αποτέλεσμα την κατακόρυφη άνοδο της  $g(t)$ . Επίσης η υπερύψωση εμφανίζει μία από τις μικρότερες τιμές των μετρήσεων και είναι  $\beta=0,362$ .



**Εικόνα 28: Εξομάλυνση με 8 γειτονικά σημεία και 4 επαναλήψεις για απόκριση συστήματος με αντιστάθμιση**



**Εικόνα 29: Χρόνος απόκρισης συστήματος με αντιστάθμιση έπειτα από εξομάλυνση με 8 γειτονικά σημεία και 4 επαναλήψεις**

Παράμετρος	Τιμή
$O_1$	36.918
Υπερύψωση $b$	0.362
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.462
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.611
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.569

**Πίνακας 3: Παράμετροι απόκρισης συστήματος με αντιστάθμιση έπειτα από εξομάλυνση με 8 γειτονικά σημεία και 4 επαναλήψεις**

Γενικά παρατηρούμε ότι τα καλύτερα αποτελέσματα προέκυψαν από εξομάλυνση των καμπυλών με χρήση πολλών γειτονικών σημείων (6 και 8 αντίστοιχα) και έπειτα από αρκετές επαναλήψεις (4 και στις δύο περιπτώσεις).



## Πηγές – Αναφορές

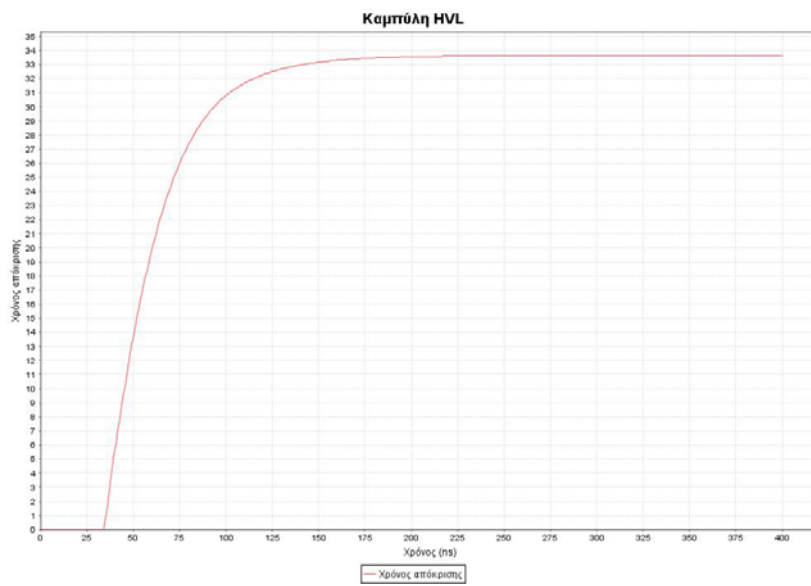
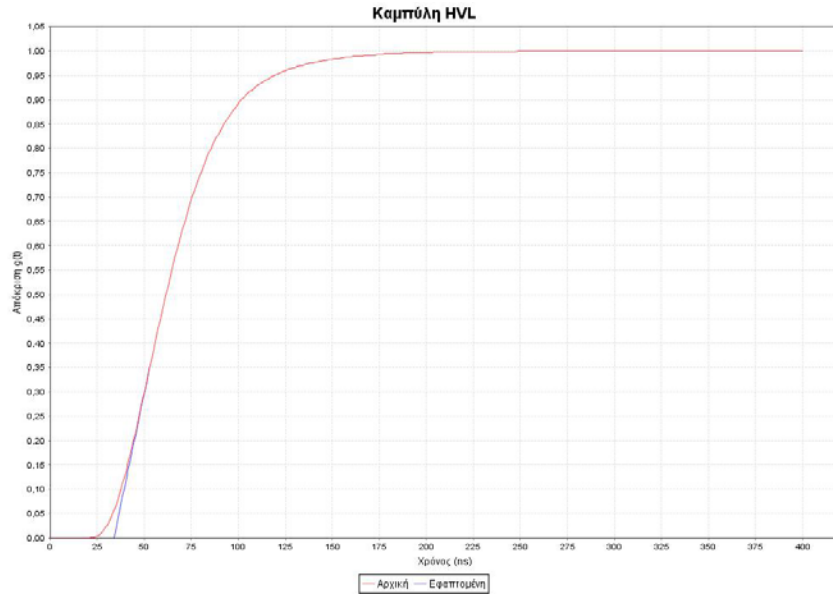
- [1] “International cloud atlas”, volume 1, Secretariat of the World Meteorological Organization, ISBN 9263104077, Geneva, 1975
- [2] DIN VDE-0101 2000-01
- [3] Ι. Αθ. Σταθόπουλος, “Υψηλές Τάσεις Ι”, Εκδόσεις Συμεών, ISBN 960-7346-07-6
- [4] Φ. Β. Τοπαλής, “Υψηλές Τάσεις”, Εκδόσεις ΕΜΠ, Αθήνα 1991
- [5] Π. Κούκος, “Τεχνολογία Υψηλών Τάσεων”, ΤΕΙ Χαλκίδας, Χαλκίδα 2004
- [6] Β. Κ. Παπαδιά, “Γραμμές μεταφοράς ηλεκτρικής ενέργειας”, Εκδόσεις Συμμετρία, Αθήνα 1999, ISBN 960-266-068-6
- [7] Γ. Κυραναστάσης, “Σημειώσεις Βιομηχανικών Συστημάτων Ηλεκτρικής Ενέργειας”, ΤΕΙ Καβάλας, Καβάλα 2001
- [8] T. R. McComb and J. E. Lagnese, “Calculating the parameters of full lighting impulses using model based curve fitting”, IEEE transaction on Power Delivery, vol 6, no 4, pp 1386-1394, Oct. 1991
- [9] Y. M. Li et al., “Exponential fitting algorithms for digitally recorded high voltage impulse parameter evaluation”, IEEE transaction on Power Delivery, vol 8, no 4, pp 1727-1735, Oct. 1993
- [10] F. Garnacho et al., “Evaluation procedure for lightning impulse parameters in case of waveforms with oscillations and/or an overshoot”, IEEE transaction on Power Delivery, vol 12, no 2, pp 640-649, Apr. 1997
- [11] J. Perez and J. Martinez, “Digitally recorded lightning impulse with overshoot parameter evaluation by using the kalman filtering method”, IEEE transaction on Power Delivery, vol 13, no 4, pp 1005-1014, Oct. 1998
- [12] Yi Li and Juris Rungis, “Precision digital filters for high voltage impulse measurement systems”, IEEE transaction on Power Delivery, vol 14, no 4, pp 1213-1220, Oct. 1999
- [13] A. Brede et al., “A new method of determining the mean curve of lightning impulses according to IEC 60060-1”, 11<sup>th</sup> ISH, London UK 1999
- [14] S. Venkatesan, P. V. Ranjan, D. Ashokaraju, “A comparative study on methods for evaluation of lightning impulse parameters”, in proc. of IEEE TENCON intl. conf., Bangalore, India, 2003
- [15] European Standard IEC 60060-2, CENELEC, Dec. 1994
- [16] Y. Li, J. Rungis, R. Sheedy, “Impulse divider unit step response evaluation”, in proc. of 9<sup>th</sup> Intl. Symposium on High Voltage Engineering, Graz, Austria, Aug. 1995
- [17] I. Lopes, W. Boaventura, R. Godinho, “Evaluation of parameters of digitally recorded HV impulses: Frequency and time domain approaches”, 9<sup>th</sup> Intl. Symposium on High Voltage Engineering, Graz, Austria, Aug. 1995
- [18] <http://www.jfree.org/jfreechart/>
- [19] <http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/>



# Παράρτημα Α1

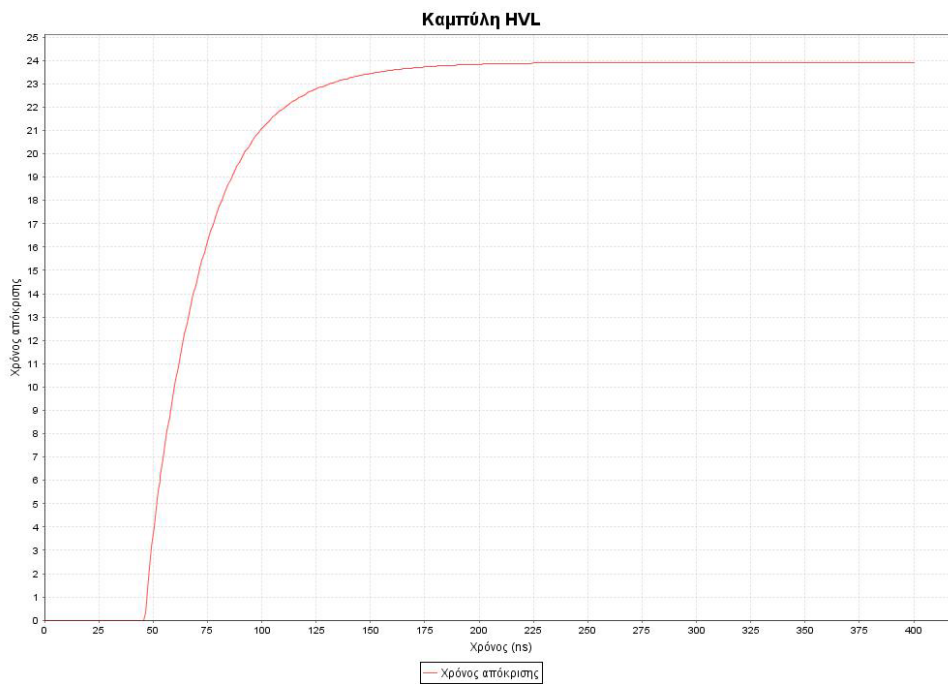
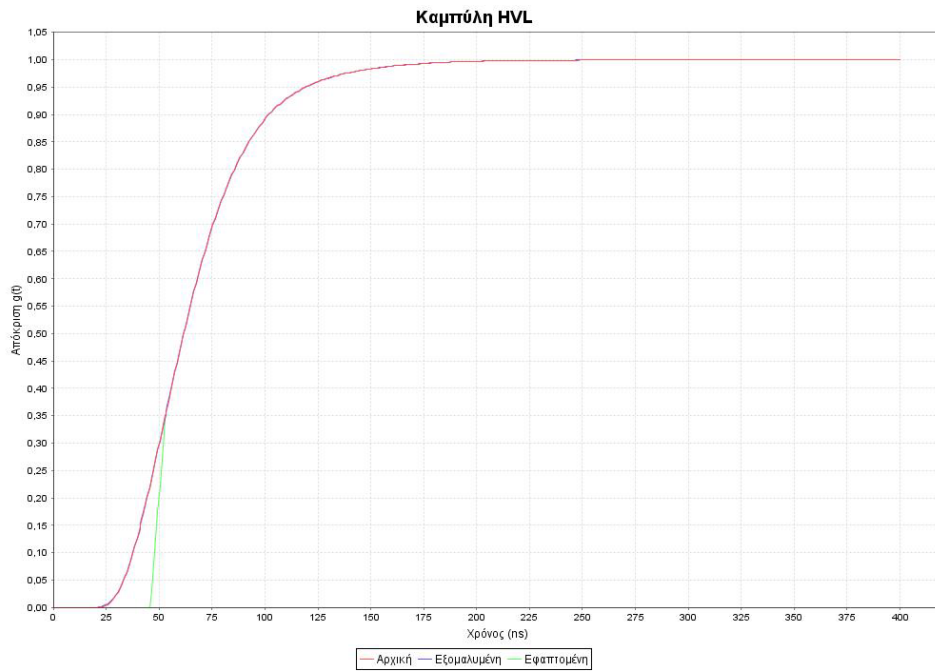
## Ανάλυση παραμέτρων συστήματος μέτρησης χωρίς αντιστάθμιση

Αριθμός γειτονικών σημείων	0
Αριθμός επαναλήψεων	0



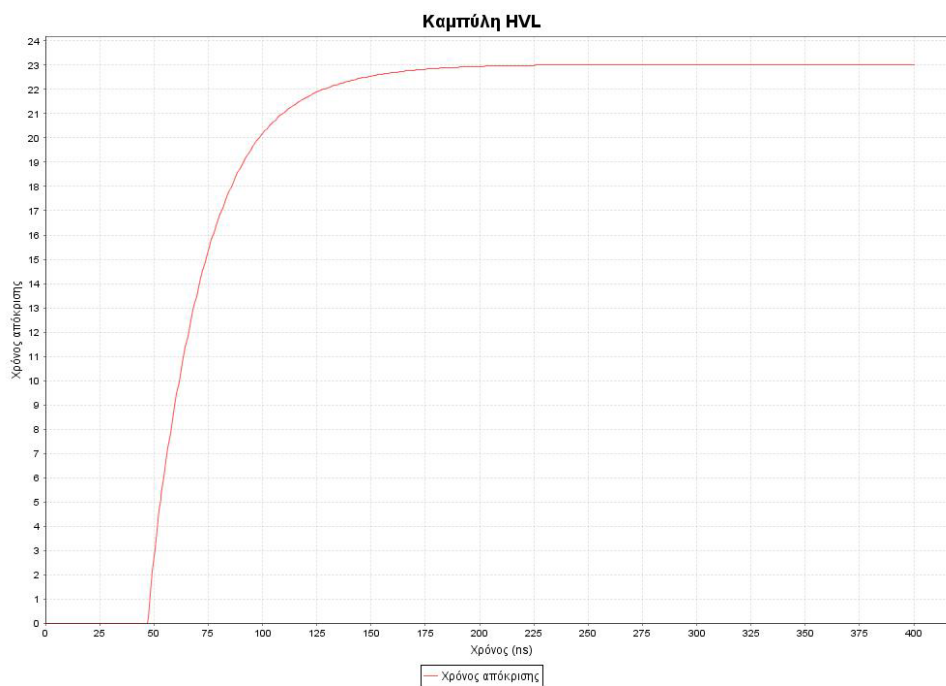
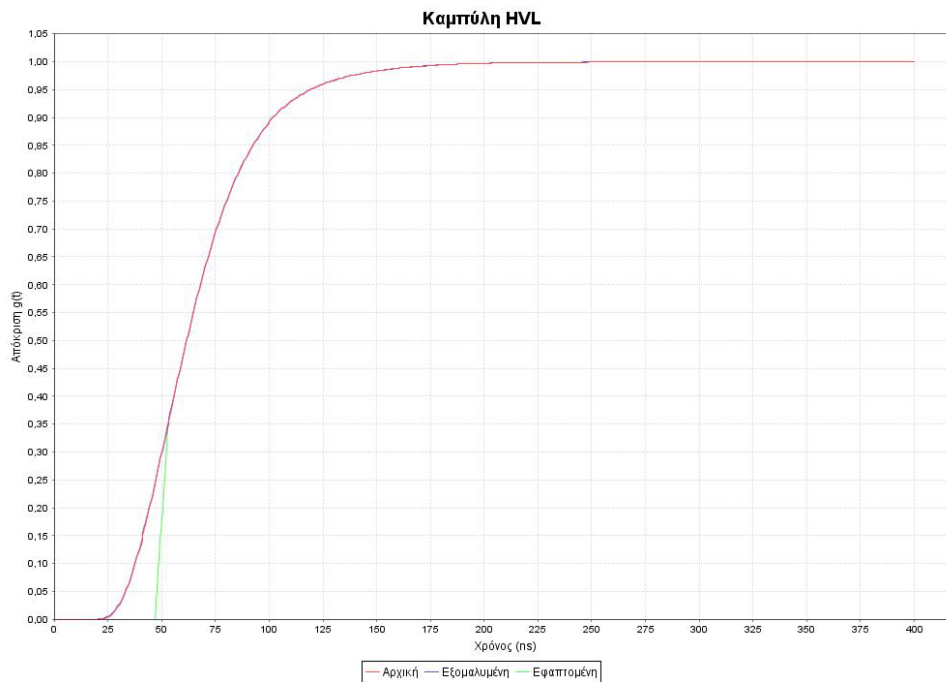
Παράμετρος	Τιμή
$O_1$	33.854
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	33.643
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	33.643
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	0.522

Αριθμός γειτονικών σημείων	2
Αριθμός επαναλήψεων	2



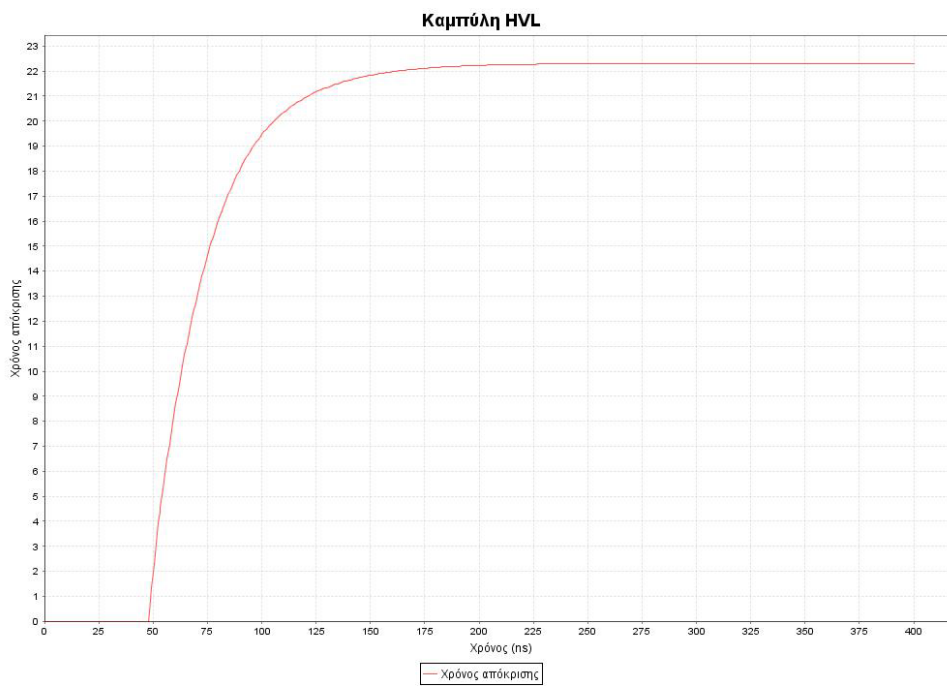
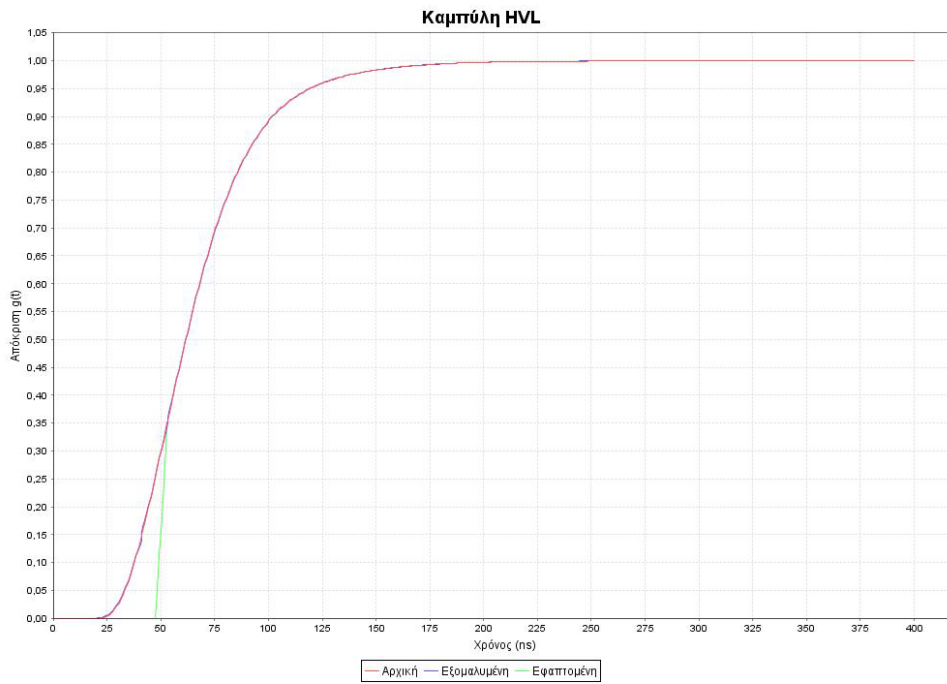
Παράμετρος	Τιμή
$O_1$	45.753
Υπερύψωση $b$	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	23.905
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	23.905
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	2.639

Αριθμός γειτονικών σημείων	2
Αριθμός επαναλήψεων	3



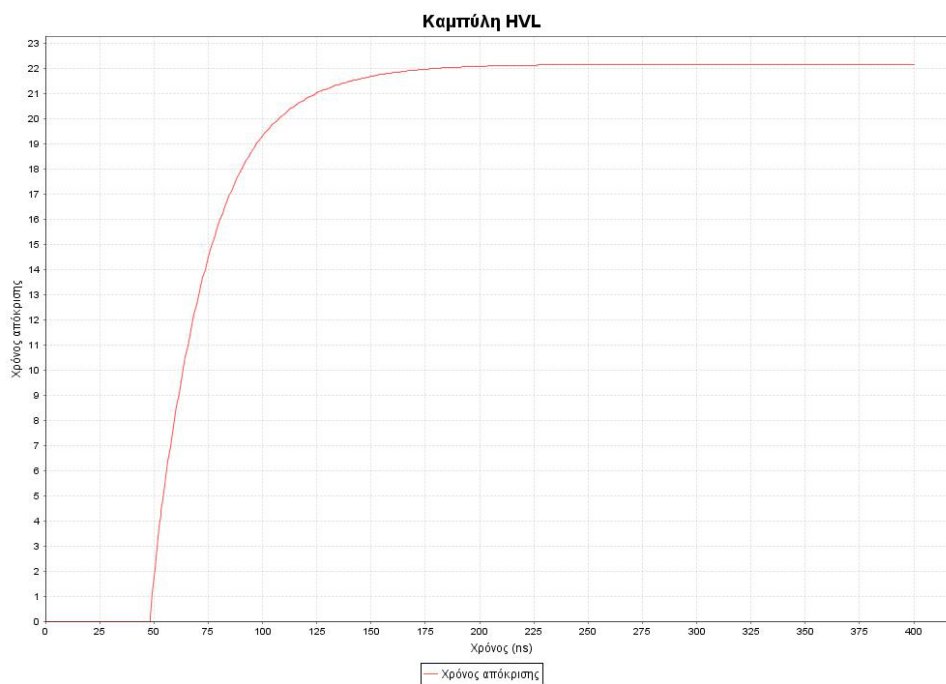
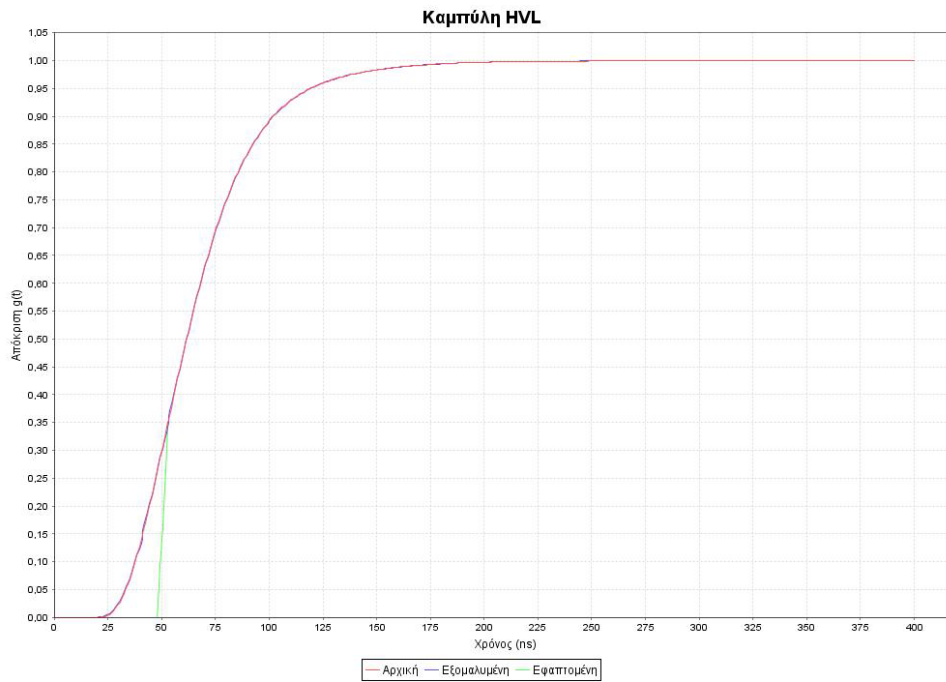
Παράμετρος	Τιμή
$O_1$	46.939
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	23.018
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	23.018
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	2.852

Αριθμός γειτονικών σημείων	2
Αριθμός επαναλήψεων	4



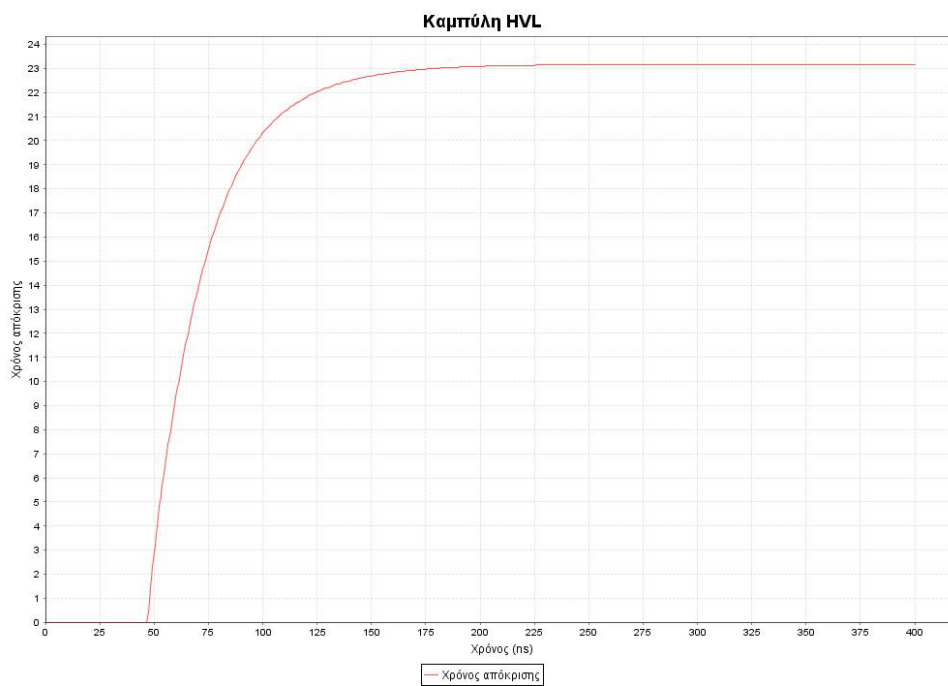
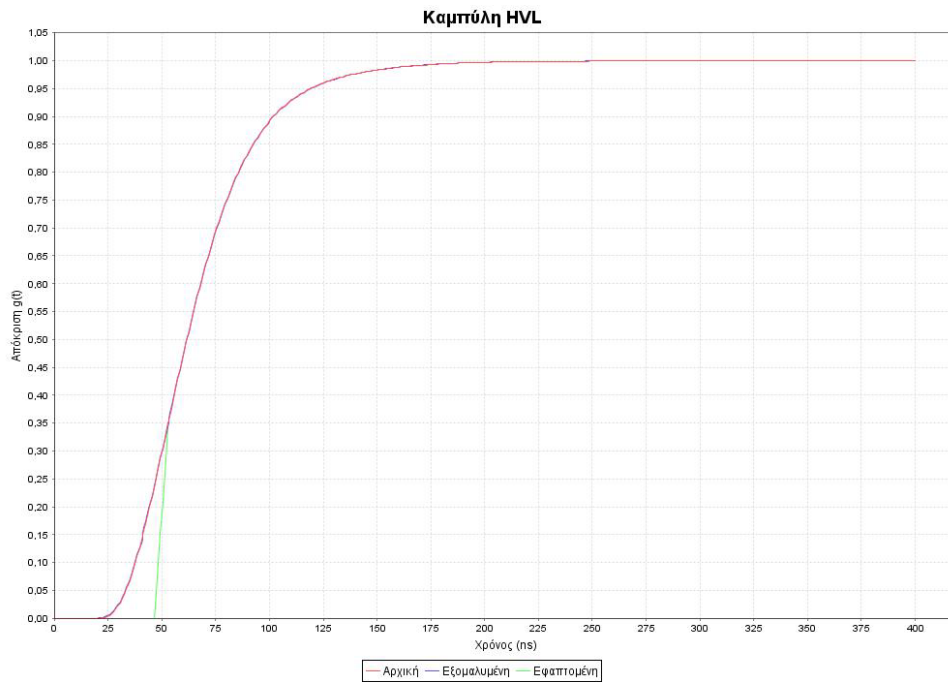
Παράμετρος	Τιμή
$O_1$	47.626
Υπερύψωση $b$	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	22.298
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	22.298
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	2.981

Αριθμός γειτονικών σημείων	2
Αριθμός επαναλήψεων	5



Παράμετρος	Τιμή
$O_1$	48.068
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	22.154
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	22.154
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.061

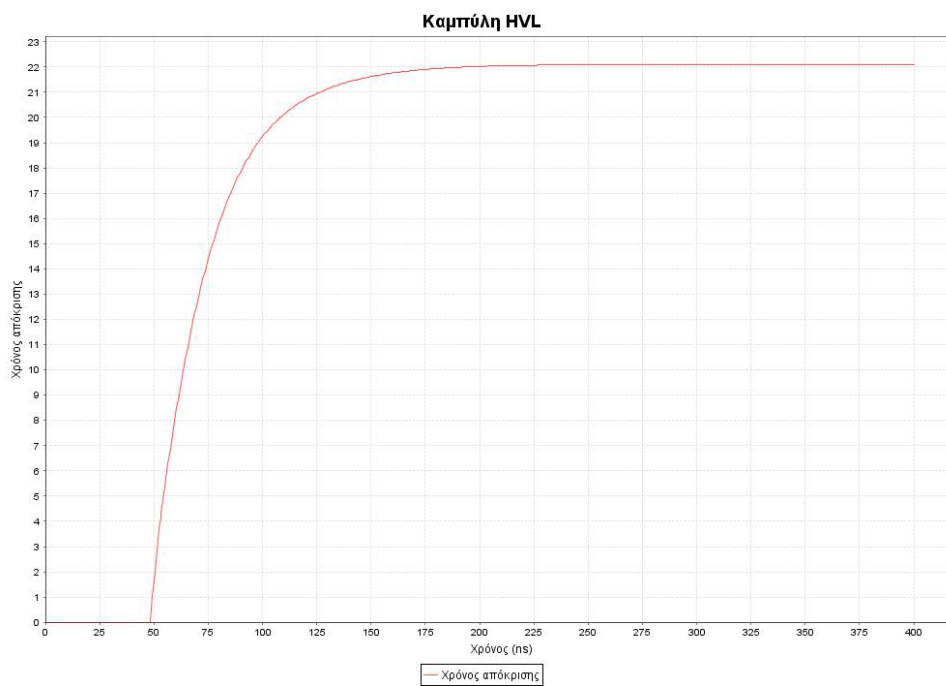
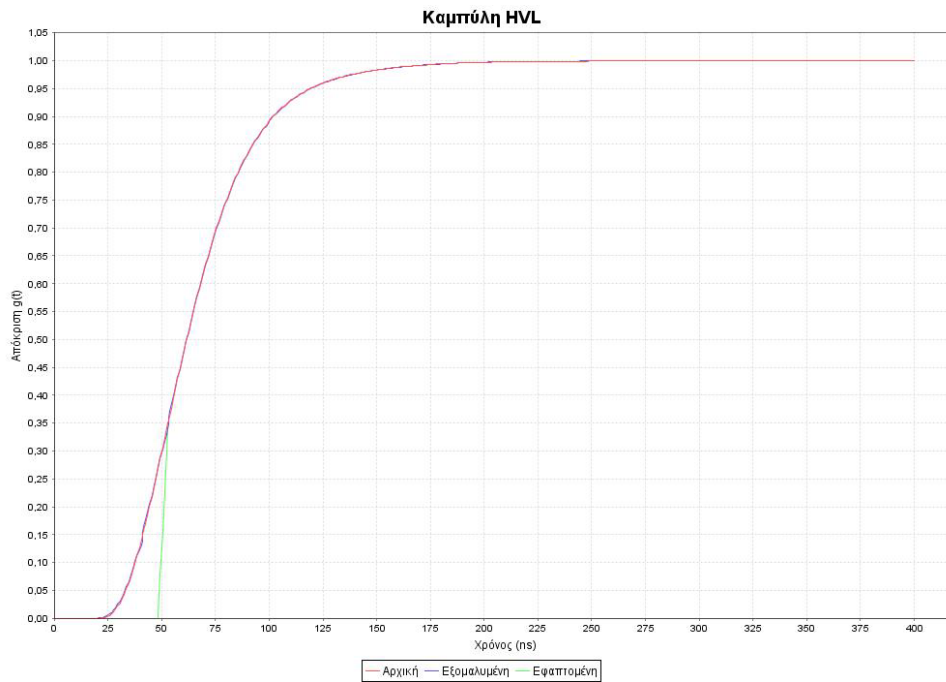
Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	1



Παράμετρος	Τιμή
$O_1$	46.541
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	23.158
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	23.158
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	2.795

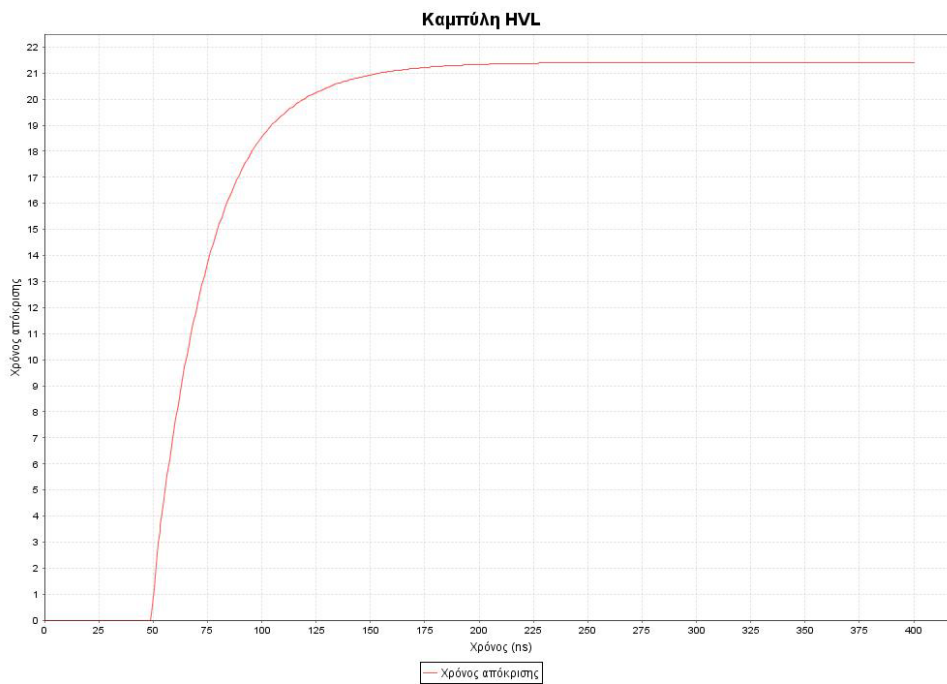
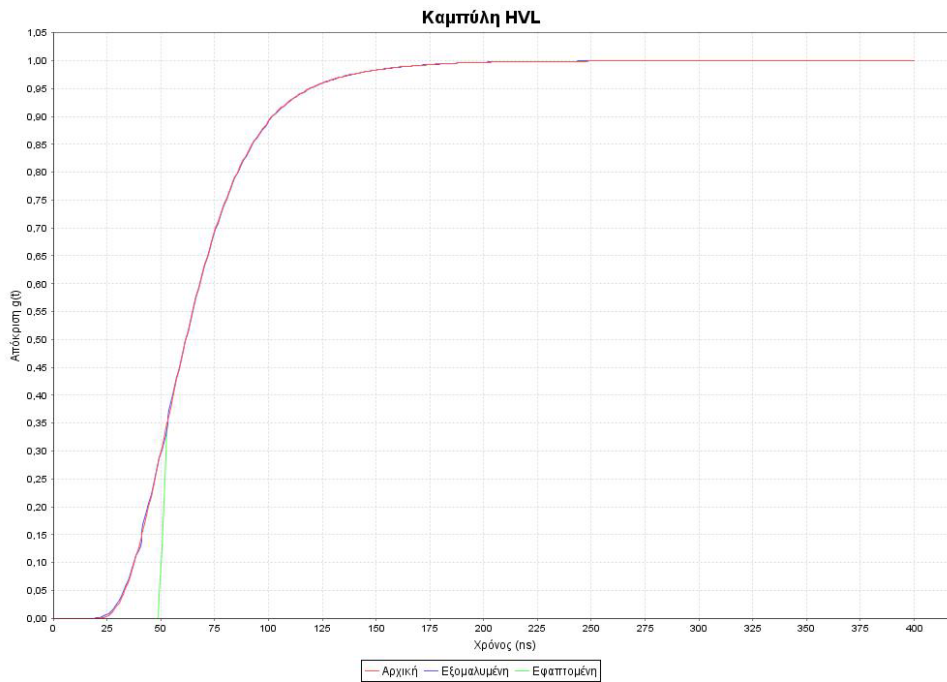


Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	2



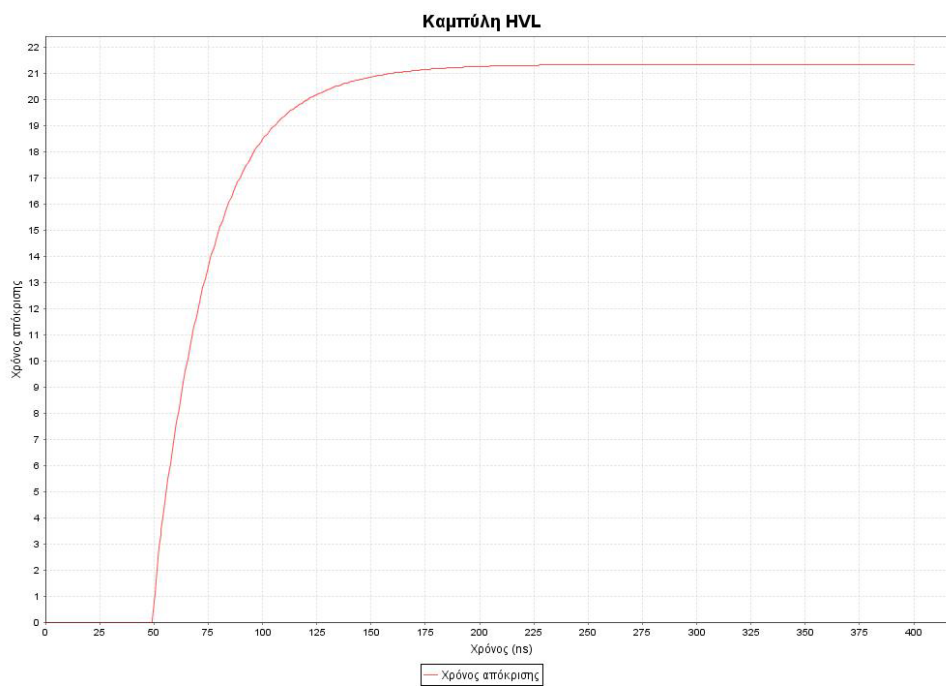
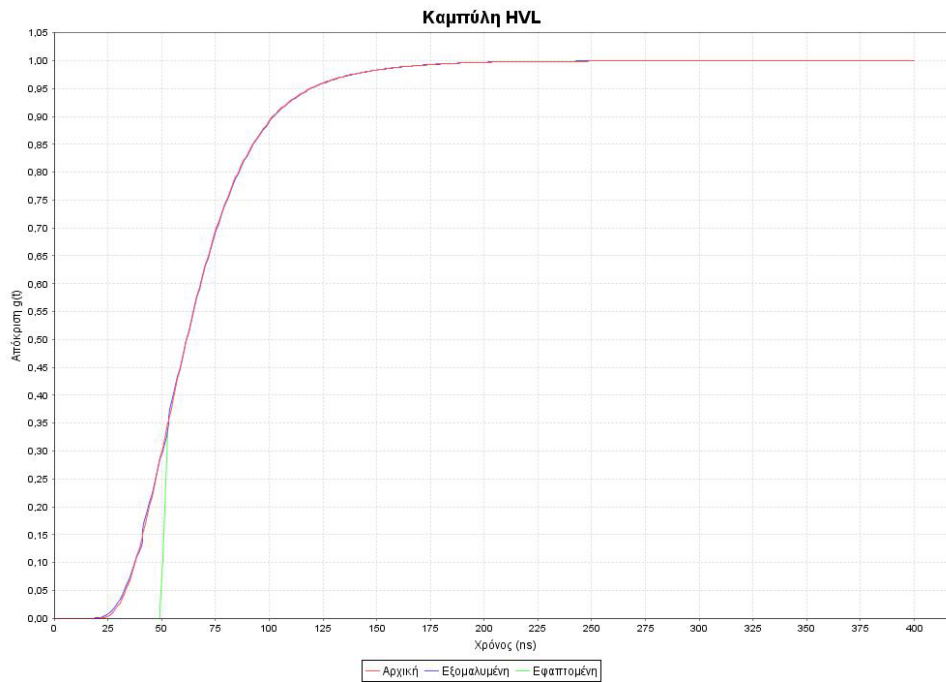
Παράμετρος	Τιμή
$O_1$	48.300
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	22.092
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	22.092
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.116

Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	3



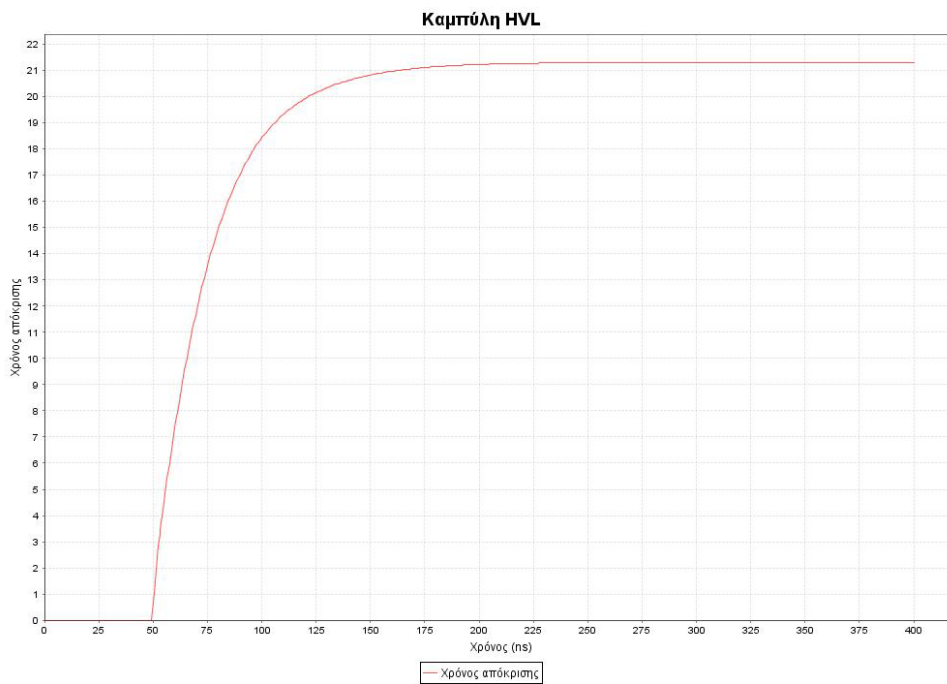
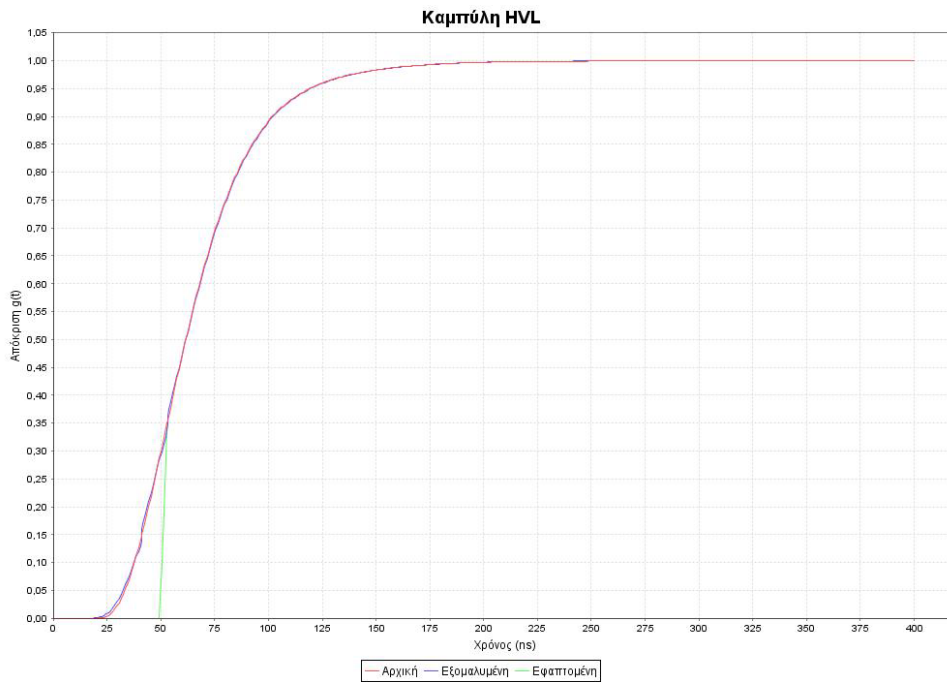
Παράμετρος	Τιμή
$O_1$	48.907
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.399
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.399
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.243

Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	4



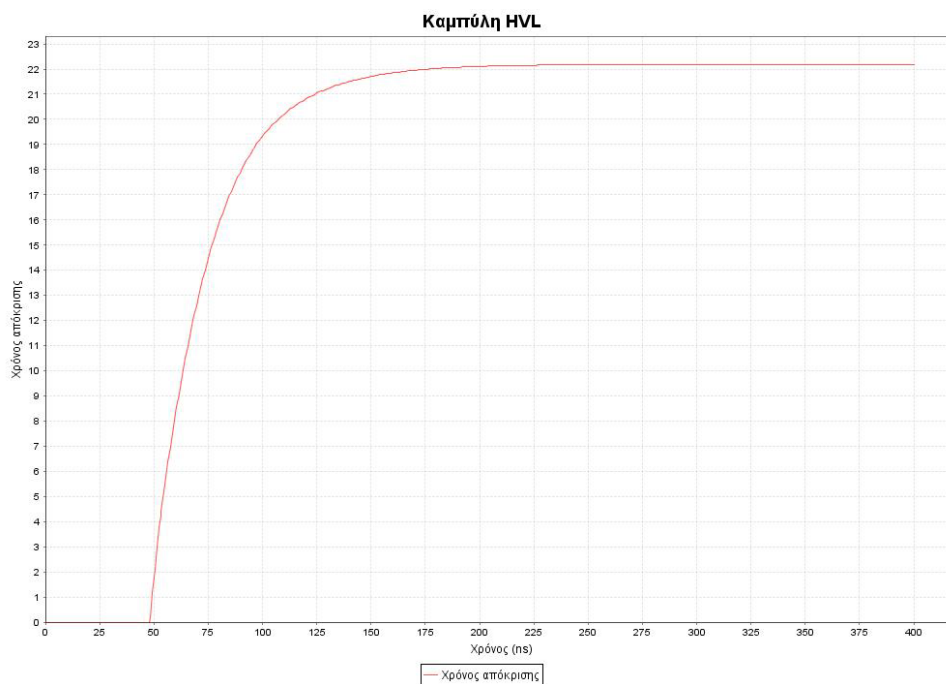
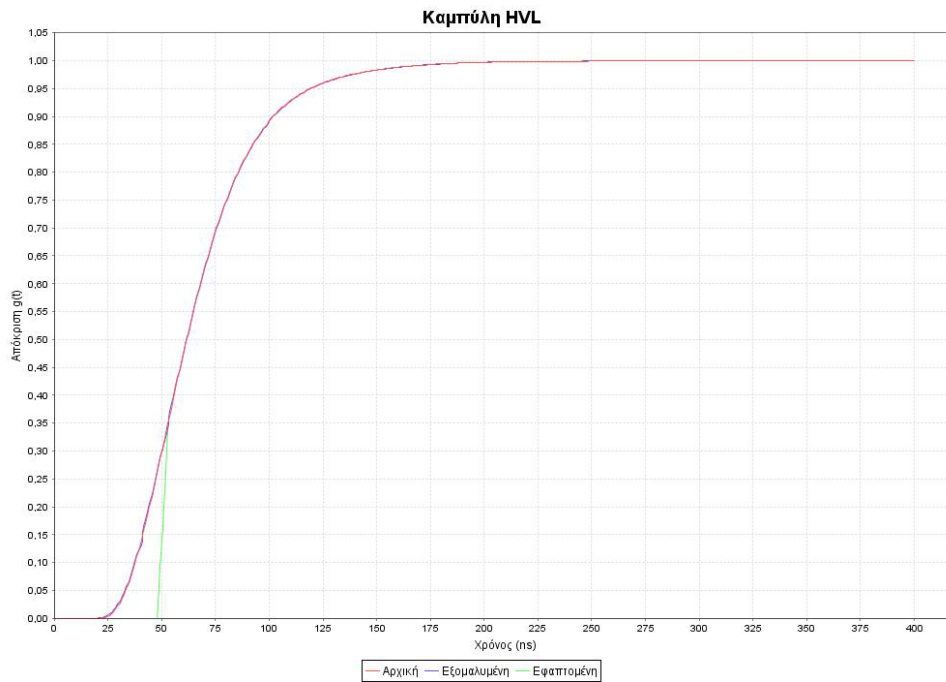
Παράμετρος	Τιμή
$O_1$	49.198
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.324
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.324
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.309

Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	5



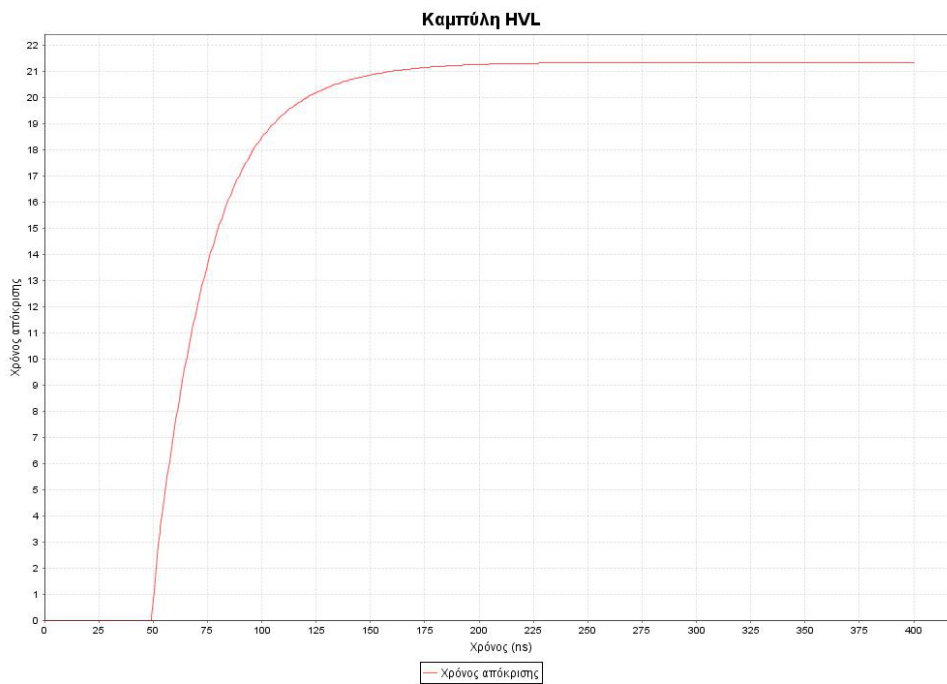
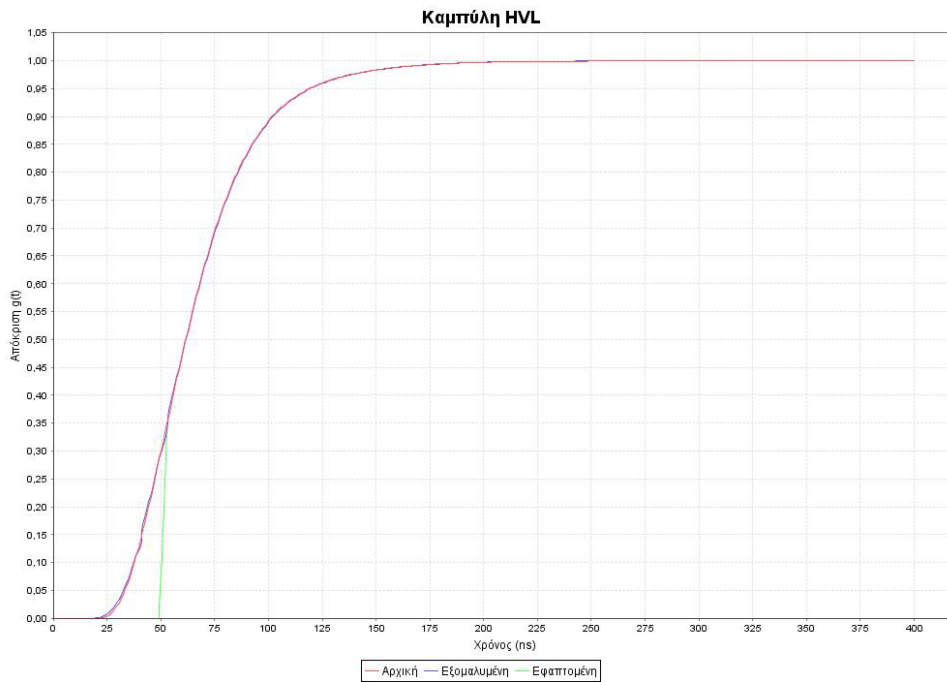
Παράμετρος	Τιμή
$O_1$	49.365
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.293
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.293
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.355

Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	1



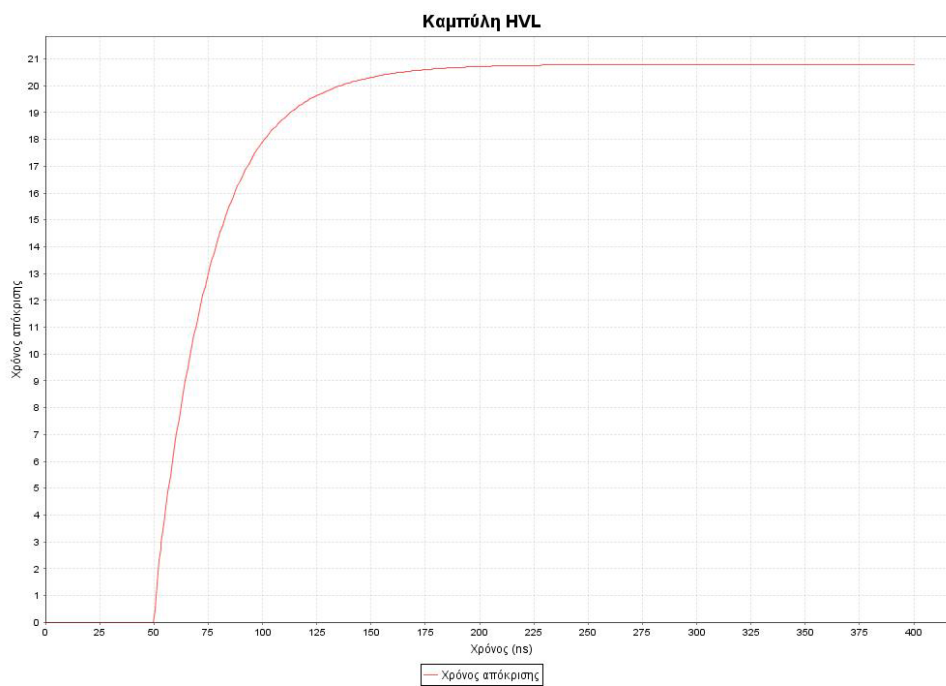
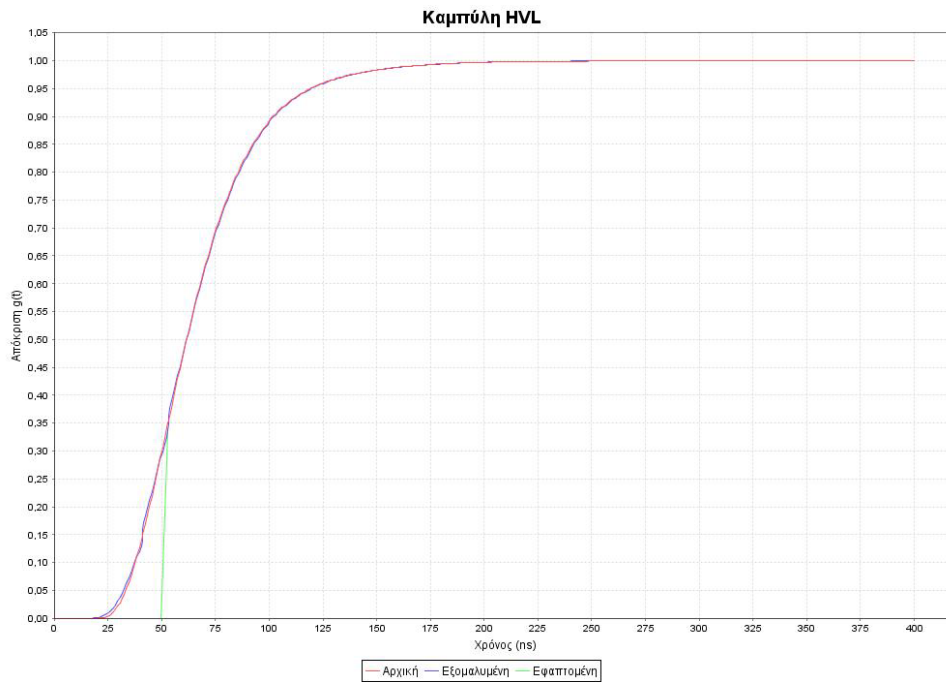
Παράμετρος	Τιμή
$O_1$	48.045
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	22.176
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	22.176
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.082

Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	2



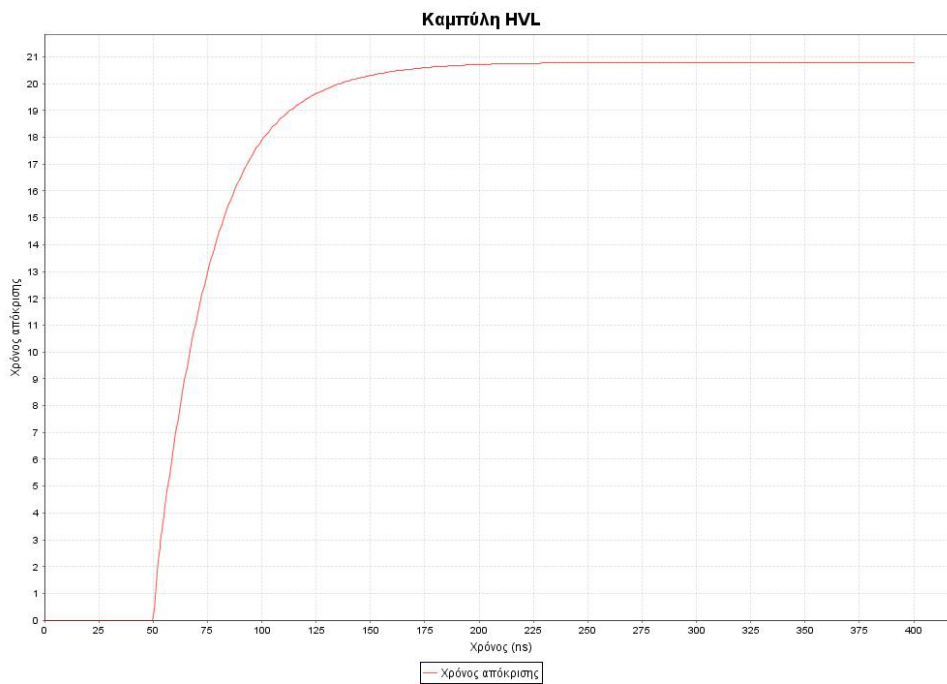
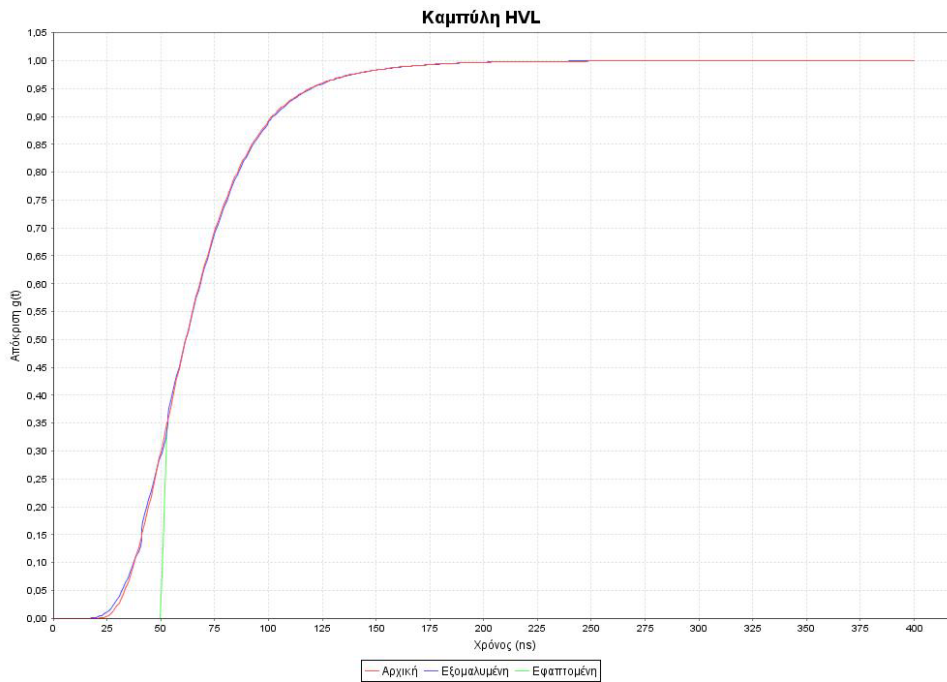
Παράμετρος	Τιμή
$O_1$	49.153
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.343
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.343
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.312

Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	3



Παράμετρος	Τιμή
$O_1$	49.465
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.786
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.786
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.408

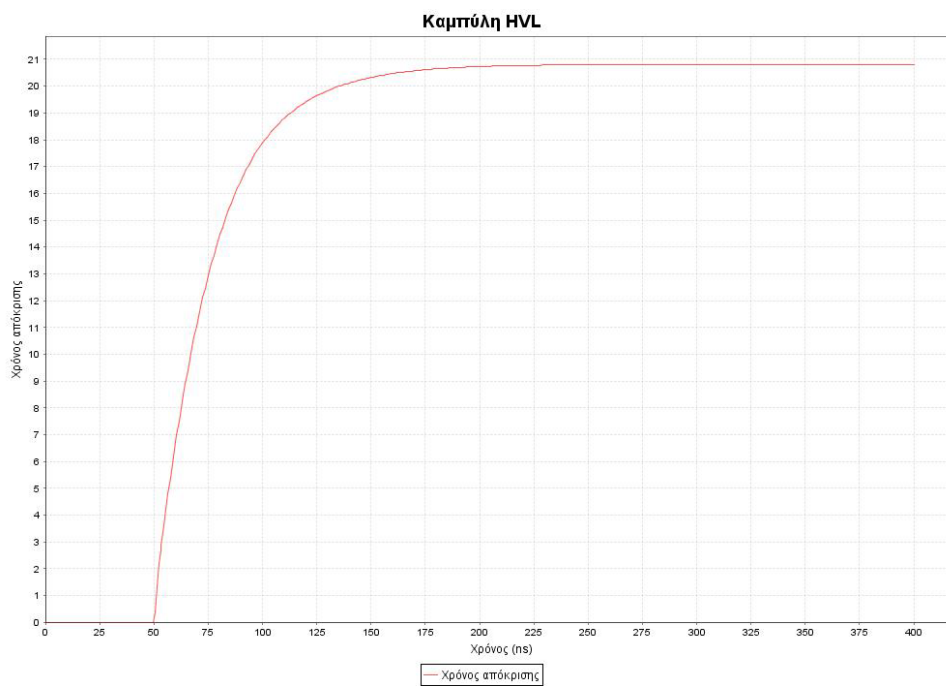
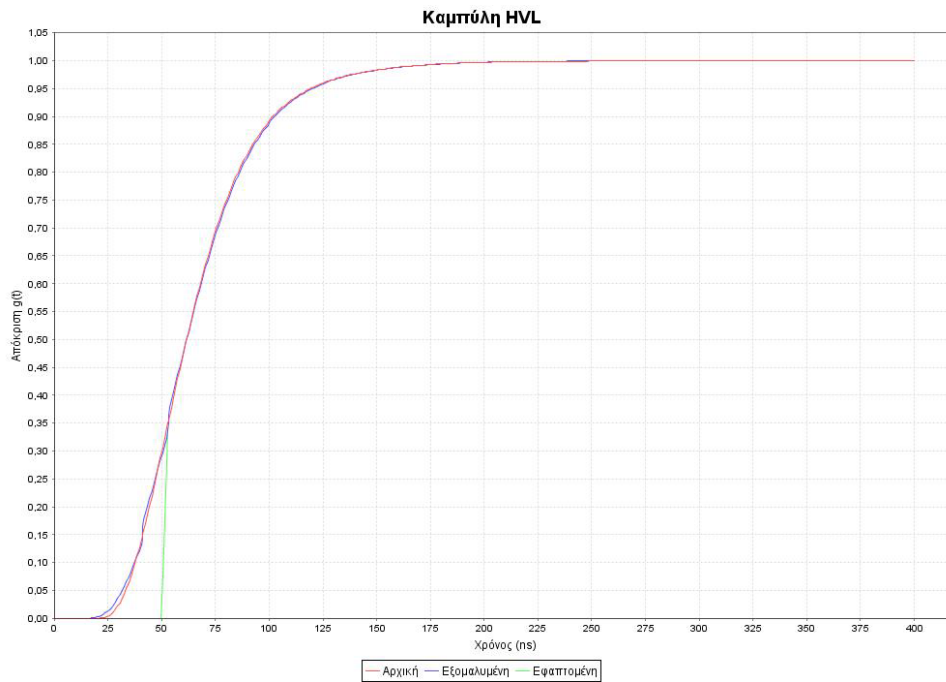
Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	4



Παράμετρος	Τιμή
$O_1$	49.599
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.781
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.781
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.468

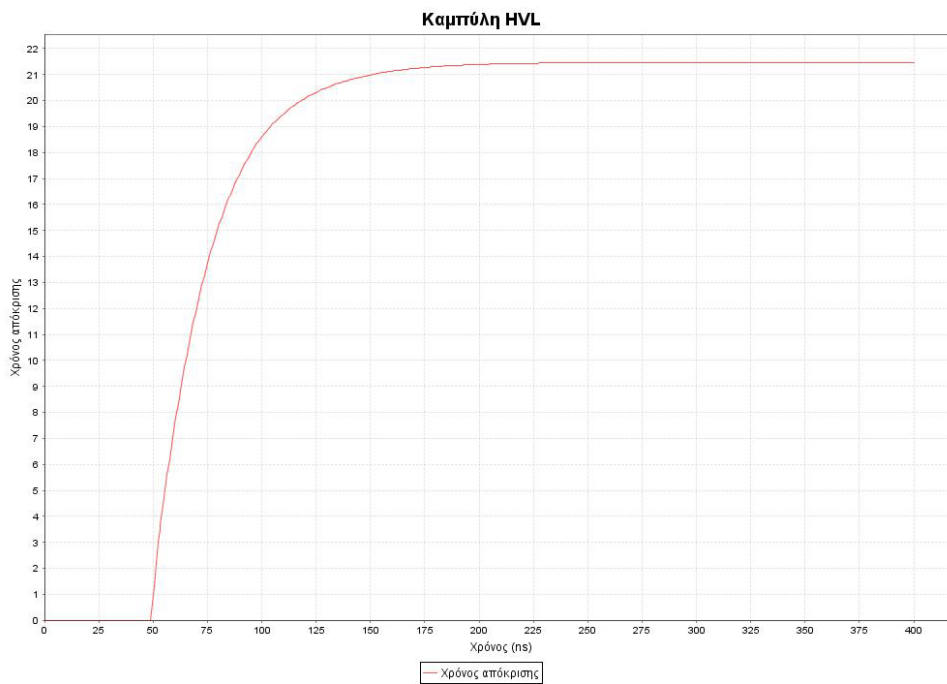
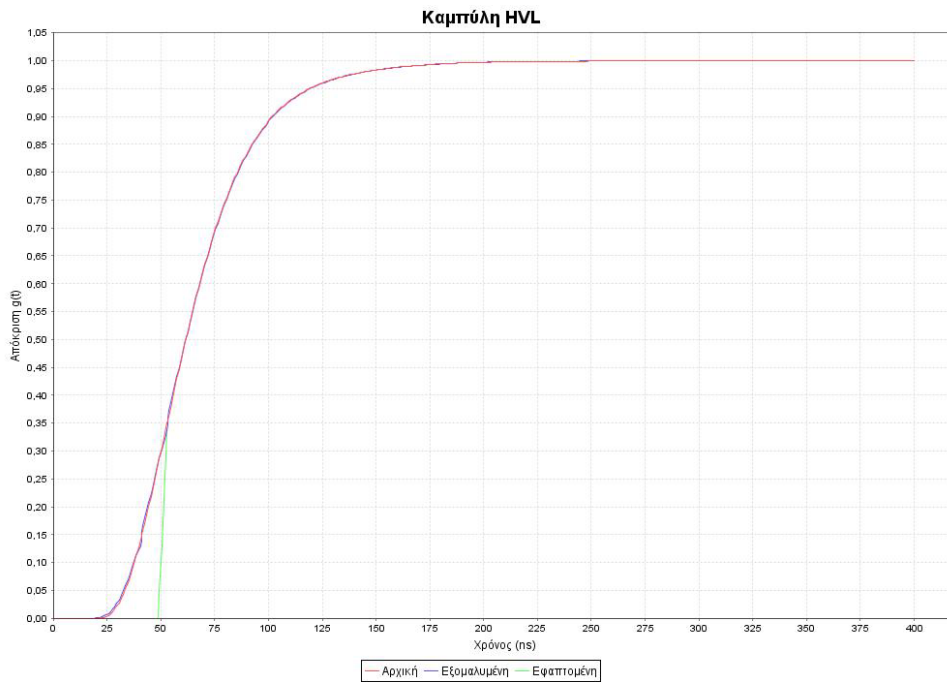


Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	5



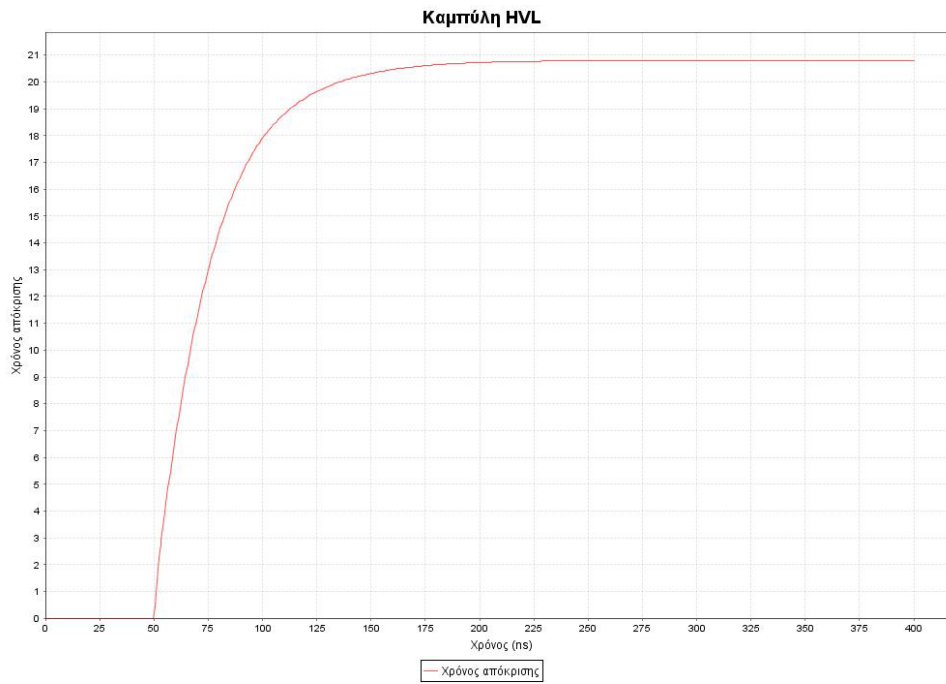
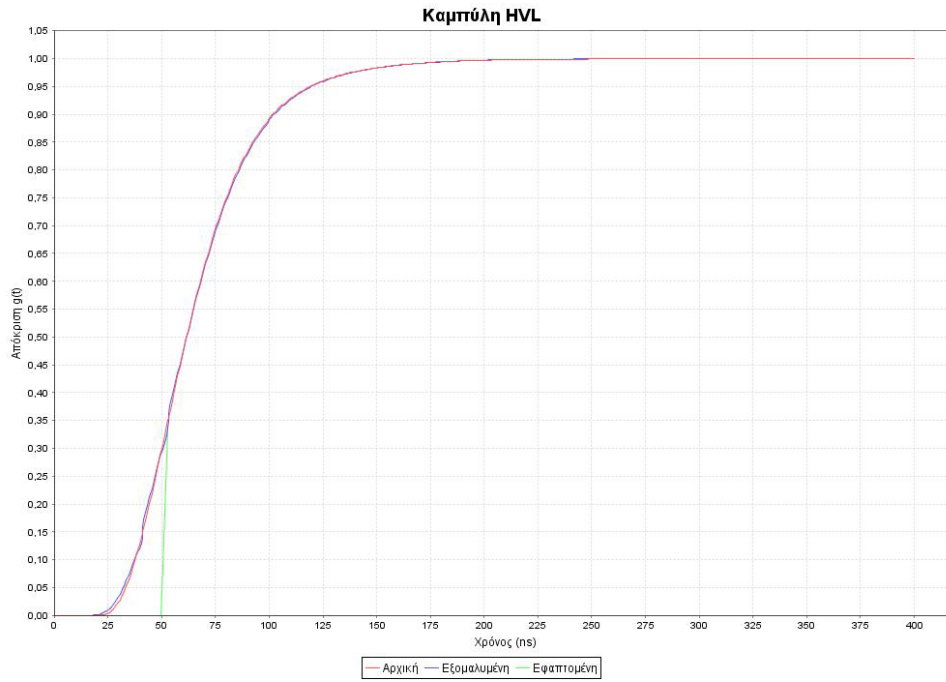
Παράμετρος	Τιμή
$O_1$	49.669
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.802
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.802
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.519

Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	1



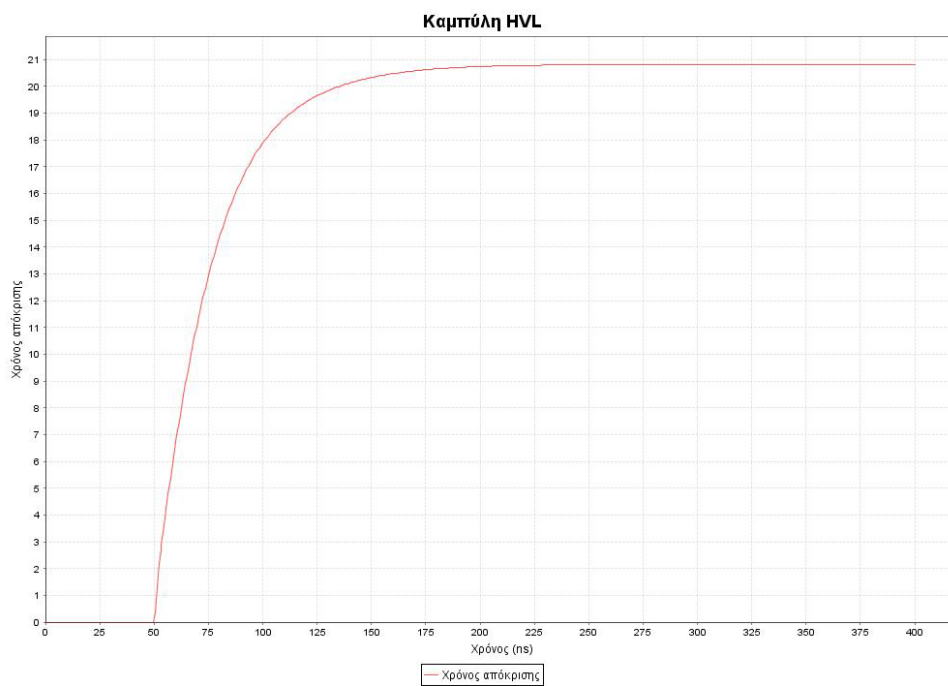
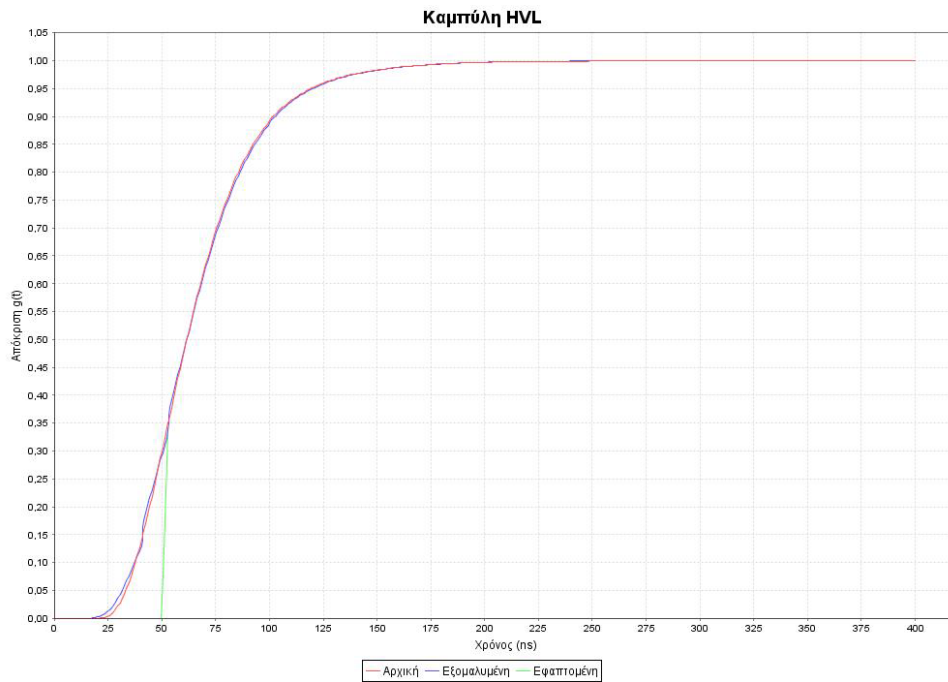
Παράμετρος	Τιμή
$O_1$	48.777
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.459
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.459
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.244

Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	2



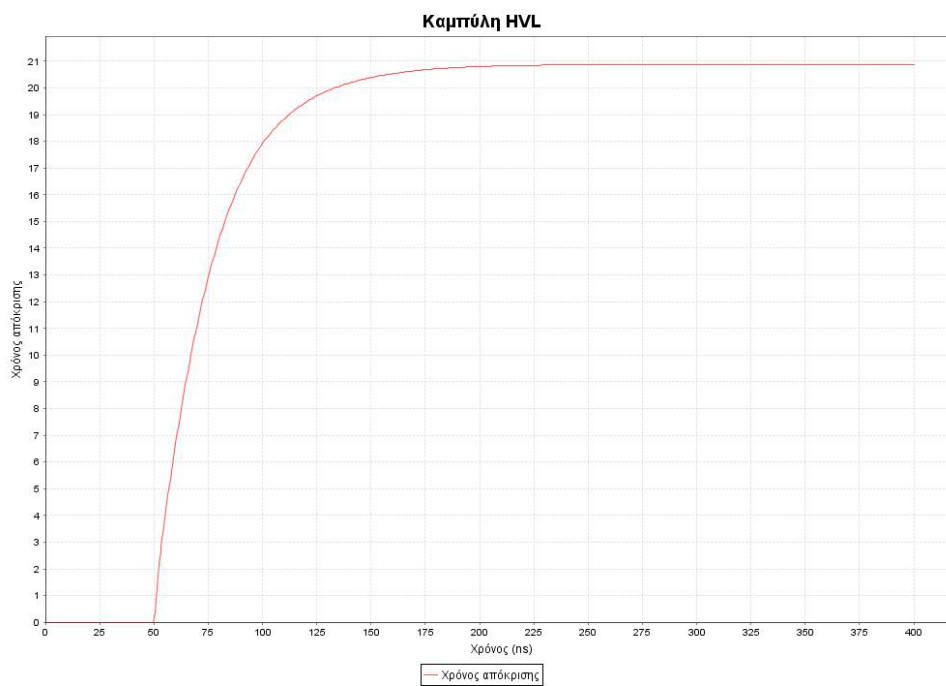
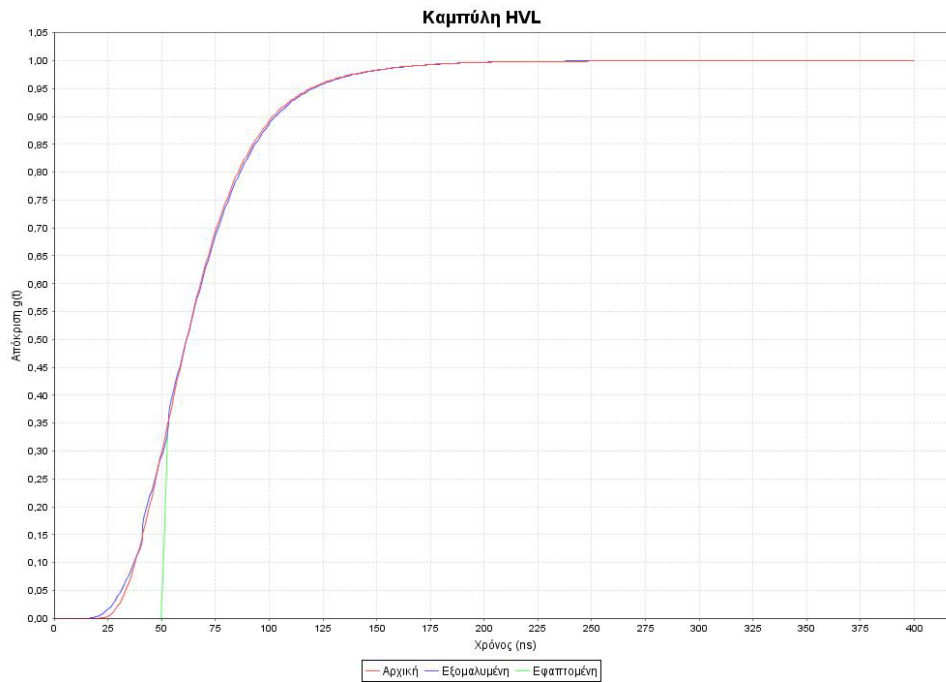
Παράμετρος	Τιμή
$O_1$	49.498
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.793
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.793
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.434

Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	3



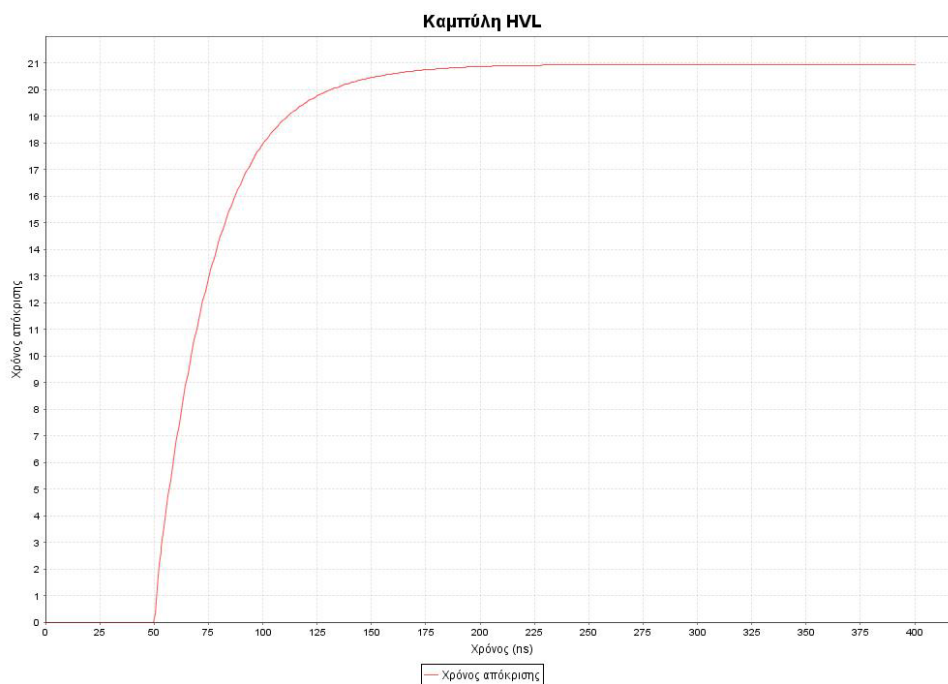
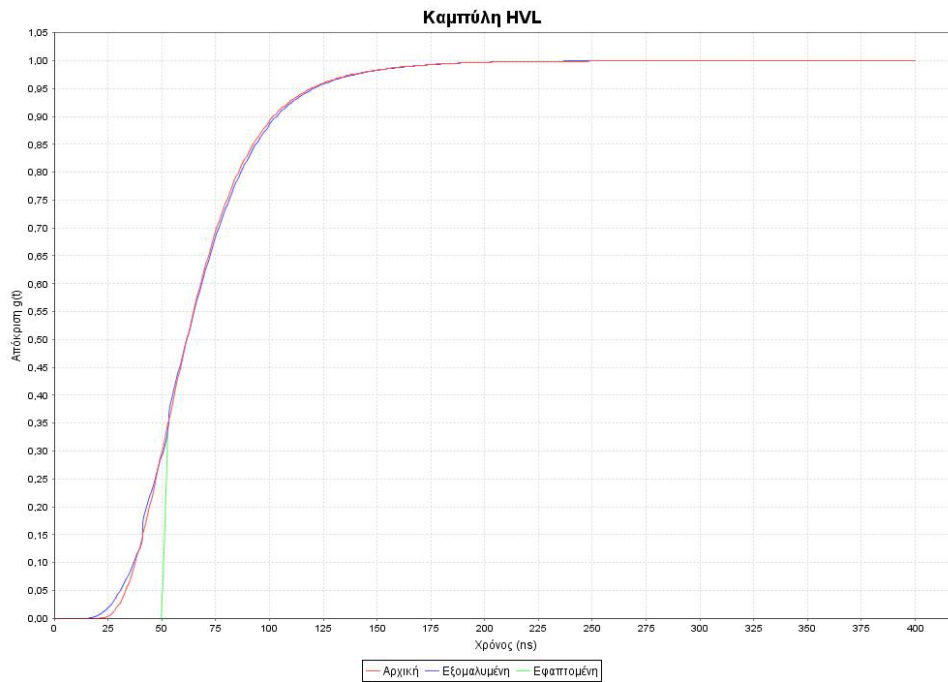
Παράμετρος	Τιμή
$O_1$	49.664
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.810
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.810
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.526

Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	4



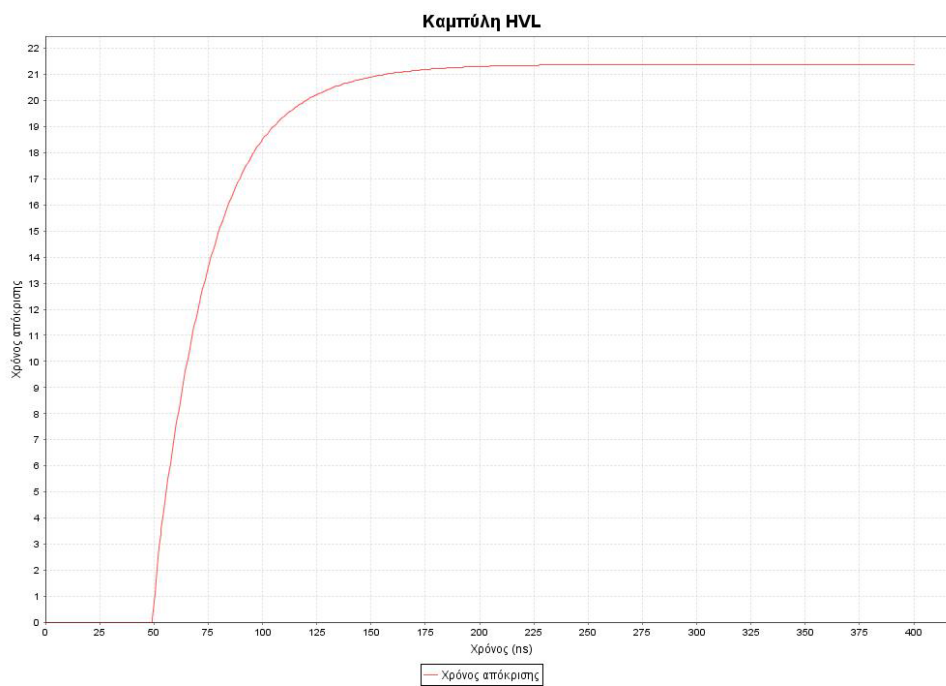
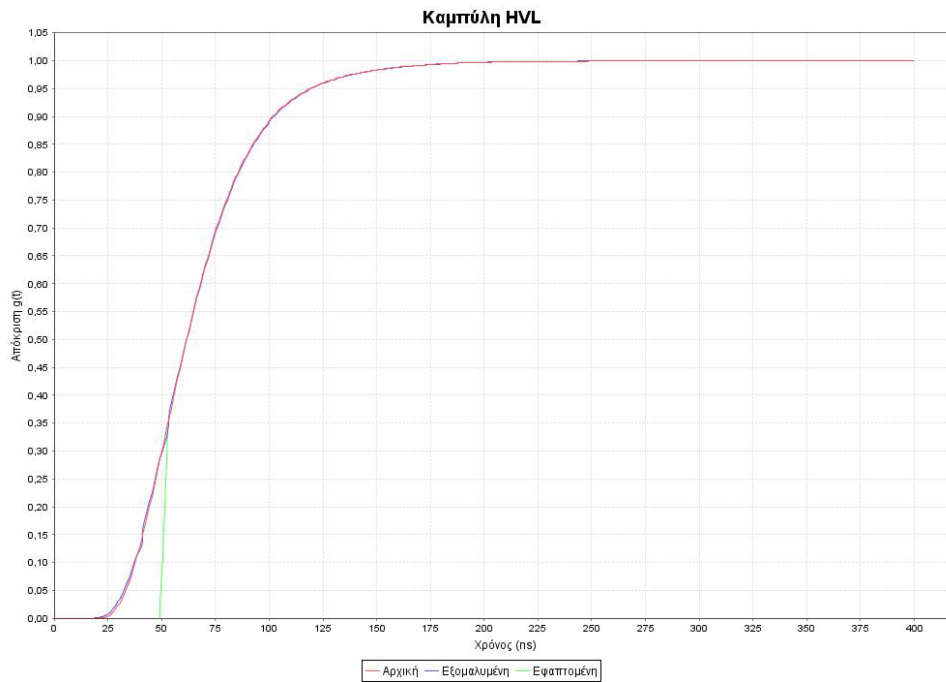
Παράμετρος	Τιμή
$O_1$	49.720
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.870
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.870
Χρόνος ισορρόπησης $T_s$ (ns)	109.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.602

Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	5



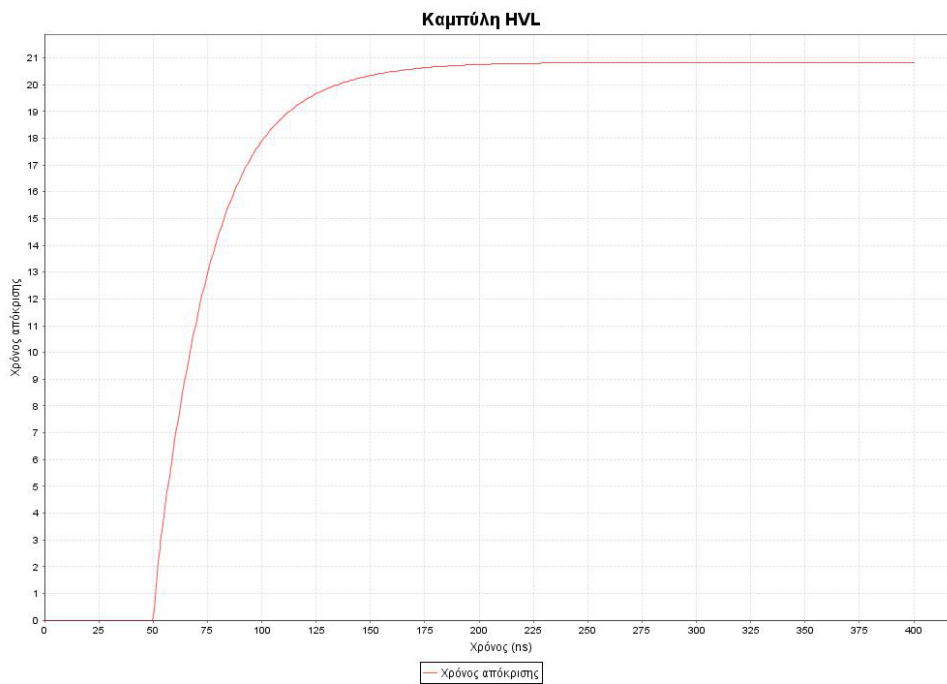
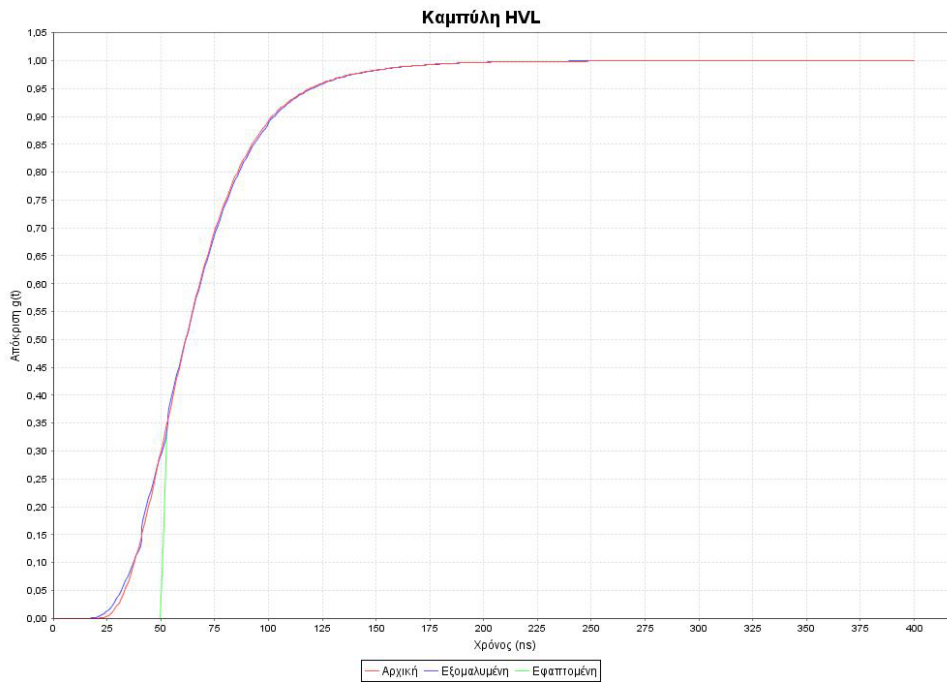
Παράμετρος	Τιμή
$O_1$	49.738
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.945
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.945
Χρόνος ισορρόπησης $T_s$ (ns)	109.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.673

Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	1



Παράμετρος	Τιμή
$O_1$	49.135
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.378
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.378
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.340

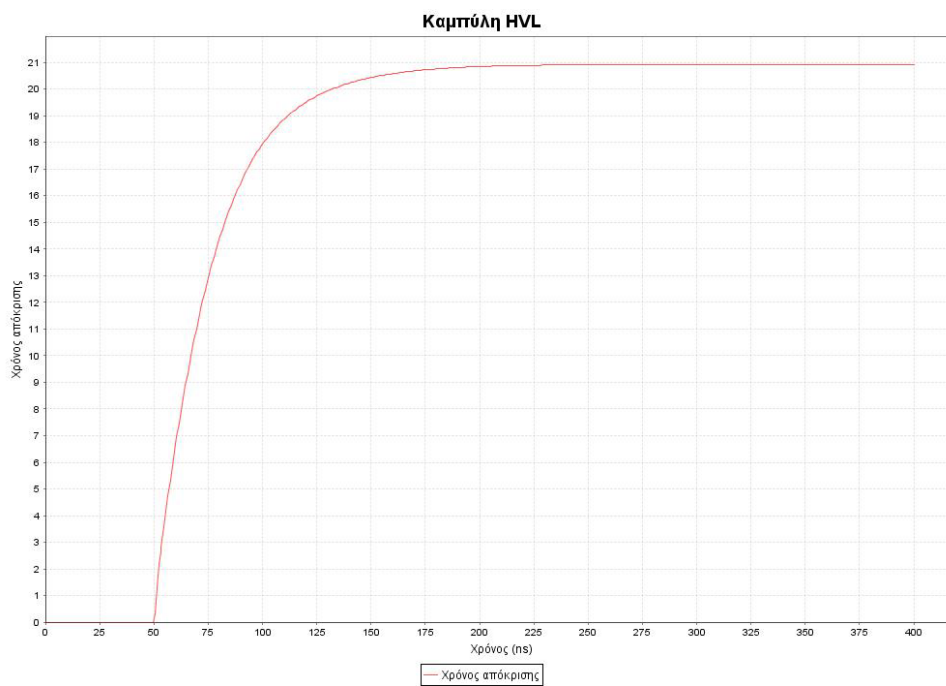
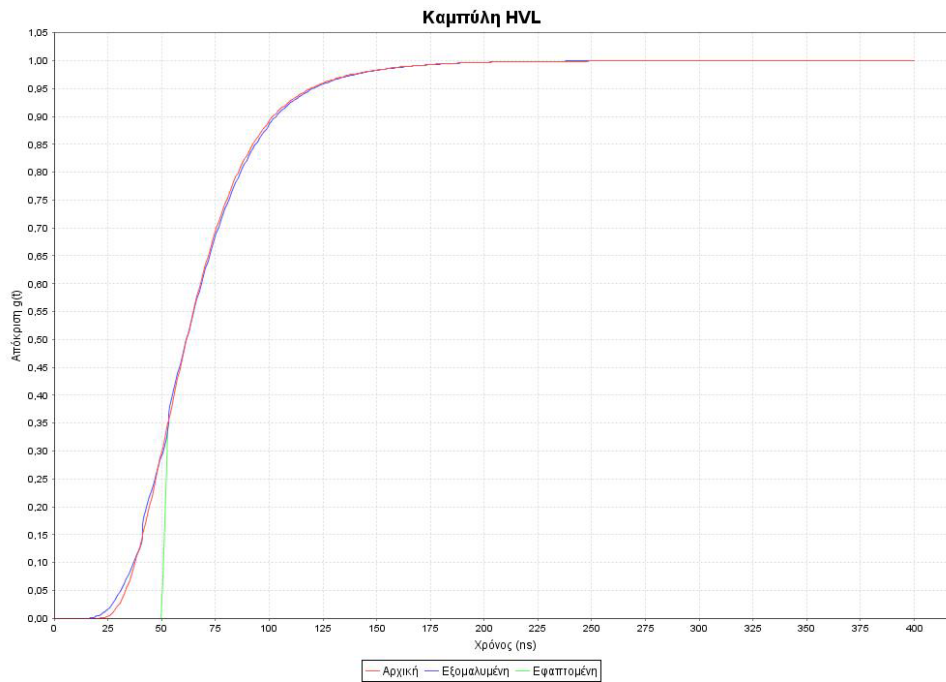
Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	2



Παράμετρος	Τιμή
$O_1$	49.648
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.819
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.819
Χρόνος ισορρόπησης $T_s$ (ns)	109.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.530

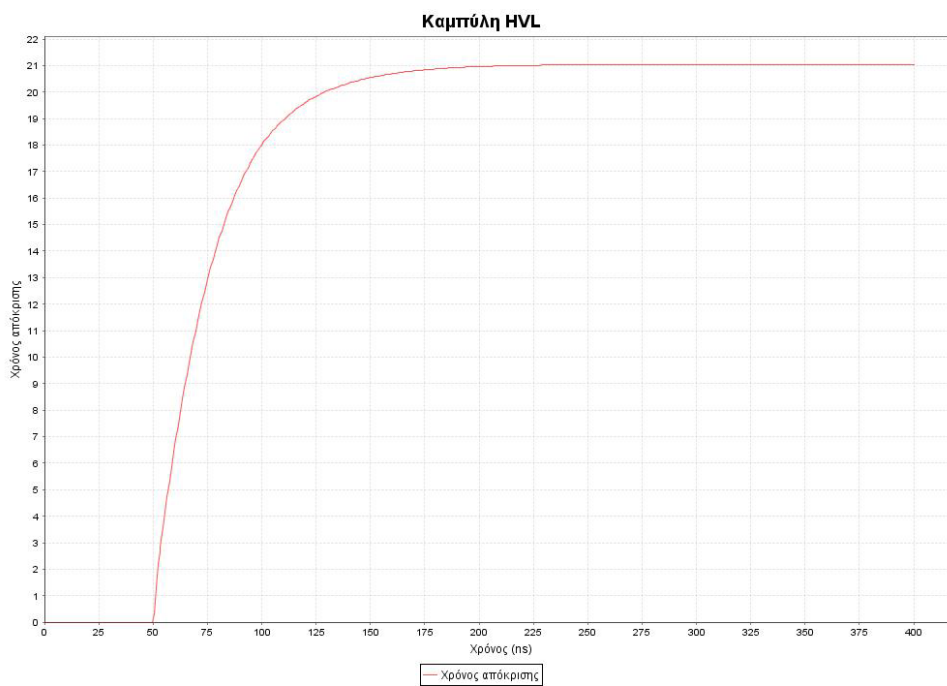
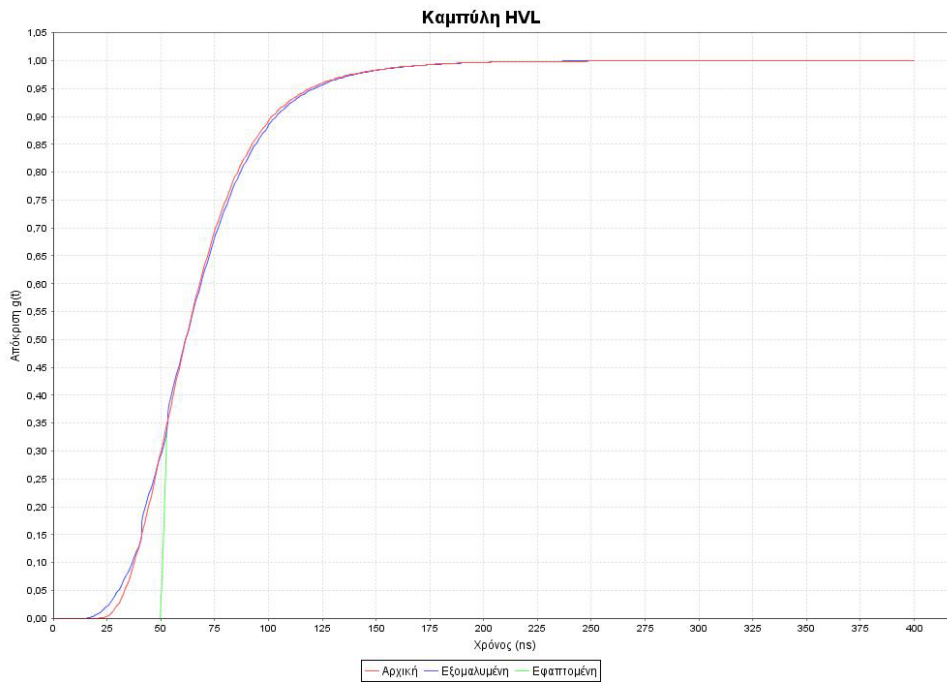


Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	3



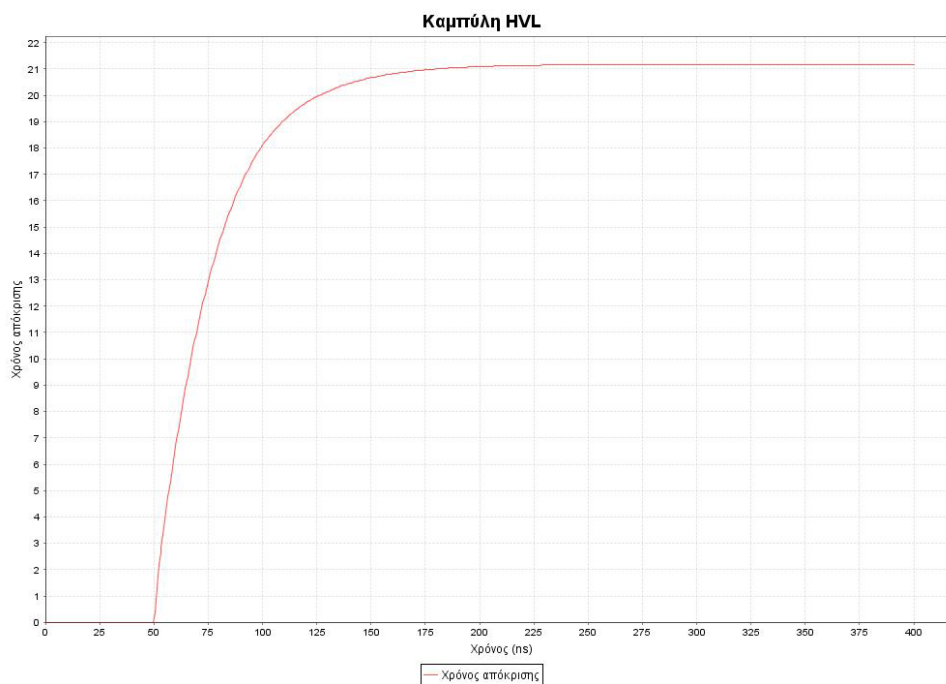
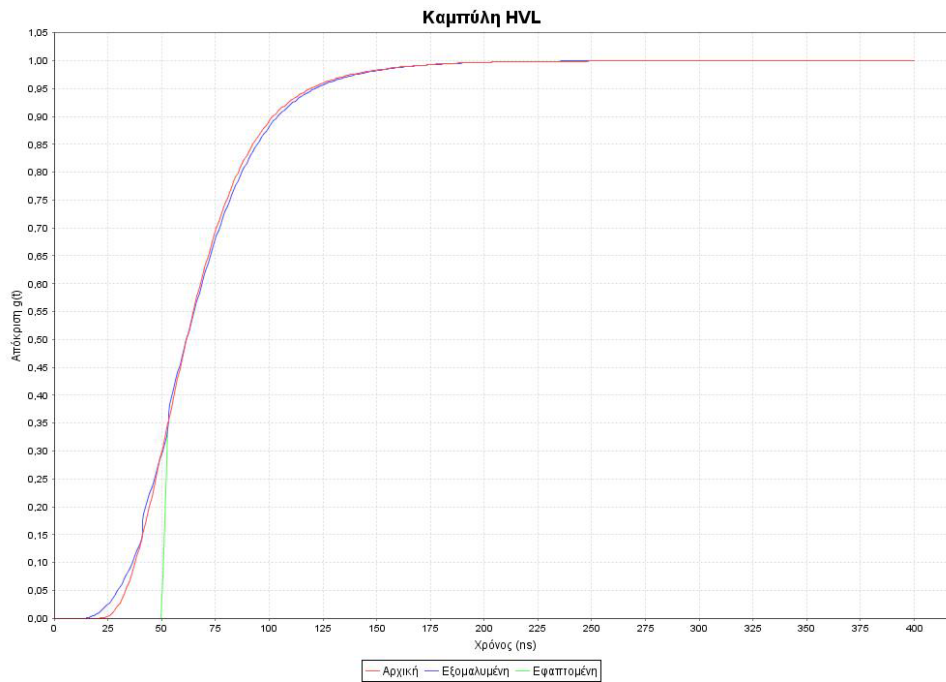
Παράμετρος	Τιμή
$O_1$	49.730
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.912
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.912
Χρόνος ισορρόπησης $T_s$ (ns)	109.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.643

Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	4



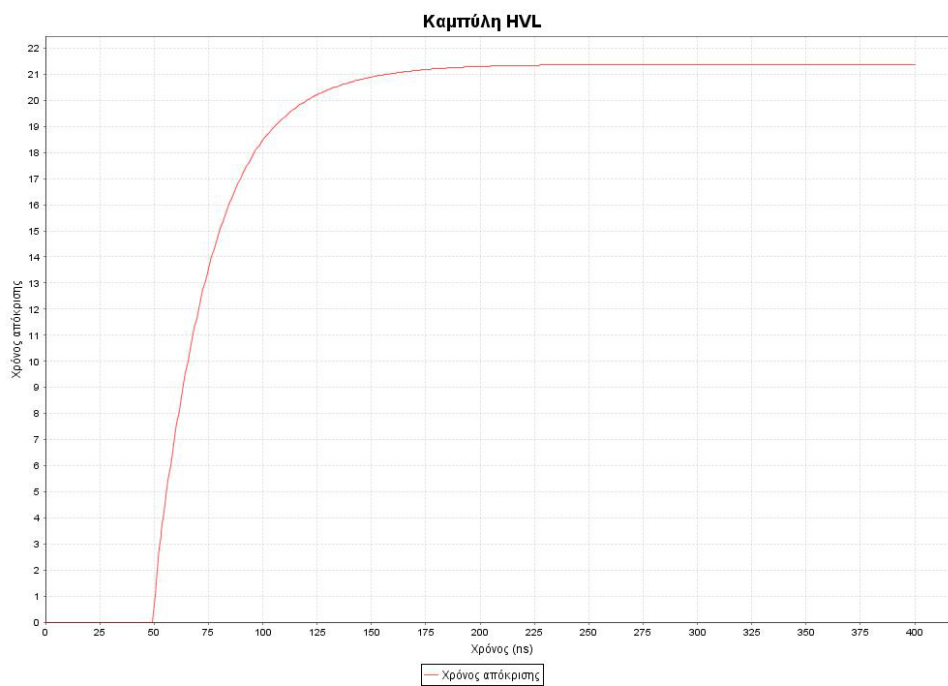
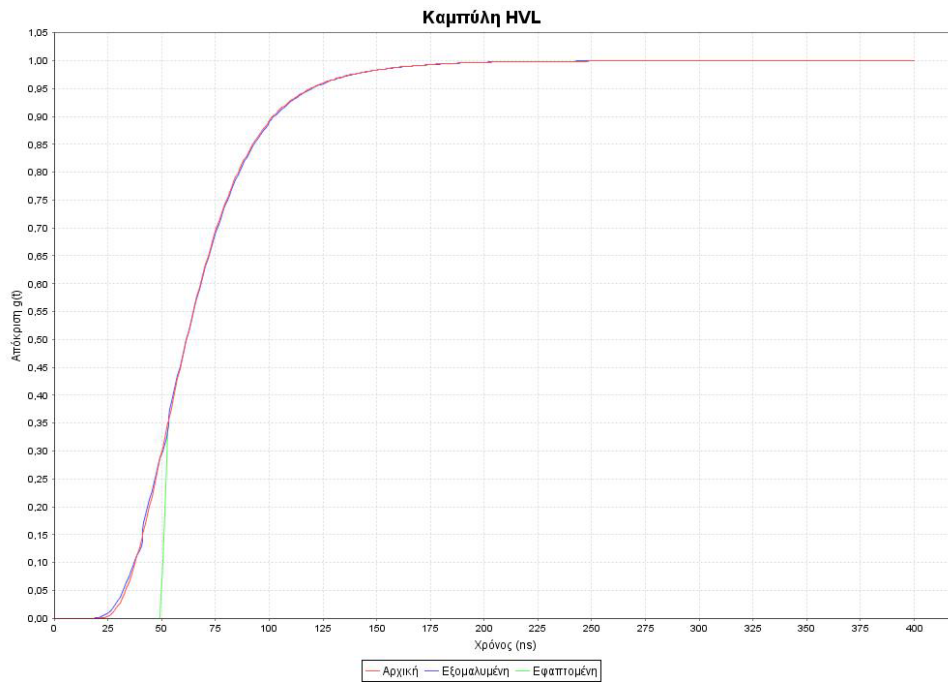
Παράμετρος	Τιμή
$O_1$	49.736
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.034
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.034
Χρόνος ισορρόπησης $T_s$ (ns)	109.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.746

Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	5



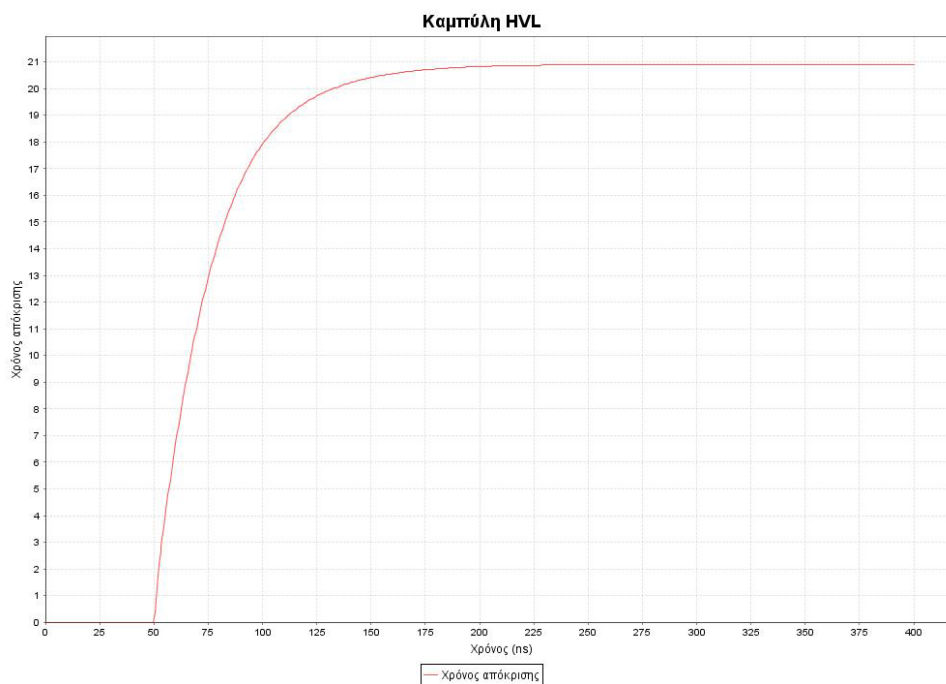
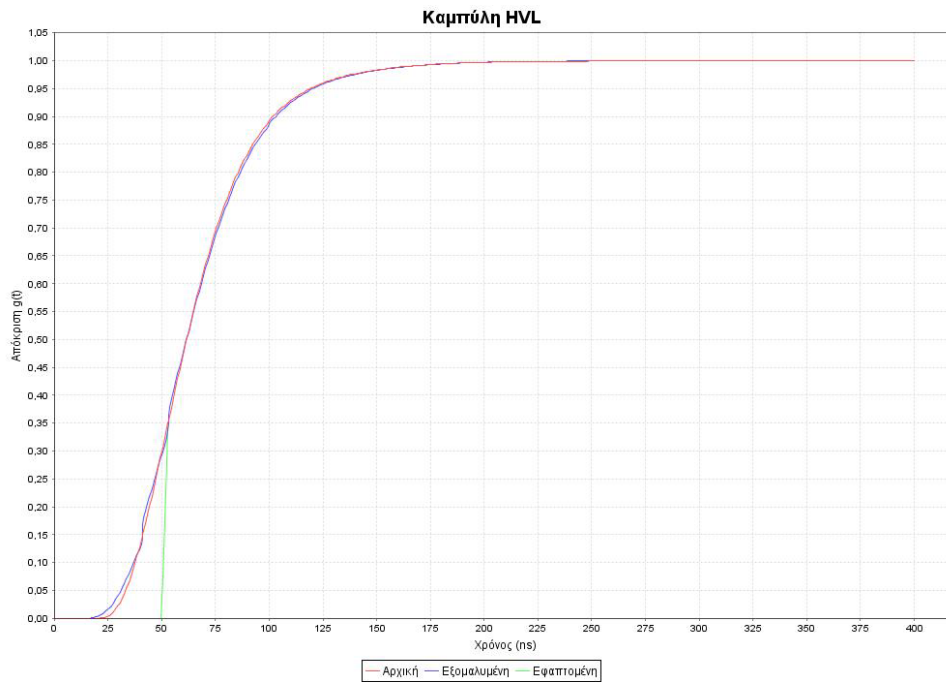
Παράμετρος	Τιμή
$O_1$	49.715
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.166
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.166
Χρόνος ισορρόπησης $T_s$ (ns)	110.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.846

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	1



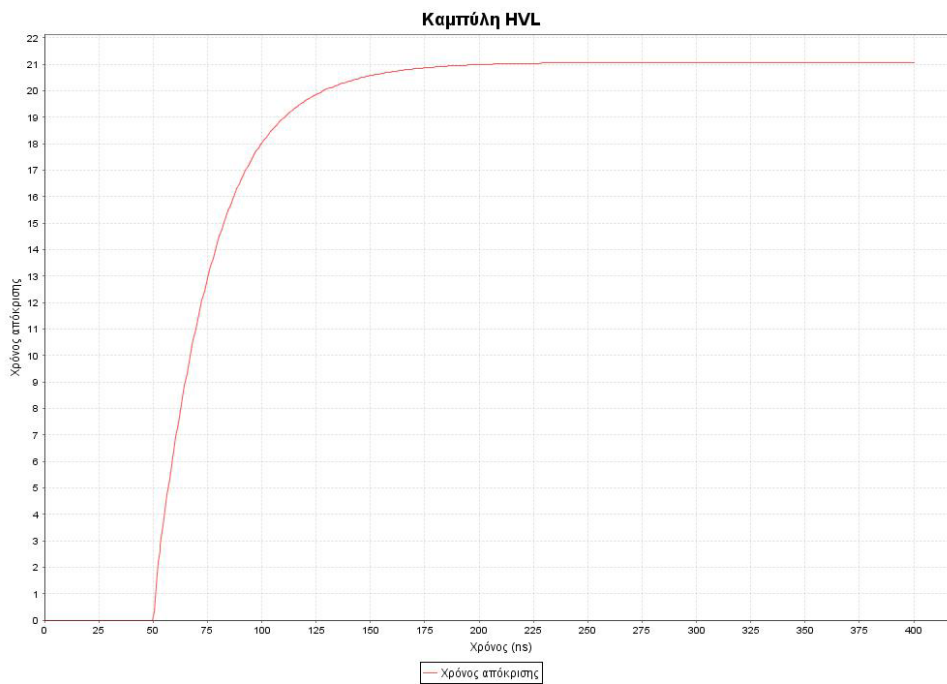
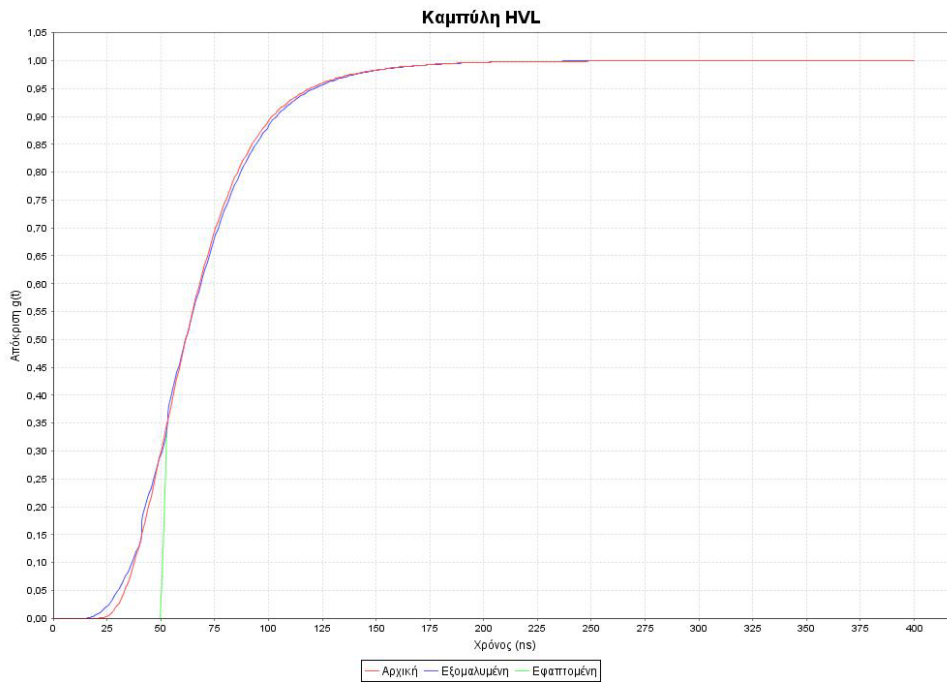
Παράμετρος	Τιμή
$O_1$	49.326
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.367
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.367
Χρόνος ισορρόπησης $T_s$ (ns)	108.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.416

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	2



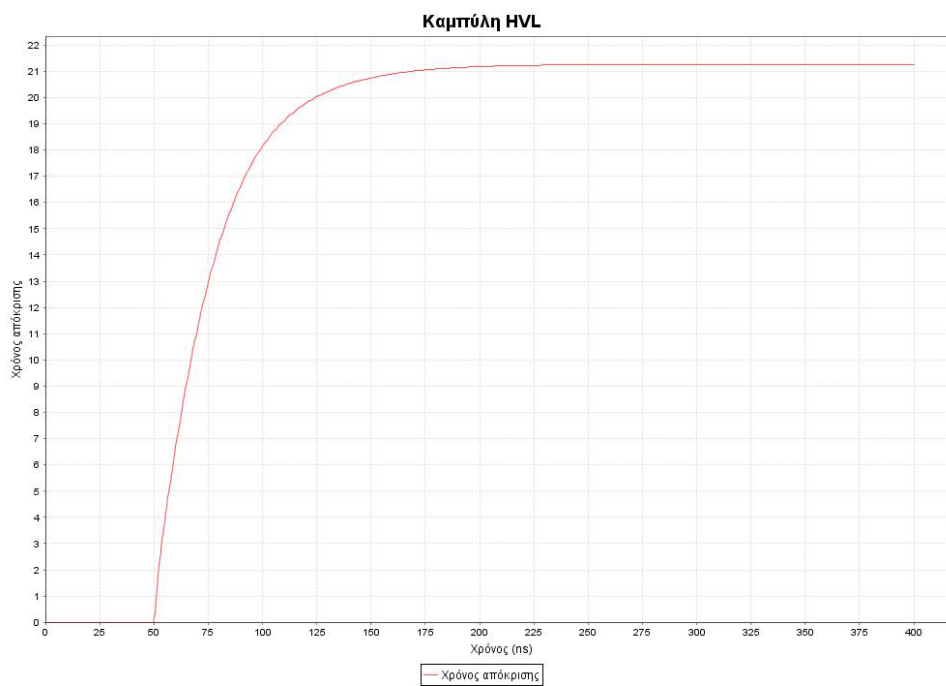
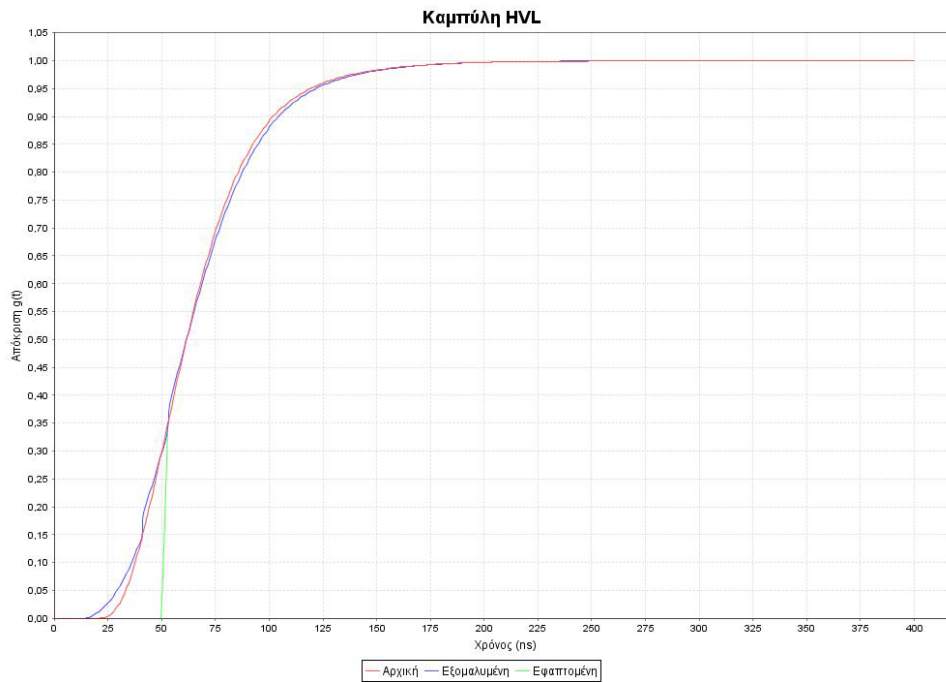
Παράμετρος	Τιμή
$O_1$	49.714
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	20.896
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	20.896
Χρόνος ισορρόπησης $T_s$ (ns)	109.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.624

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	3



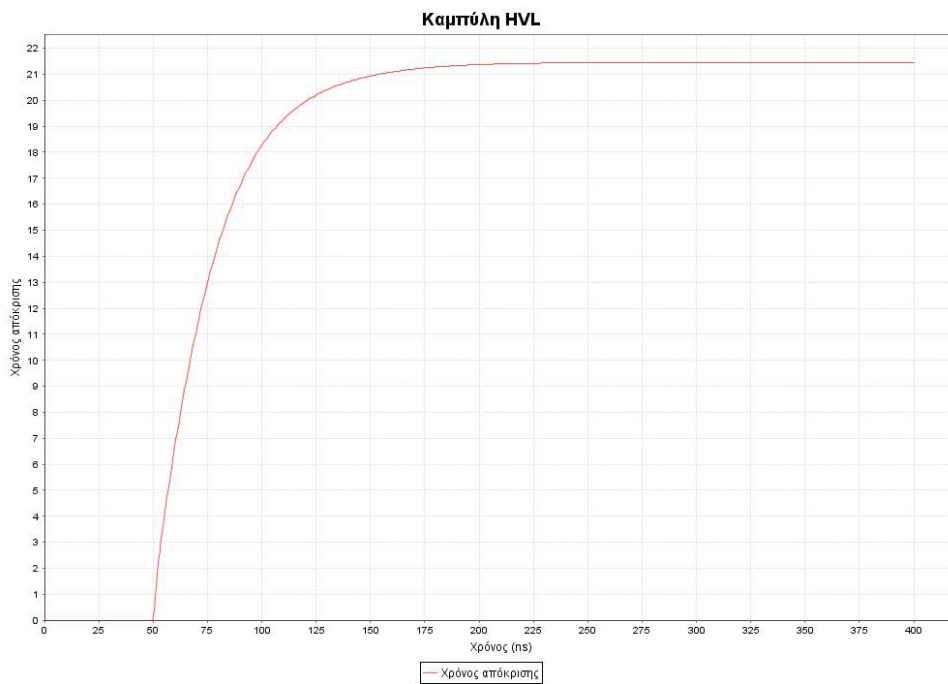
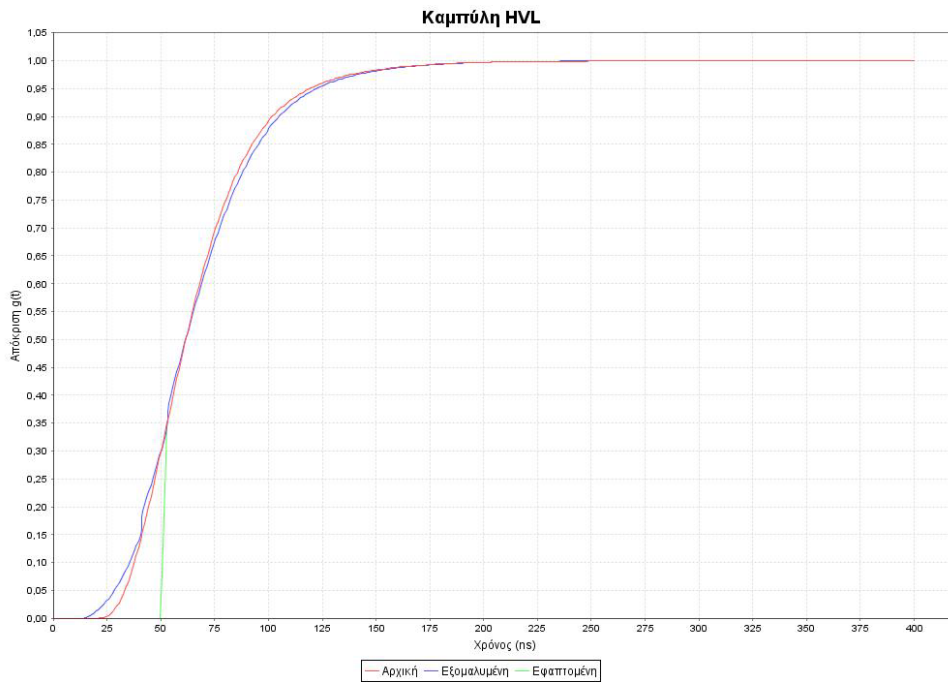
Παράμετρος	Τιμή
$O_1$	49.733
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.063
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.063
Χρόνος ισορρόπησης $T_s$ (ns)	109.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.770

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	4



Παράμετρος	Τιμή
$O_1$	49.696
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.252
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.252
Χρόνος ισορρόπησης $T_s$ (ns)	110.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	3.908

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	5



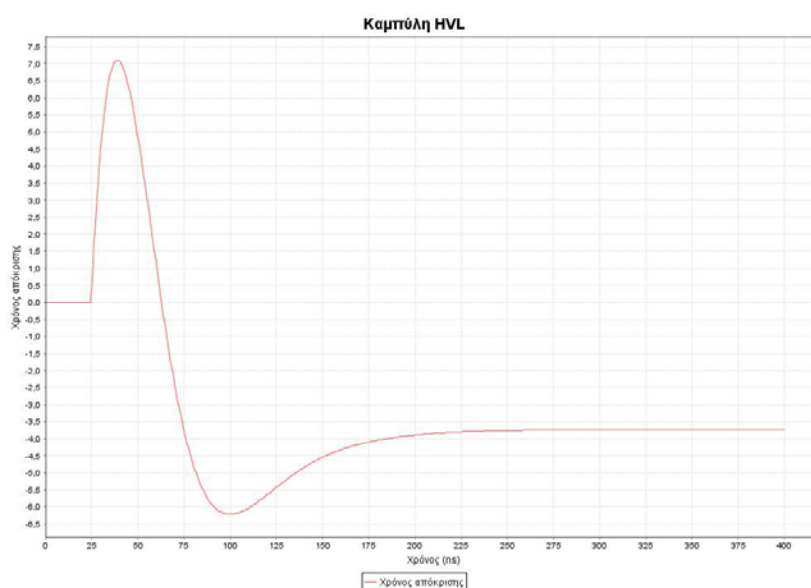
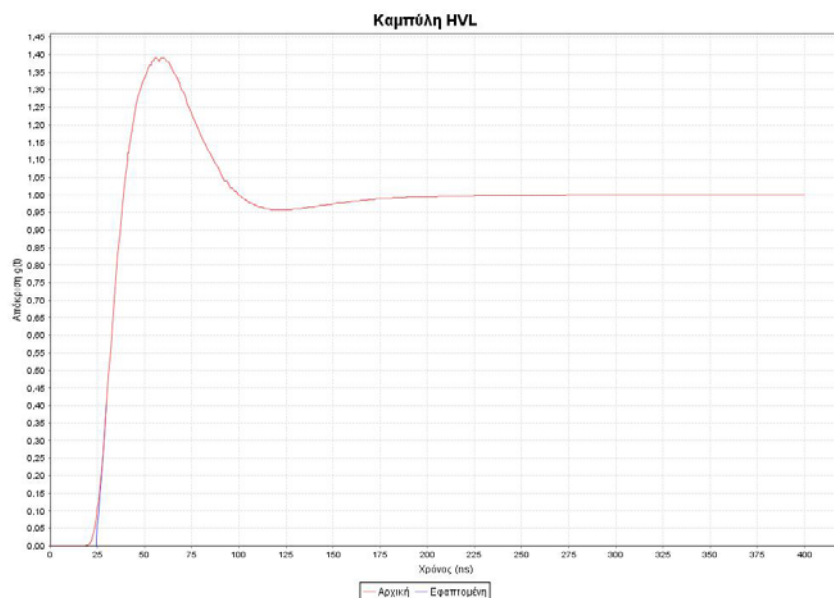
Παράμετρος	Τιμή
$O_1$	49.642
Υπερύψωση b	0.0
Μερικός χρόνος απόκρισης $T_a$ (ns)	21.445
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	21.445
Χρόνος ισορρόπησης $T_s$ (ns)	110.0
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	4.043



## Παράρτημα Α2

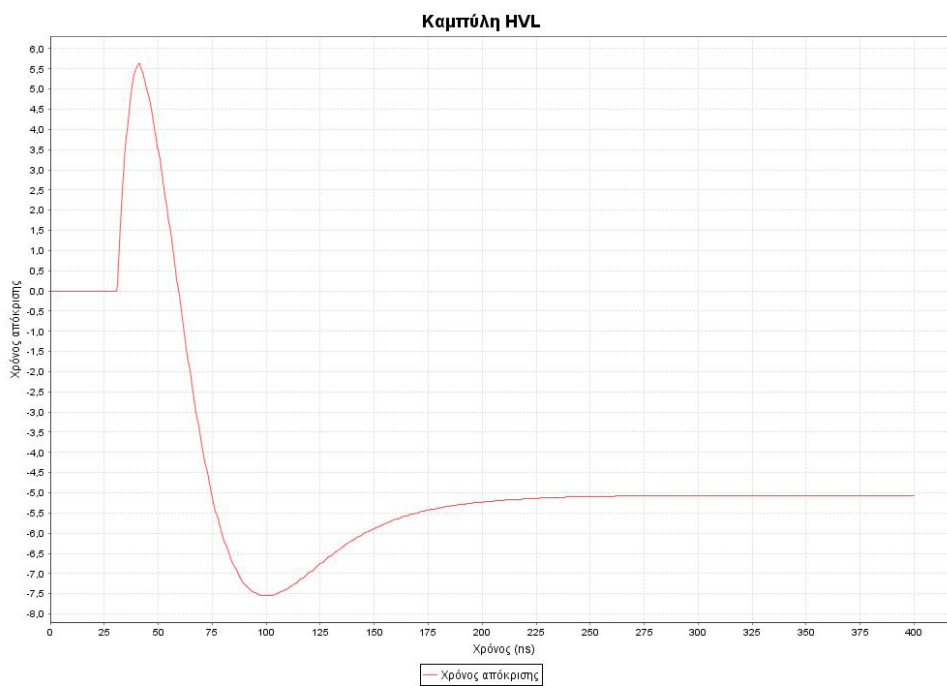
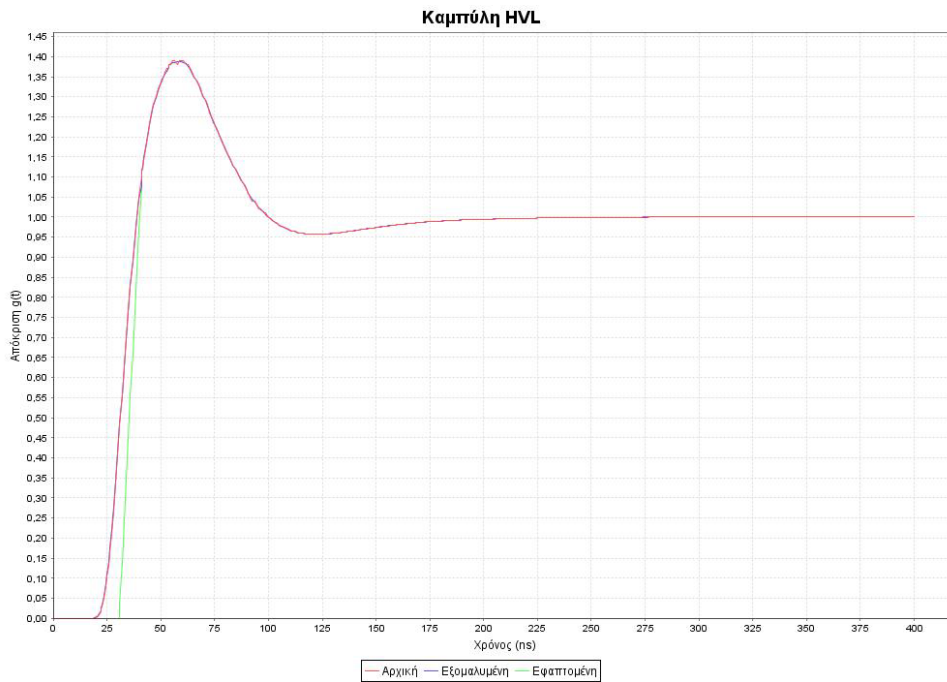
### Ανάλυση παραμέτρων συστήματος μέτρησης με αντιστάθμιση

Αριθμός γειτονικών σημείων	0
Αριθμός επαναλήψεων	0



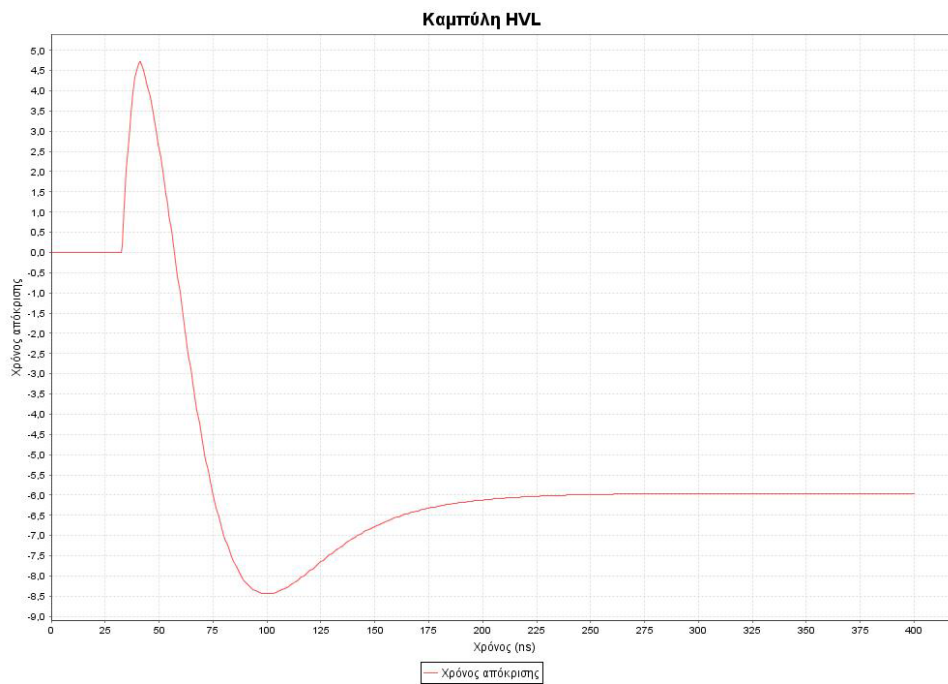
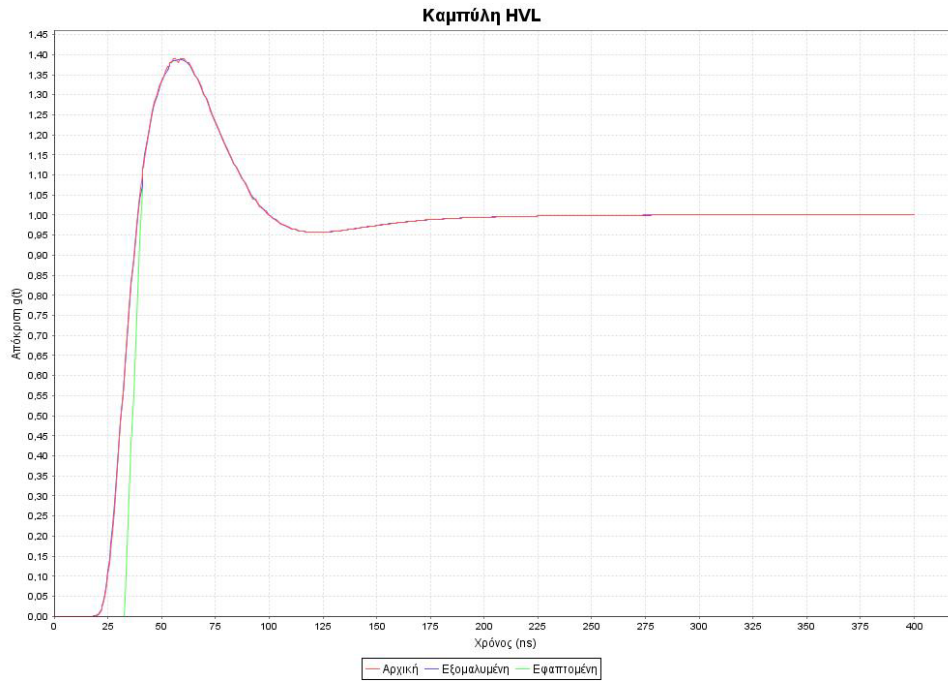
Παράμετρος	Τιμή
$O_1$	24.328
Υπερύψωση b	0.389
Μερικός χρόνος απόκρισης $T_a$ (ns)	7.115
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-3.735
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	0.227

Αριθμός γειτονικών σημείων	2
Αριθμός επαναλήψεων	2



Παράμετρος	Τιμή
$O_1$	30.548
Υπερύψωση b	0.388
Μερικός χρόνος απόκρισης $T_a$ (ns)	5.552
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-5.071
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	4.442

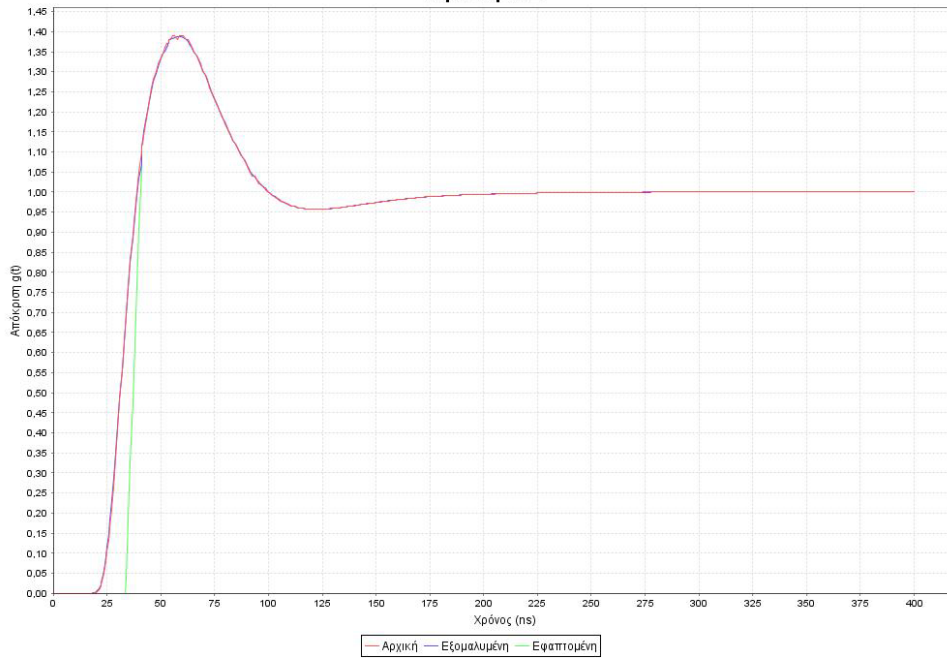
Αριθμός γειτονικών σημείων	2
Αριθμός επαναλήψεων	3



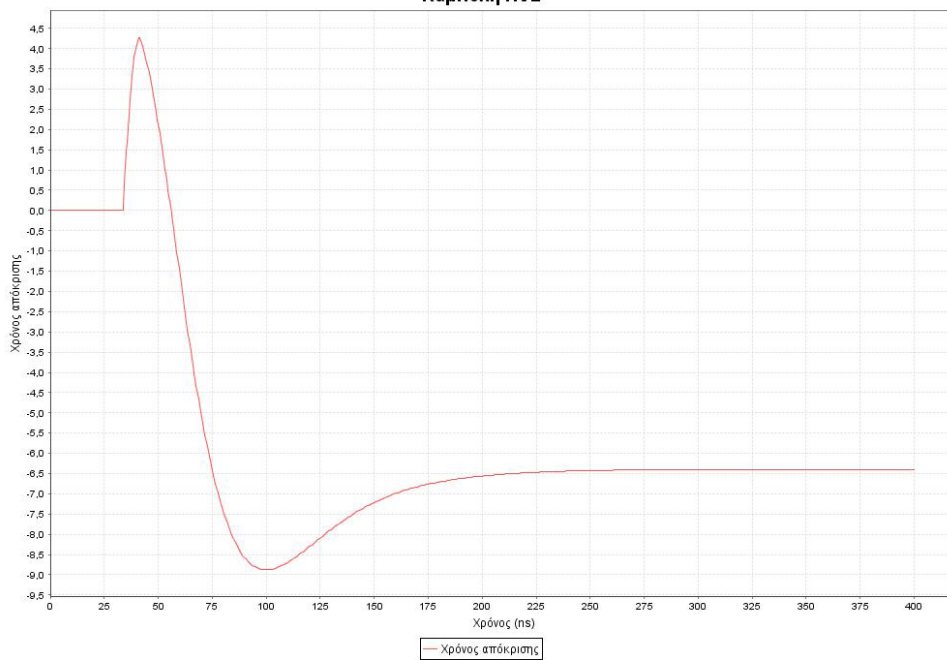
Παράμετρος	Τιμή
$O_1$	32.550
Υπερύψωση b	0.387
Μερικός χρόνος απόκρισης $T_a$ (ns)	4.597
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-5.964
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	5.565

Αριθμός γειτονικών σημείων	2
Αριθμός επαναλήψεων	4

Καμπύλη HVL

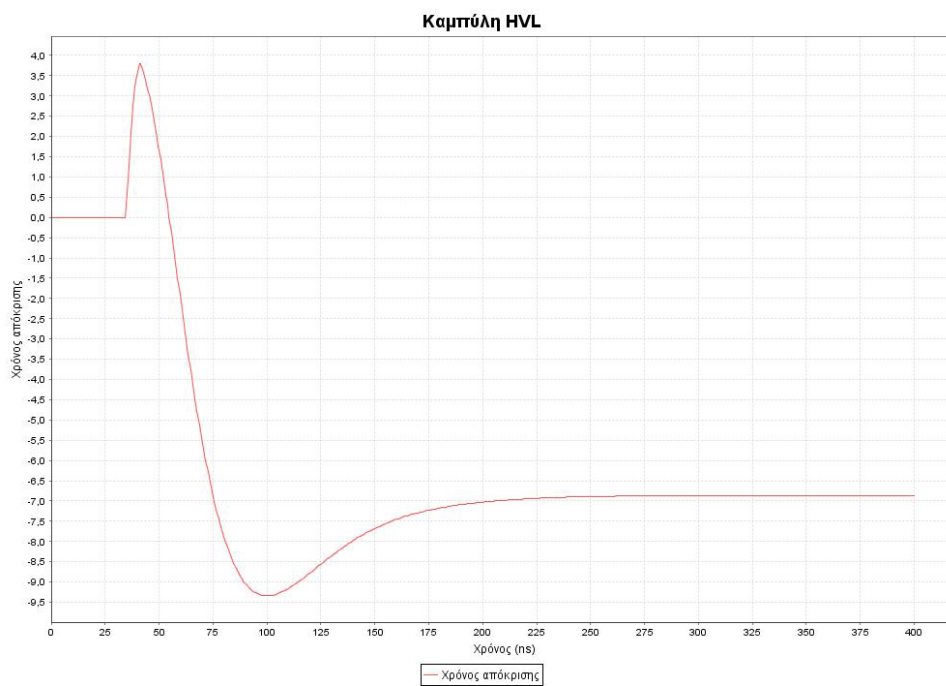
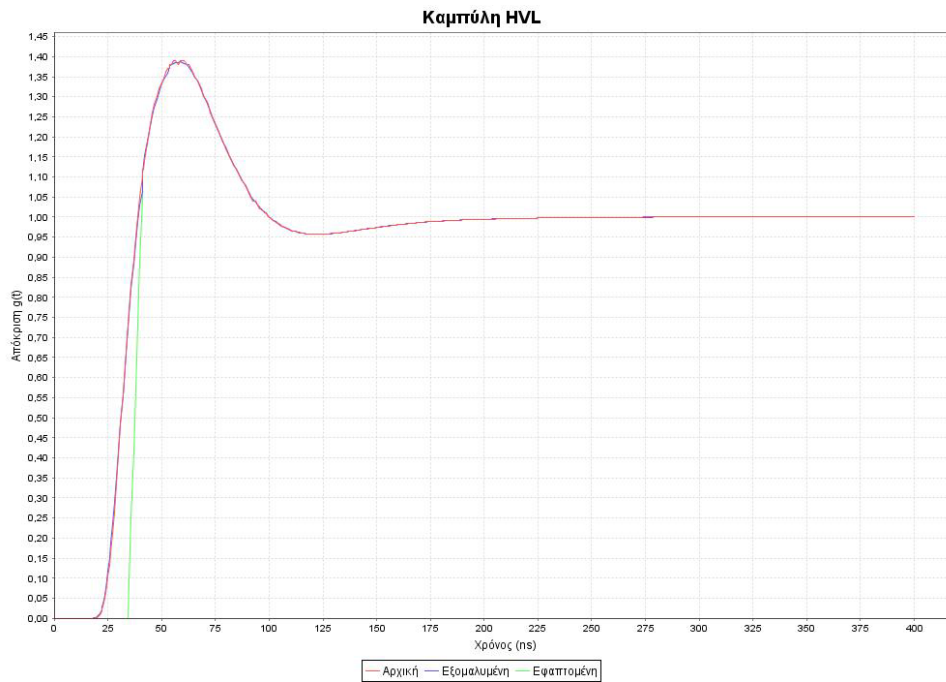


Καμπύλη HVL



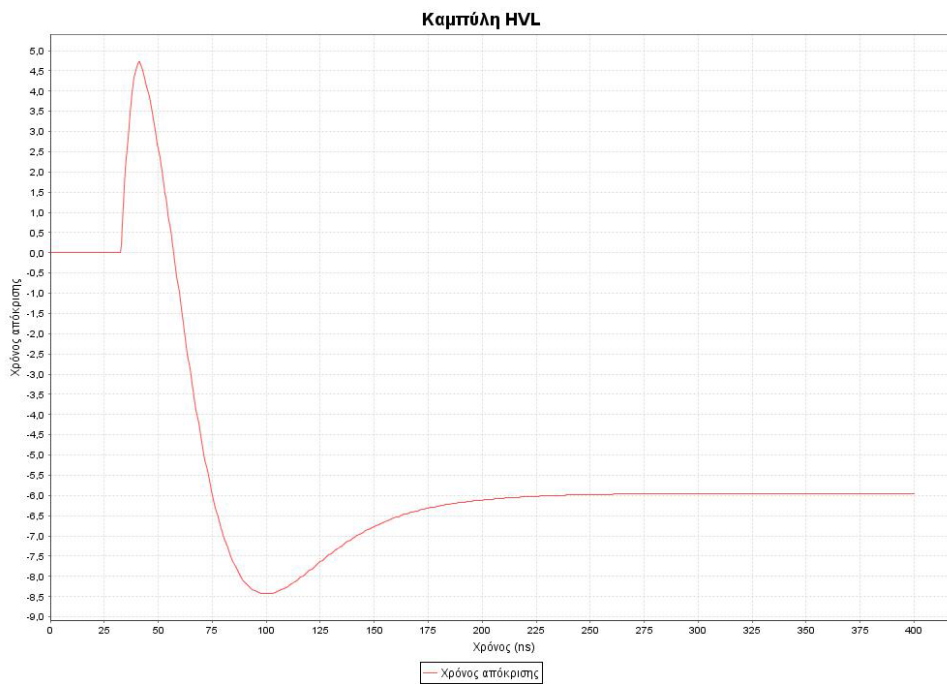
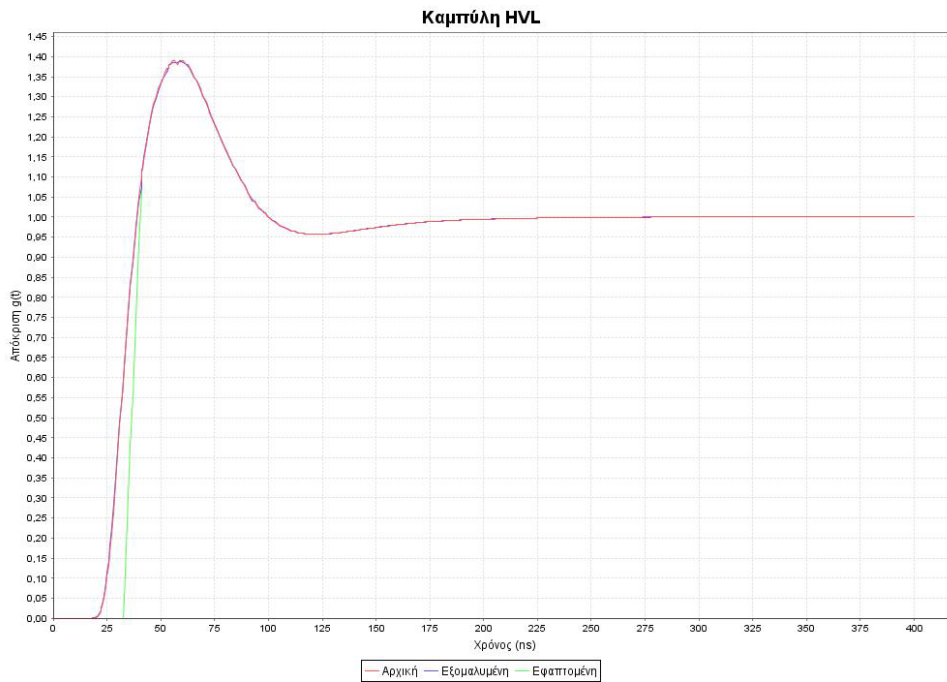
Παράμετρος	Τιμή
$O_1$	33.622
Υπερύψωση b	0.387
Μερικός χρόνος απόκρισης $T_a$ (ns)	4.108
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-6.403
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	6.167

Αριθμός γειτονικών σημείων	2
Αριθμός επαναλήψεων	5



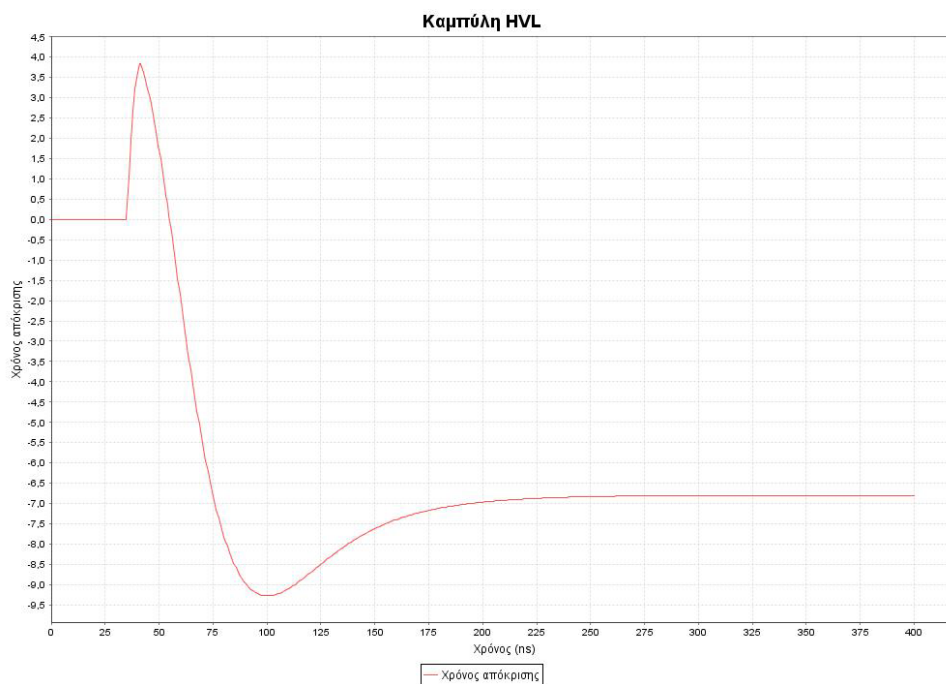
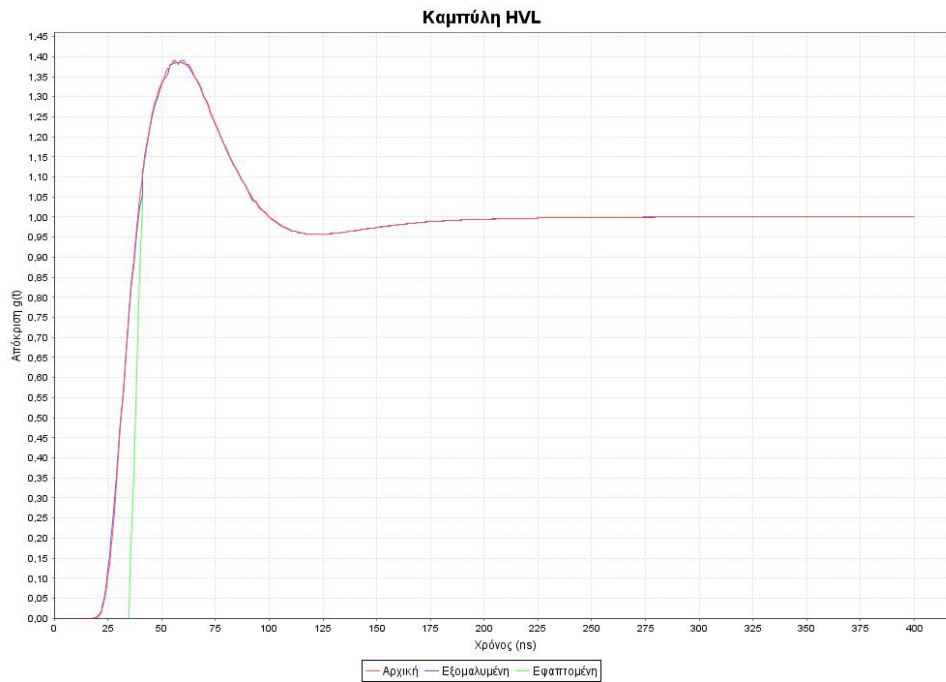
Παράμετρος	Τιμή
$O_1$	34.290
Υπερύψωση b	0.386
Μερικός χρόνος απόκρισης $T_a$ (ns)	3.603
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-6.868
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	6.557

Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	1



Παράμετρος	Τιμή
$O_1$	32.484
Υπερύψωση b	0.388
Μερικός χρόνος απόκρισης $T_a$ (ns)	4.603
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-5.954
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	5.556

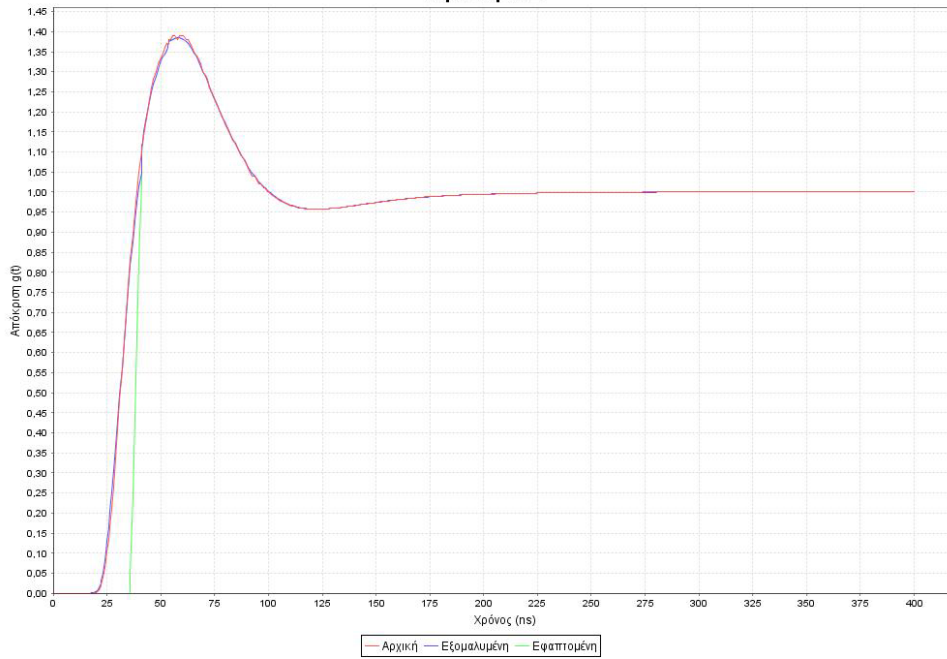
Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	2



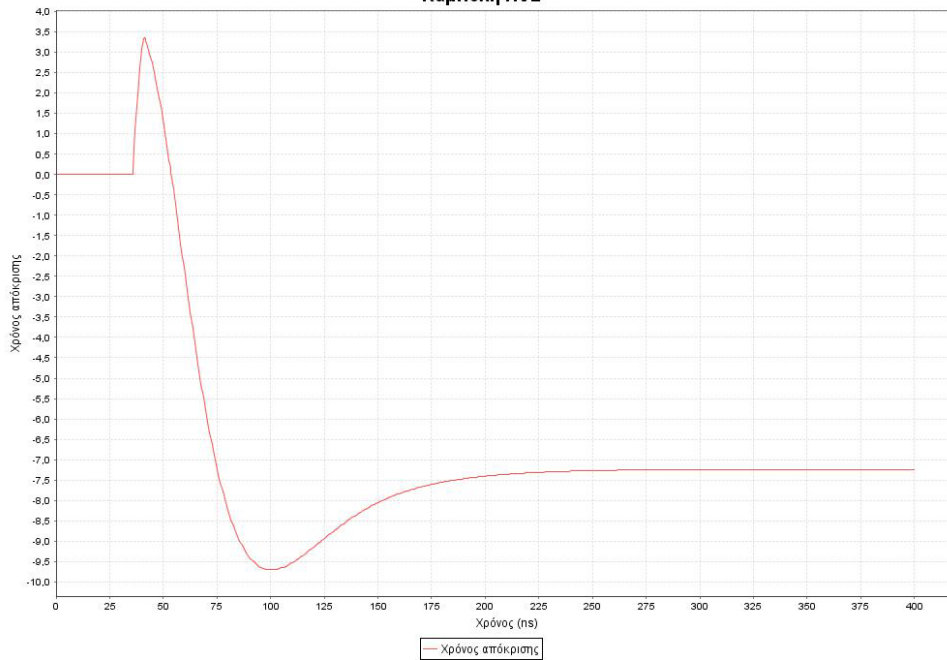
Παράμετρος	Τιμή
$O_1$	34.800
Υπερύψωση b	0.386
Μερικός χρόνος απόκρισης $T_a$ (ns)	3.613
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-6.807
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	6.864

Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	3

Καμπύλη HVL



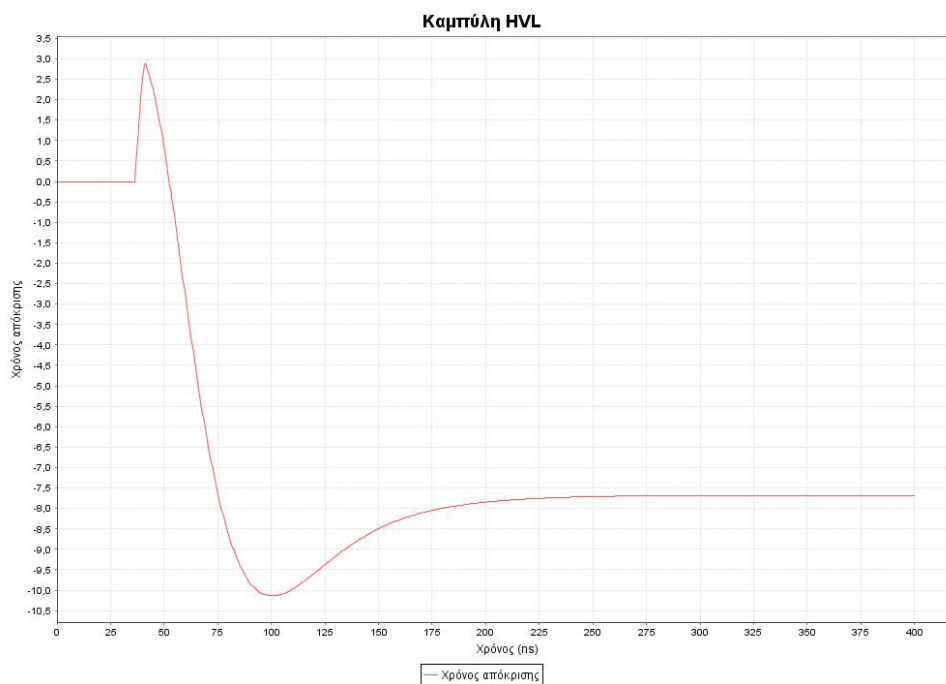
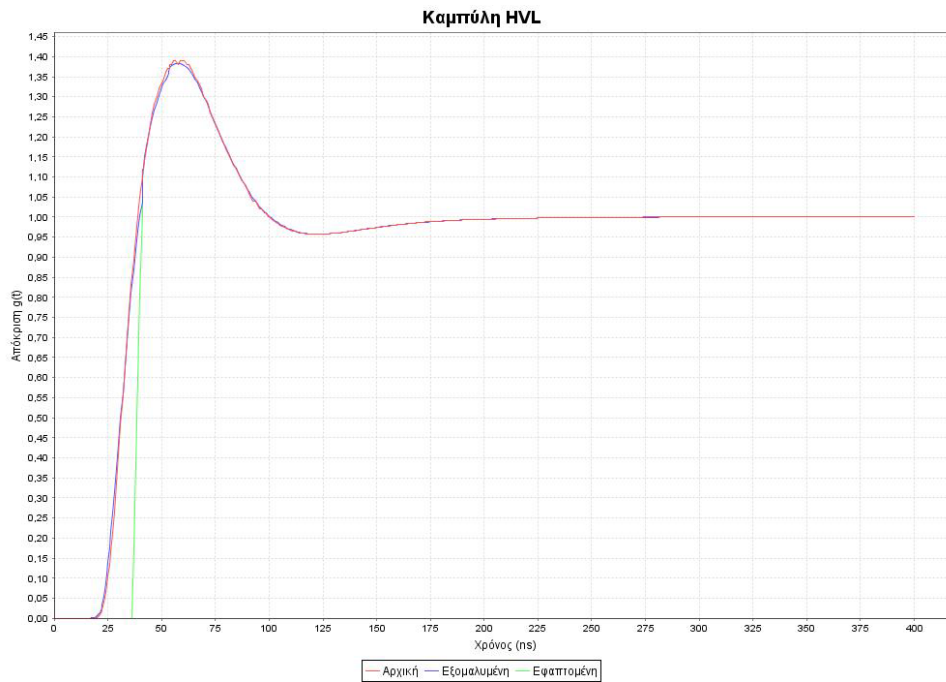
Καμπύλη HVL



Παράμετρος	Τιμή
$O_1$	35.647
Υπερύψωση b	0.384
Μερικός χρόνος απόκρισης $T_a$ (ns)	3.077
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.246
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	7.369



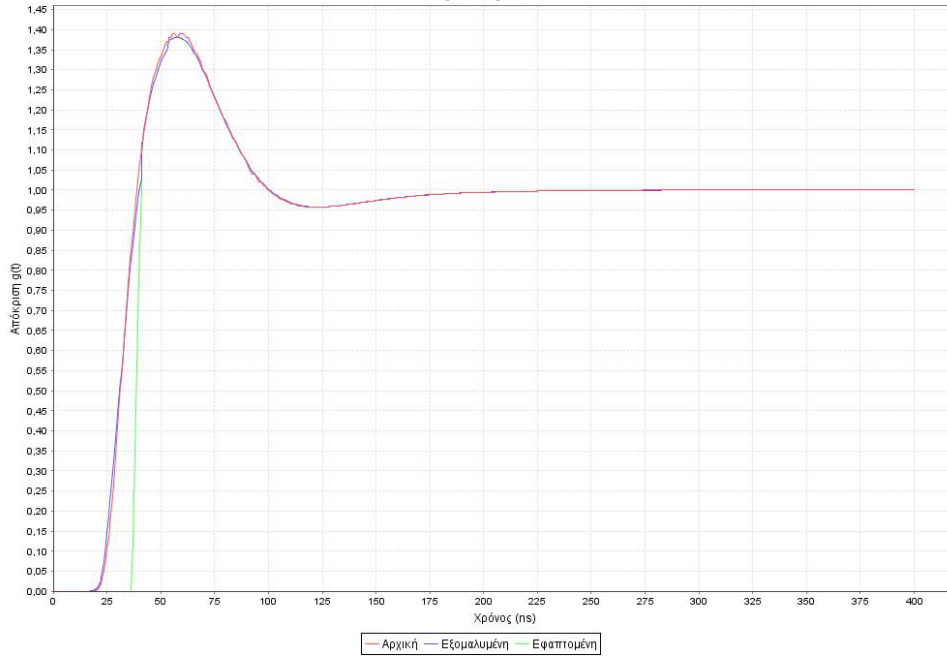
Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	4



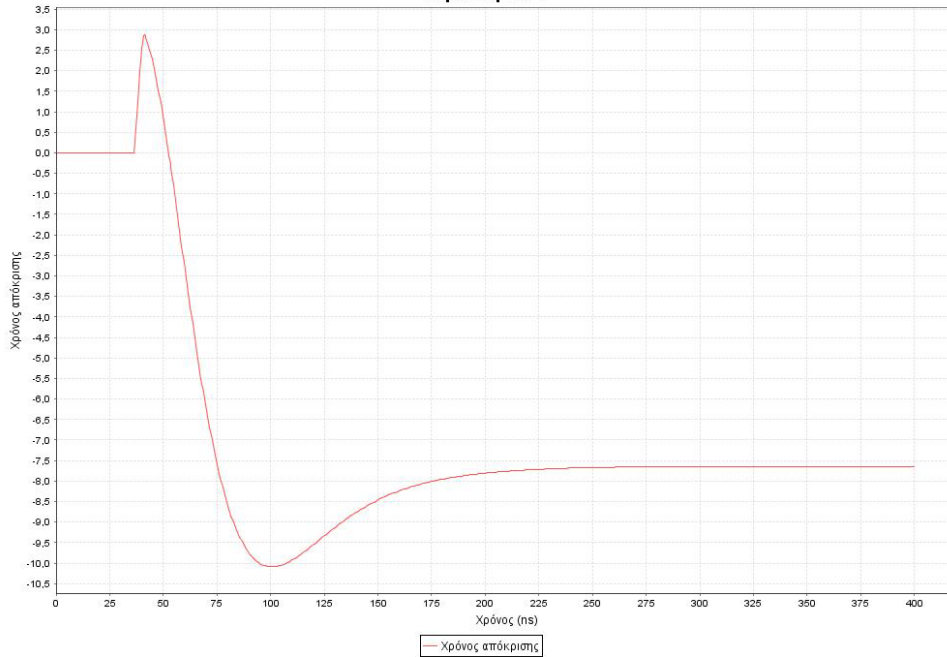
Παράμετρος	Τιμή
$O_1$	36.076
Υπερύψωση b	0.382
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.559
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.687
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	7.654

Αριθμός γειτονικών σημείων	4
Αριθμός επαναλήψεων	5

Καμπύλη HVL

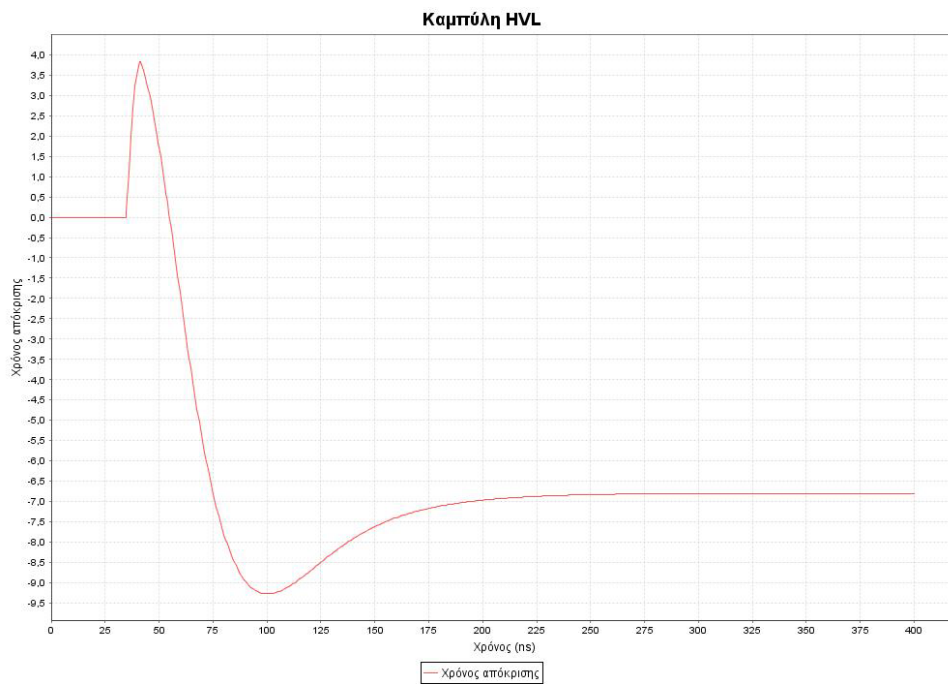
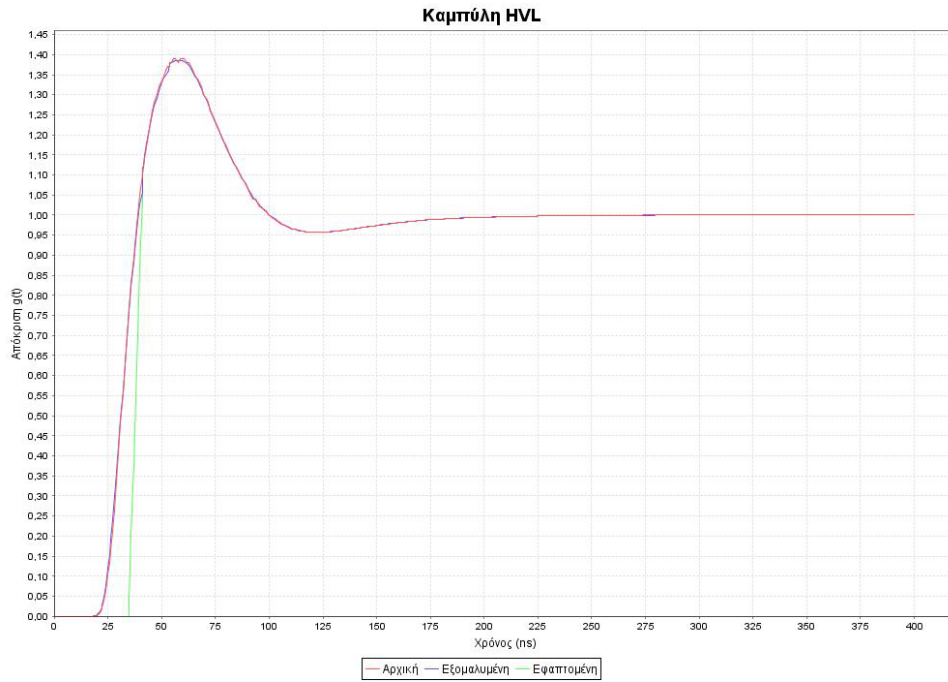


Καμπύλη HVL



Παράμετρος	Τιμή
$O_1$	36.336
Υπερύψωση b	0.380
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.855
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.647
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	7.823

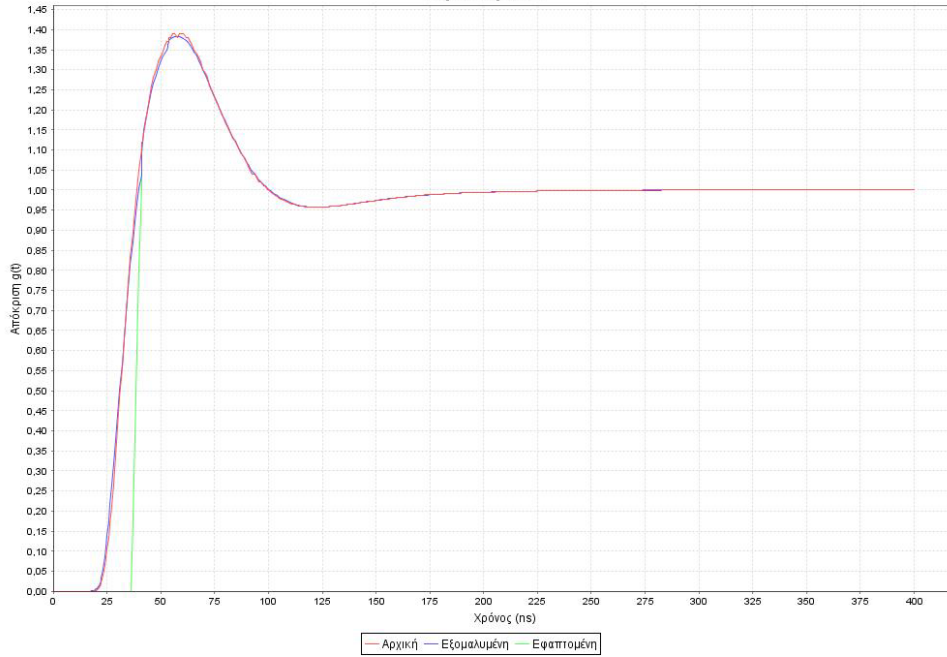
Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	1



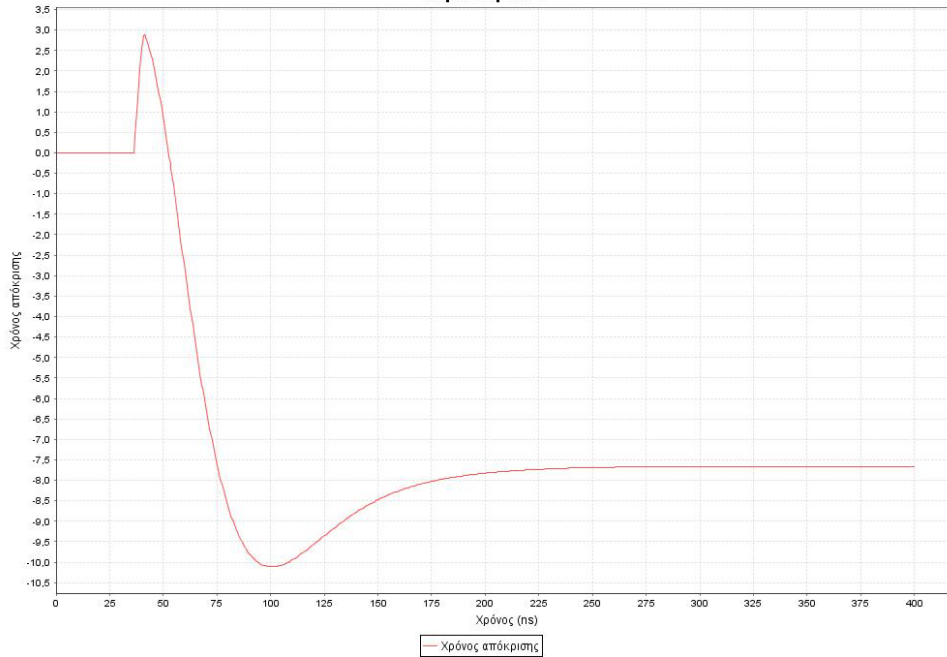
Παράμετρος	Τιμή
$O_1$	34.636
Υπερύψωση b	0.386
Μερικός χρόνος απόκρισης $T_a$ (ns)	3.612
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-6.807
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	6.806

Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	2

Καμπύλη HVL



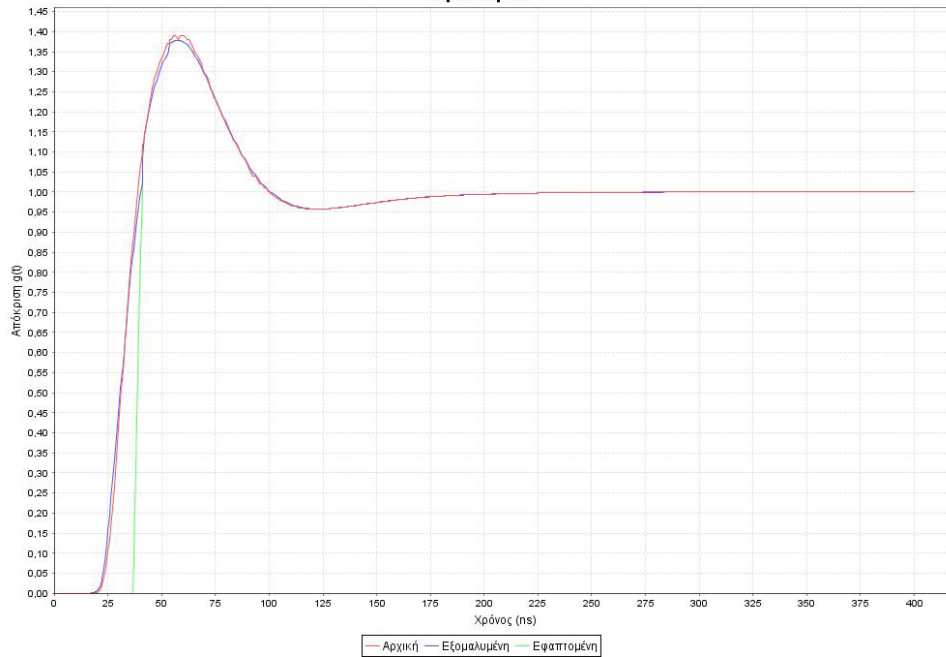
Καμπύλη HVL



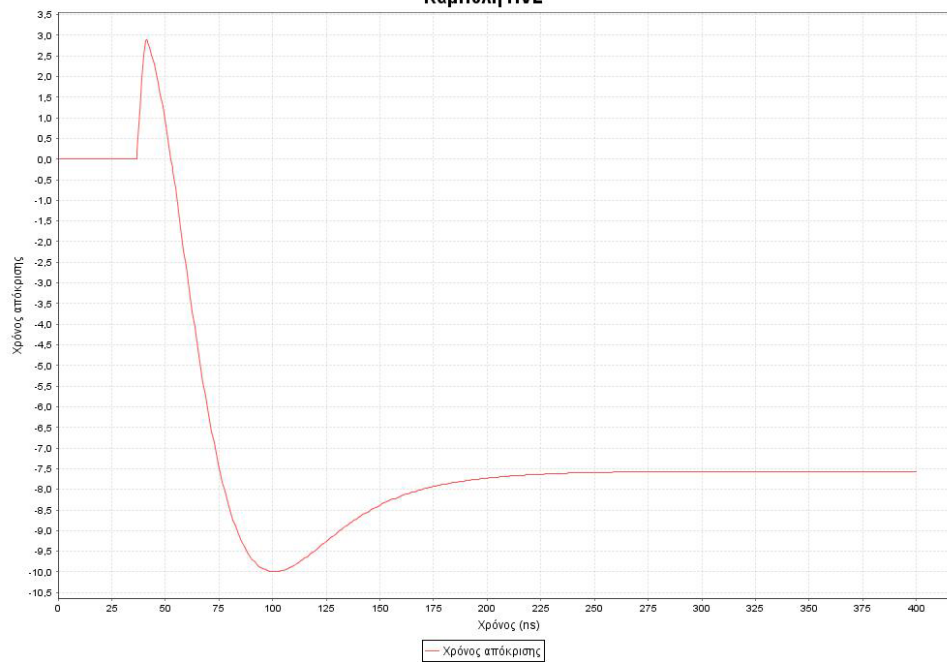
Παράμετρος	Τιμή
$O_1$	36.062
Υπερύψωση b	0.382
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.563
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.665
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	7.685

Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	3

Καμπύλη HVL



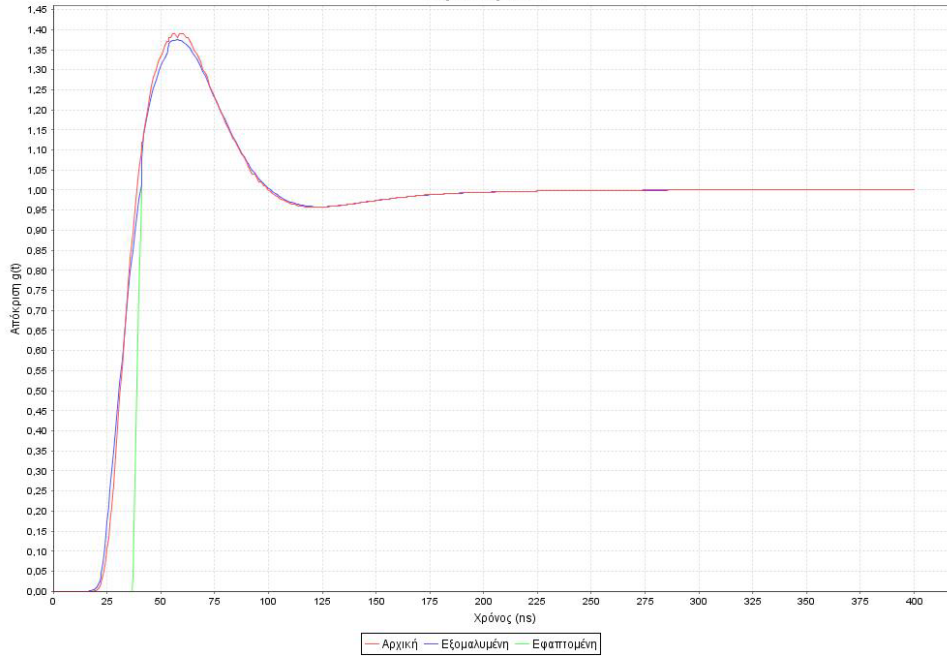
Καμπύλη HVL



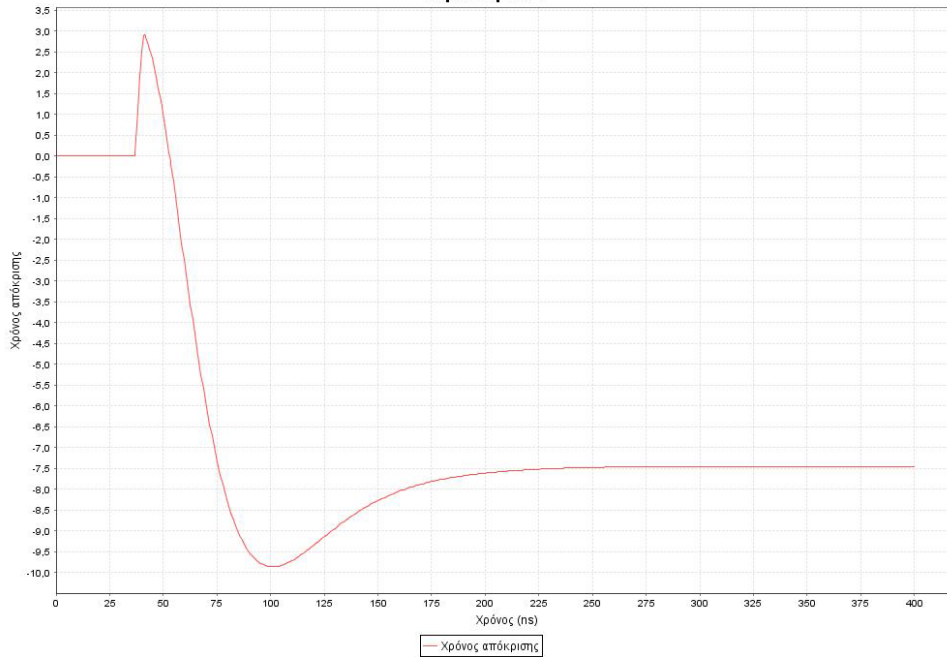
Παράμετρος	Τιμή
$O_1$	36.514
Υπερύψωση b	0.378
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.864
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.568
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	7.996

Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	4

Καμπύλη HVL



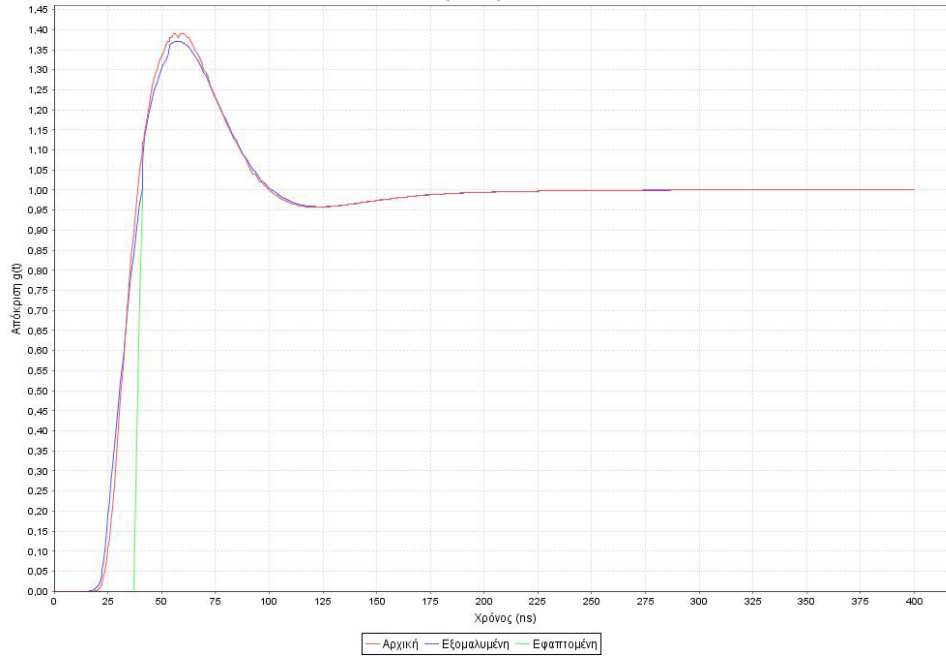
Καμπύλη HVL



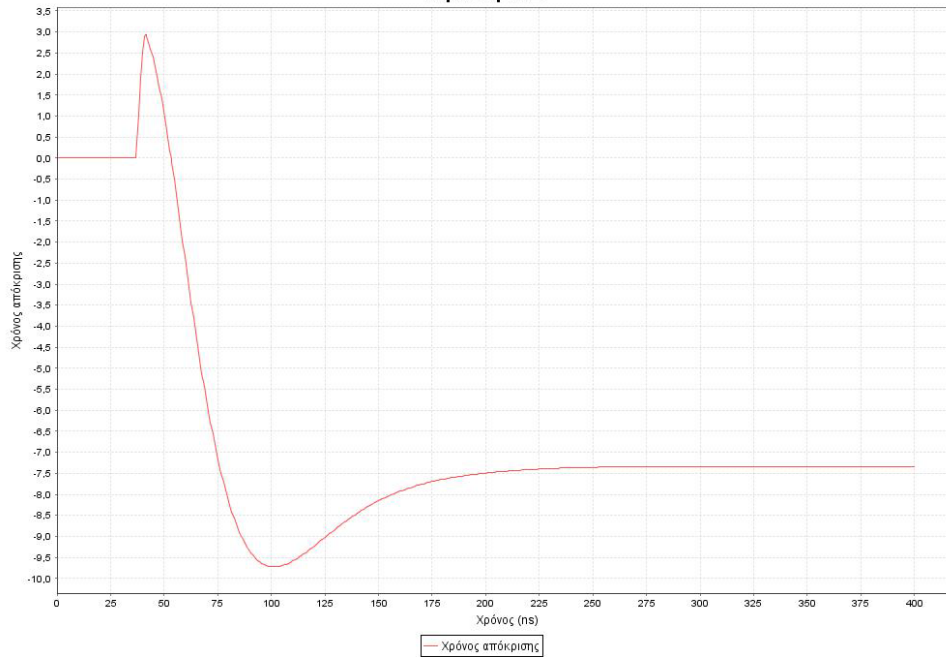
Παράμετρος	Τιμή
$O_1$	36.727
Υπερύψωση b	0.374
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.882
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.451
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.185

Αριθμός γειτονικών σημείων	6
Αριθμός επαναλήψεων	5

Καμπύλη HVL



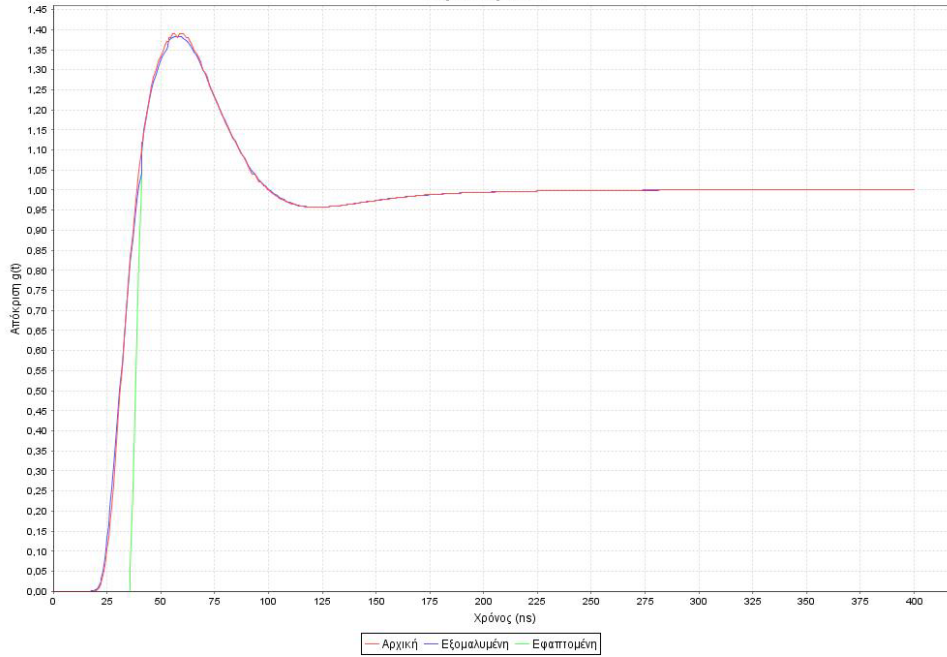
Καμπύλη HVL



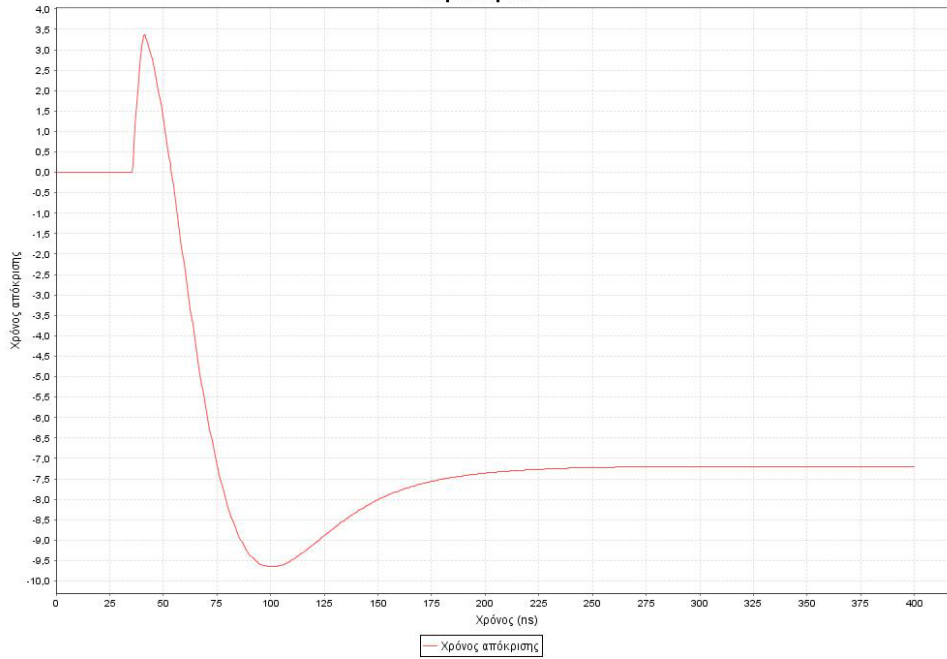
Παράμετρος	Τιμή
$O_1$	36.850
Υπερύψωση b	0.370
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.899
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.333
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.327

Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	1

Καμπύλη HVL



Καμπύλη HVL

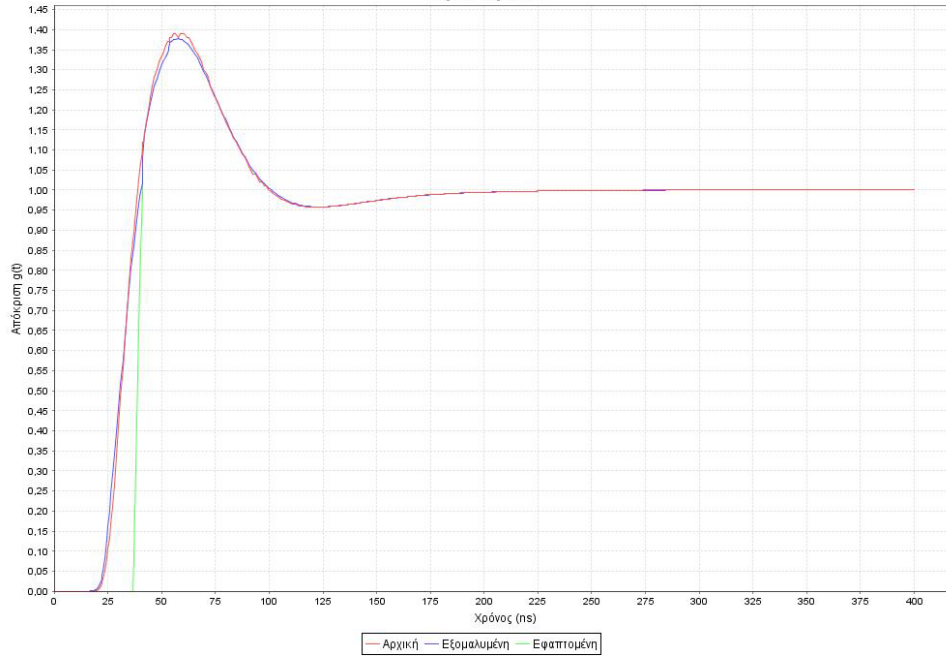


Παράμετρος	Τιμή
$O_1$	35.572
Υπερύψωση $b$	0.383
Μερικός χρόνος απόκρισης $T_a$ (ns)	3.085
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.202
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	7.407

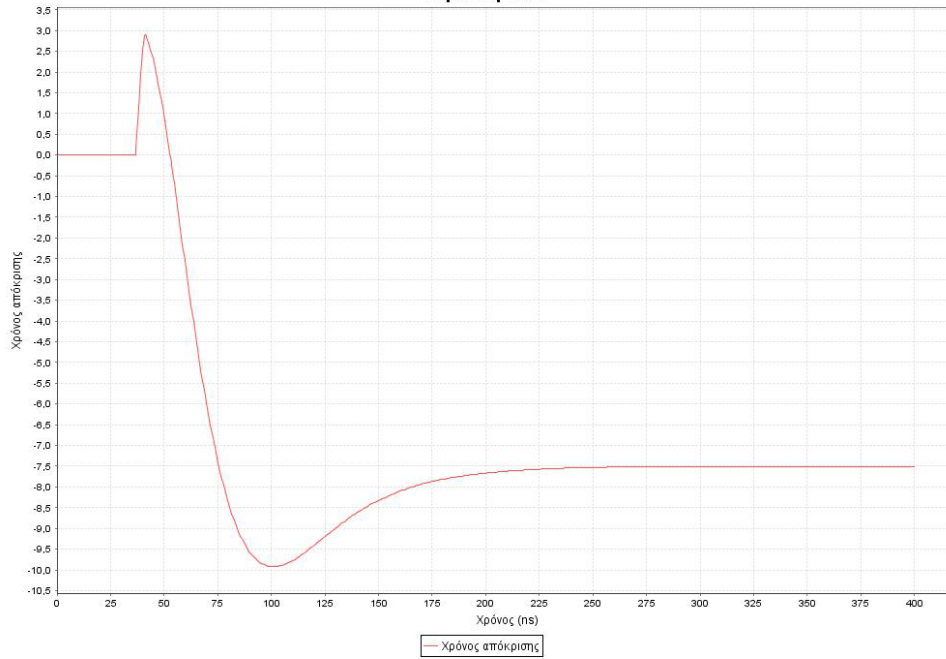


Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	2

Καμπύλη HVL



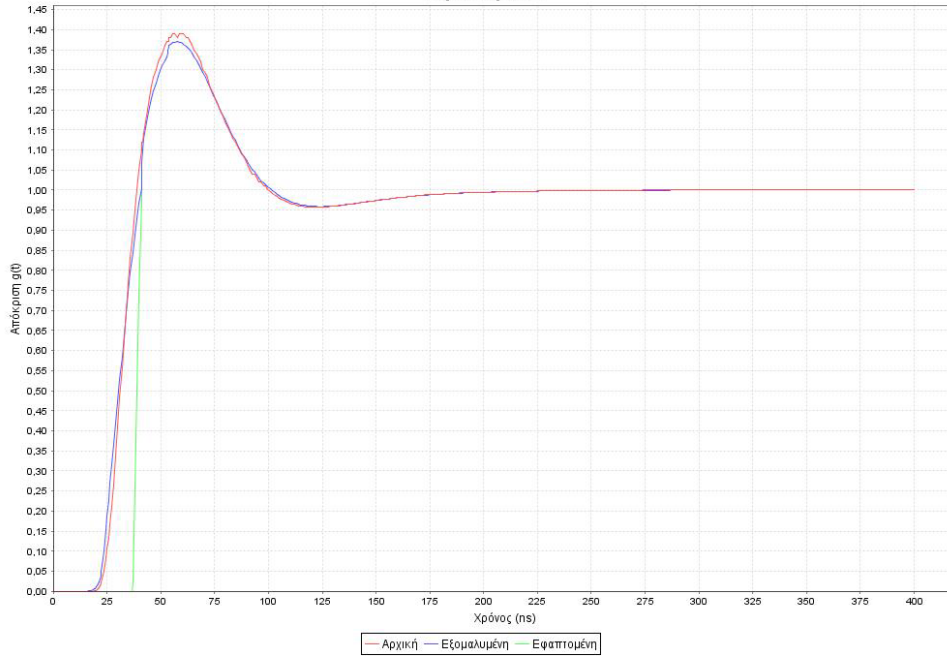
Καμπύλη HVL



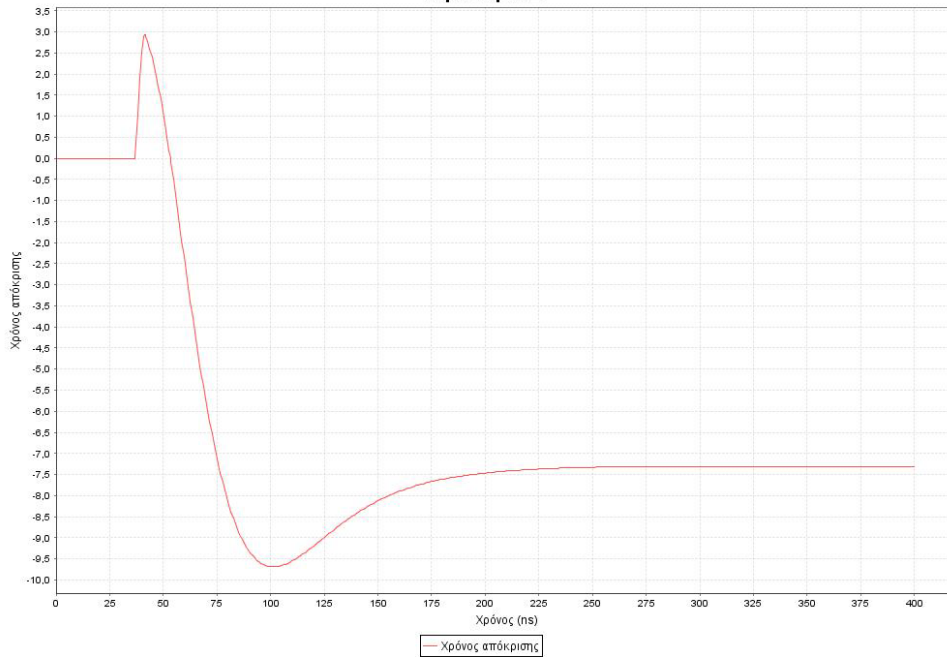
Παράμετρος	Τιμή
$O_1$	36.563
Υπερύψωση b	0.376
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.871
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.507
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.089

Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	3

Καμπύλη HVL

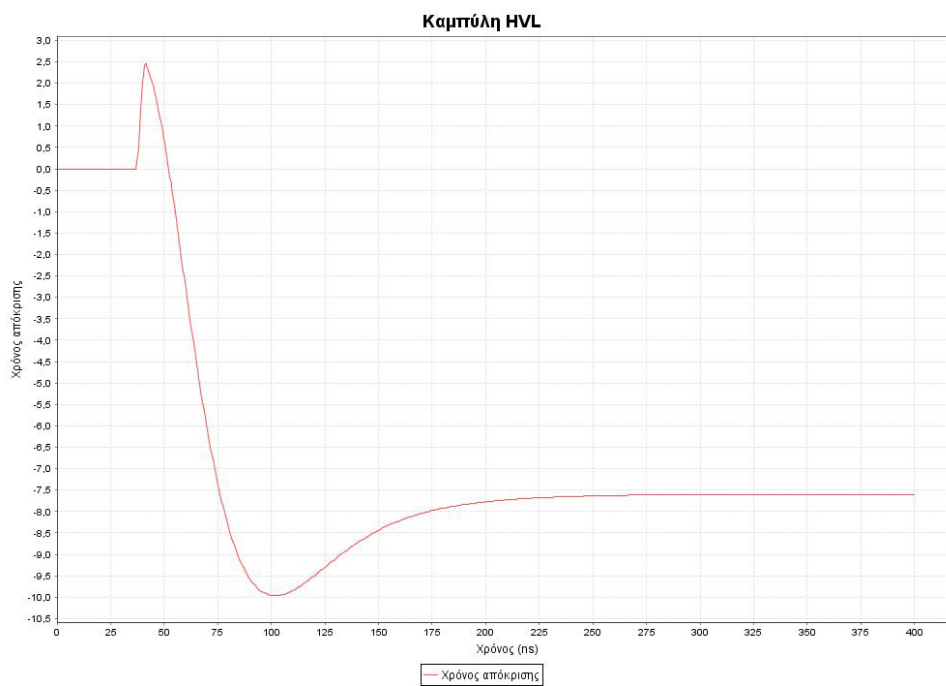
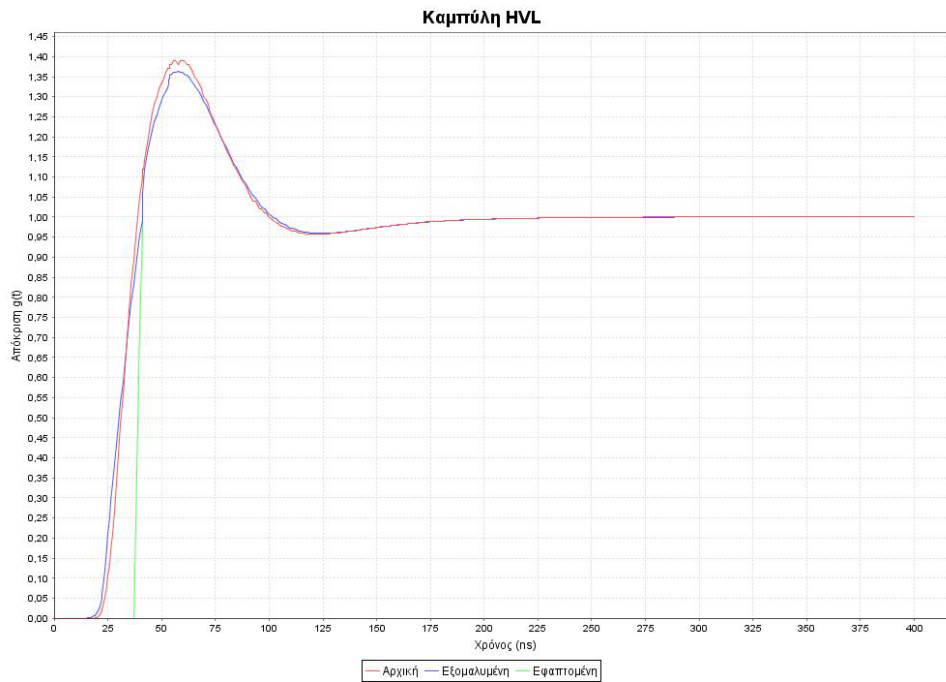


Καμπύλη HVL



Παράμετρος	Τιμή
$O_1$	36.816
Υπερύψωση $b$	0.369
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.898
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.304
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.359

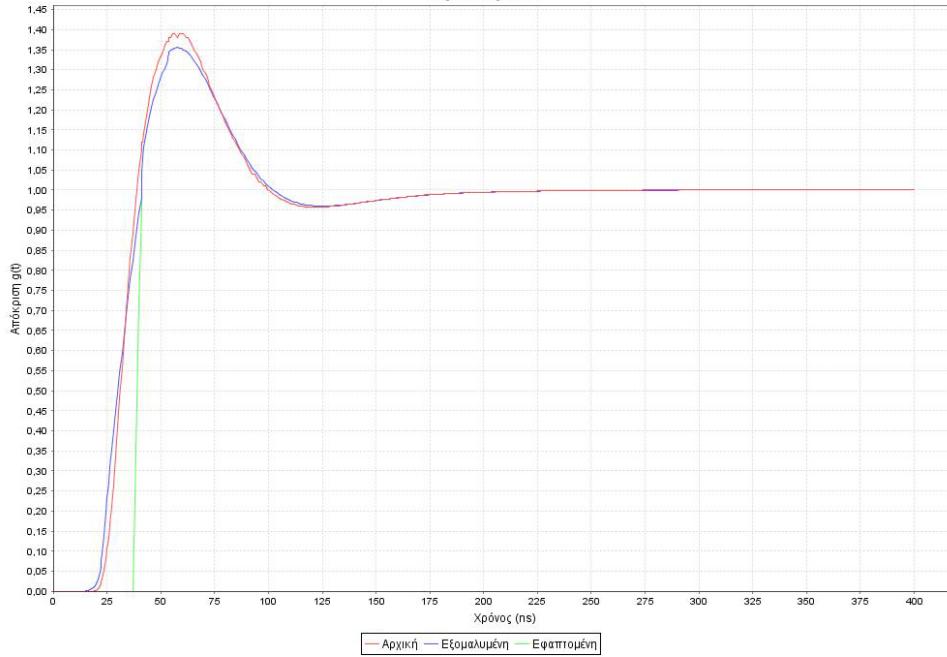
Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	4



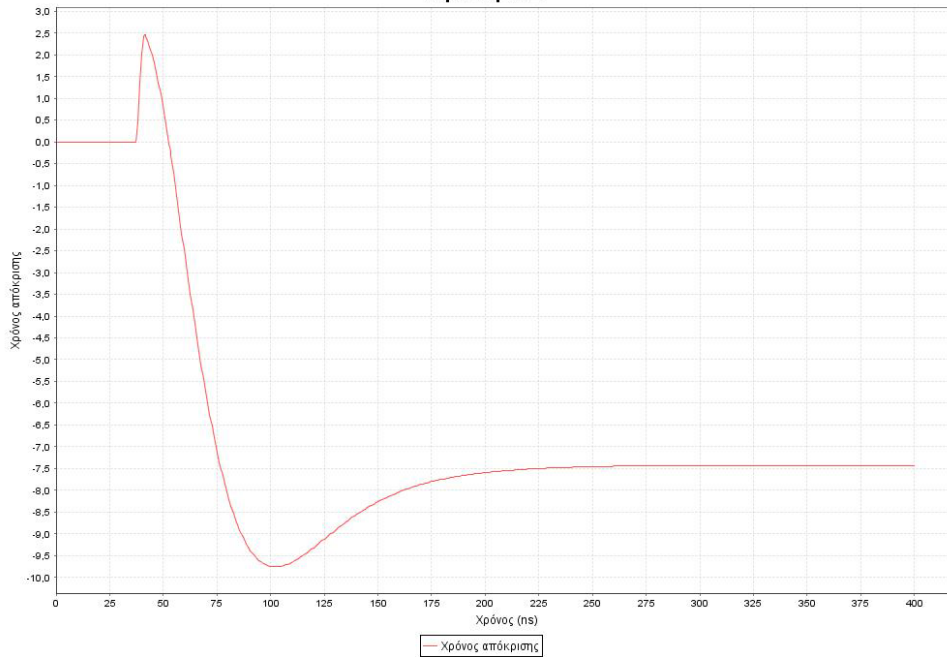
Παράμετρος	Τιμή
$O_1$	36.918
Υπερύψωση b	0.362
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.462
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.611
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.569

Αριθμός γειτονικών σημείων	8
Αριθμός επαναλήψεων	5

Καμπύλη HVL

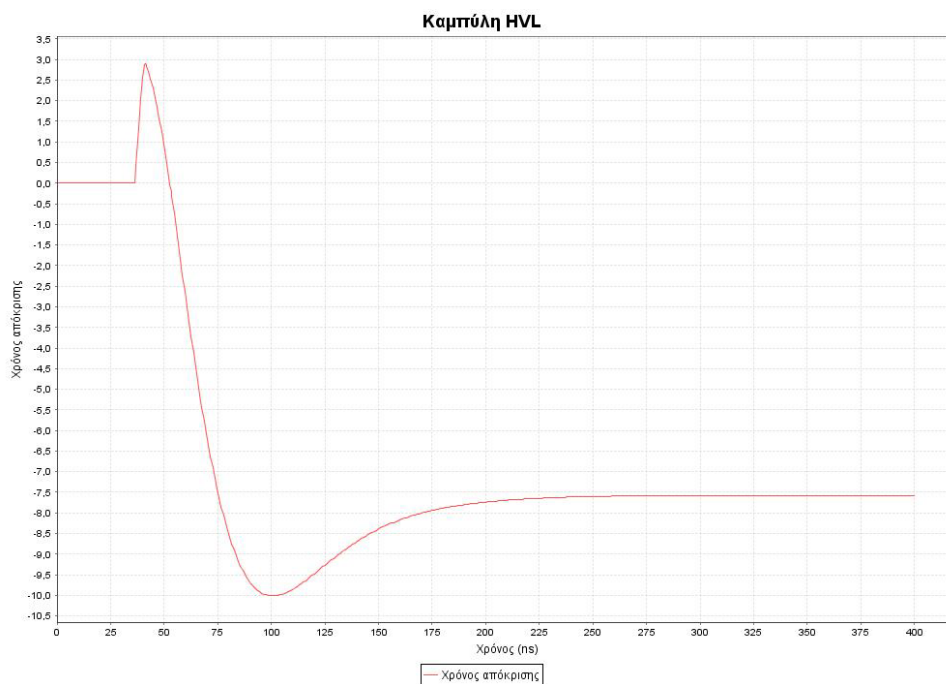
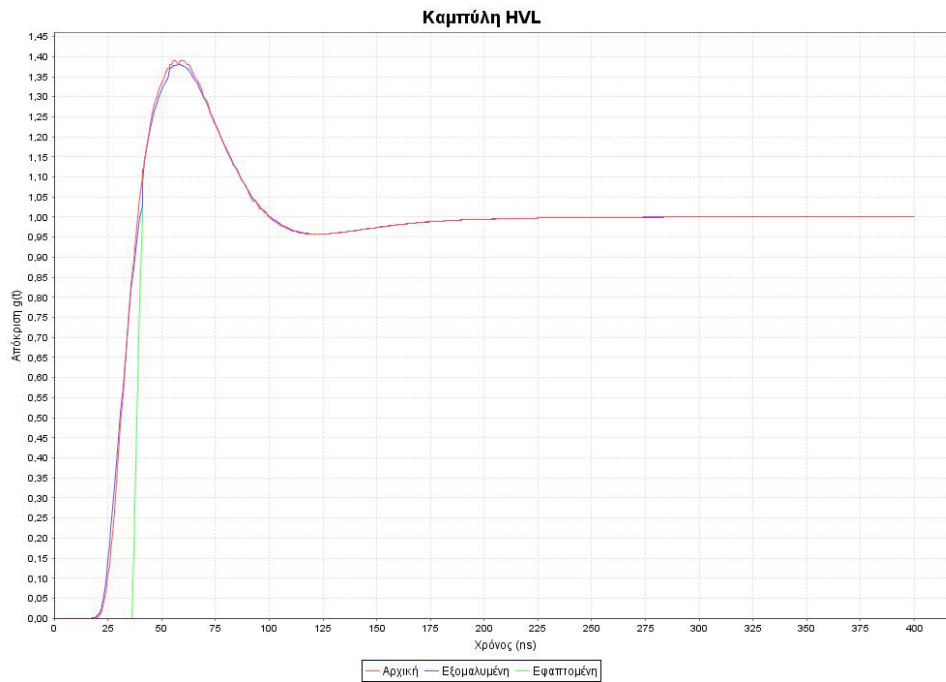


Καμπύλη HVL



Παράμετρος	Τιμή
$O_1$	36.959
Υπερύψωση $b$	0.354
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.471
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.433
Χρόνος ισορρόπησης $T_s$ (ns)	71.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.722

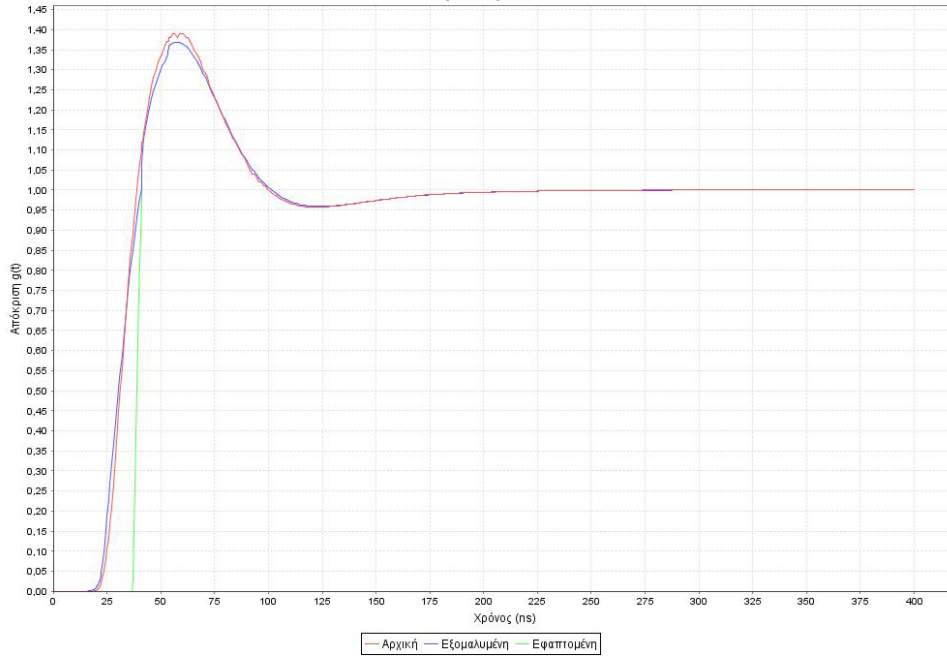
Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	1



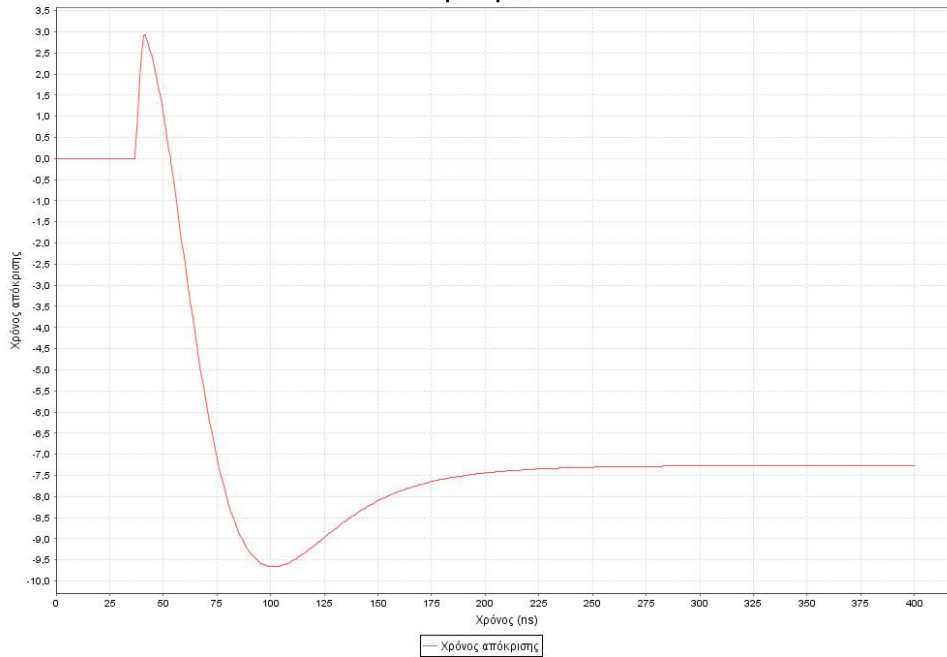
Παράμετρος	Τιμή
$O_1$	36.091
Υπερύψωση b	0.379
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.875
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.579
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	7.806

Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	2

Καμπύλη HVL

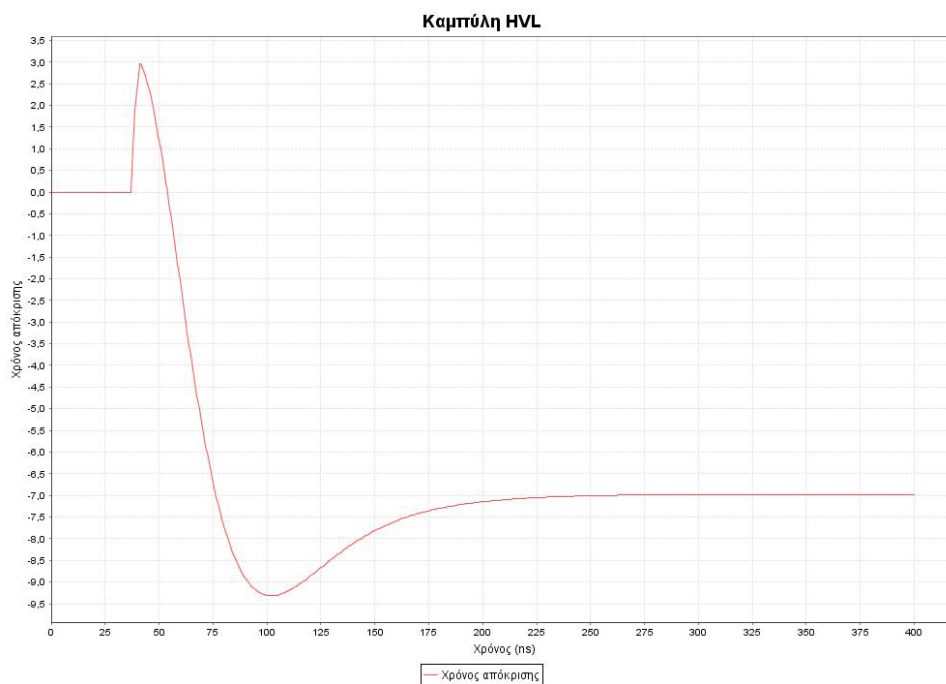
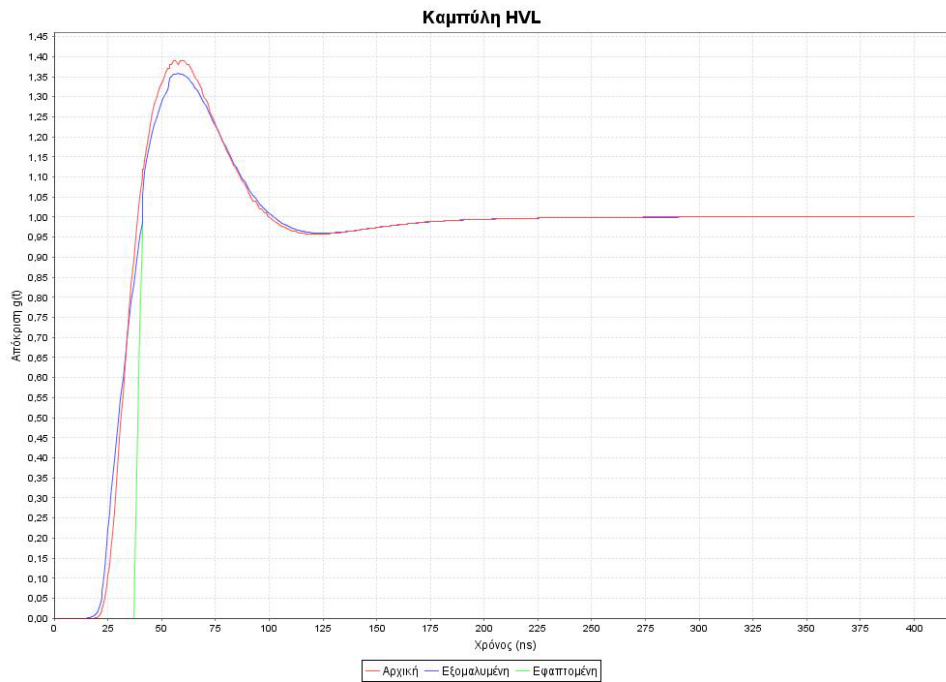


Καμπύλη HVL



Παράμετρος	Τιμή
$O_1$	36.769
Υπερύψωση b	0.368
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.895
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.283
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.377

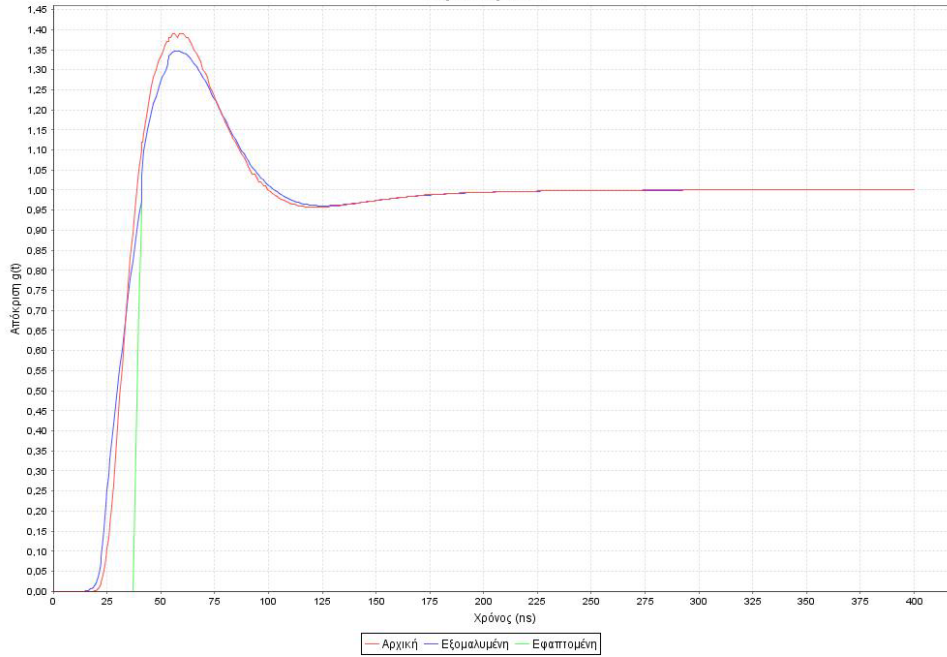
Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	3



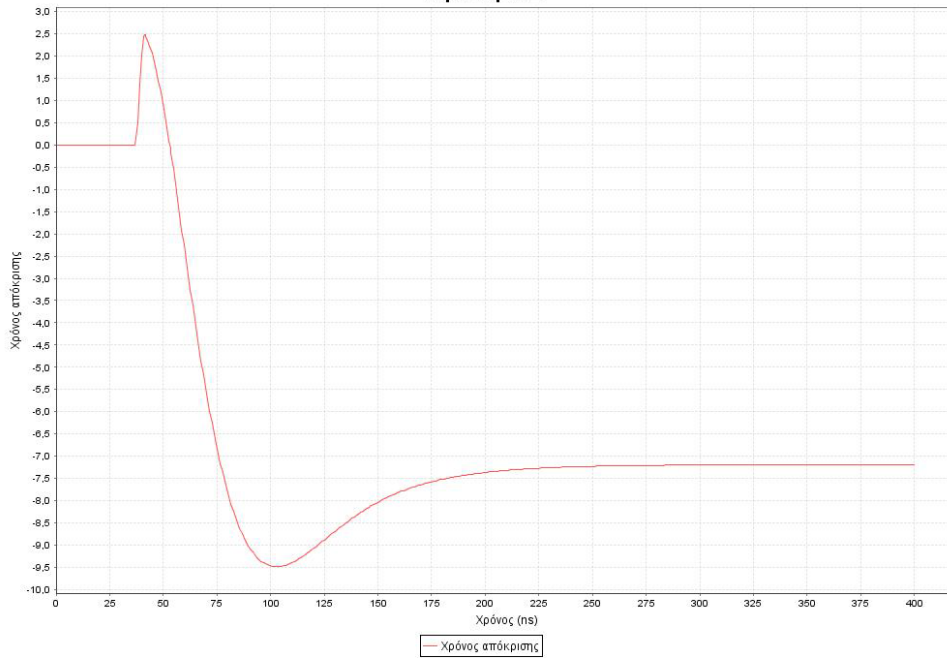
Παράμετρος	Τιμή
$O_1$	36.894
Υπερύψωση b	0.357
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.971
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-6.984
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.651

Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	4

Καμπύλη HVL



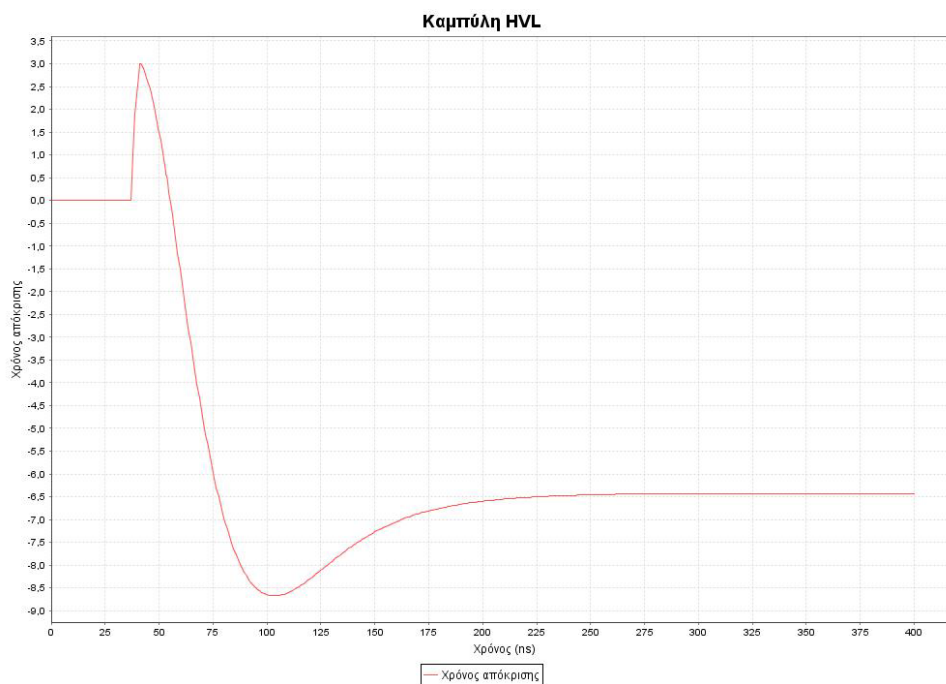
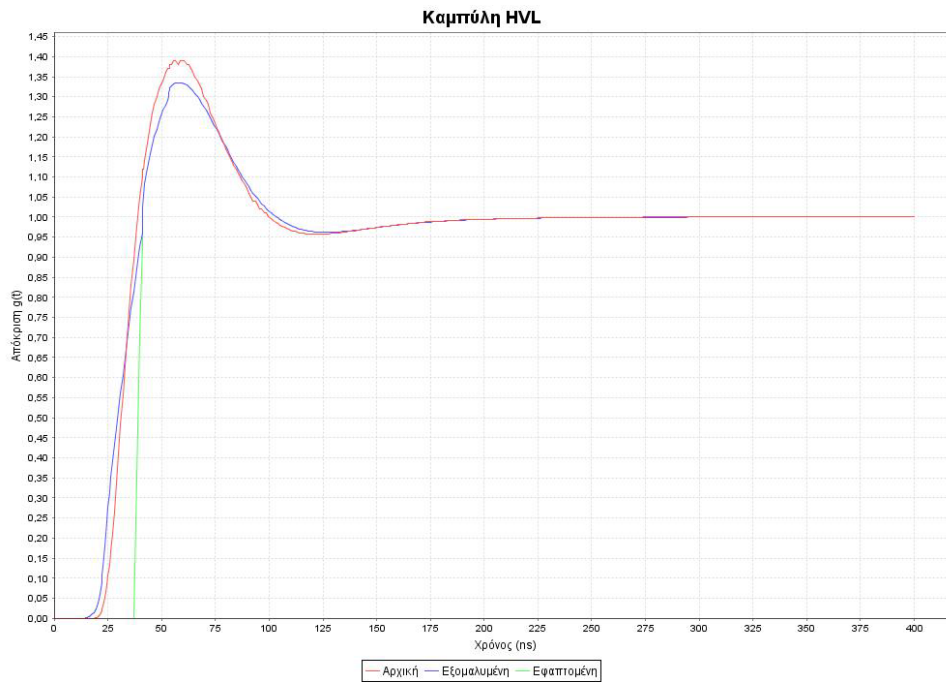
Καμπύλη HVL



Παράμετρος	Τιμή
$O_1$	36.911
Υπερύψωση $b$	0.346
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.488
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.205
Χρόνος ισορρόπησης $T_s$ (ns)	71.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.879



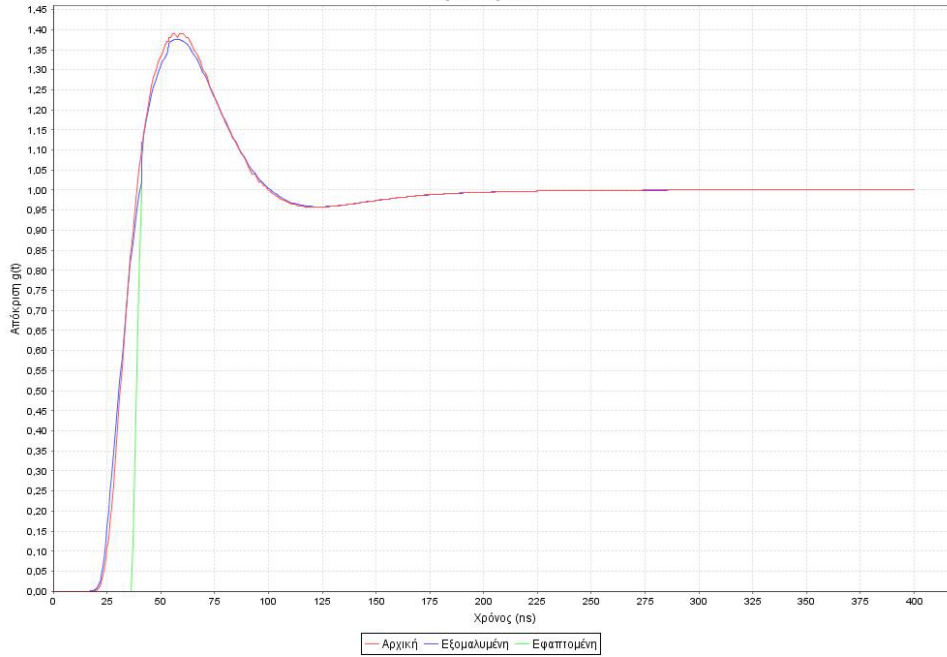
Αριθμός γειτονικών σημείων	10
Αριθμός επαναλήψεων	5



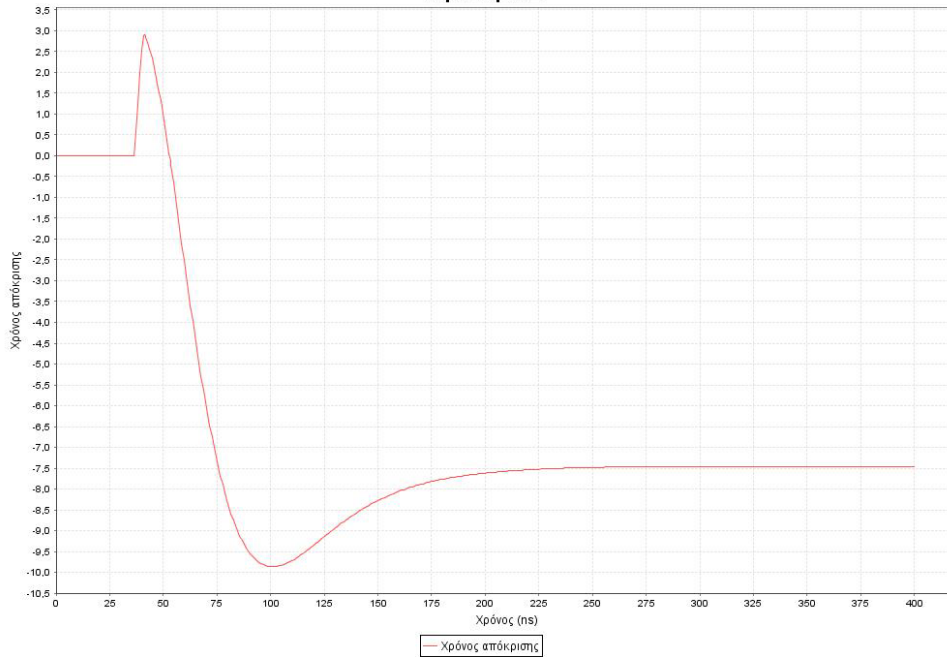
Παράμετρος	Τιμή
$O_1$	36.889
Υπερύψωση b	0.334
Μερικός χρόνος απόκρισης $T_a$ (ns)	3.008
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-6.432
Χρόνος ισορρόπησης $T_s$ (ns)	71.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	9.038

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	1

Καμπύλη HVL

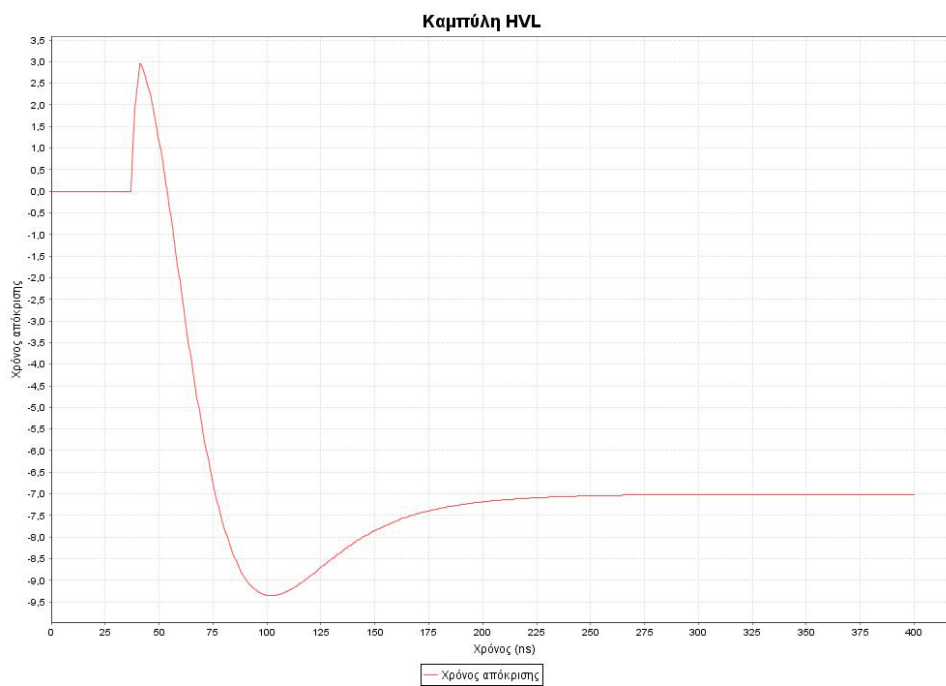
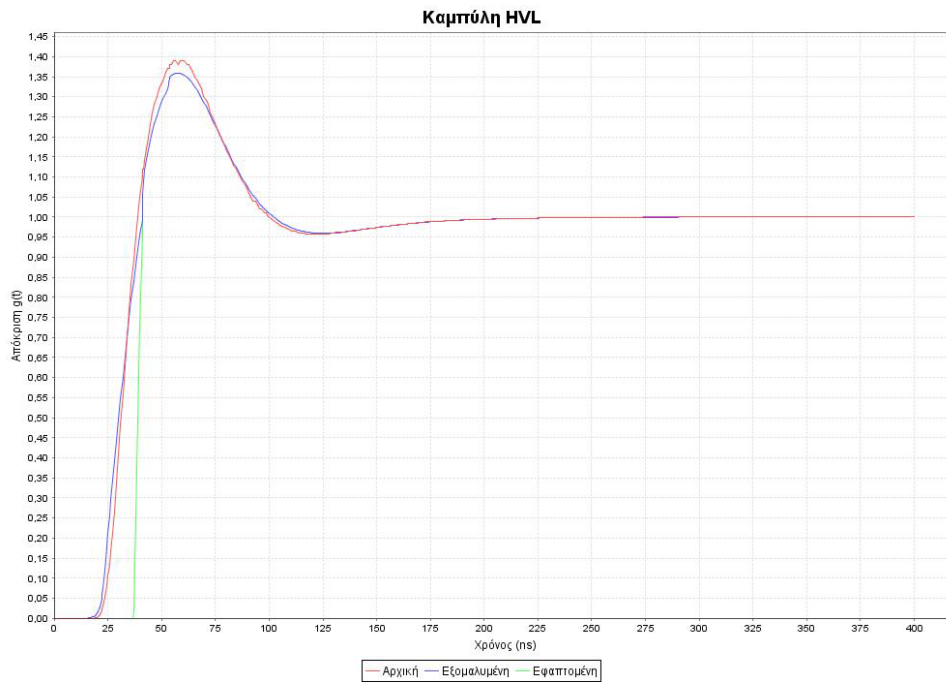


Καμπύλη HVL



Παράμετρος	Τιμή
$O_1$	36.367
Υπερύψωση b	0.375
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.874
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.455
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.069

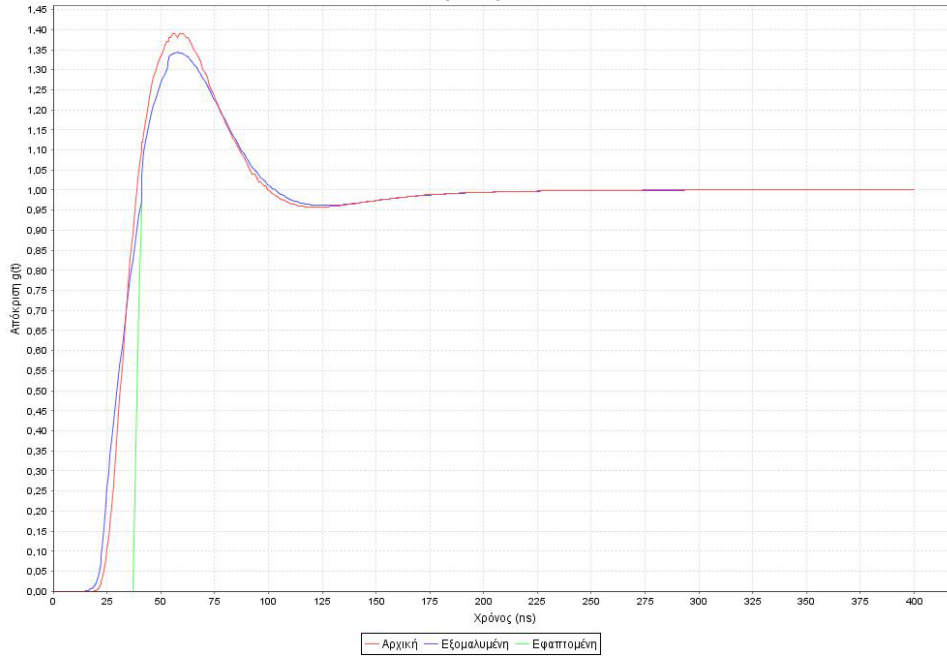
Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	2



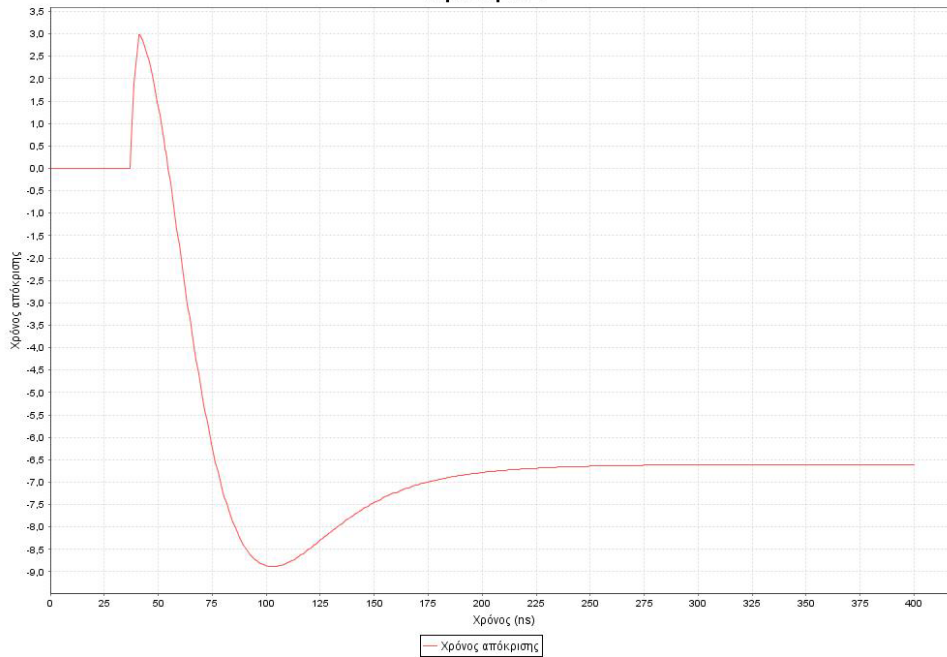
Παράμετρος	Τιμή
$O_1$	36.832
Υπερύψωση b	0.359
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.960
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-7.022
Χρόνος ισορρόπησης $T_s$ (ns)	70.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.620

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	3

Καμπύλη HVL

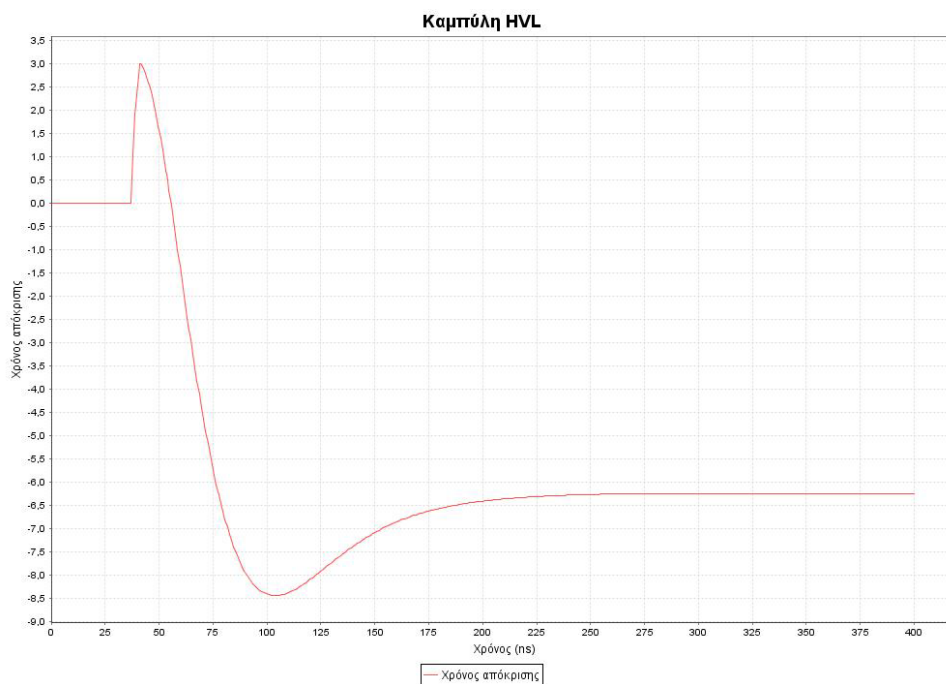
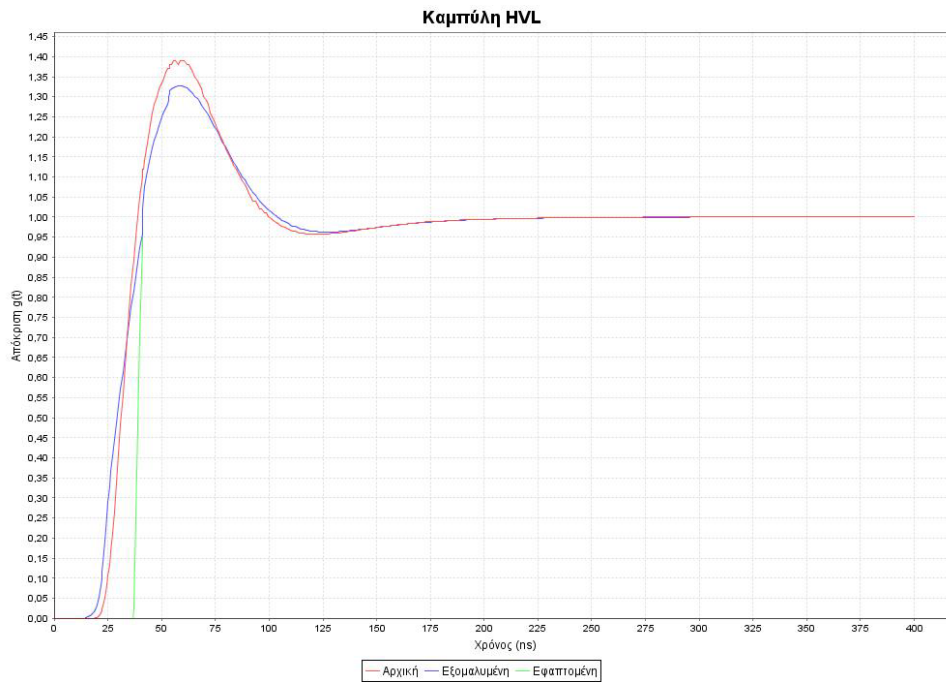


Καμπύλη HVL



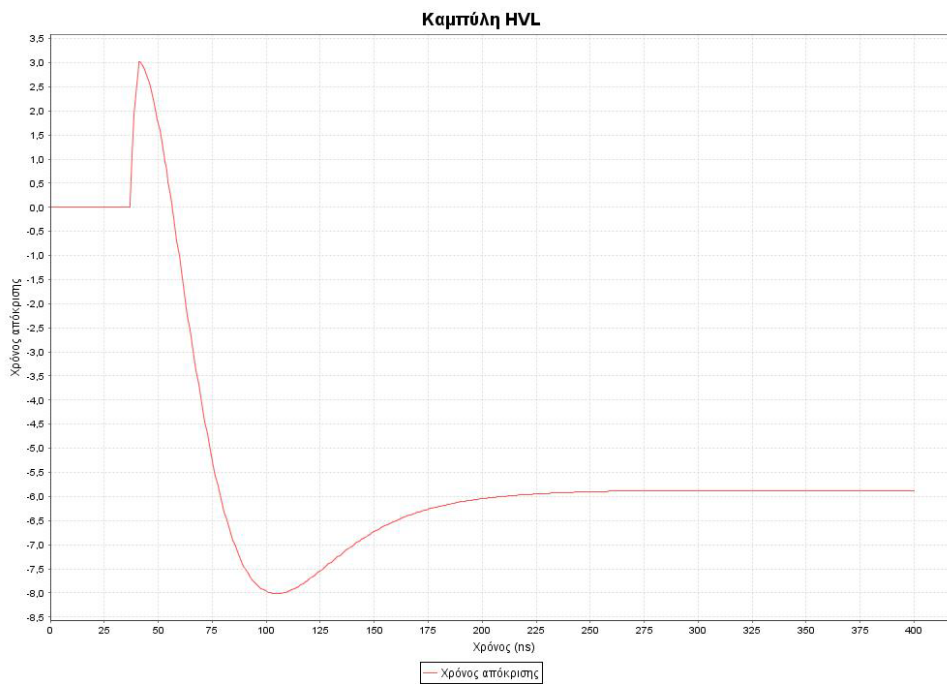
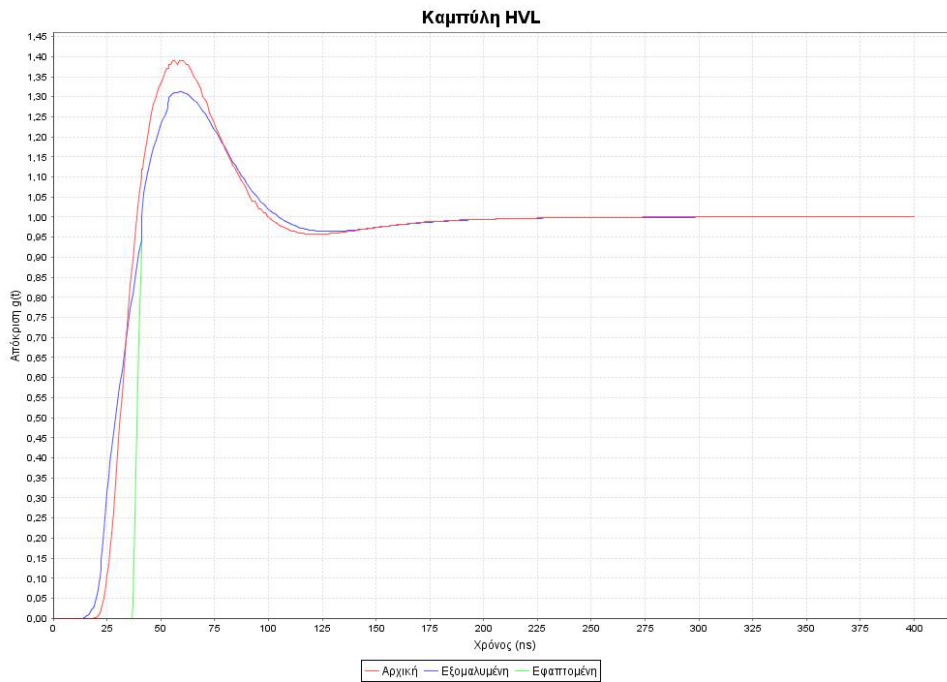
Παράμετρος	Τιμή
$O_1$	36.852
Υπερύψωση b	0.342
Μερικός χρόνος απόκρισης $T_a$ (ns)	2.987
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-6.620
Χρόνος ισορρόπησης $T_s$ (ns)	71.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	8.916

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	4



Παράμετρος	Τιμή
$O_1$	36.800
Υπερύψωση $b$	0.326
Μερικός χρόνος απόκρισης $T_a$ (ns)	3.008
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-6.240
Χρόνος ισορρόπησης $T_s$ (ns)	71.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	9.155

Αριθμός γειτονικών σημείων	12
Αριθμός επαναλήψεων	5



Παράμετρος	Τιμή
$O_1$	36.726
Υπερύψωση b	0.311
Μερικός χρόνος απόκρισης $T_a$ (ns)	3.027
Πειραματικός χρόνος απόκρισης $T_N$ (ns)	-5.876
Χρόνος ισορρόπησης $T_s$ (ns)	71.6
Αρχικός χρόνος παραμόρφωσης $T_o$ (ns)	9.367

## Παράρτημα Β

### *Ο πλήρης κώδικας της εφαρμογής*

#### Η κλάση HVLgui

```
package hvl;

import java.util.*;
import java.awt.*;
import java.awt.event.*;
import java.awt.image.*;
import javax.imageio.*;
import javax.swing.*;
import javax.swing.border.*;
import java.io.*;

public class HVLgui extends JFrame implements ActionListener {

    Color backgrndColor = new Color(202,202,255);
    Color buttonColor = new Color(157,157,255);
    Color mobColor = new Color(130,100,255);

    //choose file panel
    File inputFile;
    JTextField inputFileFld = new JTextField("");
    JButton browseBut = new JButton("...");
    JButton loadBut = new JButton(" Φόρτωση ");

    //smooth data panel
    JSpinner spinner1, spinner2;
    JButton computeBut = new JButton(" Εφαρμογή ");

    int lastComputedSmoothRange = 0, lastComputedTimes = 0;
    String lastInputFileName = "";

    //result labels
    JLabel o1Lbl1, bLbl1, TaLbl1, TNlbl1, TsLbl1, ToLbl1, o1Lbl2, bLbl2,
    TaLbl2, TNlbl2, TsLbl2, ToLbl2;

    //control but panel
    JButton clearChartsBut, saveChartsBut, massiveMeasBut;

    //chart panel
    HVLchart hvlChart, hvlChart2;

    //Info dialog
    InfoDialog idg = new InfoDialog(this);
    MassiveMeasurementsTask mmtask;
    javax.swing.Timer timer;

    //state
    int state = 0;

    int computationIndex = 0; //index ypologismou gia emfanisi
    apotelesmatwn stin katallili stili
```

```

//data
Vector xVector = new Vector();
Vector yVector = new Vector();

public HVLgui(){
    enableEvents(AWTEvent.WINDOW_EVENT_MASK);

    //graphics
    Border raisedBorder = BorderFactory.createRaisedBevelBorder();
    Border loweredBorder =
BorderFactory.createLoweredBevelBorder();
    Border emptyBorder = BorderFactory.createEmptyBorder(5,5,5,5);
    Border emptyBorderR = BorderFactory.createEmptyBorder(5,5,5,0);
    Border emptyBorderL = BorderFactory.createEmptyBorder(5,0,5,5);
    Border lineBorder = BorderFactory.createLineBorder(new
Color(50,0,255));
    Border etchedBorder =
BorderFactory.createEtchedBorder(EtchedBorder.RAISED);
    Border etchedBorderSimple =
BorderFactory.createEtchedBorder(EtchedBorder.LOWERED);

    //choose file panel
    browseBut.setBackground(buttonColor);

    browseBut.setBorder(BorderFactory.createCompoundBorder(raisedBorder,e
mptyBorder));
    browseBut.addActionListener(this);

    loadBut.setBackground(buttonColor);
    loadBut.setBorder(raisedBorder);
    loadBut.addActionListener(this);

    JPanel upbutPanel = new JPanel();
    upbutPanel.setOpaque(false);
    upbutPanel.setLayout(new BorderLayout(3,2));
    upbutPanel.add(browseBut, BorderLayout.CENTER);
    upbutPanel.add(loadBut, BorderLayout.EAST);

    JPanel inputFilePanel = new JPanel();
    inputFilePanel.setOpaque(false);

    inputFilePanel.setBorder(BorderFactory.createCompoundBorder(emptyBord
er,BorderFactory.createTitledBorder(lineBorder,"Αρχείο δεδομένων"));
    inputFilePanel.setLayout(new BorderLayout(3,2));
    inputFilePanel.add(new JLabel("Διαδρομή στον δίσκο :
",JLabel.RIGHT), BorderLayout.WEST);
    inputFilePanel.add(inputFileFld, BorderLayout.CENTER);
    inputFilePanel.add(upbutPanel, BorderLayout.EAST);

    //smooth data panel
    spinner1 = new JSpinner(new SpinnerNumberModel(3, 0, 6, 1));
    spinner2 = new JSpinner(new SpinnerNumberModel(1, 0, 5, 1));

    computeBut.setBackground(buttonColor);

    computeBut.setBorder(BorderFactory.createCompoundBorder(raisedBorder,
emptyBorder));
    computeBut.addActionListener(this);

    JPanel smoothDataPanel = new JPanel();
    smoothDataPanel.setOpaque(false);

```



```

smoothDataPanel.setBorder(BorderFactory.createCompoundBorder(emptyBorder,
BorderFactory.createTitledBorder(lineBorder, "Εξομάλυνση δεδομένων")));
smoothDataPanel.setLayout(new FlowLayout(FlowLayout.LEFT, 5, 2));
smoothDataPanel.add(new JLabel("Αριθμός σημείων :
", JLabel.RIGHT));
smoothDataPanel.add(spinner1);
smoothDataPanel.add(new JLabel("Αριθμός εκτελέσεων :
", JLabel.RIGHT));
smoothDataPanel.add(spinner2);
smoothDataPanel.add(computeBut);

//north panel
JPanel northPanel = new JPanel();
northPanel.setOpaque(false);
northPanel.setLayout(new GridLayout(2, 1, 2, 4));
northPanel.add(inputFilePanel);
northPanel.add(smoothDataPanel);

//result labels panel
o1Lb1 = new JLabel(" ", JLabel.CENTER);

o1Lb1.setBorder(BorderFactory.createCompoundBorder(etchedBorder,
emptyBorder));

bLb1 = new JLabel(" ", JLabel.CENTER);
bLb1.setBorder(BorderFactory.createCompoundBorder(etchedBorder,
emptyBorder));

TaLb1 = new JLabel(" ", JLabel.CENTER);
TaLb1.setBorder(etchedBorder);

TNl1b1 = new JLabel(" ", JLabel.CENTER);
TNl1b1.setBorder(etchedBorder);

TsLb1 = new JLabel(" ", JLabel.CENTER);
TsLb1.setBorder(etchedBorder);

ToLb1 = new JLabel(" ", JLabel.CENTER);
ToLb1.setBorder(etchedBorder);

o1Lb12 = new JLabel(" ", JLabel.CENTER);

o1Lb12.setBorder(BorderFactory.createCompoundBorder(etchedBorder,
emptyBorder));

bLb12 = new JLabel(" ", JLabel.CENTER);

bLb12.setBorder(BorderFactory.createCompoundBorder(etchedBorder,
emptyBorder));

TaLb12 = new JLabel(" ", JLabel.CENTER);
TaLb12.setBorder(etchedBorder);

TNl1b12 = new JLabel(" ", JLabel.CENTER);
TNl1b12.setBorder(etchedBorder);

TsLb12 = new JLabel(" ", JLabel.CENTER);
TsLb12.setBorder(etchedBorder);

ToLb12 = new JLabel(" ", JLabel.CENTER);

```

```

ToLbl2.setBorder(etchedBorder);

JLabel header1Lbl = new JLabel(" ΠΑΡΑΜΕΤΡΟΣ ", JLabel.CENTER);
header1Lbl.setBorder(etchedBorderSimple);
header1Lbl.setOpaque(true);
header1Lbl.setBackground(mobColor);

JPanel paramNamesPanel = new JPanel();
paramNamesPanel.setOpaque(false);
paramNamesPanel.setLayout(new GridLayout(7,1,2,3));
paramNamesPanel.add(header1Lbl);
paramNamesPanel.add(new JLabel("O1 : ", JLabel.RIGHT));
paramNamesPanel.add(new JLabel("Υπερύψωση b :
", JLabel.RIGHT));
paramNamesPanel.add(new JLabel("<html><body><p align=right>
Μερικός χρόνος απόκρισης Ta : </p></body></html>", JLabel.RIGHT));
paramNamesPanel.add(new JLabel("<html><body><p align=right>
Πειραματικός χρόνος απόκρισης TN : </p></body></html>", JLabel.RIGHT));
paramNamesPanel.add(new JLabel("<html><body><p align=right>
Χρόνος ισορρόπησης Ts : </p></body></html>", JLabel.RIGHT));
paramNamesPanel.add(new JLabel("<html><body><p align=right>
Αρχικός χρόνος παραμόρφωσης To : </p></body></html>", JLabel.RIGHT));

JLabel header2Lbl = new JLabel(" ΥΠΟΛΟΓΙΣΜΟΣ 1
", JLabel.CENTER);
header2Lbl.setBorder(etchedBorderSimple);
header2Lbl.setOpaque(true);
header2Lbl.setBackground(mobColor);

JLabel header3Lbl = new JLabel(" ΥΠΟΛΟΓΙΣΜΟΣ 2
", JLabel.CENTER);
header3Lbl.setBorder(etchedBorderSimple);
header3Lbl.setOpaque(true);
header3Lbl.setBackground(mobColor);

JPanel resultLabelsPanel = new JPanel();
resultLabelsPanel.setOpaque(false);
resultLabelsPanel.setLayout(new GridLayout(7,2,3,3));
resultLabelsPanel.add(header2Lbl);
resultLabelsPanel.add(header3Lbl);
resultLabelsPanel.add(o1Lbl);
resultLabelsPanel.add(o1Lbl2);
resultLabelsPanel.add(bLbl);
resultLabelsPanel.add(bLbl2);
resultLabelsPanel.add(TaLbl);
resultLabelsPanel.add(TaLbl2);
resultLabelsPanel.add(TN1bl);
resultLabelsPanel.add(TN1bl2);
resultLabelsPanel.add(TsLbl);
resultLabelsPanel.add(TsLbl2);
resultLabelsPanel.add(ToLbl);
resultLabelsPanel.add(ToLbl2);

JPanel resultPanel = new JPanel();
resultPanel.setOpaque(false);

resultPanel.setBorder(BorderFactory.createCompoundBorder(emptyBorder,
BorderFactory.createTitledBorder(lineBorder, "Αποτελέσματα")));
resultPanel.setLayout(new GridLayout(1,2,3,2));
resultPanel.add(paramNamesPanel);

```

```

resultPanel.add(resultLabelsPanel);

//center panel
JPanel centerPanel = new JPanel();
centerPanel.setOpaque(false);
centerPanel.setLayout(new BorderLayout(2,20));
centerPanel.add(resultPanel, BorderLayout.NORTH);
centerPanel.add(new JLabel(" ΨΗΛΕΣ ΤΑΞΕΙΣ ", JLabel.CENTER),
BorderLayout.CENTER);

//control buttons panel
clearChartsBut = new JButton("<html><body><p align=center>
ΚΑΘΑΡΙΣΜΟΣ <br> ΥΠΟΛΟΓΙΣΜΩΝ </p></body></html>");
clearChartsBut.setBackground(buttonColor);
clearChartsBut.setBorder(raisedBorder);
clearChartsBut.addActionListener(this);

saveChartsBut = new JButton("<html><body><p align=center>
ΑΠΟΘΗΚΕΥΣΗ <br> ΑΠΟΤΕΛΕΣΜΑΤΩΝ </p></body></html>");
saveChartsBut.setBackground(buttonColor);
saveChartsBut.setBorder(raisedBorder);
saveChartsBut.addActionListener(this);

massiveMeasBut = new JButton("MyTHESIS");
massiveMeasBut.setBackground(buttonColor);
massiveMeasBut.setBorder(raisedBorder);
massiveMeasBut.addActionListener(this);

JPanel controlPanel = new JPanel();
controlPanel.setOpaque(false);

controlPanel.setBorder(BorderFactory.createCompoundBorder(BorderFacto
ry.createTitledBorder(lineBorder,"Ελεγχος"), emptyBorder));
controlPanel.setLayout(new GridLayout(1,3,10,2));
controlPanel.add(clearChartsBut);
controlPanel.add(saveChartsBut);
controlPanel.add(massiveMeasBut);

//south panel
JPanel southPanel = new JPanel();
southPanel.setOpaque(false);
southPanel.setBorder(emptyBorder);
southPanel.setLayout(new BorderLayout(2,5));
southPanel.add(controlPanel, BorderLayout.NORTH);
southPanel.add(new FooterPanel(this), BorderLayout.SOUTH);

//main panel
JPanel mainPanel = new JPanel();
mainPanel.setBackground(backgrndColor);

mainPanel.setBorder(BorderFactory.createCompoundBorder(emptyBorderR,e
tchedBorder));
mainPanel.setLayout(new BorderLayout());
mainPanel.add(northPanel, BorderLayout.NORTH);
mainPanel.add(centerPanel, BorderLayout.CENTER);
mainPanel.add(southPanel, BorderLayout.SOUTH);

//chart panel
hvlChart = new HVLchart("Καμπύλη HVL", "Χρόνος (ns)", "Απόκριση
g(t)");

```

```

        JPanel chartPanel = new JPanel();
        chartPanel.setBackground(backgrndColor);

        chartPanel.setBorder(BorderFactory.createCompoundBorder(emptyBorderL,
etchedBorder));
        chartPanel.setLayout(new BorderLayout());
        chartPanel.add(hvlChart, BorderLayout.CENTER);

        //chart panel 2
        hvlChart2 = new HVLchart("Καμπύλη HVL", "Χρόνος (ns)", "Χρόνος
απόκρισης");

        JPanel chartPanel2 = new JPanel();
        chartPanel2.setBackground(backgrndColor);

        chartPanel2.setBorder(BorderFactory.createCompoundBorder(emptyBorderL
,etchedBorder));
        chartPanel2.setLayout(new BorderLayout());
        chartPanel2.add(hvlChart2, BorderLayout.CENTER);

        //chart panels
        JPanel chartPanels = new JPanel();
        chartPanels.setOpaque(false);

        chartPanels.setBorder(BorderFactory.createCompoundBorder(emptyBorder,
etchedBorder));
        chartPanels.setLayout(new GridLayout(2,1));
        chartPanels.add(hvlChart);
        chartPanels.add(hvlChart2);

        //HVLgui
        getContentPane().setBackground(backgrndColor);
        getContentPane().setLayout(new GridLayout(1,2));
        getContentPane().add(mainPanel);
        getContentPane().add(chartPanels);

        setTitle("Ανάλυση παραμέτρων απόκρισης συστημάτων μέτρησης YF
v1.0");

        Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
        setSize(screenSize.width, screenSize.height-30);
        setExtendedState(Frame.MAXIMIZED_BOTH);

        setLocation((int) ((screenSize.width/2) - (getWidth()/2)),
(int) ((screenSize.height/2) - (getHeight()/2)));
        show();
    }

    public void actionPerformed(ActionEvent e){
        Object src = e.getSource();

        if(src == browseBut){
            chooseFileAction();
        }
        else if(src == loadBut){
            loadFileAction();
        }
        else if(src == computeBut){
            int smoothRange = ((Integer)
spinner1.getValue()).intValue();

```

```

        int times = ((Integer) spinner2.getValue()).intValue();

        if((smoothRange != lastComputedSmoothRange)|| (times !=
lastComputedTimes)) state = 1;

        //if smoothRange or smoothing times have been changed,
proceed
        if((yVector.size()>0)&&(state == 1)){
            computeAction(smoothRange, times);
        }
        else {
            if(state == 0) System.out.println("No data loaded!!
No processing can be done!!");
            else if(state == 2) System.out.println("Data
processing already done!!");
        }
    }
    else if(src == clearChartsBut){
        clearChartsAction();
    }
    else if(src == saveChartsBut){
        saveChartsAction();
    }
    else if(src == massiveMeasBut){
        if(inputFile != null){
            mmtask = new MassiveMeasurementsTask(this);

            timer = new javax.swing.Timer(300, new
ActionListener() {
                public void actionPerformed(ActionEvent evt)
                {
                    try{

                        idg.progressBar.setValue(mmtask.getCompletionPercent());

                        idg.statusLbl.setText(mmtask.getMessage());

                                if (mmtask.isDone()) {
                                    timer.stop();
                                    idg.setVisible(false);
                                }
                                }
                                catch(Exception qq){
                                    System.out.println("Check timer action
in gui! "+qq.getMessage());
                                }
                            }
                        });

                        idg.show();
                        timer.start();
                        mmtask.go();
                    }
                    else {
                        System.out.println("MMB: File not
specified!");
                    }
                }
            }

            public void computeAction(int smoothRange, int times){

```

```

try{
    System.out.println("\n **** START PROCESSING ****\n");

    lastInputFileName = inputFile.getAbsolutePath();
    lastComputedSmoothRange = smoothRange;
    lastComputedTimes = times;

    computationIndex++; //1 tin prwti fora, 2 tin deyteri kt1

    Vector yVectorSm = (Vector) yVector.clone();
    Vector xVectorSm = (Vector) xVector.clone(); //epeidi
kanoume ena insert parakatw, as exoume kai to arxiko

    try{
        double val = 0;

        //SMOOTHING
        if((smoothRange > 0)&&(times > 0)){
            System.out.print("Smoothing...");

            for(int t=0;t<times;t++){
                for(int
i=smoothRange;i<yVectorSm.size()-smoothRange;i++){
                    val = 0;
                    for(int j=-
smoothRange;j<=smoothRange;j++){
                        val += ((Double)
yVectorSm.get(i+j)).doubleValue();
                    }
                    yVectorSm.set(i, new
Double(val/(2*smoothRange+1)));
                }
            }

            hvlChart.addSeries("Εξομαλυσμένη", xVectorSm,
yVectorSm);

            System.out.println("OK\n");
        }
        else {
            System.out.println("Smoothing not needed!");
        }

        //FIND MAX SLOPE, maxG(t) and t1
        double maxGt = 0, t1 = 0;
        int t1index = 0;
        double x_prev, x, x_next, y_prev, y, y_next;
        double maxSlope = 0, slope = 0, x_maxSlope = 0,
y_maxSlope = 0;

        int maxSlopeIndex = 0;

        //init
        x_prev = ((Double) xVectorSm.get(0)).doubleValue();
        x      = ((Double) xVectorSm.get(1)).doubleValue();

        y_prev = ((Double) yVectorSm.get(0)).doubleValue();
        y      = ((Double) yVectorSm.get(1)).doubleValue();

        for(int i=2;i<yVectorSm.size();i++){

            if(y_prev > maxGt) maxGt = y_prev;

```

```

                if((t1 == 0)&&(y_prev >= 1)) {
                    t1 = ((Double) xVectorSm.get(i-
2)).doubleValue();
                    t1index = i-2;
                    System.out.println("t1 point = "+t1+"
at index "+t1index+"\n");
                }

                x_next = ((Double)
xVectorSm.get(i)).doubleValue();
                y_next = ((Double)
yVectorSm.get(i)).doubleValue();

                if(((x_next - x)>0)&&((x - x_prev)>0)){ //mi
bgalei apeiro
                    slope = (((y_next - y)/(x_next -
x))+((y - y_prev)/(x - x_prev)))/2;

                    if(Math.abs(slope)>maxSlope){
                        maxSlope = slope;
                        maxSlopeIndex = i-1;
                        x_maxSlope = x;
                        y_maxSlope = y;
                    }
                }

                x_prev = x;
                x      = x_next;

                y_prev = y;
                y      = y_next;
            }

            System.out.println("Max slope = "+maxSlope+" at
point "+maxSlopeIndex);
            System.out.println("x = "+x_maxSlope+" y =
"+y_maxSlope+"\n");

            //O1 point
            double xO1 = (maxSlope * x_maxSlope -
y_maxSlope)/maxSlope;
            System.out.println("O1 point = "+xO1+"\n");
            if(computationIndex%2==0)
o1Lbl2.setText(getVal(xO1,3));
            else o1Lbl1.setText(getVal(xO1,3));

            //find the nearest xIndex, we will create a y01
value
            int O1index = 0;
            val = 0;
            for(int z=0;z<xVectorSm.size();z++){
                val = ((Double)
xVectorSm.get(z)).doubleValue();
                if(val>=xO1) {
                    O1index = z;
                    System.out.println("xO1 = "+xO1+"
val = "+val);
                    break;
                }
            }
        }

```

```

y_prev = ((Double) yVectorSm.get(Olindex-
1)).doubleValue();
y = ((Double)
yVectorSm.get(Olindex)).doubleValue();
double y01 = (y_prev + y)/2;
//insert point x01,y01
xVectorSm.insertElementAt(new Double(x01),
Olindex);
yVectorSm.insertElementAt(new Double(y01),
Olindex);

//dior8wse twra ta indexes pou brikes prin...
if(tlindex>=Olindex) tlindex++;
if(maxSlopeIndex>=Olindex) maxSlopeIndex++;

//tangent  $y = \lambda x + \beta$ 
// $\beta = y\_maxSlope - maxSlope * x\_maxSlope$ 
double tanB = y_maxSlope - maxSlope * x_maxSlope;
//again x01 if we want
// $x01 = -tanB/maxSlope$ ;
int tanOffset = -1; //it's needed below...
Vector xVectorTan = new Vector(); //ti na
kanoume...

Vector yVectorTan = new Vector();

xVectorTan.add(new Double(x01));
yVectorTan.add(new Double(0));

for(int t=0;t<xVectorSm.size();t++){
    val = ((Double)
xVectorSm.get(t)).doubleValue();
    y = maxSlope * val + tanB;
    if((val >= x01 )&&(y <= y_maxSlope)){
        if(tanOffset == -1) tanOffset = t;
        xVectorTan.add(new Double(val));
        yVectorTan.add(new Double(y));
    }
}
hvlChart.addSeries("Εφαπτομένη", xVectorTan,
yVectorTan);

//compute 1-g(t)
val = 0;
Vector yVector1minus = new Vector();
for(int j=0;j<yVectorSm.size();j++){
    if(j<=Olindex){
        yVector1minus.add(new Double(0));
    }
    else if(j<maxSlopeIndex){
        val = ((Double) yVectorTan.get(j-
tanOffset)).doubleValue();
        yVector1minus.add(new Double(1 - val));
    }
    else {
        val = ((Double)
yVectorSm.get(j)).doubleValue();
        yVector1minus.add(new Double(1 - val));
    }
}

```



```

        //overshooting b
        double o_b = maxGt - 1;
        System.out.println("Overshooting b = "+o_b+"\n");
        if(computationIndex%2==0)
bLb12.setText(getVal(o_b,3));
        else bLb1.setText(getVal(o_b,3));

        //integral vars
        double b, B, h;

        //T(t) integral
        Vector yVectorTt = new Vector();
        for(int d=0;d<O1index;d++) yVectorTt.add(new
Double(0));

        double tt = 0;
        for(int v=0index;v<yVectorlminus.size();v++){
            b = ((Double) yVectorlminus.get(v-
1)).doubleValue();
            B = ((Double)
yVectorlminus.get(v)).doubleValue();
            h = ((Double) xVectorSm.get(v)).doubleValue()
- ((Double) xVectorSm.get(v-1)).doubleValue();
            tt += (b+B)*h/2;
            yVectorTt.add(new Double(tt));
        }
        hv1Chart2.addSeries("Χρόνος απόκρισης", xVectorSm,
yVectorTt);

        //partial response time
        double partialRespTime = ((Double)
yVectorTt.get(t1index));
        System.out.println("Partial Response Time Ta =
T(t1) = "+partialRespTime+"\n");
        if(computationIndex%2==0)
TaLb12.setText(getVal(partialRespTime,3));
        else TaLb1.setText(getVal(partialRespTime,3));

        //experimental response time
        double experRespTime = tt;
        System.out.println("Experimental Response Time TN =
T(tmax) = "+experRespTime+"\n");
        if(computationIndex%2==0)
TNl12.setText(getVal(experRespTime,3));
        else TNl1.setText(getVal(experRespTime,3));

        //settling time Ts
        double ts = 0, xtt = 0;
        for(int q=0index;q<yVectorTt.size();q++){
            tt = ((Double)
yVectorTt.get(q)).doubleValue();
            xtt = ((Double)
xVectorSm.get(q)).doubleValue();
            if((Math.abs(experRespTime - tt)/0.02)<xtt){
                ts = xtt;
                break;
            }
        }
        System.out.println("Seetling Time Ts = "+ts+"\n");
        if(computationIndex%2==0)
TsLb12.setText(getVal(ts,3));

```

```

else TsLbl.setText(getVal(ts,3));

//initial distortion time To
double To = 0;
for(int f=1;f<=maxSlopeIndex;f++){
    b = ((Double) yVectorSm.get(f-
1)).doubleValue();
    B = ((Double)
yVectorSm.get(f)).doubleValue();
    h = ((Double) xVectorSm.get(f)).doubleValue()
- ((Double) xVectorSm.get(f-1)).doubleValue();
    To += (b+B)*h/2;
}

double triangle = (x_maxSlope - x01)*y_maxSlope/2;
To = To - triangle;
System.out.println("Initial distortion Time To =
"+To+"\n");

if(computationIndex%2==0)
ToLbl2.setText(getVal(To,3));
else ToLbl.setText(getVal(To,3));

state = 2;
}
catch(Exception q){
    q.printStackTrace();
}
System.out.println("\n ***** FINISHED *****\n");
}
catch(Exception ew){
    System.out.println("Check main computation action!
"+ew.getMessage());
    ew.printStackTrace();
}
}

public void chooseFileAction(){
    try{
        JFileChooser chooser = new JFileChooser();
        ExampleFileFilter filter = new ExampleFileFilter();
        filter.addExtension("csv");
        filter.setDescription("Comma Seperated Values file");
        chooser.setFileFilter(filter);
        chooser.setAcceptAllFileFilterUsed(false);
        chooser.setBackground(backgrndColor);
        chooser.setSelectionMode(JFileChooser.FILES_ONLY);
        chooser.setDialogTitle("Αναζήτηση αρχείου δεδομένων");
        chooser.setCurrentDirectory(new File("."));
        chooser.rescanCurrentDirectory();
        int returnVal = chooser.showOpenDialog(this);
        if(returnVal == JFileChooser.APPROVE_OPTION) {

            inputFile = chooser.getSelectedFile();

            inputFileFld.setText(inputFile.getAbsolutePath());
        }
        catch(Exception ee){
            System.out.println("Check file choosing action!
"+ee.getMessage());

```

```

        ee.printStackTrace();
    }
}

public void loadFileAction(){
    try{
        if(inputFile != null){
            xVector.clear();
            yVector.clear();

            String data = "";
            double x,y;
            BufferedReader in = new BufferedReader(new
FileReader(inputFile));
            while((data = in.readLine()) != null){

                try{
                    StringTokenizer stok = new
StringTokenizer(data, ";");
                    if(stok.hasMoreTokens()) x =
Double.parseDouble(stok.nextToken());
                    else x = 0;
                    if(stok.hasMoreTokens()) y =
Double.parseDouble(stok.nextToken());
                    else y = 0;

                    x = x * 1000000000; //x σε ns

                    xVector.add(new Double(x));
                    yVector.add(new Double(y));
                }
                catch(Exception qz){
                    System.out.println("Problem with
input! "+data+" "+qz.getMessage());
                }
            }

            hvlChart.addSeries("Αρχική", xVector,
yVector);

            state = 1;
        }
        else {
            System.out.println("File not specified!");
        }
    }
    catch(Exception ee){
        System.out.println("Check file loading action!
"+ee.getMessage());
        ee.printStackTrace();
    }
}

public void saveChartsAction(){
    saveChartsAction("USE_TIME");
}

public void saveChartsAction(String prefix){
    try{
        File testFile = new File("output");

```

```

        if(!testFile.exists()){
            testFile.mkdirs();
        }

        if(prefix.equalsIgnoreCase("USE_TIME")) prefix =
String.valueOf(Calendar.getInstance().getTime().getTime());

        System.out.print("Creating file "+prefix+".txt and jpg
images "+prefix+"_a and _b...");
        File fileA = new File(".\\output\\"+prefix+"_a.jpg");

        ImageIO.write(hvlChart.chart.createBufferedImage(1000,700), "jpg",
fileA);
        //ChartUtilities.saveChartAsJPEG(fileA, hvlChart.chart,
1000, 700);

        File fileB = new File(".\\output\\"+prefix+"_b.jpg");

        ImageIO.write(hvlChart2.chart.createBufferedImage(1000,700), "jpg",
fileB);
        //ChartUtilities.saveChartAsJPEG(fileB, hvlChart2.chart,
1000, 700);

        File fileC = new File(".\\output\\"+prefix+".txt");
        PrintWriter outC = new PrintWriter(new BufferedWriter(new
FileWriter(fileC, false), true);
        outC.println("Ανάλυση παραμέτρων απόκρισης συστημάτων
μέτρησης YT");
        outC.println("");
        outC.println("Δεδομένα από το αρχείο
"+lastInputFileName);
        outC.println("Αριθμός σημείων :
"+lastComputedSmoothRange+" Αριθμός εκτελέσεων : "+lastComputedTimes);
        outC.println("");
        if(computationIndex%2==0) {
            outC.println("O1\t: "+o1Lb12.getText());
            outC.println("b\t: "+bLb12.getText());
            outC.println("Ta\t: "+TaLb12.getText());
            outC.println("TN\t: "+TNLb12.getText());
            outC.println("Ts\t: "+TsLb12.getText());
            outC.println("To\t: "+ToLb12.getText());
        }
        else {
            outC.println("O1\t: "+o1Lb1.getText());
            outC.println("b\t: "+bLb1.getText());
            outC.println("Ta\t: "+TaLb1.getText());
            outC.println("TN\t: "+TNLb1.getText());
            outC.println("Ts\t: "+TsLb1.getText());
            outC.println("To\t: "+ToLb1.getText());
        }
        outC.println("-----");
        outC.println("");
        outC.close();

        File fileD = new File(".\\output\\ALL_RESULTS.txt");
        PrintWriter outD = new PrintWriter(new BufferedWriter(new
FileWriter(fileD, true), true);
        outD.println("Δεδομένα από το αρχείο
"+lastInputFileName);
        outD.println("Αριθμός σημείων :
"+lastComputedSmoothRange+" Αριθμός εκτελέσεων : "+lastComputedTimes);

```

```

        outD.println("");
        if (computationIndex%2==0) {
            outD.println("O1\t: "+o1Lbl2.getText());
            outD.println("b\t: "+bLbl2.getText());
            outD.println("Ta\t: "+TaLbl2.getText());
            outD.println("TN\t: "+TNlbl2.getText());
            outD.println("Ts\t: "+TsLbl2.getText());
            outD.println("To\t: "+ToLbl2.getText());
        }
        else {
            outD.println("O1\t: "+o1Lbl.getText());
            outD.println("b\t: "+bLbl.getText());
            outD.println("Ta\t: "+TaLbl.getText());
            outD.println("TN\t: "+TNlbl.getText());
            outD.println("Ts\t: "+TsLbl.getText());
            outD.println("To\t: "+ToLbl.getText());
        }
        outD.println("-----");
        outD.println("");
        outD.close();

        System.out.println("done!");
    }
    catch (Exception ew) {
        System.out.println("Check save charts action!
"+ew.getMessage());
        ew.printStackTrace();
    }
}

public void clearChartsAction() {
    try {
        hvlChart.clearAll();
        hvlChart2.clearAll();

        o1Lbl.setText(" ");
        bLbl.setText(" ");
        TaLbl.setText(" ");
        TNlbl.setText(" ");
        TsLbl.setText(" ");
        ToLbl.setText(" ");

        o1Lbl2.setText(" ");
        bLbl2.setText(" ");
        TaLbl2.setText(" ");
        TNlbl2.setText(" ");
        TsLbl2.setText(" ");
        ToLbl2.setText(" ");

        state = 0;

        computationIndex = 0;
    }
    catch (Exception ew) {
        System.out.println("Check clear charts action!
"+ew.getMessage());
        ew.printStackTrace();
    }
}
}

```

```

        public String getVal(double val, int decimals){
            try{
                //returns a string representation of a double with the
                selected number of decimals
                String valStr = String.valueOf(val);
                int comma = valStr.indexOf(".");
                if((comma != -1)&&(valStr.length()>(comma+decimals+1))){
                    valStr = valStr.substring(0,comma+decimals+1);
                }

                return valStr;
            }
            catch(Exception w){
                System.out.println("Check value prep! "+w.getMessage());
                w.printStackTrace();
            }

            return String.valueOf(val);
        }

        public static void main(String[] args){
            HVLgui hvlg = new HVLgui();
        }

        protected void processWindowEvent(WindowEvent e) {
            if(e.getID() == WindowEvent.WINDOW_CLOSING) {
                cancel();
                System.exit(1);
            }
            super.processWindowEvent(e);
        }

        void cancel() {
            dispose();
        }

    } //end of class

```

## Η κλάση HVLchart

```

/* =====
 * JFreeChart : a free chart library for the Java(tm) platform
 * =====
 */
package hvl;

import java.awt.*;
import java.awt.image.*;
import javax.swing.*;
import java.util.*;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.LogarithmicAxis;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.plot.XYPlot;
import org.jfree.chart.axis.NumberTickUnit;

```

```

import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
import org.jfree.ui.ApplicationFrame;
import org.jfree.ui.RefineryUtilities;

public class HVLchart extends JPanel {

    JFreeChart chart;
    XYSeriesCollection dataset;

    public HVLchart(String title, String xTitle, String yTitle) {

        super();

        dataset = new XYSeriesCollection();

        chart = ChartFactory.createXYLineChart(
            title,           // chart title
            xTitle,         // domain axis label
            yTitle,         // range axis label
            dataset,        // data
            PlotOrientation.VERTICAL,
            true,           // include legend
            true,
            false
        );

        XYPlot plot = chart.getXYPlot();
        NumberAxis domainAxis = new NumberAxis(xTitle);
        NumberAxis rangeAxis = new NumberAxis(yTitle);
        plot.setDomainAxis(domainAxis);
        plot.setRangeAxis(rangeAxis);
        chart.setBackgroundPaint(Color.white);
        plot.setOutlinePaint(Color.black);

        ChartPanel chartPanel = new ChartPanel(chart);
        chartPanel.setPreferredSize(new java.awt.Dimension(430,
400));

        setOpaque(false);
        setLayout(new GridLayout(1,1));
        add(chartPanel);
    }

    void addSeries(String seriesName, Vector xData, Vector yData){
        XYSeries sn = new XYSeries(seriesName);

        for(int i=0;i<xData.size();i++){
            sn.add((Double) xData.get(i), (Double) yData.get(i));
        }

        dataset.addSeries(sn);

        chart.fireChartChanged();
    }

    void clearAll(){
        dataset.removeAllSeries();
    }
}

```

## Η κλάση MassiveMeasurementsTask

```
package hvl;

import java.util.*;
import java.io.*;

public class MassiveMeasurementsTask {

    private boolean done = false;
    private String statMessage = "";
    private int completionPercent = 0;

    private HVLgui father;
    private long sleeptime = 100;

    public MassiveMeasurementsTask(HVLgui _father){
        father = _father;
    }

    public void go() {
        final SwingWorker worker = new SwingWorker() {
            public Object construct() {
                completionPercent = 0;
                done = false;
                statMessage = null;
                return new ActualTask();
            }
        };
        worker.start();
    }

    public int getCompletionPercent() {
        return completionPercent;
    }

    public boolean isDone() {
        return done;
    }

    public String getMessage() {
        return statMessage;
    }

    class ActualTask {
        ActualTask() {
            try {
                statMessage = "Άνοιγμα αρχείου";
                completionPercent+= 10;

                if(father.inputFile.exists()){

                    statMessage = "Επεξεργασία";
                    for(int sm=1;sm<=6;sm++){
                        for(int t=1;t<=5;t++){
                            Thread.sleep(sleeptime);
                            statMessage = "Φόρτωση ";
                            father.loadFileAction();

                            father.spinner1.setValue(new
Integer(sm));
```



```

Integer(t));

"+sm+ "."+t;

"+sm+ "."+t;

father.saveChartsAction(father.inputFile.getName()+"-"+sm+"-"+t);
completionPercent+= 10;

"+sm+ "."+t;

father.clearChartsAction();
completionPercent+= 5;
    }
}
else {
    father.inputFile = null;
    System.out.println("INPUT FILE NOT FOUND!");
}

done = true;
    System.out.println("done!");

} catch (Exception e) {
    System.out.println("ActualTask interrupted");
    e.printStackTrace();
}
} //end of Actual Task constructor
}
}

```

## Η κλάση SwingWorker

```

package hvl;

import javax.swing.SwingUtilities;

/**
 * This is the 3rd version of SwingWorker (also known as
 * SwingWorker 3), an abstract class that you subclass to
 * perform GUI-related work in a dedicated thread. For
 * instructions on and examples of using this class, see:
 *
 * http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html
 *
 * Note that the API changed slightly in the 3rd version:
 * You must now invoke start() on the SwingWorker after
 * creating it.
 */
public abstract class SwingWorker {

```

```

private Object value; // see getValue(), setValue()

/**
 * Class to maintain reference to current worker thread
 * under separate synchronization control.
 */
private static class ThreadVar {
    private Thread thread;
    ThreadVar(Thread t) { thread = t; }
    synchronized Thread get() { return thread; }
    synchronized void clear() { thread = null; }
}

private ThreadVar threadVar;

/**
 * Get the value produced by the worker thread, or null if it
 * hasn't been constructed yet.
 */
protected synchronized Object getValue() {
    return value;
}

/**
 * Set the value produced by worker thread
 */
private synchronized void setValue(Object x) {
    value = x;
}

/**
 * Compute the value to be returned by the <code>get</code> method.
 */
public abstract Object construct();

/**
 * Called on the event dispatching thread (not on the worker thread)
 * after the <code>construct</code> method has returned.
 */
public void finished() {
}

/**
 * A new method that interrupts the worker thread. Call this method
 * to force the worker to stop what it's doing.
 */
public void interrupt() {
    Thread t = threadVar.get();
    if (t != null) {
        t.interrupt();
    }
    threadVar.clear();
}

/**
 * Return the value created by the <code>construct</code> method.
 * Returns null if either the constructing thread or the current
 * thread was interrupted before a value was produced.
 *
 * @return the value created by the <code>construct</code> method
 */

```

```

public Object get() {
    while (true) {
        Thread t = threadVar.get();
        if (t == null) {
            return getValue();
        }
        try {
            t.join();
        }
        catch (InterruptedException e) {
            Thread.currentThread().interrupt(); // propagate
            return null;
        }
    }
}

/**
 * Start a thread that will call the <code>construct</code> method
 * and then exit.
 */
public SwingWorker() {
    final Runnable doFinished = new Runnable() {
        public void run() { finished(); }
    };

    Runnable doConstruct = new Runnable() {
        public void run() {
            try {
                setValue(construct());
            }
            finally {
                threadVar.clear();
            }

            SwingUtilities.invokeLater(doFinished);
        }
    };

    Thread t = new Thread(doConstruct);
    threadVar = new ThreadVar(t);
}

/**
 * Start the worker thread.
 */
public void start() {
    Thread t = threadVar.get();
    if (t != null) {
        t.start();
    }
}
}

```

## Η κλάση InfoDialog

```

package hvl;

import java.io.*;

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class InfoDialog extends JDialog {

    HVLgui father;
    JLabel statusLbl = new JLabel("", JLabel.CENTER);
    JProgressBar progressBar = new JProgressBar(JProgressBar.HORIZONTAL,
0, 1000);

    public InfoDialog(HVLgui _father){
        super(_father, "Μετρήσεις", false);

        enableEvents(AWTEvent.WINDOW_EVENT_MASK);

        father = _father;

        progressBar.setStringPainted(true);
        progressBar.setOpaque(false);
        progressBar.setValue(0);

        JPanel mainPanel = new JPanel();
        mainPanel.setBackground(father.backgrndColor);

        mainPanel.setBorder(BorderFactory.createEmptyBorder(10,10,10,10));
        mainPanel.setLayout(new GridLayout(2,1,2,5));
        mainPanel.add(statusLbl);
        mainPanel.add(progressBar);

        getContentPane().setBackground(father.backgrndColor);
        getContentPane().setLayout(new GridLayout(1,1));
        getContentPane().add(mainPanel);

        pack();

        Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
        setLocation((int) ((screenSize.width/2) -
(getWidth()/2)), (int) ((screenSize.height/2) - (getHeight()/2)));

        //show();
    }

    protected void processWindowEvent(WindowEvent e) {
        if (e.getID() == WindowEvent.WINDOW_CLOSING) {
            //cancel();
            setVisible(false);
        }
        //super.processWindowEvent(e);
    }

    void cancel() {
        dispose();
    }
}

```

## Η κλάση FooterPanel

```
package hvl;

import java.awt.*;
import javax.swing.*;
import javax.swing.border.*;

public class FooterPanel extends JPanel {

    HVLgui father;

    public FooterPanel(HVLgui _father){
        father = _father;

        //load images
        JLabel infoLbl = new JLabel(new
ImageIcon("images/pyrforos.gif"));
        infoLbl.setFont(new Font("Times New Roman", Font.BOLD, 28));
        infoLbl.setForeground(new Color(64,0,128));

        //LEFT PANEL
        JPanel infoPanel = new JPanel();
        infoPanel.setOpaque(false);
        infoPanel.setBorder(BorderFactory.createRaisedBevelBorder());
        infoPanel.setLayout(new GridLayout(1,1));
        infoPanel.add(infoLbl);

        JLabel infoFullLbl = new JLabel(" Εθνικό Μετσόβιο Πολυτεχνείο
", JLabel.CENTER);
        infoFullLbl.setFont(new Font("Arial", Font.BOLD, 12));

        JLabel infoCommentLbl = new JLabel(" Σχολή Ηλεκτρ. Μηχ. & Μηχ.
H/Y ", JLabel.CENTER);
        infoCommentLbl.setFont(new Font("Arial", Font.PLAIN, 10));

        JLabel infoAddrLbl = new JLabel("Εργαστήριο Υψηλών Τάσεων",
JLabel.CENTER);
        infoAddrLbl.setFont(new Font("Arial", Font.PLAIN, 10));

        JPanel infoDataPanel = new JPanel();
        infoDataPanel.setOpaque(false);
        infoDataPanel.setLayout(new GridLayout(3,1));
        infoDataPanel.add(infoFullLbl);
        infoDataPanel.add(infoCommentLbl);
        infoDataPanel.add(infoAddrLbl);

        JPanel leftPanel = new JPanel();
        leftPanel.setOpaque(false);
        leftPanel.setLayout(new BorderLayout(3,2));
        leftPanel.add(infoPanel, BorderLayout.WEST);
        leftPanel.add(infoDataPanel, BorderLayout.EAST);

        //RIGHT PANEL
        JLabel nameLbl = new JLabel("Ασημίνα Σαρλή", JLabel.CENTER);
        nameLbl.setFont(new Font("Arial", Font.BOLD, 12));

        JLabel thesisLbl = new JLabel(" Διπλωματική Εργασία ",
JLabel.CENTER);
        thesisLbl.setFont(new Font("Arial", Font.PLAIN, 10));
```

```

JLabel yearLbl = new JLabel("Μάιος 2008", JLabel.CENTER);
yearLbl.setFont(new Font("Arial", Font.PLAIN, 10));

JPanel rightPanel = new JPanel();
rightPanel.setOpaque(false);
rightPanel.setBorder(BorderFactory.createRaisedBevelBorder());
rightPanel.setLayout(new GridLayout(3,1));
rightPanel.add(nameLbl);
rightPanel.add(thesisLbl);
rightPanel.add(yearLbl);

//FOOTER PANEL
setBackground(father.backgrndColor);

setBorder(BorderFactory.createEtchedBorder(EtchedBorder.RAISED));
setLayout(new BorderLayout());
add(leftPanel, BorderLayout.WEST);
add(rightPanel, BorderLayout.EAST);

}
} //end of class

```

## Η κλάση ExampleFileFilter

```

/*
 * Copyright (c) 2003 Sun Microsystems, Inc. All Rights Reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * -Redistributions of source code must retain the above copyright
 * notice, this list of conditions and the following disclaimer.
 *
 * -Redistribution in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in
 * the documentation and/or other materials provided with the
distribution.
 *
 * Neither the name of Sun Microsystems, Inc. or the names of contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * This software is provided "AS IS," without a warranty of any kind. ALL
 * EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING
 * ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR
PURPOSE
 * OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN AND ITS LICENSORS SHALL
NOT
 * BE LIABLE FOR ANY DAMAGES OR LIABILITIES SUFFERED BY LICENSEE AS A
RESULT
 * OF OR RELATING TO USE, MODIFICATION OR DISTRIBUTION OF THE SOFTWARE OR
ITS
 * DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY
LOST
 * REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL,
CONSEQUENTIAL,

```

```

* INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE
THEORY
* OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE,
EVEN
* IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
*
* You acknowledge that Software is not designed, licensed or intended for
* use in the design, construction, operation or maintenance of any nuclear
* facility.
*/

/*
 * @(#)ExampleFileFilter.java 1.14 03/01/23
 */

package hvl;

import java.io.File;
import java.util.Hashtable;
import java.util.Enumeration;
import javax.swing.*;
import javax.swing.filechooser.*;

/**
 * A convenience implementation of FileFilter that filters out
 * all files except for those type extensions that it knows about.
 *
 * Extensions are of the type ".foo", which is typically found on
 * Windows and Unix boxes, but not on Macintosh. Case is ignored.
 *
 * Example - create a new filter that filters out all files
 * but gif and jpg image files:
 *
 *     JFileChooser chooser = new JFileChooser();
 *     ExampleFileFilter filter = new ExampleFileFilter(
 *         new String{"gif", "jpg"}, "JPEG & GIF Images")
 *     chooser.addChoosableFileFilter(filter);
 *     chooser.showOpenDialog(this);
 *
 * @version 1.14 01/23/03
 * @author Jeff Dinkins
 */
public class ExampleFileFilter extends FileFilter {

    private static String TYPE_UNKNOWN = "Type Unknown";
    private static String HIDDEN_FILE = "Hidden File";

    private Hashtable filters = null;
    private String description = null;
    private String fullDescription = null;
    private boolean useExtensionsInDescription = true;

    /**
     * Creates a file filter. If no filters are added, then all
     * files are accepted.
     *
     * @see #addExtension
     */
    public ExampleFileFilter() {
        this.filters = new Hashtable();
    }
}

```

```

/**
 * Creates a file filter that accepts files with the given extension.
 * Example: new ExampleFileFilter("jpg");
 *
 * @see #addExtension
 */
public ExampleFileFilter(String extension) {
    this(extension, null);
}

/**
 * Creates a file filter that accepts the given file type.
 * Example: new ExampleFileFilter("jpg", "JPEG Image Images");
 *
 * Note that the "." before the extension is not needed. If
 * provided, it will be ignored.
 *
 * @see #addExtension
 */
public ExampleFileFilter(String extension, String description) {
    this();
    if(extension!=null) addExtension(extension);
    if(description!=null) setDescription(description);
}

/**
 * Creates a file filter from the given string array.
 * Example: new ExampleFileFilter(String {"gif", "jpg"});
 *
 * Note that the "." before the extension is not needed and
 * will be ignored.
 *
 * @see #addExtension
 */
public ExampleFileFilter(String[] filters) {
    this(filters, null);
}

/**
 * Creates a file filter from the given string array and description.
 * Example: new ExampleFileFilter(String {"gif", "jpg"}, "Gif and JPG
Images");
 *
 * Note that the "." before the extension is not needed and will be
ignored.
 *
 * @see #addExtension
 */
public ExampleFileFilter(String[] filters, String description) {
    this();
    for (int i = 0; i < filters.length; i++) {
        // add filters one by one
        addExtension(filters[i]);
    }
    if(description!=null) setDescription(description);
}

/**
 * Return true if this file should be shown in the directory pane,
 * false if it shouldn't.

```



```

*
* Files that begin with "." are ignored.
*
* @see #getExtension
* @see FileFilter#accepts
*/
public boolean accept(File f) {
    if(f != null) {
        if(f.isDirectory()) {
            return true;
        }
        String extension = getExtension(f);
        if(extension != null && filters.get(getExtension(f)) != null) {
            return true;
        }
    }
    return false;
}

/**
 * Return the extension portion of the file's name .
 *
 * @see #getExtension
 * @see FileFilter#accept
 */
public String getExtension(File f) {
    if(f != null) {
        String filename = f.getName();
        int i = filename.lastIndexOf('.');
        if(i>0 && i<filename.length()-1) {
            return filename.substring(i+1).toLowerCase();
        }
    }
    return null;
}

/**
 * Adds a filetype "dot" extension to filter against.
 *
 * For example: the following code will create a filter that filters
 * out all files except those that end in ".jpg" and ".tif":
 *
 * ExampleFileFilter filter = new ExampleFileFilter();
 * filter.addExtension("jpg");
 * filter.addExtension("tif");
 *
 * Note that the "." before the extension is not needed and will be
 * ignored.
 */
public void addExtension(String extension) {
    if(filters == null) {
        filters = new Hashtable(5);
    }
    filters.put(extension.toLowerCase(), this);
    fullDescription = null;
}

/**
 * Returns the human readable description of this filter. For
 * example: "JPEG and GIF Image Files (*.jpg, *.gif)"

```

```

*
* @see setDescription
* @see setExtensionListInDescription
* @see isExtensionListInDescription
* @see FileFilter#getDescription
*/
public String getDescription() {
    if(fullDescription == null) {
        if(description == null || isExtensionListInDescription()) {
            fullDescription = description==null ? "(" : description + " (";
            // build the description from the extension list
            Enumeration extensions = filters.keys();
            if(extensions != null) {
                fullDescription += "." + (String) extensions.nextElement();
                while (extensions.hasMoreElements()) {
                    fullDescription += ", ." + (String)
extensions.nextElement();
                }
            }
            fullDescription += ")";
        } else {
            fullDescription = description;
        }
    }
    return fullDescription;
}

/**
 * Sets the human readable description of this filter. For
 * example: filter.setDescription("Gif and JPG Images");
 *
 * @see setDescription
 * @see setExtensionListInDescription
 * @see isExtensionListInDescription
 */
public void setDescription(String description) {
    this.description = description;
    fullDescription = null;
}

/**
 * Determines whether the extension list (.jpg, .gif, etc) should
 * show up in the human readable description.
 *
 * Only relevent if a description was provided in the constructor
 * or using setDescription();
 *
 * @see getDescription
 * @see setDescription
 * @see isExtensionListInDescription
 */
public void setExtensionListInDescription(boolean b) {
    useExtensionsInDescription = b;
    fullDescription = null;
}

/**
 * Returns whether the extension list (.jpg, .gif, etc) should
 * show up in the human readable description.
 *
 * Only relevent if a description was provided in the constructor

```

```
* or using setDescription();
*
* @see getDescription
* @see setDescription
* @see setExtensionListInDescription
*/
public boolean isExtensionListInDescription() {
    return useExtensionsInDescription;
}
}
```