



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

## Ανάπτυξη πειραματικής διάταξης χαρακτηρισμού πιεζοηλεκτρικών υλικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΙΩΑΝΝΗ Α. ΚΟΥΚΟΒΙΝΗ ΠΛΑΤΙΑ**

**Επιβλέπων :** Ηλίας Κουκούτσης  
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2014

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ  
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

## Ανάπτυξη πειραματικής διάταξης χαρακτηρισμού πιεζοηλεκτρικών υλικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΙΩΑΝΝΗ Α. ΚΟΥΚΟΒΙΝΗ ΠΛΑΤΙΑ**

**Επιβλέπων :** Ηλίας Κουκούτσης  
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11<sup>η</sup> Μαΐου 2014.

.....  
Ηλίας Κουκούτσης  
Επίκουρος Καθηγητής

.....  
Κωνσταντίνος Παπαοδυσσεύς  
Αναπληρωτής Καθηγητής

.....  
Γεώργιος Καμπουράκης  
Αναπληρωτής Καθηγητής

Αθήνα, Ιούνιος 2014

.....

**ΙΩΑΝΝΗΣ Α. ΚΟΥΚΟΒΙΝΗΣ ΠΛΑΤΙΑΣ**

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

Copyright © Ιωάννης Α. Κουκοβίνης Πλατιάς, 2014.  
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.

## Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την ανάπτυξη πειραματικής διάταξης για τον χαρακτηρισμό πιεζοηλεκτρικών υλικών. Η ολοένα και αυξανόμενη ζήτηση για μια άμεση και αξιόπιστη διαδικασία χαρακτηρισμού πιεζοηλεκτρικών υλικών οδήγησε στην ανάγκη ανάπτυξης μιας πειραματικής διάταξης, η οποία θα μπορεί να πραγματοποιήσει μετρήσεις υστέρησης (hysteresis) και ερπυσμού (creep) των πιεζοηλεκτρικών υλικών μεταξύ των λειτουργιών της. Η διάταξη που αναπτύχθηκε θα χρησιμοποιηθεί μεταξύ άλλων στα πλαίσια του UA9 Project, το οποίο είναι μια διακρατική σύμπραξη που σκοπό έχει την βελτίωση του συστήματος collimation του LHC (Large Hadron Collider – Μεγάλος Επιταχυντής Αδρονίων), του μεγαλύτερου και ισχυρότερου επιταχυντή σωματιδίων παγκοσμίως που βρίσκεται στο CERN.

Το λογισμικό για την διάταξη αναπτύχθηκε στο περιβάλλον οπτικού προγραμματισμού LabVIEW της National Instruments ενώ τα scripts για την μεταγενέστερη επεξεργασία των δεδομένων που συλλέχθηκαν σε Matlab. Όλες οι σημαντικές πτυχές της σχεδίασης του λογισμικού αναλύονται με λεπτομέρεια, μεταξύ των οποίων η σχεδιαστική αρχιτεκτονική, η διαχείριση, απεικόνιση και μετεπεξεργασία των δεδομένων, ο συγχρονισμός της διαδικασίας λήψης δεδομένων, όλες οι επιλογές εκτέλεσης καθώς και η αυτόματη δημιουργία αναφοράς.

**Λέξεις Κλειδιά:** Πιεζοηλεκτρικά υλικά, LabVIEW, Matlab, υστέρηση, ερπυσμός, UA9, CERN, LHC, Μεγάλος Επιταχυντής Αδρονίων

Η σελίδα αυτή είναι σκόπιμα λευκή.

## **Abstract**

The present diploma thesis deals with the development of a test-bench control for piezoelectric actuators characterization. The growing demand for a fast and reliable procedure for piezoelectric actuators' characterization has subsequently led to the need for development of a test-bench structure that can perform hysteresis and creep measurements among its other functions. The developed test bench will be used to provide results that will be utilized within the UA9 project, a collaborative initiative that has undertaken the task of improving the collimation system, a critical component of the Large Hadron Collider (LHC), the world's largest and most powerful particle accelerator at CERN.

The software part of the project is written in National Instruments' LabVIEW visual programming development environment and the scripts for the post-processing of the collected data in Matlab. All of the important software design aspects are covered in detail and those include among others the design architecture, the data management, representation and post-processing, the synchronization of the data acquisition, all the execution options and the automatic report generation.

**Keywords:** Piezoelectric, actuator, LabVIEW, Matlab, hysteresis, creep, UA9, CERN, LHC

Η σελίδα αυτή είναι σκόπιμα λευκή.



## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Ηλία Κουκούτση για την συνεχή υποστήριξη και καθοδήγησή του καθώς και την παρότρυνση του να αποκτήσω πολύτιμες εμπειρίες δουλεύοντας στο CERN. Επίσης θα ήθελα να εκφράσω την ευγνωμοσύνη μου στον επιβλέποντά μου στο CERN, Mario Di Castro, για την εμπιστοσύνη που μου έδειξε και την βοήθεια. Επιπλέον θα ήθελα να εκφράσω την εκτίμηση μου προς τους συναδέλφους μου στο CERN, χωρίς την βοήθεια των οποίων δεν θα μπορούσα να προχωρήσω.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την αγάπη και υποστήριξη μέσα στα χρόνια.

Η σελίδα αυτή είναι σκόπιμα λευκή.

## Πίνακας Περιεχομένων

<b>1</b>	<b>Εισαγωγή.....</b>	<b>19</b>
1.1	Εισαγωγή στο CERN.....	20
1.2	Εισαγωγή στον LHC.....	20
1.2.1	Σκοπός του LHC.....	22
1.2.2	LHC Layout.....	23
1.3	Το LHC Collimation System.....	26
1.4	Το UA9 Project.....	28
1.5	Εισαγωγή στα Πιεζοηλεκτρικά Υλικά.....	31
1.5.1	Φαινόμενο υστέρησης.....	35
1.5.2	Φαινόμενο ερπυσμού.....	36
1.5.3	Λοιπές έννοιες.....	37
1.6	Εισαγωγή στο LabVIEW.....	38
<b>2</b>	<b>Διάταξη &amp; Υλικό.....</b>	<b>41</b>
2.1	Εισαγωγή στην πειραματική διάταξη.....	41
2.2	State of the art.....	43
2.3	Τεχνικές Προδιαγραφές.....	46
2.3.1	Μηχανική διάταξη.....	46
2.3.2	Διάταξη κινητήρα.....	46
2.3.3	Αισθητήρες και περιφερειακές συσκευές.....	48
2.3.4	Σύστημα PXI.....	51
2.4	Λειτουργική ανάλυση.....	58
<b>3</b>	<b>Ανάλυση LabVIEW.....</b>	<b>61</b>
3.1	Βασικές δομικές έννοιες.....	61
3.1.1	State Machine.....	61
3.1.2	Functional Global Variable (FGV).....	63
3.2	Κύρια ανάλυση.....	65
3.2.1	Επισκόπηση.....	65
3.2.2	Εφαρμογή Server.....	67
3.2.3	Εφαρμογή Client.....	72
3.3	Επιλογές εκτέλεσης.....	78
3.3.1	Creep τεστ.....	78
3.3.2	Static-execution Hysteresis τεστ.....	81
3.3.3	Dynamic-execution Hysteresis τεστ.....	84

3.4	Διαχείριση δεδομένων .....	86
3.5	Συγχρονισμός λήψης δεδομένων .....	95
3.6	Γραφική αναπαράσταση δεδομένων .....	99
3.7	Μετεπεξεργασία Matlab .....	103
3.8	Αυτόματη δημιουργία αναφοράς .....	105
<b>4</b>	<b>Αποτελέσματα .....</b>	<b>111</b>
<b>5</b>	<b>Επίλογος .....</b>	<b>117</b>
5.1	Σύνοψη και συμπεράσματα.....	117
5.2	Μελλοντικές επεκτάσεις .....	118
<b>6</b>	<b>Βιβλιογραφία .....</b>	<b>119</b>
<b>Παράρτημα Α Κώδικας Matlab.....</b>		<b>123</b>
<b>Παράρτημα Β Οδηγίες για το LabVIEW Project.....</b>		<b>127</b>

## Κατάλογος Σχημάτων

Σχήμα 1. 1: Τοποθεσία του υπόγειου LHC τούνελ .....	20
Σχήμα 1. 2: Το σύμπλεγμα επιταχυντών στο CERN.....	21
Σχήμα 1. 3: Συνολική εποπτεία των πειραμάτων του LHC.....	24
Σχήμα 1. 4: Περίληψη του LHC layout.....	26
Σχήμα 1. 5: Collimator base-support assembly .....	27
Σχήμα 1. 6: Ένα LHC collimator με την δεξαμενή κενού ανοιχτή, αριστερά, και κοντινό πλάνο του ανοίγματος, δεξιά.....	27
Σχήμα 1. 7: Διαδικασία collimation με κρύσταλλο.....	29
Σχήμα 1. 8: Πιεζοηλεκτρικό γωνιόμετρο .....	31
Σχήμα 1. 9: Συμπεριφορά πιεζοηλεκτρικών υλικών .....	32
Σχήμα 1. 10: Γραμμικός ενεργοποιητής.....	33
Σχήμα 1. 11: Bender ενεργοποιητής.....	34
Σχήμα 1. 12: Τυπική γραφική παράσταση υστέρησης.....	34
Σχήμα 1. 13: Πλήρης καμπύλη υστέρησης .....	36
Σχήμα 1. 14: Φαινόμενο ερπυσμού .....	37
Σχήμα 1. 15: Τυπικό παράδειγμα VI.....	39
Σχήμα 2. 1: Κεντρικό μέρος της διάταξης.....	42
Σχήμα 2. 2: Περιφερειακές συσκευές- National Instruments PXI chassis και modules (κάτω-αριστερά), ενισχυτής σερβοκινητήρα και τροφοδοσία του μηχανικού φρένου (πάνω-αριστερά) και γραμμικοί ενισχυτές για διέγερση του πιεζοηλεκτρικού στοιχείου και μετρήσεις του capacitive sensor(right).....	43
Σχήμα 2. 3: Διάταξη για μονή στοίβα PZT (πάνω-αριστερά) και διάταξη για δύο κυλινδρικές PZT στοίβες (πάνω-δεξιά και κάτω) .....	44
Σχήμα 2. 4: Instron 8516 σερβοϋδραυλικό σύστημα .....	45
Σχήμα 2. 5α (αριστερά) και Σχήμα 2. 5β (δεξιά).....	45
Σχήμα 2. 6: Ενισχυτής σερβοκινητήρα (αριστερά) και σερβοκινητήρας (δεξιά).....	47
Σχήμα 2. 7: Neugart PLPE 70 μπροστινή όψη (αριστερά), πίσω (μέση) και εσωτερική (δεξιά) .....	47
Σχήμα 2. 8: R+W SK2.....	48
Σχήμα 2. 9: Fogale MCC-10 συνδεδεμένο στο MC900 (αριστερά) και διαθέσιμα modules (δεξιά).....	49

Σχήμα 2. 10: Cedrat LA75B και LC75B .....	49
Σχήμα 2. 11: Althen SG-KP-12E-420 (αριστερά), Tedeo Huntleigh 615 (μέση) και Althen ALF256 (δεξιά) .....	50
Σχήμα 2. 12: PCB Piezotronics 356B18 επιταχυνσιόμετρο .....	50
Σχήμα 2. 13: Γεμάτο και άδειο PXIe-1078 chassis .....	52
Σχήμα 2. 14: NI PXIe-8133 .....	53
Σχήμα 2. 15: PXIe-4357 (πάνω-αριστερά), TB-4357 (πάνω-δεξιά), TB-4357 συνδεδεμένο στο PXIe-4357 (κάτω-αριστερά) και TB-4357 εσωτερική όψη (κάτω-δεξιά).....	54
Σχήμα 2. 16: PXIe-4496 .....	55
Σχήμα 2. 17: PXI-4461 .....	55
Σχήμα 2. 18: PXI-7841R (αριστερά) και SCB-68 connector block (δεξιά) .....	56
Σχήμα 2. 19: Πλακέτα ηλεκτρονόμου και εξωτερική τροφοδοσία για τη λειτουργία του φρένου .....	57
Σχήμα 2. 20: Δευτερεύουσα διάταξη για μετρήσεις ερπυσμού (αριστερά) και κοντινό της διάταξης με τον capacitive sensor και το πιεζοηλεκτρικό στοιχείο τοποθετημένα (δεξιά).....	58
Σχήμα 2. 21: Μορφή κυματομορφής διέγερσης στοιχείου για τεστ ερπυσμού.....	59
Σχήμα 2. 22: Κύρια διάταξη (αριστερά και κέντρο) και κοντινή όψη του πιεζοηλεκτρικού ενεργοποιητή και του capacitive sensor (δεξιά).....	60
Σχήμα 3. 1: Αρχιτεκτονική state machine .....	62
Σχήμα 3. 2α (αριστερά): Single default transition και Σχήμα 3. 2β (δεξιά): Δύο πιθανές μεταβάσεις όπου το αποτέλεσμα της έκβασης καθορίζεται από τη συνάρτηση select .....	63
Σχήμα 3. 3: Γενική μορφή Functional Global Variable .....	63
Σχήμα 3. 4: FGV με “Set” και “Get” καταστάσεις.....	64
Σχήμα 3. 5: Διάγραμμα ροής των server και client εφαρμογών.....	66
Σχήμα 3. 6: Εφαρμογή server – κατάσταση Initialization.....	68
Σχήμα 3. 7: Εφαρμογή server – κατάσταση DAQ Configuration και Front Panel .....	69
Σχήμα 3. 8: Εφαρμογή server – Data acquisition και data transmission state με παραλλαγές case .....	70
Σχήμα 3. 9: Εφαρμογή server – Close references κατάσταση.....	71
Σχήμα 3. 10: Εφαρμογή server – Error κατάσταση.....	71
Σχήμα 3. 11: Εφαρμογή client – Initialization κατάσταση.....	73
Σχήμα 3. 12: Εφαρμογή client – Κατάσταση data reception και data logging.....	74
Σχήμα 3. 13: Εφαρμογή client: Post-processing options και Front Panel case με παραλλαγές case .....	76
Σχήμα 3. 14: Εφαρμογή client – Close references κατάσταση.....	77
Σχήμα 3. 15: Εφαρμογή client – Error κατάσταση.....	78

Σχήμα 3. 16: Creep τεστ Front Panel .....	79
Σχήμα 3. 17: Δημιουργία Creep κυματομορφής εξόδου .....	80
Σχήμα 3. 18: Hysteresis τεστ Front Panel .....	82
Σχήμα 3. 19: Static-execution load setting Front Panel.....	83
Σχήμα 3. 20: Dynamic-execution load setting Front Panel .....	85
Σχήμα 3. 21: “ <i>SyncedAcquisitionProcess</i> ” subvi .....	86
Σχήμα 3. 22: “Enqueue” κατάσταση των data FGVs στην εφαρμογή server.....	87
Σχήμα 3. 23: “ <i>PT100AcquisitionProcess</i> ” subvi.....	87
Σχήμα 3. 24: “ <i>accelerometerAcquisitionProcess</i> ” subvi.....	88
Σχήμα 3. 25: “ <i>accelerometerAcquisitionProcess</i> ” subvi - FFT.....	88
Σχήμα 3. 26: Αποστολή δεδομένων μέσω network stream .....	89
Σχήμα 3. 27: “Dequeue” κατάσταση για τα περισσότερα data FGVs στην εφαρμογή server .89	
Σχήμα 3. 28: PT100 “Dequeue” κατάσταση στην εφαρμογή server .....	90
Σχήμα 3. 29: “Dequeue” FGV κατάσταση των επιταχυνσιόμετρων και FFT στη server εφαρμογή.....	91
Σχήμα 3. 30: Λήψη δεδομένων στη client εφαρμογή.....	91
Σχήμα 3. 31: “Enqueue” κατάσταση των data FGVs στην client εφαρμογή .....	92
Σχήμα 3. 32: “Dequeue” κατάσταση για τα περισσότερα data FGVs στην client εφαρμογή ..92	
Σχήμα 3. 33: “Dequeue” κατάσταση FGV επιταχυνσιόμετρων και FFT στην client εφαρμογή .....	93
Σχήμα 3. 34: Data logging στην client εφαρμογή .....	94
Σχήμα 3. 35: “ <i>initSyncStartDynamic</i> ” subvi .....	96
Σχήμα 3. 36: “ <i>initSyncStartStatic</i> ” subvi.....	97
Σχήμα 3. 37: “ <i>removeJunkSamples</i> ” subvi .....	98
Σχήμα 3. 38: “ <i>SyncedAcquisitionProcess</i> ” subvi 1 <sup>ο</sup> iteration.....	99
Σχήμα 3. 39: “ <i>graphCreationClient</i> ” subvi.....	100
Σχήμα 3. 40: Κώδικας “ <i>graphCreationClient</i> ” subvi.....	101
Σχήμα 3. 41: αρχική μορφή δεδομένων FFT (αριστερά) και επιθυμητή (δεξιά) .....	102
Σχήμα 3. 42: “ <i>FFTGraphClient</i> ” subvi .....	102
Σχήμα 3. 43: “ <i>unifiedLengthArrayClient</i> ” subvi .....	103
Σχήμα 3. 44: “ <i>normalizeArrayClient</i> ” subvi .....	103
Σχήμα 3. 45: MathScript κόμβος.....	104
Σχήμα 3. 46: Front Panel του “ <i>reportOptionsClient</i> ” subvi με όλες τις πιθανές επιλογές.....	105
Σχήμα 3. 47: “ <i>reportGenerationClient</i> ” subvi .....	106
Σχήμα 3. 48: “ <i>hysteresisReportClient</i> ” subvi.....	107
Σχήμα 3. 49: “ <i>creepReportClient</i> ” subvi.....	108
Σχήμα 3. 50: “ <i>leakageCurrentReportClient</i> ” subvi.....	108





## *Πρόλογος*

Η παρούσα εργασία εκπονήθηκε σε συνεργασία με τον Ευρωπαϊκό Οργανισμό Πυρηνικών Ερευνών (CERN).



# 1

## *Εισαγωγή*

Σκοπός του παρόντος κεφαλαίου είναι η εισαγωγή του αναγνώστη στις έννοιες που πραγματεύεται η παρούσα διπλωματική.

Αρχικά, γίνεται μια εισαγωγή στο CERN, ακολουθούμενη από μία περιγραφή του Μεγάλου Επιταχυντή Αδρονίων (LHC), της δομής του και των κύριων συστατικών που τον απαρτίζουν. Επακόλουθα, επικεντρωνόμαστε σε ένα από τα πιο κρίσιμα υποσυστήματα του LHC, το Collimation system, παρέχοντας λεπτομέρειες σχετικά με την λειτουργία του καθώς και τις μελλοντικές τάσεις. Συγκεκριμένα, εστιάζουμε στο προτεινόμενο crystal collimation UA9 project, το οποίο θα βελτιώσει τη λειτουργία και απόδοση του collimation system και περιλαμβάνει την χρησιμοποίηση πιεζοηλεκτρικών στοιχείων. Εν συνεχεία, κάνουμε μια μικρή εισαγωγή στην επιστήμη πιεζοηλεκτρικών υλικών, εστιάζοντας κυρίως στις ιδιότητες των υλικών που μελετώνται στην παρούσα διπλωματική, καθώς και μια σύντομη εισαγωγή στο λογισμικό LabVIEW της National Instruments, το οποίο επέτρεψε το έργο της διπλωματικής να περατωθεί.

## ***1.1 Εισαγωγή στο CERN***

Το CERN, ο Ευρωπαϊκός Οργανισμός Πυρηνικών Ερευνών, θεωρείται ένα από τα μεγαλύτερα και πιο καταξιωμένα κέντρα επιστημονικής έρευνας και βρίσκεται στα νοτιοδυτικά προάστια της Γενεύης στα Γάλλο-ελβετικά σύνορα. Προς το παρόν έχει είκοσι ευρωπαϊκά κράτη μέλη, καθώς και το Ισραήλ, που είναι το πρώτο και μοναδικό μη-ευρωπαϊκό κράτος μέλος.

Το CERN ιδρύθηκε το 1954 και έκτοτε βρίσκεται στο προσκήνιο της επιστημονικής έρευνας με πολλές σπουδαίες ανακαλύψεις να έχουν λάβει χώρα σε αυτό, όπως η εφεύρεση του World Wide Web.

Αυτή τη στιγμή, πάνω από 10000 επιστήμονες και 600 ιδρύματα από πάνω από 100 χώρες συμμετέχουν στις δραστηριότητες του CERN, καθιστώντας το ένα από τα πιο πολυπολιτισμικά περιβάλλοντα στον ερευνητικό χώρο. [1][2][3]

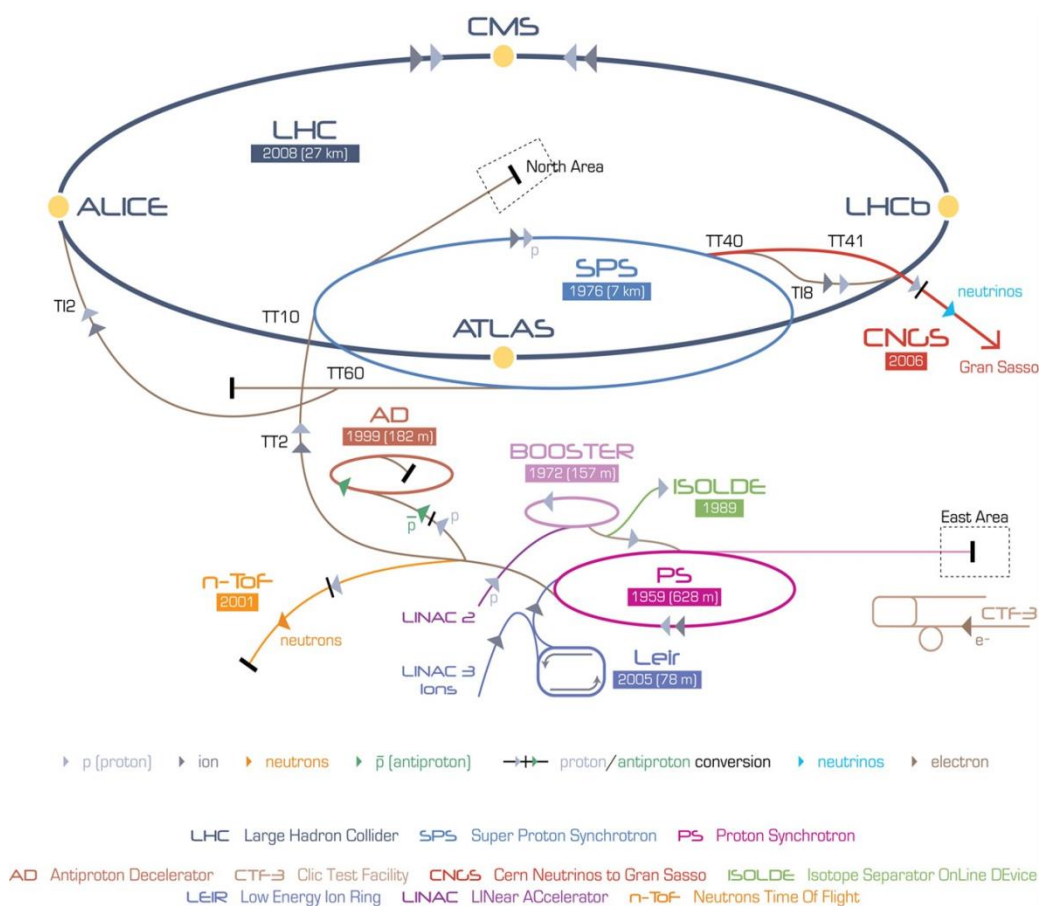
## ***1.2 Εισαγωγή στον LHC***

Ο LHC (Μεγάλος Επιταχυντής Αδρονίων) είναι ένας κυκλικός επιταχυντής με διάμετρο 27 km, που βρίσκεται 175 m κάτω από τα Γάλλο-ελβετικά σύνορα κοντά στη Γενεύη και κατασκευάστηκε από το CERN.



**Σχήμα 1. 1:** Τοποθεσία του υπόγειου LHC τούνελ

Ο LHC χρησιμοποιεί το ίδιο τούνελ με τον προηγούμενο επιταχυντή του CERN, Large Electron Positron (LEP), τα αποτελέσματα του οποίου συνετέλεσαν στην καθιέρωση του Καθιερωμένου Μοντέλου<sup>1</sup> κατά τη διάρκεια λειτουργίας του από το 1989 ως το 2000. [4]



Σχήμα 1. 2: Το σύμπλεγμα επιταχυντών στο CERN

Μόλις τα σωματίδια παράγονται από τον ιονισμό του υδρογόνου, γραμμικοί επιταχυντές (LINAC), ενισχύουν την ενέργειά τους προτού αυτά εγχυθούν στο Proton Synchrotron (PS) διαμέτρου 630 m. Τα πρωτόνια εν συνεχεία εγχέονται στο 7 km διαμέτρου Super Proton Synchrotron (SPS) το οποίο τα επιταχύνει στα 450 GeV. Ο

<sup>1</sup> Το Καθιερωμένο Μοντέλο (Standard Model - SM) είναι ένα σύνολο θεωριών που περιγράφουν τα θεμελιώδη σωματίδια και την αλληλεπίδραση τους. Παρά την πειραματική επιτυχία του, το Καθιερωμένο Μοντέλο δεν θεωρείται ακόμα πλήρης θεωρία καθώς δεν περιλαμβάνει τη βαρύτητα, τη σκοτεινή ύλη και τη σκοτεινή ενέργεια.

LHC τότε επιταχύνει τα σωματίδια στα 7 TeV <sup>2</sup>. Δύο δέσμες πρωτονίων, με ενέργεια 7 TeV η κάθε μια συγκρούονται, οδηγώντας σε ενέργεια 14 TeV ανά σύγκρουση.

Κάθε δέσμη περιέχει 2808 bunches με ελάχιστη απόσταση 7 m στη μέγιστη λειτουργία. Κάθε bunch περιέχει περίπου  $1.15 \times 10^{11}$  πρωτόνια με μήκος περίπου 11 cm και διάμετρο 400 μm. Κοντά στα σημεία αλληλεπίδρασης, τα bunches συμπιέζονται μέχρι το μήκος τους να είναι 7.5 cm και η διάμετρός τους 16 μm ώστε να αυξηθεί η πιθανότητα σύγκρουσης. Είκοσι συγκρούσεις πρωτονίου-πρωτονίου ανά bunch προσδοκούνται σε ένα τρέξιμο υψηλής φωτεινότητας, αλλά δεδομένης της συχνότητας σύγκρουσης 40 MHz, ο συνολικός αριθμός συγκρούσεων ξεπερνά τα 800 εκατομμύρια ανά δευτερόλεπτο. Η φωτεινότητα, που ισούται με τον αριθμό σωματιδίων στη διατομή της δέσμης ανά μονάδα χρόνου, φτάνει τη μέγιστη τιμή των  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  ύστερα από δύο χαμηλότερες φάσεις,  $5.0 \cdot 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$  και  $2 \cdot 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ .

Για να εκτραπούν οι δέσμες πρωτονίων στο δαχτυλίδι του LHC, 1232 υπεραγωγμοί (superconducting - SC) διπολικοί μαγνήτες μήκους 14.3 m ο καθένας χρησιμοποιούνται καθώς και 858 υπεραγωγμοί τετραπολικοί μαγνήτες για να τις εστιάσουν. Για τις χαμηλές θερμοκρασίες που απαιτούνται για την επίτευξη της υπεραγωγμής ιδιότητας στους μαγνήτες χρησιμοποιείται υγρό ήλιο. Οι διατάξεις κρυογονικής που χρησιμοποιούνται για να παρέχουν το ήλιο αποτελούν το μεγαλύτερο σύστημα ψύξης στον κόσμο, λειτουργώντας στους 1.9 °K. Ο LHC θεωρείται το πιο πολύπλοκο επιστημονικό πείραμα στην ιστορία. [5][6][7][8]

### **1.2.1 Σκοπός του LHC**

Ένας από τους σημαντικότερους στόχους του LHC ήταν η απόδειξη ύπαρξης του σωματιδίου Higgs. Το μποζόνιο Higgs, που προτάθηκε θεωρητικά το 1964, είναι ένα στοιχειώδες σωματίδιο του οποίου η ύπαρξη ανακοινώθηκε στο CERN στις 4 Ιουλίου 2012. Η σημασία της ανακάλυψης έγκειται στην επιβεβαίωση του πεδίου Higgs, το οποίο είναι κρίσιμο στο Καθιερωμένο Μοντέλο και σε άλλες θεωρίες στη φυσική

---

<sup>2</sup> Το TeV είναι μονάδα μέτρησης της ενέργειας που χρησιμοποιείται στη φυσική σωματιδίων. 1 TeV είναι περίπου η ενέργεια κίνησης ενός ιπτάμενου κουνουπιού. Αυτό που κάνει τον LHC τόσο ξεχωριστό είναι ότι στριμώνει την ενέργεια σε ένα χώρο περίπου ένα εκατομμύριο φορές μικρότερο από ένα κουνούπι.

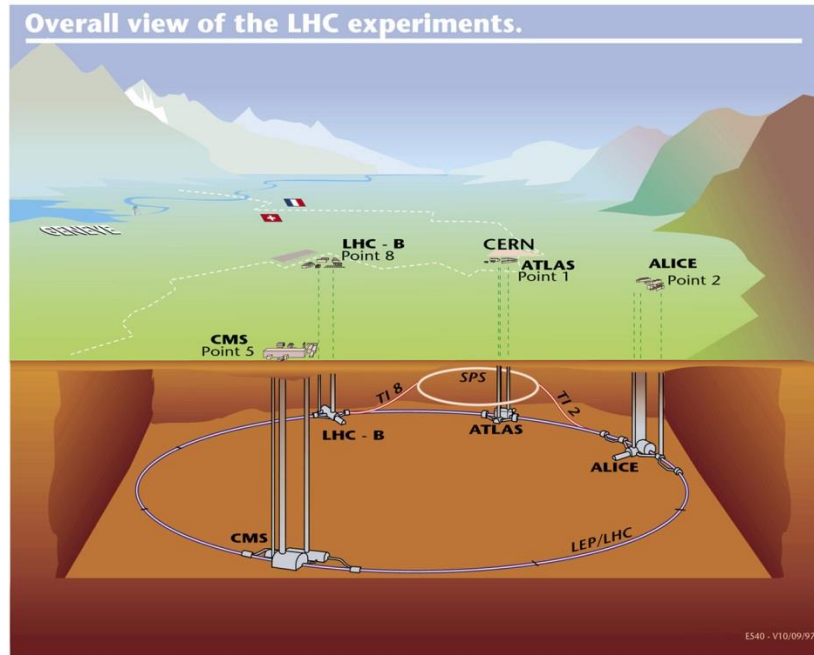
σωματιδίων. Με το Καθιερωμένο Μοντέλο, οι φυσικοί έχουν καταφέρει να περιγράψουν τα στοιχειώδη σωματίδια και τις αλληλεπιδράσεις τους. Το μποζόνιο Higgs, ένα massive scalar στοιχειώδες σωματίδιο, εξηγεί γιατί τα άλλα στοιχειώδη σωματίδια έχουν μάζα και στην Ηλεκτρασθενή Θεωρία (electroweak theory) παράγει τις μάζες των λεπτονίων (ηλεκτρόνια, μίονια, ταυ) και κουάρκς. [9]

Δεδομένα από όλους τους ανιχνευτές θα αναλυθούν επίσης για την παρατήρηση επιπλέον διαστάσεων. Η Θεωρία Χορδών (String theory) προβλέπει ότι υπάρχουν πρόσθετες χωρικές διαστάσεις και ίσως μπορούν να παρατηρηθούν σε πολύ υψηλές ενέργειες.

### **1.2.2 LHC Layout**

Ο LHC αποτελείται από πολλές διατάξεις που επιτελούν διακεκριμένα έργα. Παρότι μεγάλα σε αριθμό, τα βασικά στοιχεία του LHC μπορούν να συνοψιστούν ως ακολούθως.

Τα βασικά στοιχεία φυσικά είναι οι **ανιχνευτές**. Τέσσερις ανιχνευτές σωματιδίων έχουν κατασκευαστεί και τοποθετηθεί σε μεγάλες υπόγειες στοές, που αντιστοιχούν στα τέσσερα σημεία συγκρούσεων των δύο δεσμών. Αυτοί είναι οι ATLAS (A Toroidal LHC ApparatuS), CMS (Compact Muon Solenoid), ALICE (An LHC Ion Collision Experiment) και LHCb (LHC beauty). Οι ATLAS και CMS είναι σχεδιασμένοι για να μελετούν ένα μεγάλο εύρος φυσικής και θεωρούνται πειράματα γενικού σκοπού. Ο ALICE είναι αφιερωμένος στις συγκρούσεις βαριών ιόντων και αναδημιουργεί συνθήκες παρόμοιες με αυτές ακριβώς μετά το Big Bang ώστε να μελετηθούν και αναλυθούν οι ιδιότητες των quark-gluon πλάσματος, ενώ ο LHCb είναι αφιερωμένος στα φαινόμενα B-physics – με άλλα λόγια, εστιάζει στα B-hadron decays τα οποία μπορούν να μας διαφωτίσουν σχετικά με τις διαφορές μεταξύ ύλης και αντιύλης καθώς και πιθανότατα να μας παρέχουν έμμεσες αποδείξεις για διαδικασίες που παραβιάζουν το Καθιερωμένο Μοντέλο. Επιπλέον υπάρχουν οι TOTEM (Total Cross Section, Elastic Scattering and Diffraction Dissociation at the LHC), LHCf (Large Hadron Collider forward) και MoEDAL (the Monopole and Exotics Detector At the LHC). [10]



Σχήμα 1. 3: Συνολική εποπτεία των πειραμάτων του LHC

Σημαντικά στοιχεία στην λειτουργία του LHC είναι οι **υπεραγωγίμοι μαγνήτες**. Αυτοί είναι υπεύθυνοι για να κρατάνε τις δέσμες σταθερές και ευθυγραμμισμένες. Συγκεκριμένα οι **διπολικοί μαγνήτες** αποτελούν ένα από τα πιο πολύπλοκα κομμάτια του LHC και χρησιμοποιούνται για την κάμψη της τροχιάς των σωματιδίων. Αυτό είναι απαραίτητο αφού ο LHC είναι *κυκλικός επιταχυντής*, που σημαίνει ότι η δέσμη περιστρέφεται πολλές φορές σε κλειστή τροχιά πριν φτάσει το μέγιστο επίπεδο ενέργειας, σε αντίθεση με έναν *γραμμικό επιταχυντή*, όπου τα φορτισμένα σωματίδια κινούνται σε ευθείες τροχιές και περνούν από αριθμό επιταχυντικών διατάξεων. Υπάρχουν 1232 κύριοι διπολικοί μαγνήτες εγκατεστημένοι στον LHC και ο καθένας έχει 14.3m μήκος και ζυγίζει 35 τόνους.

Άλλα ζωτικά στοιχεία είναι οι **τετραπολικοί μαγνήτες**. Αυτοί βοηθούν ώστε να κρατηθούν τα σωματίδια σε μια σφιχτή δέσμη – με άλλα λόγια, εστιάζουν τη δέσμη. Η ιδέα πίσω από αυτό είναι ότι όταν τα σωματίδια συγκεντρώνονται, είναι πιο πιθανό να συγκρουστούν σε μεγαλύτερες ποσότητες όταν φτάσουν τους ανιχνευτές του LHC. Επιπλέον υπάρχουν μικρότεροι εξαπολικοί, οκταπολικοί and δεκαπολικοί μαγνήτες για τη διόρθωση μικρών ατελειών των μαγνητικών πεδίων στα άκρα των διπολικών μαγνητών. [11][12]



Μεγάλης σημασίας είναι επίσης οι **κοιλότητες ραδιοσυχνότητας (radiofrequency RF cavities)**. Πρόκειται για μεταλλικούς θαλάμους που περιέχουν ηλεκτρομαγνητικό πεδίο και κύριος σκοπός τους είναι να επιταχύνουν φορτισμένα σωματίδια. Είναι κατασκευασμένοι με τέτοιο τρόπο ώστε τα ηλεκτρομαγνητικά κύματα να συντονίζονται και να ενισχύονται στο εσωτερικό τους. Το αποτέλεσμα είναι φορτισμένα σωματίδια που περνούν μέσα απ' τις κοιλότητες να δέχονται την ώθηση του εναλλασσόμενου ηλεκτρομαγνητικού πεδίου και να επιταχύνονται. Αξίζει να σημειωθεί ότι το πεδίο αλλάζει κατεύθυνση σε συγκεκριμένη συχνότητα ώστε να καθίσταται δυνατή η επιτάχυνση. [13]

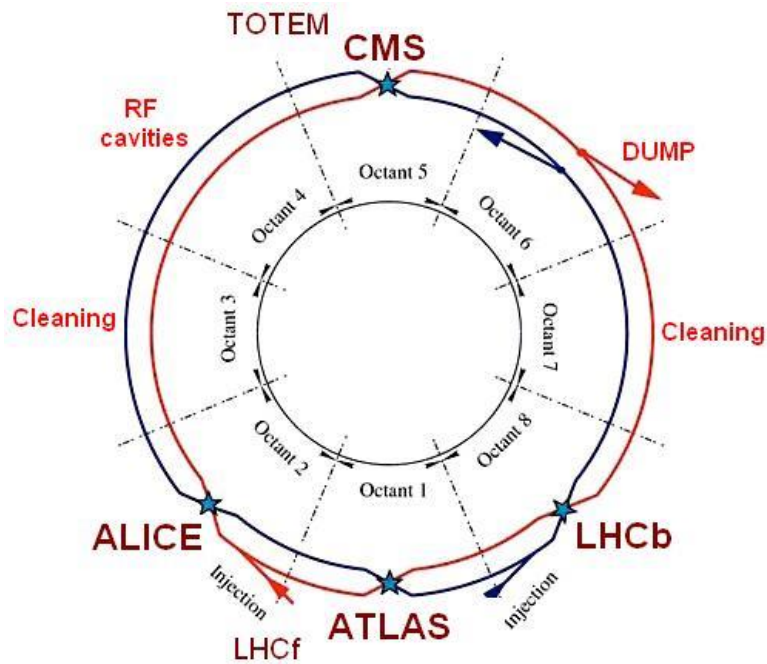
Ένα άλλο σημαντικό στοιχείο του LHC είναι το **σύστημα απόρριψης δέσμης (beam dump system)**. Οι δέσμες δεν κυκλοφορούν μέσα στον επιταχυντή επ' άπειρο. Καθώς σωματίδια συγκρούονται με το πλάι του σωλήνα της δέσμης (beam pipe) ή μεταξύ τους, οι δέσμες 'φθίνουν' με αποτέλεσμα να γίνεται όλο και πιο απίθανο να οδηγήσουν σε ενδιαφέρουσες συγκρούσεις. Οι φυσικοί μπορούν να απορρίψουν τις δέσμες, αφαιρώντας τις από τον επιταχυντή και στέλνοντάς τις να απορριφθούν με ασφάλεια στο σύστημα απόρριψης δέσμης – συνήθως ένα θωρακισμένο ενάντια στην ακτινοβολία τετράγωνο βαθιά κάτω από το έδαφος. Για να γίνει κατανοητή η αναγκαιότητα του συστήματος, μια ονομαστική δέσμη του LHC περιέχει αποθηκευμένη ενέργεια 350 MJ, που περιλαμβάνεται σε 2808 bunches με τυπική απόκλιση  $\sigma$  της τάξης των 0.3 mm. Η καταστροφική δύναμη μιας τέτοιας δέσμης επιτάσσει ένα σύστημα απόρριψης, με το οποίο αφαιρείται η δέσμη εξ' ολοκλήρου από τον LHC, ελαττώνεται η πυκνότητα μέγιστης δύναμης και εν τέλει απορροφάται η δέσμη από ένα εξειδικευμένο σύστημα. [14][15]

Τέλος αναπόσπαστο κομμάτι του LHC είναι το **collimation system**. Η υψηλής φωτεινότητας απόδοση του LHC βασίζεται στην αποθήκευση, επιτάχυνση και σύγκρουση δεσμών με πρωτοφανή ενέργεια. Η πυκνότητα ενέργειας της δέσμης είναι 1000 φορές υψηλότερη απ' ότι είχε επιτευχθεί σε παρόμοια δαχτυλίδια αποθήκευσης πρωτονίων. Ελάχιστη διαρροή αυτής της αποθηκευμένης ενέργειας επαρκεί για να επιφέρει quench<sup>3</sup> σε έναν υπεραγωγίμο μαγνήτη ή ακόμα και να καταστρέψει μέρη του επιταχυντή. Το LHC collimation system παρέχει προστασία στον επιταχυντή

---

<sup>3</sup> Το "Quenching" συμβαίνει όταν οποιοδήποτε μέρος ενός υπεραγωγίμου καλωδίου ενός μαγνήτη μεταβαίνει σε μη-υπεραγωγίμη κατάσταση. Συμβαίνει όταν ξεπερνιέται η κρίσιμη θερμοκρασία ή σε περίπτωση αποτυχίας ψύξης ή απωλειών δέσμης.

έναντι στις αναπόφευκτες συνηθισμένες και απρόβλεπτες απώλειες και επιτυγχάνει το απαραίτητο περιφερειακό καθάρισμα (halo cleaning) της δέσμης. [16]



Σχήμα 1. 4: Περίληψη του LHC layout

### 1.3 To LHC Collimation System

Όπως αναφέρθηκε εν συντομία, το LHC collimation system αποτελεί την άμυνα του LHC απέναντι στις αναπόφευκτες απώλειες της δέσμης. Σε αντίθεση με ένα ιδανικό πείραμα, η εμπειρία έχει δείξει ότι συγκεκριμένες διεργασίες οδηγούν σε απώλειες. Οι χειρισμοί που απαιτούνται για την προετοιμασία της δέσμης για σύγκρουση, καθώς και οι ίδιες οι συγκρούσεις, οδηγούν σε απώλειες. Όπως γίνεται εύκολα κατανοητό, αυτές οι απώλειες μεγαλώνουν καθώς αυξάνεται η φωτεινότητα της δέσμης. Επιπλέον, εξ' αιτίας του υπεραγωγίμου περιβάλλοντος του LHC, ένα αποδοτικό σύστημα καθαρισμού που θ' αποτρέπει τα "quenches" είναι αναγκαίο. [17]

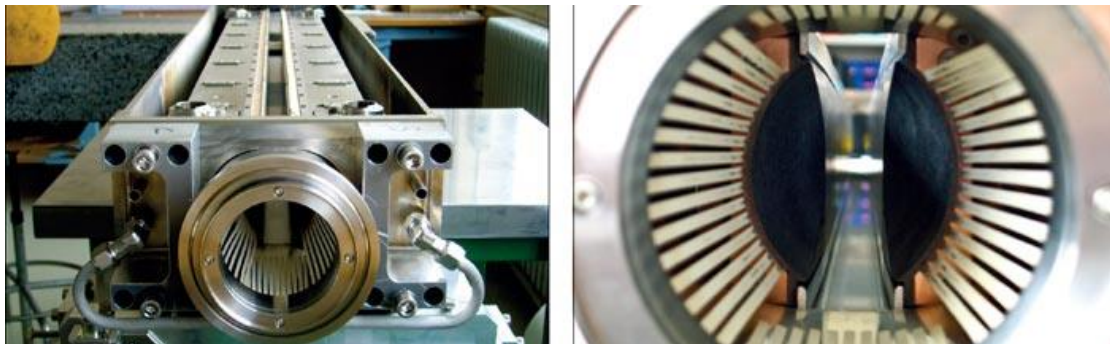
Ο κύριος σκοπός του collimation system είναι το καθάρισμα του φωτοστέφανου (beam halo), διατηρώντας ταυτόχρονα τις απώλειες σε ευαίσθητα σημεία κάτω του επιτρεπτού ορίου. Το υπάρχον σύστημα εξασφαλίζει ότι κάτω του 0.01% των απωλειών της δέσμης εναποτίθεται στους κρούς μαγνήτες.

Το εγκατεστημένο collimation system αποτελείται από 5 μέρη: (1) τη βάση, που στηρίζει το σύστημα στο τούνελ, (2) την πλάκα ευθυγράμμισης, που επιτρέπει τη ρύθμιση του collimator και (3) το lower plugin, που παρέχει ψυκτικό υγρό και τις ηλεκτρικές ενώσεις. Τα προαναφερθέντα μέρη αποτελούν το base-support assembly, που φαίνεται στο Σχ. 1.5.



Σχήμα 1. 5: Collimator base-support assembly

Το υπόλοιπο collimation system αποτελείται από (4) τη δεξαμενή κενού με δύο jaws και κινητήρες για την κίνησή τους και (5) το upper support assembly, που φιλοξενεί όλα τα αναγκαία για την εγκατάσταση μέρη. Ο διαχωρισμός μεταξύ base-support και upper-support assembly εξυπηρετεί την γρήγορη εγκατάσταση και αντικατάσταση του collimator σε περίπτωση βλάβης.



Σχήμα 1. 6: Ένα LHC collimator με την δεξαμενή κενού ανοιχτή, αριστερά, και κλειστή, δεξιά.

Η δεξαμενή αποτελείται από δύο jaws που επιτρέπουν ένα άνοιγμα για τη δέσμη, πρακτικά περιορίζοντας το beam halo και από τις δύο πλευρές. Αισθητήρες ακριβείας χρησιμοποιούνται για να παρακολουθείται η θέση των jaws και η απόστασή τους. Πάνω στα jaws είναι τοποθετημένοι αισθητήρες θερμοκρασίας. Όλα τα παραπάνω είναι συνδεδεμένα σε ένα beam-interlock σύστημα που πυροδοτεί την απόρριψη της δέσμης αν ανιχνευθούν πιθανώς επικίνδυνες συνθήκες.

Το collimation system του LHC περιλαμβάνει μια πολυεπίπεδη διαδικασία καθαρισμού. Πρωτοβάθμια, δευτεροβάθμια και τριτοβάθμια collimators και absorbers αξιοποιούνται για να μειωθούν τα halo σωματίδια σε ανεκτά επίπεδα. Συμπαγή υλικά από άνθρακα αλλά και μη-συμπαγή αλλά υψηλής απορρόφησης μεταλλικά υλικά χρησιμοποιούνται για διάφορους σκοπούς.

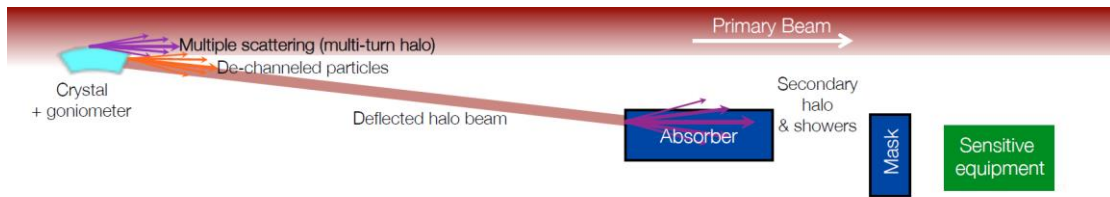
Το collimation system του LHC είναι το μεγαλύτερο και πιο προηγμένο σύστημα καθαρισμού που κατασκευάστηκε για επιταχυντή σωματιδίων. Αποτελείται από 84 κινούμενους collimators δύο όψεων διαφόρων σχεδίων και υλικών, που μαζί με τους injection protection collimators επιτρέπουν 396 βαθμούς ελευθερίας.

Στη μέγιστη ενέργεια του LHC, μια δέσμη μεγέθους μικρότερου από 200  $\mu\text{m}$  μεταφράζεται στην ανάγκη τα collimators να λειτουργούν με μεγάλη ακρίβεια. Η εγγενής πολυπλοκότητα των παρόντων collimators καθώς και η επιθυμία επίτευξης ακόμα υψηλότερου καθαρίσματος halo, οδήγησαν στο UA9 project. [18][19]

## ***1.4 To UA9 Project***

Το UA9 Project, μια σύμπραξη πανεπιστημίων και ερευνητικών κέντρων από την Γαλλία, Ιταλία, Ρωσία και Ηνωμένο Βασίλειο, ερευνά πως μικροσκοπικοί λυγισμένοι κρύσταλλοι μπορούν να βελτιώσουν το collimation σε μεγάλους επιταχυντές, όπως ο LHC.

Αυτό που κάνει τους λυγισμένους κρυστάλλους χρήσιμους είναι ότι η κρυσταλλική δομή μπορεί να περιορίσει τις κατευθύνσεις των φορτισμένων σωματιδίων που περνούν από μέσα τους. Αυτή η ιδιότητα μπορεί να χρησιμοποιηθεί για την απομάκρυνση σωματιδίων. Αφού εκτραπούν από τον κρύσταλλο, τα halo σωματίδια αναχαιτίζονται από δευτερεύουσες διατάξεις collimators που δρουν ως absorbers.



Σχήμα 1. 7: Διαδικασία collimation με κρύσταλλο

Όπως αναφέρθηκε, τα υπάρχοντα collimators είναι ογκώδη και τα absorbers πολύ κοντά στη δέσμη, κάνοντας την ευθυγράμμιση πολύπλοκη. Η χρήση κρυστάλλων λύνει αυτά τα προβλήματα, καθώς ένας κρύσταλλος που χρησιμοποιείται ως πρωτεύων collimator μπορεί να εκτρέψει τα halo σωματίδια σε μεγάλες γωνίες και να τα κατευθύνει σε δευτερεύουσες διατάξεις collimator-absorber. Κατ' αυτό τον τρόπο, το ογκώδες collimator-absorber μπορεί να τοποθετηθεί σε μεγαλύτερη απόσταση απ' την δέσμη, ελαττώνοντας της πολυπλοκότητα του συστήματος και αυξάνοντας ταυτόχρονα την απόδοση του.

Οι πρώτες δοκιμές της ιδέας έγιναν το 2009, χρησιμοποιώντας δέσμες από το SPS (Super Proton Synchrotron) και ήταν επιτυχημένες. Ανέκδοτα αποτελέσματα προσομοιώσεων για το προτεινόμενο collimation system του LHC, πράγματι προβλέπουν μια απόδοση καθαρισμού 5 με 10 φορές καλύτερη σε σύγκριση με το υπάρχον σύστημα. Η προσαρμογή, ωστόσο, της ιδέας σε έναν επιταχυντή υψηλών ενεργειών όπως ο LHC ενέχει προκλήσεις. Συγκεκριμένα, σε συνθήκες ηρεμίας ένας λυγισμένος κρύσταλλος μπορεί να εναποθέτει μέχρι 0.5 MW ισχύος σε ένα μικρό σημείο του collimator-absorber, πράγμα που σημαίνει ότι το collimator-absorber πρέπει να μπορεί να αντέχει αυτή τη δύναμη για πολλά δευτερόλεπτα χωρίς ζημιά. Επιπρόσθετα, ο ρυθμός ανάπτυξης του halo της δέσμης είναι τόσο αργός που οι πρώτες προσκρούσεις στον κρύσταλλο συμβαίνουν σε μια περιοχή που μόλις ξεπερνά τα λίγα ατομικά στρώματα. Αυτό μεταφράζεται στην απαίτηση να υπάρχει μια επίπεδη επιφάνεια παράλληλη στα κρυσταλλικά επίπεδα με πρωτοφανή ανοχή. Τέλος, ισχύει πως όσο μεγαλύτερη είναι η ενέργεια σωματιδίου, τόσο μικρότερη είναι η γωνιακή ανοχή. Αυτό επιτάσσει μηχανισμούς ευθυγράμμισης με μεγάλη γωνιακή ακρίβεια και σε αυτό το πλαίσιο έγκειται η ανάγκη κατασκευής διάταξης για τον χαρακτηρισμό πιεζοηλεκτρικών στοιχείων.

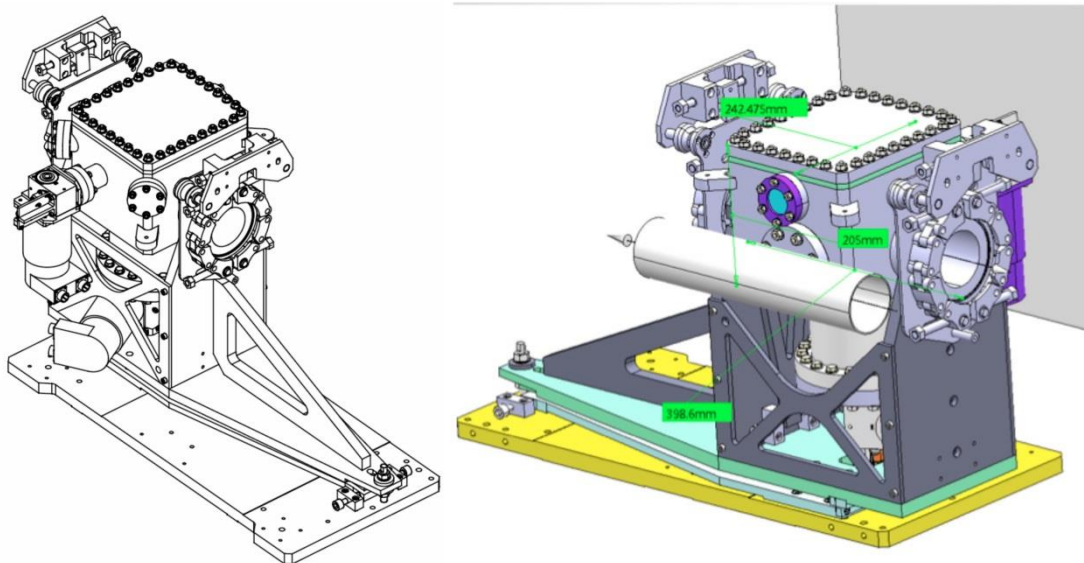
Σε αυτό το σημείο είναι απαραίτητο να παραθέσουμε μερικές λεπτομέρειες του προτεινόμενου από το UA9 Project συστήματος collimator κρυστάλλου για τον LHC.

Το σχέδιο περιστρέφεται γύρω από ένα γωνιόμετρο κατάλληλο για τον προσανατολισμό και την τοποθέτηση του κρυστάλλου με γωνιακή ακρίβεια της τάξης του 1  $\mu\text{rad}$  που απαιτείται από τη λειτουργία του LHC. Η προτεινόμενη λύση αποτελείται από δύο διατάξεις, μια μηχανολογική, γραμμική διάταξη και μια πιεζοηλεκτρική, περιστροφική διάταξη που εγγυάται γραμμική μεταβολή του γωνιακού προσανατολισμού σε όλες τις περιπτώσεις. Το πλάνο που έχει προταθεί εισάγει τον ελάχιστο αριθμό επιπλέον συσκευών στο υπάρχον πλάνο του LHC, εκμεταλλευόμενο τα υπάρχοντα collimators και όργανα.

Μέχρι τώρα, η περιστροφική κίνηση επιτυγχάνεται μετατρέποντας τη γραμμική κίνηση ενός μοχλού, χρησιμοποιώντας υβριδικούς κινητήρες και LVDTs για την παρακολούθηση της γραμμικής τοποθέτησης. Αυτή η διάταξη σε γενικές γραμμές επιτυγχάνει το σκοπό της σε σταθερές συνθήκες, ωστόσο, κάτω από συγκεκριμένες συνθήκες παρατηρείται υπέρβαση τόσο στη γραμμική όσο και στη γωνιακή τοποθέτηση.

Για να ξεπεραστεί το πρόβλημα της γωνιακής ακρίβειας, προτάθηκε το πιεζοηλεκτρικό γωνιόμετρο του Σχ. 1.8. Η προαναφερθείσα λύση εξασφαλίζει τα παρακάτω:

- Μετακίνηση του κρυστάλλου γραμμικά και κάθετα στη δέσμη με βήμα 60 mm και γραμμική ανάλυση 5  $\mu\text{m}$ .
- Περιστροφή του κρυστάλλου με εύρος γωνιακής εκτροπής  $\pm 10$  mrad με γωνιακή ανάλυση 0.1  $\mu\text{rad}$  και γωνιακή ακρίβεια  $\pm 1$   $\mu\text{rad}$ .
- Γωνιακή εκτροπή με ανακρίβεια  $< 10$  % και settling time στο 99% της τελικής τιμής των 20 ms.
- Σωστή λειτουργία μετά από ψήσιμο στους 250 ° C.
- Σωστή λειτουργία σε επίπεδα ακτινοβολίας 10 MGy.
- Εύκολη εγκατάσταση στο LHC Collimator quick plug support system.
- Διαφανές στη κανονική λειτουργία του LHC.



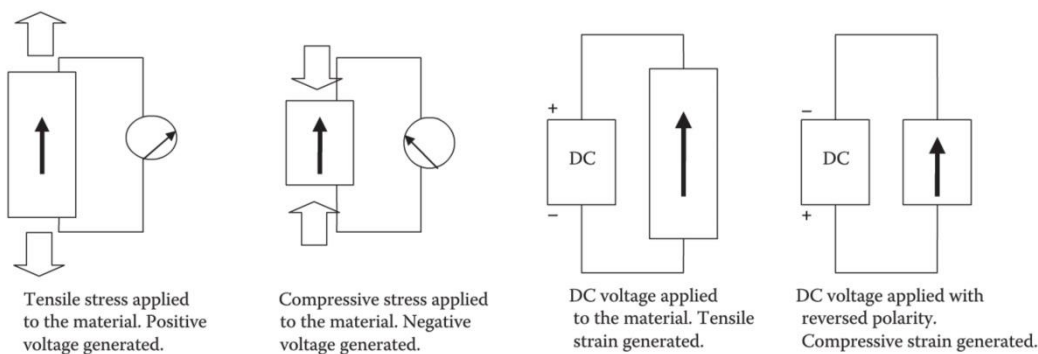
Σχήμα 1. 8: Πιεζοηλεκτρικό γωνιόμετρο

Συνοψίζοντας, το σχέδιο περιλαμβάνει μια πιεζοηλεκτρική περιστροφική διάταξη τοποθετημένη πάνω σε ένα υψηλής ακρίβειας γραμμικό άξονα, που μαζί εκπληρώνουν τις προδιαγραφές.

Γι' αυτό το λόγο, κρίθηκε απαραίτητη η ανάπτυξη μιας πειραματικής διάταξης για τον χαρακτηρισμό πιεζοηλεκτρικών υλικών, που θα πιστοποιούσε υπάρχουσες μετρήσεις και θα επέτρεπε τη μελέτη σύνθετων θεμάτων, όπως η επίδραση της ακτινοβολίας στα πιεζοηλεκτρικά υλικά. [20][21][22][23]

## 1.5 Εισαγωγή στα Πιεζοηλεκτρικά Υλικά

Η λέξη piezo προέρχεται από την ελληνική λέξη “πιέζειν”. Τα πιεζοηλεκτρικά υλικά κατέχουν δύο αντίστροφες ιδιότητες: την ικανότητα να αναπτύσσουν ηλεκτρικό φορτίο στην επιφάνειά τους όταν μηχανική πίεση ασκείται σε αυτά και την ικανότητα να εμφανίζουν μηχανική παραμόρφωση όταν διοχετεύονται με ηλεκτρικό φορτίο. Οι ιδιότητες αυτές απεικονίζονται στο Σχ. 1.9.



Σχήμα 1. 9: Συμπεριφορά πιεζοηλεκτρικών υλικών

Υπάρχουν φυσικά πιεζοηλεκτρικά υλικά όπως τα quartz και tourmaline αλλά το πιεζοηλεκτρικό φαινόμενο σε αυτά δεν είναι πολύ ισχυρό και το υλικό πρέπει να κοπεί σε συγκεκριμένα σχήματα. Τεχνητοί κρύσταλλοι όπως τα Rochelle salt και lithium sulphate μπορούν να αποκτήσουν την πιεζοηλεκτρική ιδιότητα με βιομηχανικές διαδικασίες αλλά κάτι τέτοιο απαιτεί μεγάλο χρόνο διεργασίας. Αυτό μας οδηγεί στα κεραμικά υλικά όπως το Lead Zirconate Titanate (PZT)<sup>4</sup>, που δεν κατέχουν πιεζοηλεκτρικές αρετές εκ φύσεως, αλλά μπορούν να τις αποκτήσουν μέσω της διαδικασίας που ονομάζεται *poling*. Υπάρχουν πολλά πλεονεκτήματα στα κεραμικά piezos όπως η εύκολη κατασκευή και ευελιξία τους όσον αφορά το σχήμα.

Το πιεζοηλεκτρικό φαινόμενο ανακαλύφθηκε από τους Jacques και Pierre Curie το 1880 και έκτοτε πιεζοηλεκτρικά υλικά χρησιμοποιούνται σε πληθώρα εφαρμογών. Τέτοια είναι ενεργοποιητές (actuators), αισθητήρες, πηγές ενέργειας και υψηλής τάσης, πιεζοηλεκτρικοί κινητήρες, συστήματα ελέγχου θορύβου και κραδασμών, στη βιομηχανία πληροφορικής (ρολόγια quartz για την παραγωγή παλμών ή στους σκληρούς δίσκους), στη βιομηχανία μουσικών οργάνων (μαγνήτες οργάνων και μικρόφωνα) και στη βιομηχανία ιατρικής (παρακολούθηση πίεσης και παλμών, απεικόνιση υπερήχων μεταξύ άλλων).

Όπως αναφέρθηκε, το UA9 project επικεντρώνεται στη χρησιμοποίηση πιεζοηλεκτρικών actuators (ενεργοποιητών) στο γωνιόμετρο και ως εκ τούτου είναι χρήσιμο να παραθέσουμε μερικές πληροφορίες σχετικά με αυτούς.

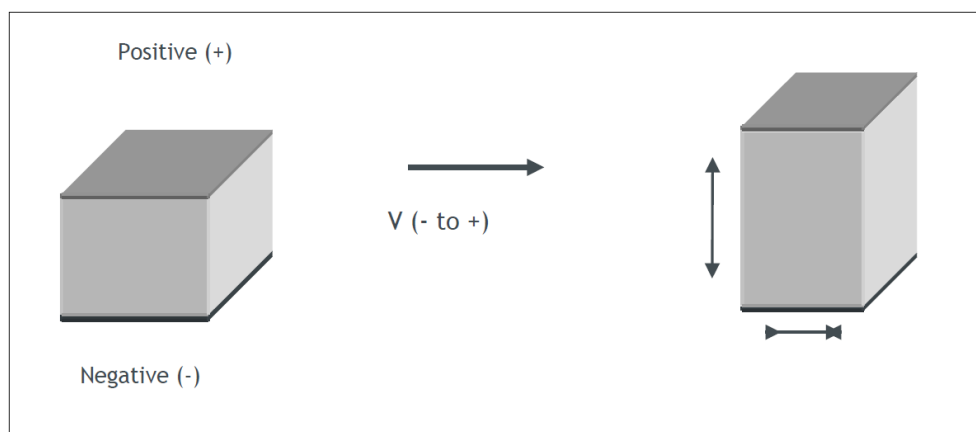
Το έμμεσο πιεζοηλεκτρικό φαινόμενο, κατά το οποίο ένα ηλεκτρικό ερέθισμα μεταβάλλει τις διαστάσεις του πιεζοηλεκτρικού υλικού, έχει ως αποτέλεσμα την

<sup>4</sup> PZT είναι ο χημικός τύπος, όπου το P αντιπροσωπεύει το “Plombus”, δηλαδή Μόλυβδο.



ανάπτυξη πιεζοηλεκτρικών ενεργοποιητών για πληθώρα εφαρμογών. Το πλεονέκτημά τους είναι ότι δεν απαιτούν πολύπλοκα σχέδια και μπορούν να παράγουν σημαντικά επίπεδα δύναμης με γρήγορη απόκριση για χαμηλές τάσεις. Υπάρχουν τρεις κύριοι τύποι: οι γραμμικοί, στοιβαγμένοι και εύκαμπτοι ενεργοποιητές.

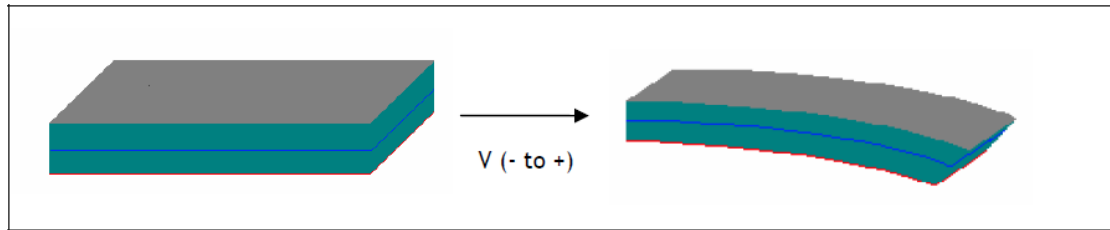
Οι **γραμμικοί ενεργοποιητές (linear)** εμφανίζουν γραμμική παραμόρφωση αναλογικά με την εφαρμοζόμενη τάση. Όπως φαίνεται στο Σχ. 1.10, η παραμόρφωση συμβαίνει τόσο στον X όσο και στον Y άξονα.



Σχήμα 1. 10: Γραμμικός ενεργοποιητής

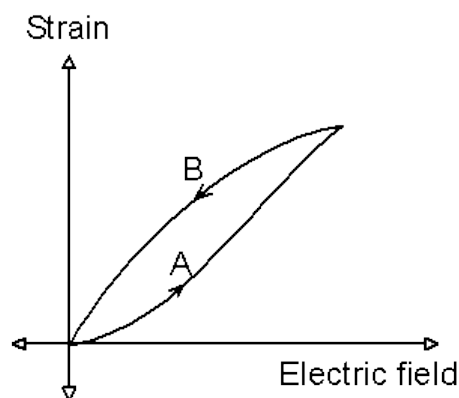
Οι **στοιβαγμένοι ενεργοποιητές πολλών στρωμάτων (stacked multilayer)** απ' την άλλη, είναι γραμμικοί ενεργοποιητές κολλημένοι ο ένας πάνω στον άλλο. Το πλεονέκτημά τους είναι ότι ενώ η εφαρμοζόμενη τάση είναι όση σε έναν απλό γραμμικό ενεργοποιητή, η παραμόρφωση είναι το άθροισμα για όλους τους στοιβαγμένους γραμμικούς. Παραδείγματος χάριν, αν ο στοιβαγμένος αποτελείται από 10 γραμμικούς, η παραμόρφωσή του θα είναι 10 φορές αυτή του ενός γραμμικού.

Τέλος, οι **εύκαμπτοι ενεργοποιητές (bender)**, όπως υπονοεί το όνομα τους, λυγίζουν αναλογικά με την τάση. Αποτελούνται από δύο στρώματα με αντίθετες φορές παραμόρφωσης. Στο Σχ. 1.11 φαίνεται ένα τέτοιο παράδειγμα. Η μπλε γραμμή διαχωρίζει τα δύο στρώματα και είναι στην πραγματικότητα το σημείο γείωσης. Όταν εφαρμόζεται τάση, εξ' αιτίας της αντίστροφης πολικότητας των δύο στρωμάτων, το ένα διαστέλλεται ενώ το άλλο συστέλλεται, με αποτέλεσμα ο ενεργοποιητής να λυγίζει.



Σχήμα 1. 11: Bender ενεργοποιητής

Η εκμετάλλευση των πιεζοηλεκτρικών υλικών προσφέρει πολλά πλεονεκτήματα, ωστόσο κανείς πρέπει να λαμβάνει υπόψη του και τα ιδιαίτερα φαινόμενα που εμφανίζουν – ειδικά αυτά της υστέρησης και του ερπυσμού. Όλα τα πιεζοηλεκτρικά υλικά εμφανίζουν μια μηχανική υστέρηση καθώς η παραμόρφωση δεν ακολουθεί ακριβώς την φόρτιση και αποφόρτιση. Εκφράζεται ως ποσοστό και αντιπροσωπεύει το αποτέλεσμα της διαίρεσης μεταξύ μέγιστης παραμόρφωσης και μέγιστης διαφοράς των δύο μονοπατιών (όπως φαίνεται στο παράδειγμα του Σχ. 1.12). Η υστέρηση εξαρτάται από το είδος του κεραμικού και μπορεί να κυμαίνεται μεταξύ 4% και 20%. Απ' την άλλη πλευρά, ο ερπυσμός μεταφράζεται ως η συνέχιση διαστολής του υλικού για ένα μικρό διάστημα αφού φορτιστεί. Αντίστοιχα, το υλικό δεν επιστρέφει στην αρχική του κατάσταση όταν αποφορτιστεί. Ο ερπυσμός μετριέται σε δευτερόλεπτα και αντιπροσωπεύει τον απαιτούμενο χρόνο για να φτάσει το υλικό στην τελική του κατάσταση.



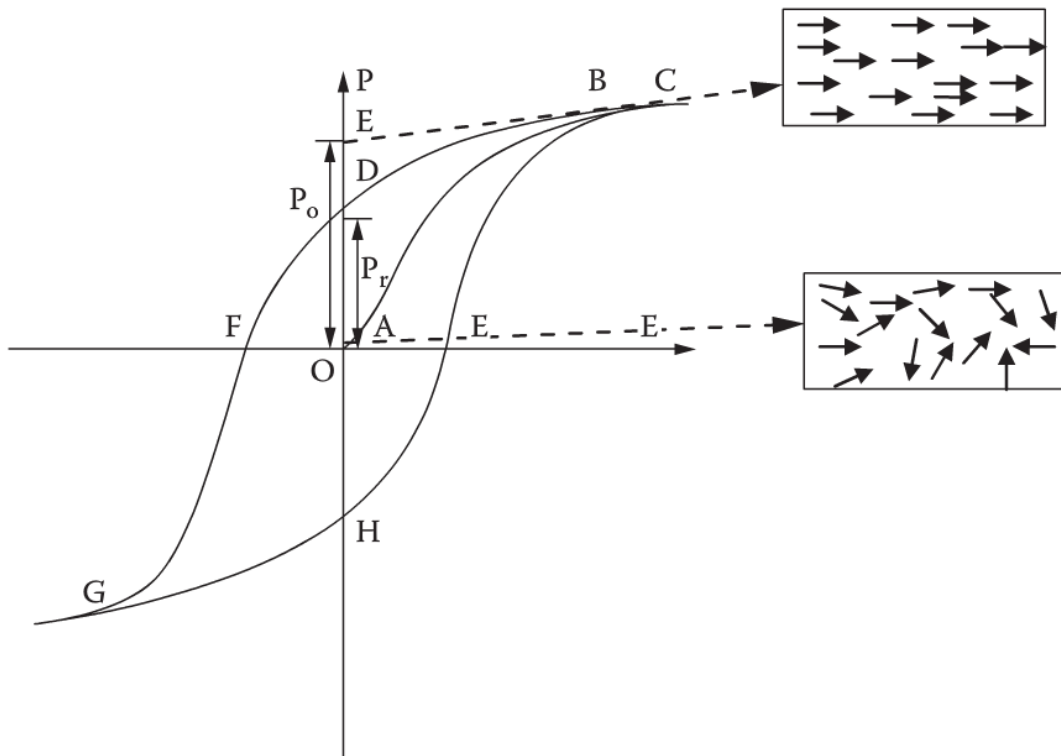
Σχήμα 1. 12: Τυπική γραφική παράσταση υστέρησης

Σε αυτό το σημείο κρίνεται σκόπιμο τα φαινόμενα αυτά να αναλυθούν διεξοδικά.  
[24][25][26]

### 1.5.1 Φαινόμενο υστέρησης

Η φύση των πιεζοηλεκτρικών υλικών είναι άρρηκτα συνδεδεμένη με την σημαντική ποσότητα ηλεκτρικών δίπολων στο εσωτερικό των υλικών. Το δίπολο είναι ένα διάνυσμα, συνεπώς έχει κατεύθυνση και τιμή σύμφωνα με τα φορτισμένα σωματίδια γύρω του. Όταν δίπολα είναι τοποθετημένα το ένα δίπλα στο άλλο τείνουν να έχουν την ίδια κατεύθυνση και συνολικά σχηματίζουν περιοχές με το όνομα “Weiss”. Οι περιοχές αυτές γενικά είναι τυχαία προσανατολισμένες αλλά μπορούν να ευθυγραμμιστούν με την προαναφερθείσα διαδικασία *poling*, κατά την οποία ένα ισχυρό ηλεκτρικό πεδίο εφαρμόζεται στο υλικό.

Το φαινόμενο της υστέρησης βήμα-βήμα είναι το εξής. Αρχικά, όταν δεν εφαρμόζεται τάση, οι φερροηλεκτρικές περιοχές είναι τυχαία προσανατολισμένες και η πόλωση είναι 0 (σημείο O). Καθώς το πεδίο αυξάνεται, οι περιοχές προσανατολίζονται στην κατεύθυνση του πεδίου και η πόλωση αυξάνεται γραμμικά αρχικώς (τμήμα OA). Καθώς το πεδίο αυξάνεται επιπλέον, όλο και περισσότερες περιοχές προσανατολίζονται, η καμπύλη γίνεται μη-γραμμική και τελικά όταν όλες οι περιοχές προσανατολίζονται, η πόλωση γίνεται μέγιστη (σημείο B). Η πόλωση σε αυτό το σημείο ( $P_0$  στο Σχ. 1.13) αντιπροσωπεύει το τμήμα OE και καλείται *πόλωση κορεσμού*. Απ’ αυτό το σημείο, αν το πεδίο μειωθεί σταδιακά, η πόλωση μειώνεται αλλά χωρίς να ακολουθεί την ίδια καμπύλη. Η μείωση της πόλωσης είναι αργή διαδικασία, που σημαίνει ότι καθυστερεί σε σχέση με το ηλεκτρικό πεδίο, κάτι που φαίνεται στο τμήμα BD. Όταν το πεδίο μηδενίζεται, παραμένει μια ποσότητα πόλωσης, το λεγόμενο *κατάλοιπο πόλωσης*, που φαίνεται στο τμήμα OE ( $P_R$  στο Σχ. 1.13). Για να εξαφανιστεί, ένα ηλεκτρικό πεδίο αντίθετης κατεύθυνσης πρέπει να εφαρμοστεί. Η πόλωση μηδενίζεται (σημείο F) όταν το πεδίο έχει τιμή  $-E_c$ , που καλείται *coercive field*. Αν το πεδίο συνεχίσει να αυξάνεται στην αντίθετη κατεύθυνση πέραν του  $E_c$ , οι περιοχές προσανατολίζονται στην κατεύθυνση του πεδίου και η πόλωση αυξάνει καθώς το πεδίο αυξάνει, στη νέα αντίθετη κατεύθυνση. Η πόλωση γίνεται μέγιστη (πόλωση κορεσμού  $-P_s$ ) στην αντίθετη κατεύθυνση στο σημείο G. Αν το πεδίο επιστρέψει στο 0, η καμπύλη ακολουθεί το μονοπάτι GH και θα υπάρχει κατάλοιπο πόλωσης  $-P_R$  (σημείο H). Αν το πεδίο αυξηθεί απ’ το 0 προς τα θετικά, το κατάλοιπο πόλωσης εξαφανίζεται όταν το πεδίο φτάσει την τιμή  $+E_c$ . Περαιτέρω αύξηση του πεδίου θα οδηγήσει στο μονοπάτι  $E_cB$ , κλείνοντας το βρόχο. Ο κλειστός βρόχος ονομάζεται *καμπύλη υστέρησης (hysteresis curve)*. [27][28]



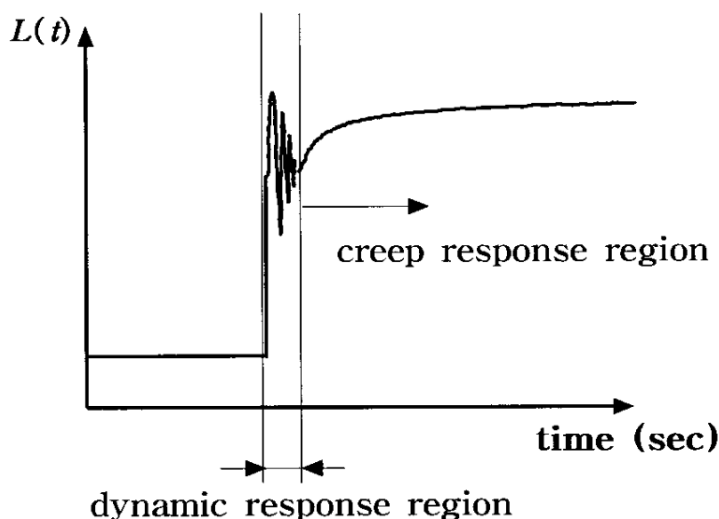
Σχήμα 1. 13: Πλήρης καμπύλη υστέρησης

### 1.5.2 Φαινόμενο ερπυσμού

Οι ίδιες ιδιότητες υλικών υπεύθυνες για την υστέρηση προκαλούν και τον ερπυσμό. Όπως αναφέρθηκε, ο ερπυσμός είναι η μεταβολή στη μετατόπιση χωρίς αντίστοιχη μεταβολή στην τάση. Αν αλλάξει η τάση λειτουργίας, το κατάλοιπο πόλωσης συνεχίζει να μεταβάλλεται, μεταφραζόμενο σε μεταβολή της παραμόρφωσης. Ο ρυθμός του ερπυσμού μειώνεται λογαριθμικά με το χρόνο. Το Σχ. 1.14 απεικονίζει την τυπική απόκριση σε βηματική διέγερση. Αρχικά, όταν εφαρμόζεται τάση, ο ενεργοποιητής ανταποκρίνεται με μια δυναμική μεταβατική συμπεριφορά εντός λίγων milliseconds ακολουθούμενη από τον ερπυσμό, που απεικονίζεται ως μια πιο αργή ολίσθηση σε σχέση με πριν. Η σχέση που περιγράφει τον ερπυσμό είναι η εξής:

$$L(t) = L_o \left[ 1 + \gamma \log_{10} \left( \frac{t}{0.1} \right) \right],$$

όπου  $L(t)$  είναι η μετατόπιση για δεδομένη διέγερση,  $L_0$  είναι μια ονομαστική σταθερά που αντιστοιχεί στην μετατόπιση 0.1 s μετά την εφαρμογή τάσης και  $\gamma$  είναι ο παράγοντας ερπυσμού που καθορίζει το ρυθμό του αλγορίθμου. [29][30]



Σχήμα 1. 14: Φαινόμενο ερπυσμού

### 1.5.3 Λοιπές έννοιες

Κατά τη σχεδίαση ενός συστήματος με πιεζοηλεκτρικούς ενεργοποιητές, η υστέρηση και ο ερπυσμός δεν είναι οι μόνες ιδιότητες που λαμβάνονται υπόψη. Αντιθέτως, πρέπει να ληφθούν υπόψη και ιδιότητες όπως το ρεύμα διαρροής, η εμπέδηση και η θερμοκρασία φερροηλεκτρικής μετάβασης (ferroelectric transition temperature).

Όσον αφορά το ρεύμα διαρροής (leakage current), όταν λειτουργεί κάτω απ' τη συχνότητα συντονισμού, ένας ενεργοποιητής λειτουργεί ως πυκνωτής. Η χωρητικότητά του εξαρτάται από την επιφάνεια και το πάχος του κεραμικού, καθώς και τις ιδιότητες του υλικού. Για τους στοιβαγμένους, εξαρτάται επίσης και από τον αριθμό των στρωμάτων. Το πραγματικό ηλεκτρικό ανάλογο ενός πιεζοηλεκτρικού ενεργοποιητή είναι η παράλληλη σύνδεση ενός πυκνωτή  $C_{act}$  και μιας αντίστασης  $R_{act}$ . Σύμφωνα με το νόμο του Ohm, αν  $V$  είναι η εφαρμοζόμενη τάση, το ρεύμα διαρροής ισούται με:

$$I_l = \frac{V}{R_{act}}$$

Η ηλεκτρική εμπέδηση (impedance) από την άλλη είναι ένα μέτρο αντίστασης στο ρεύμα που το κύκλωμα παρουσιάζει όταν εφαρμόζεται τάση σε αυτό. Παριστάνεται ως  $Z$  και είναι το μιγαδικό κλάσμα της τάσης με το ρεύμα. Η εμπέδηση εκφράζεται ως μιγαδικός αριθμός με την παρακάτω σχέση:

$$Z = R + jX,$$

όπου το πραγματικό μέρος,  $R$ , αντιπροσωπεύει την αντίσταση και το φανταστικό,  $X$ , την αντίδραση.

Τέλος, ένα σημαντικό στοιχείο είναι η θερμοκρασία φερροηλεκτρικής μετάβασης (ferroelectric transition temperature). Αυτή αντιπροσωπεύει την θερμοκρασία πέρα απ' την οποία τα φερροηλεκτρικά υλικά χάνουν την στιγμιαία πόλωσή τους. [31][32][33]

## ***1.6 Εισαγωγή στο LabVIEW***

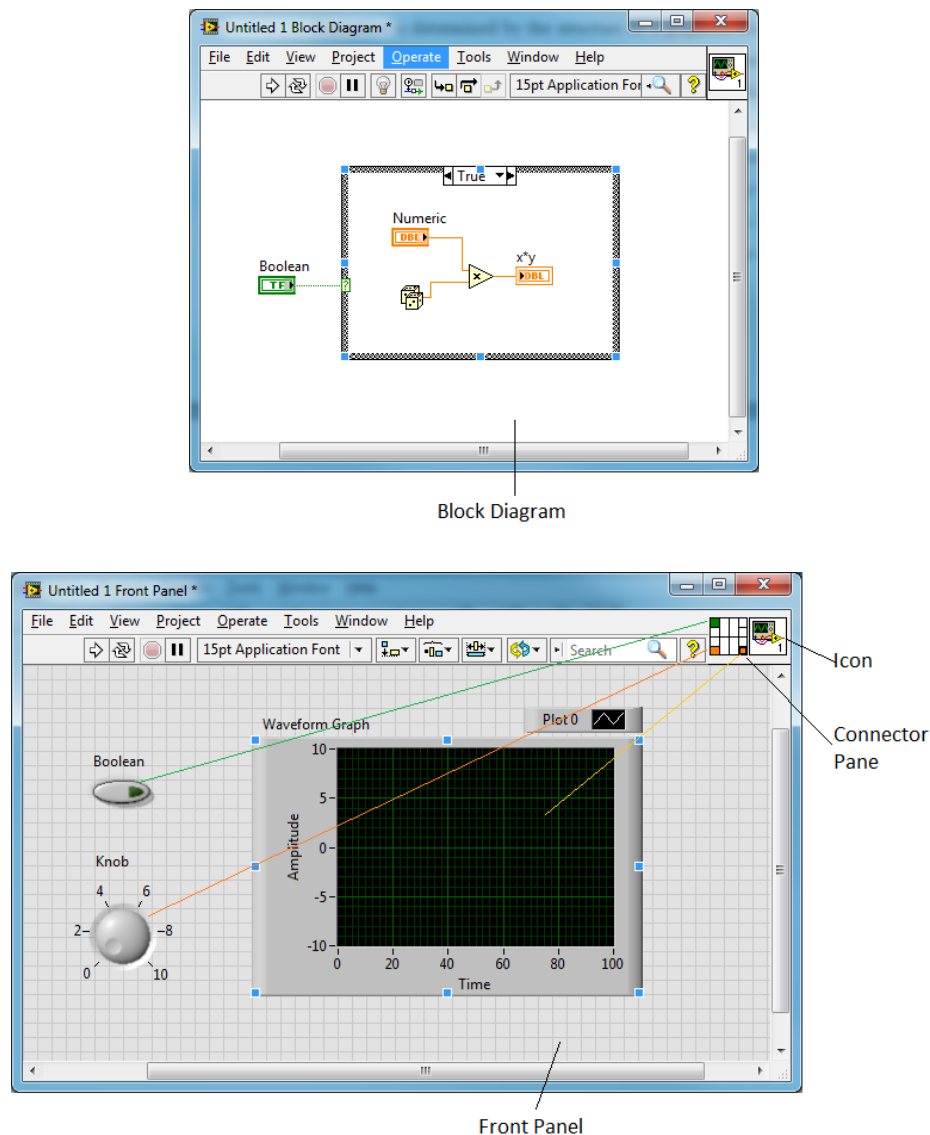
Το LabVIEW είναι η συντομογραφία του “Laboratory Virtual Engineering Workbench” και είναι μια πλατφόρμα ανεπτυγμένη από την National Instruments. Το LabVIEW είναι από τα πιο διαδεδομένα περιβάλλοντα οπτικού προγραμματισμού. Η γλώσσα ονομάζεται “G” και είναι μια dataflow γλώσσα προγραμματισμού.

Το LabVIEW χρονολογείται από τη δεκαετία του 1980 καθώς αρχικά κυκλοφόρησε για Apple Macintosh το 1986. Έκτοτε, έχει εξελιχθεί και προσαρμοστεί στις σύγχρονες ανάγκες και πλέον προσφέρεται σε Windows, Linux ή Mac OS X, με την τρέχουσα έκδοση να είναι η 13.

Αυτό που διαφοροποιεί τις γλώσσες οπτικού προγραμματισμού είναι ότι ο χρήστης δημιουργεί προγράμματα χειριζόμενος σχήματα αντί για κώδικα. Η εκτέλεση καθορίζεται από τη δομή ενός γραφικού μπλοκ διαγράμματος, του πηγαίου κώδικα LabVIEW. Το πέρασμα δεδομένων μέσω κόμβων καθορίζει τη σειρά εκτέλεσης του προγράμματος. Αυτό σημαίνει ότι ένας κόμβος θα δρομολογηθεί για εκτέλεση μόνο όταν όλες οι είσοδοι του περιέχουν έγκυρα δεδομένα. Ο κόμβος εκτελείται όποτε ένας επεξεργαστής είναι διαθέσιμος. Όταν εκτελεστεί ο κόμβος, οι τιμές στις εξόδους του γίνονται διαθέσιμες και ρέουν προς τους κόμβους στους οποίους είναι συνδεδεμένες. Ο προγραμματιστής μπορεί να ελέγξει τη σειρά εκτέλεσης συνδέοντας σε αλληλουχία κόμβους με συμβατές εισόδους και εξόδους. Η γλώσσα G επιτρέπει

εγγενώς την παράλληλη επεξεργασία και εκμεταλλεύεται αυτόματα το multi-processing και multi-threading hardware με τον built-in scheduler.

Τα βασικά δομικά στοιχεία των προγραμμάτων σε LabVIEW καλούνται VIs<sup>5</sup>. Μπορούν να παρομοιαστούν με τα functions ή subroutines άλλων γλωσσών προγραμματισμού. Ένα VI περιλαμβάνει ένα Front Panel, που περιέχει όλα τα εργαλεία ελέγχου και δείκτες, ένα Block Diagram, το οποίο είναι το πλάνο των κόμβων και της μεταξύ τους καλωδίωσης, το εικονίδιο του VI και το Connector Pane, που καθορίζει τα τερματικά σύνδεσης εισόδου και εξόδου του VI. Ένα τυπικό παράδειγμα φαίνεται στο Σχ. 1.15. [34][35][36][37][38]



Σχήμα 1. 15: Τυπικό παράδειγμα VI

<sup>5</sup> Το VI είναι η συντομογραφία του Virtual Instrument





# 2

## *Διάταξη & Υλικό*

Σκοπός του κεφαλαίου αυτού είναι να εισάγει τους αναγνώστες στην πειραματική διάταξη που αναπτύχθηκε καθώς και το υλικό που χρησιμοποιήθηκε.

Το κεφάλαιο ξεκινά με μια αναφορά στο σκοπό της διάταξης, καθώς και μια σύντομη περιγραφή της και ακολουθείται από το state of the art ανάλογων διατάξεων μετρήσεων πιεζοηλεκτρικών υλικών. Η επόμενη ενότητα είναι αφιερωμένη στις τεχνικές προδιαγραφές της διάταξης και του υλικού. Τέλος, οι λειτουργίες της διάταξης περιγράφονται με λεπτομέρεια.

### *2.1 Εισαγωγή στην πειραματική διάταξη*

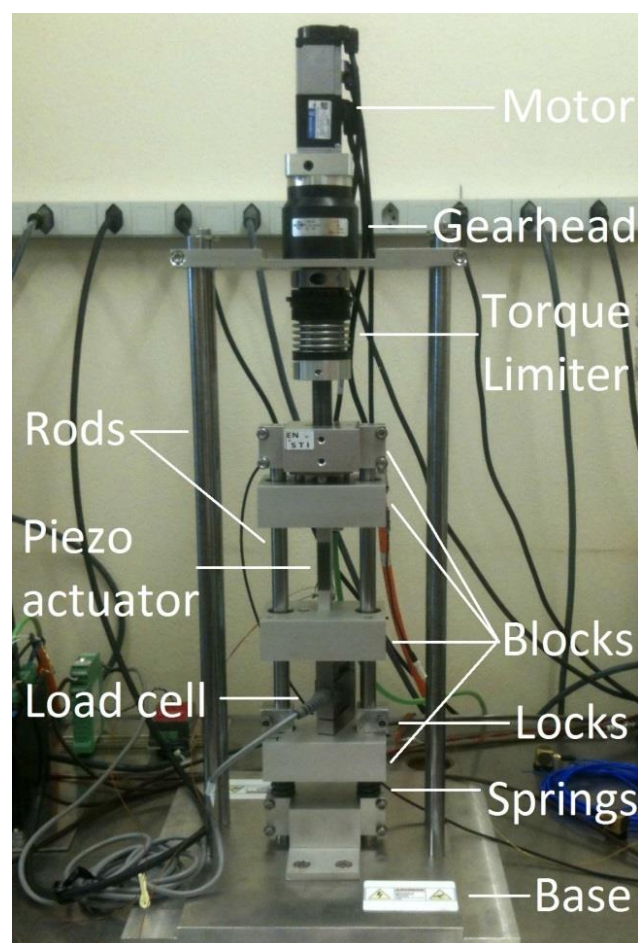
Όπως αναφέρθηκε στο πρώτο κεφάλαιο, η ανάπτυξη του προτεινόμενου στο UA9 Project γωνιομέτρου απαιτεί την ανάπτυξη ενός συστήματος που μπορεί να παρέχει ανάλυση των χαρακτηριστικών πιεζοηλεκτρικών υλικών.

Γι αυτό το σκοπό, η ανάπτυξη ενός εξειδικευμένου συστήματος προσαρμοσμένου στις απαιτήσεις αλλά και ευπροσάρμοστου σε μελλοντικές επεκτάσεις κρίθηκε απαραίτητη.

Περιληπτικά, το σύστημα αποτελείται από 5 μέρη: τη *μηχανική διάταξη*, την *διάταξη του κινητήρα*, τους *αισθητήρες* και *περιφερειακές συσκευές*, το *PXI chassis* και

*modules* και το ανεπτυγμένο σε LabVIEW λογισμικό. Το τελευταίο κομμάτι καλύπτεται εκτενώς στο 3<sup>ο</sup> κεφάλαιο, ενώ τα πρώτα τέσσερα στο παρόν.

Συνοπτικά, το κεντρικό μέρος της διάταξης αποτελείται από τη μηχανική διάταξη που αποτελείται από τη βάση, τις ράβδους, τα χαμηλής-τριβής μπλοκ, τα φράγματα και τα ελατήρια (base, rods, low-friction blocks, locks και springs), μέρος της διάταξης κινητήρα (σερβοκινητήρας, gearhead και περιοριστής ροπής) καθώς και μερικούς τοποθετημένους αισθητήρες (load cell, capacitive sensor). Η τρέχουσα διάταξη φαίνεται στο Σχ. 2.1



Σχήμα 2. 1: Κεντρικό μέρος της διάταξης

Οι περιφερειακές συσκευές αποτελούνται από το National Instruments PXI chassis και modules, που διασυνδέουν το λογισμικό με τους αισθητήρες, την διέγερση του πιεζοηλεκτρικού στοιχείου και την λειτουργία του κινητήρα, τον ενισχυτή του σερβοκινητήρα, που οδηγεί τον κινητήρα και μεταφράζει τις εντολές του PXI σε πραγματική κίνηση, τον γραμμικό ενισχυτή για την διέγερση του πιεζοηλεκτρικού

στοιχείου, που μετατρέπει ένα χαμηλό σήμα σε σήμα κατάλληλο για την διέγερση και τους υπόλοιπους αισθητήρες και περιφερειακές συσκευές (το capacitive measurement system, που μεταφράζει τις μετρήσεις του capacitive sensor σε τάση, την πλακέτα για τη μέτρηση του ρεύματος, τον ενισχυτή του load cell, τον αισθητήρα θερμοκρασίας και τα επιταχυνσιόμετρα). Μέρος αυτών φαίνονται στο Σχ. 2.2.



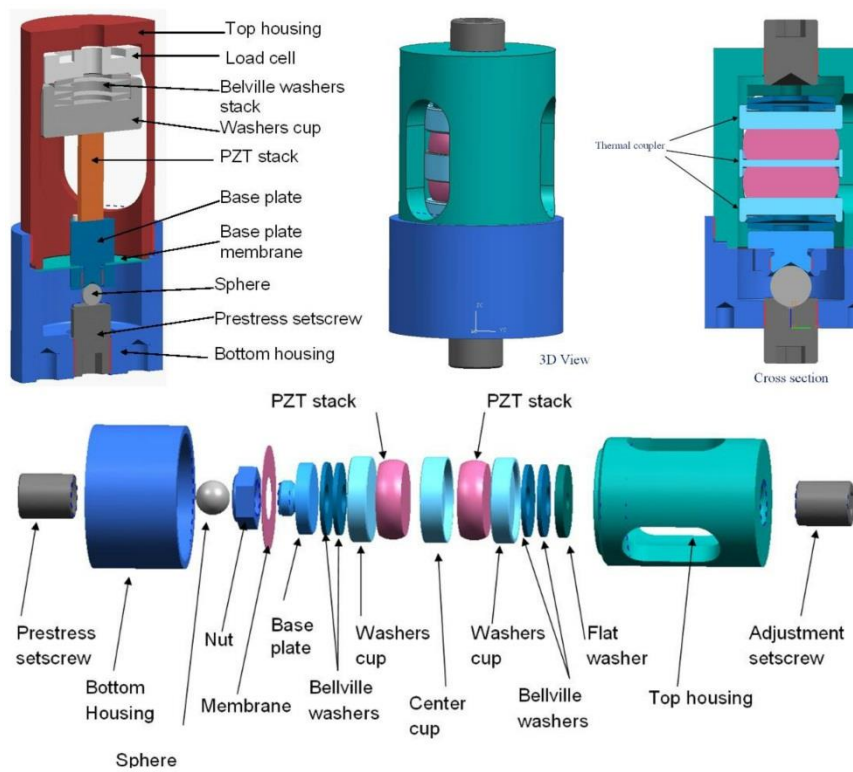
Σχήμα 2. 2: Περιφερειακές συσκευές- National Instruments PXI chassis και modules (κάτω-αριστερά), ενισχυτής σερβοκινητήρα και τροφοδοσία του μηχανικού φρένου (πάνω-αριστερά) και γραμμικοί ενισχυτές για διέγερση του πιεζοηλεκτρικού στοιχείου και μετρήσεις του capacitive sensor(right)

## 2.2 State of the art

Κατά τις περασμένες δεκαετίες, η ανάγκη μελέτης των πιεζοηλεκτρικών υλικών οδήγησε αναπόφευκτα στην ανάπτυξη αντίστοιχων μετρητικών διατάξεων.

Οι πιο συνηθισμένες διατάξεις επιτρέπουν ένα σταθερό φορτίο να ασκηθεί στο πιεζοηλεκτρικό στοιχείο. Κατ' αυτό τον τρόπο, οι επιστήμονες μπορούν να μελετήσουν τη συμπεριφορά των στοιχείων σε συγκεκριμένες στάθμες φορτίου κάτω από μεταβαλλόμενες συνθήκες ηλεκτρικής διέγερσης. Αυτές οι διατάξεις

ονομάζονται “pre-stress” και βρίσκονται στα "PZT uniaxial stress dependence: experimental results" των Audigier et al. [39], "Characterization of multilayer piezoelectric actuators for use in active isolation mounts" των Wise [40] και "Characteristics comparison of piezoelectric actuators at low electric field: analysis of strain and blocking force" των Mekid και Lim [41] μεταξύ άλλων. Η πίεση συνήθως ασκείται χειροκίνητα με ένα μπουλόνι, μια μέγγενη ή άλλο παρόμοιο μηχανισμό. Δύο αξιοσημείωτα παραδείγματα όπου χρησιμοποιείται μπουλόνι βρίσκονται στο "Extended Life PZT Stack Test Fixture" των Badescu et al. [42] και φαίνονται στο Σχ. 2.3.



Σχήμα 2.3: Διάταξη για μονή στοιβία PZT (πάνω-αριστερά) και διάταξη για δύο κυλινδρικές PZT στοιβίες (πάνω-δεξιά και κάτω)

Μια εξέλιξη αυτής της ιδέας είναι η χρήση κινητήρα για την εφαρμογή της πίεσης. Μια τέτοια προσέγγιση φαίνεται στο "Influence of uniaxial compressive stress on the ferroelectric behaviors of piezoelectric stack actuators" των Yang Gang et al. [43].

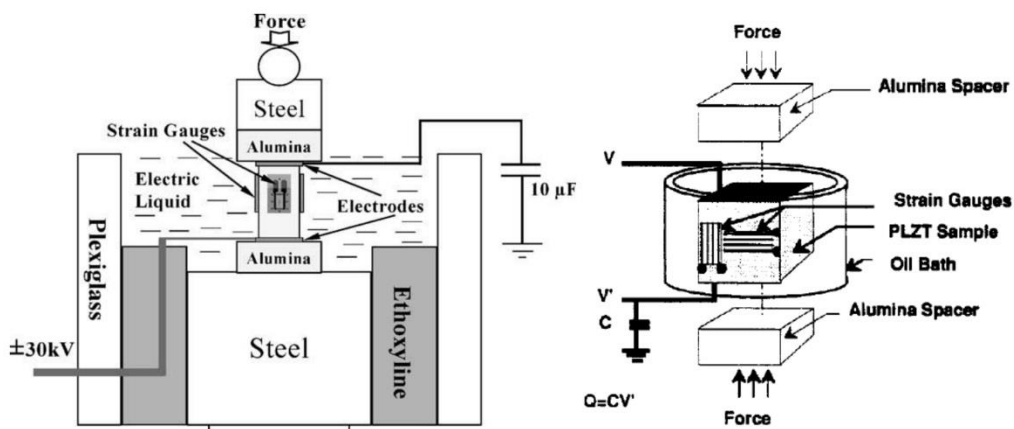
Μια διαφορετική προσέγγιση στην άσκηση πίεσης είναι η χρήση εξειδικευμένων σερβουδραυλικών διατάξεων. Τέτοια παραδείγματα υπάρχουν στα "Response of piezoelectric stack actuators under combined electro-mechanical loading" των

Mitrovic et al. [44] (Fig. 2.4.) και “*Uniaxial stress dependence of ferroelectric properties of  $x$ PMN-(1- $x$ )PZT ceramic systems*” των Yimnirun et al. [45].



Σχήμα 2. 4: Instron 8516 σεραυδοδραυλικό σύστημα

Τέλος, μια πιο εξεζητημένη προσέγγιση στο παραπάνω ζήτημα περιλαμβάνει την εμφύθιση του πιεζοηλεκτρικού στοιχείου σε κατάλληλο υγρό για να αποτραπεί η πιθανότητα ηλεκτρικού τόξου υψηλής τάσης κατά την διέγερση. Στο “*Effects of uniaxial prestress on the ferroelectric hysteretic response of soft PZT*” των Zhou et al. [46] χρησιμοποιήθηκε fluorinert electric liquid (FC-40, 3M) (Σχ. 2.5α), ενώ στο “*The effect of uniaxial stress on the electro-mechanical response of 8/65/35 PLZT*” του Lynch [47] χρησιμοποιήθηκε silicone oil (Σχ. 2.5β).



Σχήμα 2. 5α (αριστερά) και Σχήμα 2. 5β (δεξιά)

## 2.3 Τεχνικές Προδιαγραφές

### 2.3.1 Μηχανική διάταξη

Ξεκινώντας απ' την μηχανολογική πλευρά της διάταξης, όλα τα μεταλλικά μέρη είναι κατασκευασμένα από κράμα ανοξείδωτο ατσάλι 316 και ανοδιωμένο αλουμίνιο. Αυτά είναι η **βάση (base)**, οι **ράβδοι (rods)**, τα **μπλοκ (blocks)**, τα **φράγματα (locks)**, καθώς και οποιοδήποτε άλλο μεταλλικό μέρος της διάταξης. Το ανοξείδωτο ατσάλι είναι γενικά ανθεκτικό στα οξέα όμως συγκεκριμένα το 316 προσφέρει μεγαλύτερη άμυνα απέναντι στα σουλφουρικά, φωσφορικά και λιπαρά οξέα, αλδεΐδες, αμίνες και λίπη, καθώς και καλύτερη αντίσταση στη οξείδωση. Το ανοδιωμένο αλουμίνιο παρουσιάζει καλύτερη αντίσταση στη διάβρωση σε σχέση με το απλό. [48][49]

Τα **ελατήρια (springs)** είναι φτιαγμένα από *σίδηρο* με σταθερά  $k = 78 \text{ N/mm}$ . Στη διάταξη μας έχουμε δύο ελατήρια τοποθετημένα παράλληλα που σημαίνει πως η ισοδύναμη σταθερά ελατηρίου του συστήματος είναι  $k_{eq} = 2 \cdot k = 156 \text{ N/mm}$ .<sup>6</sup>

### 2.3.2 Διάταξη κινητήρα

Ο κινητήρας, το gearhead, ο περιοριστής ροπής και ο οδηγός του κινητήρα συνθέτουν την διάταξη κινητήρα. Ο **σερβοκινητήρας** είναι το μοντέλο *SANMOTION R R2AA04010FCP00M6* της *Sanyo Denki*. Πρόκειται για ένα 200 V AC σερβοκινητήρα στα 100 W με φλάντζα μεγέθους 40 mm<sup>2</sup>. Έχει 3000 RPM ονομαστική ταχύτητα, 6000 RPM μέγιστη ταχύτητα περιστροφής, 0.318 Nm ονομαστική και συνεχή stall ροπή και 1.18 Nm μέγιστη stall ροπή καθώς και 24 V DC φρένο, battery backup method absolute encoder, UL/CE έγκριση και reduction ratio ίσο με 1/3.

Ο κινητήρας οδηγείται από τον **ενισχυτή σερβοκινητήρα** της *Sanyo Denki*, μοντέλο *RS1A01AA*. Πρόκειται για ένα single-axis analog/pulse input type servo ενισχυτή, τροφοδοτούμενο από AC200 με 230 V, rated στα 15 A. Έχει wire-saving incremental encoder και battery backup method absolute encoder και NPN selectable output.

Ο σερβοκινητήρας και ο ενισχυτής του φαίνονται στο Σχ. 2.6.

---

<sup>6</sup> Γενικά, ένα σύστημα με 2 ελατήρια παράλληλα, με σταθερές  $k_1$  και  $k_2$  έχει ισοδύναμη σταθερά ελατηρίου  $k_{eq} = k_1 + k_2$ . [50]



Σχήμα 2. 6: Ενισχυτής σερβοκινητήρα (αριστερά) και σερβοκινητήρας (δεξιά)

Ο άξονας του κινητήρα συνδέεται στο **gearhead** ή **gearbox**. Η ροπή εξόδου του κινητήρα είναι γενικά χαμηλή και γι' αυτό το λόγο ένα gearhead χρησιμοποιείται για να αυξηθεί η ροπή εξόδου και να μειωθούν τα RPM. Το συγκεκριμένο gearhead είναι της *Neugart*, μοντέλο *PLPE 70* (Σχ. 2.7) με αναλογία  $i = 64$ <sup>7</sup> και χαμηλό backlash.<sup>8</sup> Σε αυτή την περίπτωση σημαίνει ότι η μέγιστη ταχύτητα των 6000 RPM του κινητήρα περιορίζεται σε  $n_{out} = 6000/64 = 93.75 \text{ RPM}$ , αλλά ταυτόχρονα η ροπή εξόδου, που είναι απαραίτητη στην άσκηση πίεσης στο πιεζοηλεκτρικό στοιχείο, αυξάνεται σημαντικά. Συγκεκριμένα, η ονομαστική ροπή γίνεται  $\tau = \tau_{motor} \cdot i = 0.318 \cdot 64 = 20.352 \text{ Nm}$ .



Σχήμα 2. 7: Neugart PLPE 70 μπροστινή όψη (αριστερά), πίσω (μέση) και εσωτερική (δεξιά)

<sup>7</sup> Η αναλογία ορίζεται ως  $i = n_{in}/n_{out}$ , όπου  $n$  είναι η ταχύτητα σε RPM

<sup>8</sup> Το backlash ορίζεται ως η χαμένη κίνηση σε ένα μηχανισμό που οφείλεται σε κενά μεταξύ των μερών του [51]

Το τελευταίο κομμάτι της διάταξης κινητήρα, ο **περιοριστής ροπής (torque limiter)**, γνωστός και ως **overload clutch**, είναι μια συσκευή που προστατεύει αυτόματα οτιδήποτε είναι συνδεδεμένο σε αυτή από ζημιά από μηχανική υπερφόρτωση αποσυνδέοντας τον οδηγό από το φορτίο. [52] Ο συγκεκριμένος περιοριστής ροπής είναι της *R+W*, μοντέλο *SK2/0030/087/W* (Σχ. 2.8). Προσφέρει προστασία υπερφόρτωσης υψηλής ακρίβειας, είναι απόλυτα backlash-free και επιτυγχάνει αποσύνδεση μέσα σε 1-3 milliseconds.



Σχήμα 2. 8: R+W SK2

### 2.3.3 Αισθητήρες και περιφερειακές συσκευές

Πολύ σημαντικές στη διαδικασία χαρακτηρισμού είναι οι μετρήσεις του χωρητικού αισθητήρα (capacitive sensor). Για να ληφθούν, πέραν του αισθητήρα απαραίτητο είναι και το capacitive measurement module.

Ο **capacitive sensor** είναι κατασκευασμένος από την *Fogale* και είναι ο *MCC-10*. Το εύρος μέτρησης είναι 0 με 1.0 mm με ανάλυση  $0.4 \text{ nm RMS/Hz}^{1/2}$ . Έχει τεχνολογία metal-resin, είναι πολύ ανθεκτικός και αντέχει θερμοκρασίες μέχρι 200 °C.

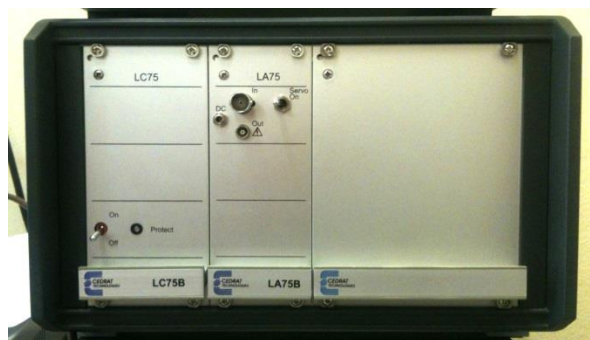
Τα **capacitive measurement modules** είναι επίσης της *Fogale*, μοντέλα *MC900*. Υπάρχουν 3 ίδια modules στεγασμένα σε δύο ξεχωριστά 19" racks, καθένα με την τροφοδοσία του. Το MC900 έχει εύρος εξόδου 0-10 V, που συνυπολογίζοντας το εύρος 0-1.0 mm του capacitive sensor, μεταφράζεται σε  $100 \mu\text{m}/\text{V}$ .





Σχήμα 2. 9: Fogale MCC-10 συνδεδεμένο στο MC900 (αριστερά) και διαθέσιμα modules (δεξιά)

Οι πιεζοηλεκτρικοί ενεργοποιητές γενικά αντέχουν υψηλά επίπεδα τάσης – π.χ. η εφαρμογή 150 V δεν είναι ασυνήθιστη και για την ακρίβεια είναι απαραίτητη ώστε να επιτευχθεί η μέγιστη παραμόρφωση. Γι αυτό το σκοπό, είναι αναγκαία η χρήση ενός **ενισχυτή τάσης**, απαραίτητα γραμμικού. Εν προκειμένω, ο ενισχυτής που διεγείρει τα πιεζοηλεκτρικά στοιχεία είναι ο *Cedrat LA75B*, μαζί με το module τροφοδοσίας του, *LC75B* (και τα δύο φαίνονται στο Σχ. 2.10). Συγκεκριμένα, ο LA75B είναι σχεδιασμένος να οδηγεί χωρητικά φορτία όπως πιεζοηλεκτρικούς ενεργοποιητές με εξαιρετικά χαμηλό θόρυβο. Το εύρος εξόδου του είναι -20/150 V, και βρίσκεται σε ένα rack μαζί με τον AC/DC μετατροπέα του LC75B. Το LC75B AC/DC module είναι σχεδιασμένο να παράγει σταθεροποιημένες DC τάσεις, απαραίτητες για την λειτουργία του ενισχυτή. Το μέγιστο ρεύμα του LA75B είναι 360 mA. Το κέρδος τάσης του ενισχυτή είναι 20 και αυτό σημαίνει πως αν η είσοδος στον ενισχυτή είναι 1 V, τότε ο ενισχυτής θα οδηγήσει το πιεζοηλεκτρικό στοιχείο με 20 V.



Σχήμα 2. 10: Cedrat LA75B και LC75B

Όπως στην περίπτωση του capacitive sensor, έτσι και για να λάβουμε μετρήσεις από το load cell πέραν του ιδίου, απαραίτητο είναι και το load cell measurement module.

Υπάρχουν δύο διαφορετικά **load cells** αναλόγως τις ανάγκες. Το πρώτο, Althen *ALF256DBR0KN* αντέχει 5 kN, ενώ το άλλο, *Tedea Huntleigh Model No. 615*, αντέχει 300 kg. Το Althen μπορεί να χρησιμοποιηθεί μόνο σε συνθήκες συμπίεσης, ενώ το Tedea τόσο σε συμπίεση όσο και σε τέντωμα.

Το **load cell measurement module** σε αυτή την περίπτωση είναι το *Althen SG-KP-12E-420*. Συνδέεται στο load cell και προσφέρει αναλογική έξοδο μεταξύ 0 και 10 V. Έτσι, στην περίπτωση του Althen load cell στα 5 kN, η αναλογία είναι  $500 \text{ N/V}$ . Στην περίπτωση του Tedea Huntleigh load cell στα 300 kg, η αναλογία διαμορφώνεται στα  $30 \text{ kg/V} \approx 294.2 \text{ N/V}$ .



Σχήμα 2. 11: Althen SG-KP-12E-420 (αριστερά), Tedea Huntleigh 615 (μέση) και Althen ALF256 (δεξιά)

Υπάρχουν δύο πανομοιότυπα **επιταχυνσιόμετρα (accelerometers)** για την μέτρηση κραδασμών στο περιβάλλον μέτρησης. Είναι κατασκευασμένα από την *PCB Piezotronics*, μοντέλο *356B18*. Είναι τριαξονικά, υψηλής ευαισθησίας, ceramic shear ICP® επιταχυνσιόμετρα, με ευαισθησία  $1000 \text{ mV/g}$  ή  $102 \text{ mV}/(\text{m/s}^2)$  στο σύστημα SI, 0.5 ως 3k Hz φάσμα συχνοτήτων και 4-pin σύνδεση.



Σχήμα 2. 12: PCB Piezotronics 356B18 επιταχυνσιόμετρο

Το **ρεύμα (current)** μετριέται με μια πλακέτα που αναπτύχθηκε βασισμένη στο ολοκληρωμένο ACPL-C87B. Το ACPL-C87B προτιμάται όταν απαιτείται υψηλή ακρίβεια και επιδεικνύει προηγμένη sigma-delta modulation τεχνολογία και  $\pm 0.5\%$  High Gain ακρίβεια. Έχει 0 ως 2 V ονομαστικό εύρος εισόδου και 1 GOhm εμπέδηση εισόδου. Η πλακέτα συνδέεται στον αριστερό ακροδέκτη του πιεζοηλεκτρικού ενεργοποιητή και παρέχει απευθείας μετατροπή του ρεύματος σε τάση. Έτσι αν το ρεύμα στην είσοδο της πλακέτας είναι 1 mA, τότε η έξοδος της θα είναι 1 mV.

Τέλος, για την μέτρηση της **θερμοκρασίας** χρησιμοποιείται ένα *PT100* resistance temperature detector (RTD). Η μέτρηση της θερμοκρασίας με ένα RTD επιτυγχάνεται συσχετίζοντας την αντίσταση του RTD στοιχείου με τη θερμοκρασία. Το PT100 είναι απ' τα πλέον διαδεδομένα RTDs και είναι φτιαγμένο από καθαρή πλατίνα (εξ ου και το πρόθεμα 'PT') και έχει ονομαστική αντίσταση 100 Ohms στους 0 °C (εξ ου και η κατάληξη '100'). Όπως όλα τα RTDs, το PT100 έχει προβλέψιμη μεταβολή στην αντίστασή του καθώς αλλάζει η θερμοκρασία και ακριβώς αυτή η ιδιότητά του χρησιμοποιείται για τον καθορισμό της θερμοκρασίας. Η ευαισθησία ενός κοινού PT100 είναι 0.385 Ohm/°C. [53]

#### **2.3.4 Σύστημα PXI**

Η τελευταία και πιο σημαντική περιφερειακή συσκευή είναι το σύστημα **PXI** της National Instruments. Το PXI προέρχεται από το PCI eXtensions for Instrumentation και είναι μια modular instrumentation πλατφόρμα που εισήχθη το 1997 από την National Instruments (NI). Χρησιμοποιείται ως βάση για πειραματικές διατάξεις, συστήματα αυτοματισμού και εργαστηριακά όργανα μεταξύ άλλων και το γεγονός ότι είναι modular επιτρέπει μεγάλη ευελιξία. Πάνω από 1150 τύποι modules από διάφορες εταιρείες είναι διαθέσιμοι και μπορούν να ενσωματωθούν σε ένα PXI chassis. Για την εξασφάλιση διαλειτουργικότητας και συμμόρφωσης με τα πρότυπα, το PXI χρησιμοποιεί τεχνολογία βασισμένη στο PCI και το βιομηχανικό πρότυπο της PXI Systems Alliance (PXISA). Κάπου το 2004, η PXI Systems Alliance ενέκρινε την προδιαγραφή PXI-5, που καθορίζει πως μπορεί να ενσωματωθεί το PCI Express στο PXI διατηρώντας την συμβατότητα προς τα πίσω. Αυτό οδήγησε στην εισαγωγή των PXI Express hybrid slots, που υποστηρίζουν τόσο PCI όσο και PCI Express. [54]

Εν προκειμένω, το σύστημα PXI αποτελείται από ένα PXI-ready chassis, ένα κεντρικό controller και τα extension modules. Το chassis είναι η ραχοκοκαλιά του PXI συστήματος αφού προσφέρει την τροφοδοσία, ψύξη και διαύλους επικοινωνίας για PCI και PCI Express στον κεντρικό controller και στα διάφορα extension modules. Ο κεντρικός controller είναι ένας υψηλής απόδοσης embedded controller που εξαλείφει την ανάγκη για εξωτερικό PC, καθώς ένα πλήρες σύστημα περιλαμβάνεται στο PXI chassis. Οι ενσωματωμένοι controllers έχουν standard χαρακτηριστικά όπως integrated CPU, σκληρό δίσκο, μνήμη, Ethernet, video, serial, USB, και άλλες περιφερειακές συνδέσεις. Τέλος, extension modules μπορούν να προστεθούν σε ένα σύστημα για να αυξηθεί η λειτουργικότητα του παρέχοντας π.χ. εισόδους και εξόδους για διασύνδεση με αισθητήρες. [55]

Το **chassis** στην προκειμένη περίπτωση είναι το *NI PXIe-1078*. Πρόκειται για 9-slot, 3U PXI Express chassis όπου στο πρώτο slot τοποθετείται ο κεντρικός controller και τα υπόλοιπα είναι διαθέσιμα για τα extension modules. Συγκεκριμένα, προσφέρει 5 hybrid slots (slots 5-9) και 3 PXI Express slots (slots 2-4). Περιλαμβάνει built-in timing και synchronization features συμπεριλαμβανομένων 10 και 100 MHz reference clocks καθώς και PXI trigger bus. Στο Σχ. 2.13 φαίνεται τόσο γεμάτο όσο και άδειο.



Σχήμα 2. 13: Γεμάτο και άδειο PXIe-1078 chassis

Ο **κεντρικός controller** που βρίσκεται στο πρώτο slot είναι ο *NI PXIe-8133*. Είναι εξοπλισμένος με έναν quad-core Intel Core i7-820QM που λειτουργεί στα 1.73 GHz και 3.06 GHz σε single-core Turbo Boost mode, 2GB DDR3 1333MHz μνήμης ram και 200GB σκληρό δίσκο. Επίσης έχει δύο θύρες 10/100/1000BASE-TX (Gigabit) Ethernet, 4 Hi-Speed USB, ExpressCard/34, GPIB, serial, DVI και external trigger.

Έχει δύο λειτουργικά εγκατεστημένα - Microsoft Windows XP και NI LabVIEW Real-Time. Η διαφορά είναι ότι το NI LabVIEW Real-Time είναι real-time λειτουργικό σύστημα, που σημαίνει πως είναι σχεδιασμένο να τρέχει εφαρμογές με πολύ ακριβές timing και μεγάλη αξιοπιστία. Αυτό είναι πολύ σημαντικό σε ένα σύστημα μετρήσεων όπως το δικό μας και η αναγκαιότητά του θα φανερωθεί στο επόμενο κεφάλαιο. Λειτουργικά συστήματα όπως τα Windows είναι σχεδιασμένα για να ανταποκρίνονται στον χρήστη ακόμα και όταν τρέχουν πολλά προγράμματα και υπηρεσίες (διασφαλίζοντας ισότητα), ενώ real-time λειτουργικά είναι σχεδιασμένα για να τρέχουν κρίσιμες εφαρμογές αξιόπιστα και με ακριβές timing (δίνοντας βάση στις προτεραιότητες του προγραμματιστή). Το NI Real-Time Hypervisor χρησιμοποιεί virtualization τεχνολογία για να τρέχει το NI LabVIEW Real-Time παράλληλα με τα Windows XP, αξιοποιώντας το real-time processing με το γραφικό περιβάλλον και τις υπηρεσίες των Windows XP. [56][57] Το PXIe-8133 φαίνεται στο Σχ. 2.14.



Σχήμα 2. 14: NI PXIe-8133

Τέλος, τα **extension modules** συνοψίζονται στον ακόλουθο πίνακα:

Αριθμός Slot	Module	Χρήση
2	PXIe-4357 + TB-4357	PT100 temperature sensor
3	PXIe-4496	Accelerometer sensors
5	PXI-4461	Load Cell and current sensors
7	PXIe-4330 + TB-4330	Strain gage
8	PXI-7841R + SCB-68	FPGA module for motor control
9	PXI-4461	Capacitive sensor, piezo excitation signal (generation και feedback acquisition)

- Το *PXIe-4357* input module προσφέρει ενσωματωμένη λήψη δεδομένων και signal conditioning για μετρήσεις με RTD. Προσφέρει 20 κανάλια και έχει πέντε 24-bit delta-sigma analog-to-digital converters με anti-aliasing και low-pass φίλτρα για μεγαλύτερη ακρίβεια, καθώς και ρυθμούς δειγματοληψίας μέχρι 100 S/s ανά κανάλι σε high-speed mode ή 1 S/s με τυπική ακρίβεια μετρήσεων 0.09 °C σε high-resolution mode. Υποστηρίζει 2-, 3- ή 4-wire RTDs. Το επιπρόσθετο *TB-4357* είναι ένα front mount terminal block με συνδεσιμότητα μέσω screw terminal για το NI *PXIe-4357*. Ο συνδυασμός τους φαίνεται στο Σχ. 2.15 και χρησιμοποιείται για τις μετρήσεις από τον PT100 αισθητήρα θερμοκρασίας.



Σχήμα 2. 15: *PXIe-4357* (πάνω-αριστερά), *TB-4357* (πάνω-δεξιά), *TB-4357* συνδεδεμένο στο *PXIe-4357* (κάτω-αριστερά) και *TB-4357* εσωτερική όψη (κάτω-δεξιά)

- Το *PXIe-4496* είναι ένα data acquisition (DAQ) module υψηλής ακρίβειας ειδικά σχεδιασμένο για πολυκάναλες ηχητικές και εφαρμογές δόνησης. Προσφέρει 16 εισόδους με ρυθμό δειγματοληψίας ως 204.8 kS/s, με 24-bit resolution ADCs με 114 dB δυναμικό εύρος. Χρησιμοποιεί delta-sigma modulation για A/D conversion. Αυτό σημαίνει πως εφαρμόζεται υπερδειγματοληψία στα δείγματα που λαμβάνονται ενώ μετά εφαρμόζεται σε αυτά ένα ψηφιακό φίλτρο, έχοντας ως αποτέλεσμα εξαιρετική γραμμικότητα φάσης διατηρώντας το θόρυβο και τις παραμορφώσεις σε πολύ χαμηλό επίπεδο. Χρησιμοποιείται για τη λήψη δεδομένων απ' τα επιταχυνσιόμετρα και απεικονίζεται στο Σχ. 2.16.



Σχήμα 2. 16: PXIe-4496

- Το *PXI-4461* είναι ένα data acquisition module υψηλής ακρίβειας. Έχει δύο αναλογικές εξόδους με ρυθμό δειγματοληψίας ως 204.8 kS/s, software-configurable AC/DC coupling και IEP conditioning, 24-bit resolution ADCs και DACs με 118 dB δυναμικό εύρος, variable anti-aliasing και anti-imaging φίλτρα, 6 επιλογές κέρδους εισόδους μεταξύ  $\pm 316$  mV και  $\pm 42.4$  V και δύο αναλογικές εισόδους με ρυθμό δειγματοληψίας μέχρι 204.8 kS/s. Όπως και το PXIe-4461 χρησιμοποιεί delta-sigma modulation τόσο στις εισόδους όσο και στις εξόδους. Υπάρχουν δύο τέτοια modules στην διάταξή μας - το πρώτο χρησιμοποιείται για τις μετρήσεις load cell και ρεύματος και το άλλο για τις μετρήσεις του capacitive sensor και τη δημιουργία και λήψη feedback του σήματος διέγερσης του πιεζοηλεκτρικού στοιχείου. Απεικονίζεται στο Σχ. 2.17.



Σχήμα 2. 17: PXI-4461

- Το *PXIe-4330* έχει οκτώ κανάλια με 24-bit analog-to-digital converters (ADCs) ανά κανάλι και 25 kS/s ρυθμό δειγματοληψίας. Για την αφαίρεση του θορύβου, κάθε module προσφέρει anti-aliasing και ψηφιακά φίλτρα ανά κανάλι. Κάθε

κανάλι προσφέρει προγραμματιζόμενη 0.625 ως 10 V τάση διέγερσης για το strain gage. Το PXIe-4330 συνοδεύεται από το NI TB-4330 front-mounting terminal block, που διευκολύνει τη διασύνδεση. Ο συνδυασμός PXIe-4330/TB-4330 χρησιμοποιείται για τις μετρήσεις του strain gage και φαίνεται στο Σχ. 2.15.

- Τέλος, το *PXI-7841R* είναι ένα multifunction reconfigurable I/O (RIO) module με user-programmable FPGA chip για on-board processing και I/O ευελιξία. Όλη η αναλογική και ψηφιακή λειτουργικότητα ρυθμίζεται με το NI LabVIEW και το LabVIEW FPGA Module. Έχει 8 αναλογικές εισόδους με ανεξάρτητους ρυθμούς δειγματοληψίας μέχρι 200 kHz, 16-bit resolution και  $\pm 10$  V εύρος, 8 αναλογικές εξόδους με ρυθμό ανανέωσης μέχρι 1 MHz, 16-bit resolution και  $\pm 10$  V εύρος εξόδου και 96 ψηφιακές γραμμές ρυθμιζόμενες ως εισόδους, εξόδους, μετρητές, ή άλλη λογική με ρυθμούς ως 40 MHz. Το PXI-7841R χρησιμοποιείται σε συνδυασμό με το NI SCB-68 connector block, που προσφέρει screw terminals για εύκολες I/O συνδέσεις. Ο συνδυασμός PXI-7841R/SCB-68 χρησιμοποιείται για τον έλεγχο του κινητήρα και απεικονίζεται στο Σχ. 2.18. Συγκεκριμένα, προσφέρει 3 ανεξάρτητες λειτουργίες: λειτουργεί τον σερβομηχανισμό και το μηχανικό φρένο του κινητήρα ενεργοποιώντας δύο ψηφιακές εξόδους και ελέγχει την ταχύτητα περιστροφής του κινητήρα μεταβάλλοντας μια αναλογική έξοδο που τροφοδοτεί τον ενισχυτή σερβοκινητήρα.



Σχήμα 2. 18: PXI-7841R (αριστερά) και SCB-68 connector block (δεξιά)

Όσον αφορά τη λειτουργία servo, μια ψηφιακή έξοδος του 7841-R/SCB-68 τροφοδοτεί την αντίστοιχη είσοδο του RS1A01AA servo ενισχυτή. Ο κινητήρας δεν λειτουργεί με τη λειτουργία servo απενεργοποιημένη, οπότε το να έχουμε αυτή τη



λειτουργία ξεχωριστά από την ρύθμιση της ταχύτητας χρησιμεύει για την αποφυγή ακούσιας λειτουργίας του κινητήρα.

Όσον αφορά το μηχανικό φρένο, και αυτό ελέγχεται από μια ψηφιακή έξοδο του 7841-R/SCB-68. Σε αυτή την περίπτωση όμως το ρεύμα και η τάση της ψηφιακής εξόδου δεν ήταν αρκετά για να λειτουργήσουν το φρένο. Σύμφωνα με τις προδιαγραφές του κατασκευαστή, το μηχανικό φρένο απαιτεί 24 V DC τάση με 270 mA ρεύμα. Για να ξεπεραστεί το εμπόδιο, χρησιμοποιήθηκε εξωτερική πλακέτα με ηλεκτρονόμους. Η πλακέτα συνδέεται ανάμεσα στο 7841-R/SCB-68 και τον ενισχυτή σερβοκινητήρα. Ο ηλεκτρονόμος ελέγχεται από μια ψηφιακή έξοδο του 7841-R/SCB-68 και όταν οπλιστεί, παρέχει 24 V DC στην είσοδο του φρένου του ενισχυτή του σερβοκινητήρα. Αυτή η τάση προέρχεται από μια εξωτερική τροφοδοσία 24 V DC στα 2300 mA. Η πλακέτα μαζί με την εξωτερική πηγή 24 V φαίνονται στο Σχ. 2.19.



Σχήμα 2. 19: Πλακέτα ηλεκτρονόμου και εξωτερική τροφοδοσία για τη λειτουργία του φρένου

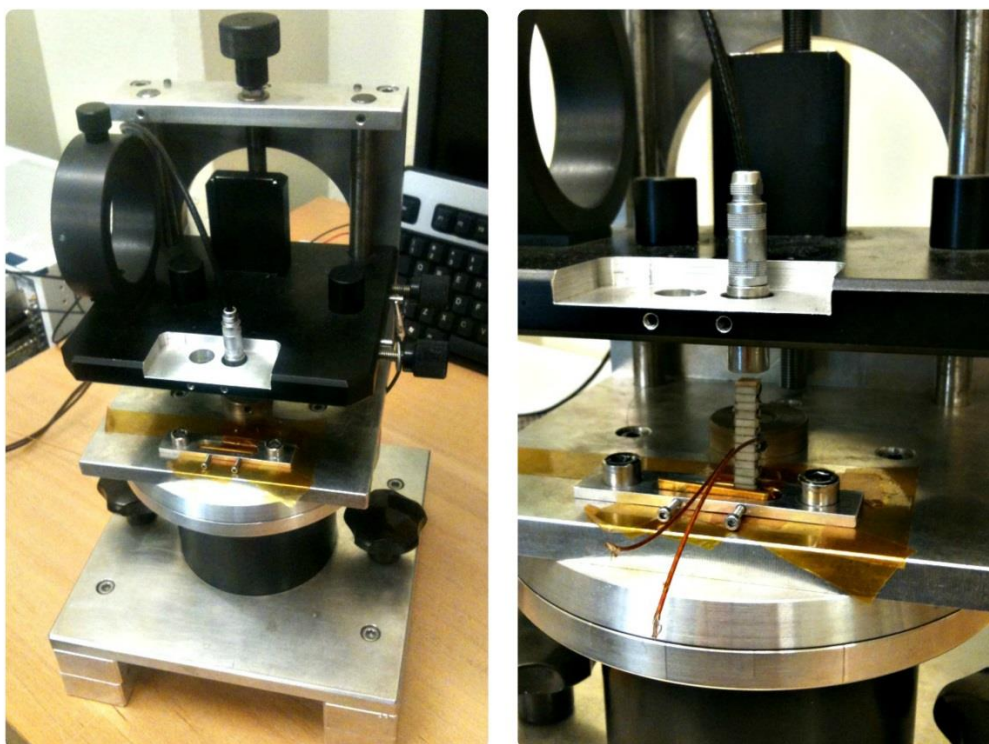
Η ταχύτητα περιστροφής του κινητήρα ελέγχεται από μια αναλογική έξοδο του 7841-R/SCB-68. Η αναλογική έξοδος έχει εύρος τιμών μεταξύ -10 V και +10 V με 16-bit resolution. Ένας 16-bit ακέραιος μπορεί να αποθηκεύσει  $2^{16}$  (ή 65,536) διακριτές τιμές. Σε unsigned αναπαράσταση, αυτές οι τιμές είναι οι ακέραιοι 0 μέχρι 65,535 και σε αναπαράσταση συμπληρώματος του δύο, οι πιθανές τιμές είναι μεταξύ -32,768 και 32,767. Αυτό σημαίνει ότι όταν η επιθυμητή έξοδος είναι -10 V, η τιμή που περνιέται στην αναλογική έξοδο πρέπει να είναι -32,768 και όταν η επιθυμητή έξοδος

είναι +10 V, η τιμή πρέπει να είναι 32,767, με όλες τις ενδιάμεσες τιμές να μεταβάλλονται γραμμικά.

## 2.4 Λειτουργική ανάλυση

Η διάταξη μπορεί να εκτελέσει τόσο μετρήσεις ερπυσμού όσο και υστέρησης με μια πληθώρα επιλογών.

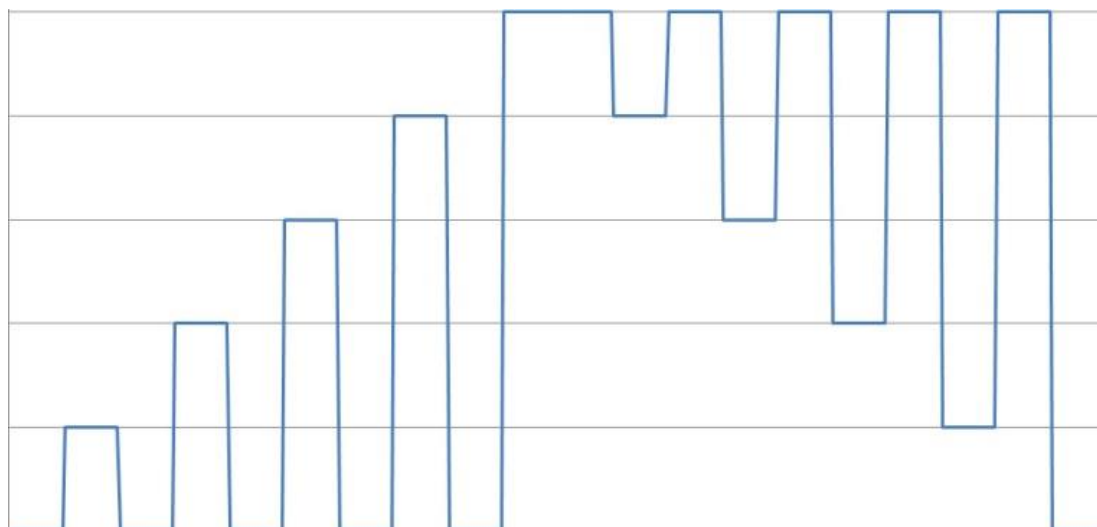
Για να εκτελεστούν μετρήσεις **ερπυσμού**, δεν πρέπει να ασκείται εξωτερική πίεση στο πιεζοηλεκτρικό στοιχείο. Παρότι τέτοιες μετρήσεις μπορούν να πραγματοποιηθούν στην διάταξη που έχει παρουσιαστεί, μια δευτερεύουσα διάταξη (Σχ. 2.20) προτιμάται καθώς προσφέρει ευκολότερη και γρηγορότερη τοποθέτηση του πιεζοηλεκτρικού στοιχείου.



Σχήμα 2. 20: Δευτερεύουσα διάταξη για μετρήσεις ερπυσμού (αριστερά) και κοντινό της διάταξης με τον capacitive sensor και το πιεζοηλεκτρικό στοιχείο τοποθετημένα (δεξιά)

Η προαναφερθείσα δευτερεύουσα διάταξη λειτουργεί στερεώνοντας το πιεζοηλεκτρικό στοιχείο στη βάση του. Ο capacitive sensor τοποθετείται πάνω από το στοιχείο και όταν εφαρμόζεται τάση σε αυτό, η απόσταση μεταξύ του πάνω μέρους

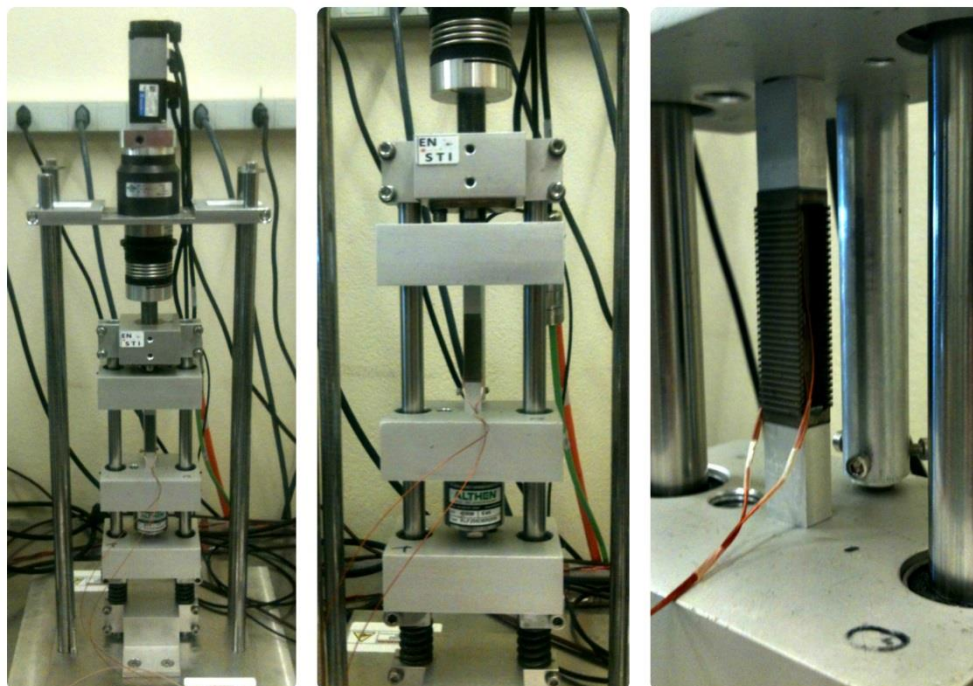
του στοιχείου και του capacitive sensor μεταβάλλεται και αυτή η μεταβολή αιχμαλωτίζεται από τον capacitive sensor. Για να πραγματοποιήσουμε ένα τεστ ερπυσμού, η διέγερση του στοιχείου ακολουθεί τη μορφή του Σχ. 2.21.



Σχήμα 2. 21: Μορφή κυματομορφής διέγερσης στοιχείου για τεστ ερπυσμού

Η εφαρμοζόμενη τάση εναλλάσσεται μεταξύ της ελάχιστης τιμής και μιας ολοένα και αυξανόμενης σε κάθε βήμα μέχρι να επιτευχθεί η μέγιστη επιθυμητή τιμή. Η διαδικασία τότε επαναλαμβάνεται αντίστροφα – η τάση εναλλάσσεται μεταξύ της μέγιστης τιμής και μιας ολοένα και φθίνουσας σε κάθε βήμα μέχρι να επιτευχθεί η αρχική ελάχιστη τιμή. Ο αριθμός βημάτων, η μέγιστη και η ελάχιστη τάση και η διάρκεια κάθε βήματος είναι παράμετροι που μπορούν να ρυθμιστούν πριν την εκτέλεση του τεστ ερπυσμού. Γενικά, η διάρκεια κάθε βήματος είναι σχετικά μεγάλη, ώστε η παραμόρφωση του πιεζοηλεκτρικού στοιχείου να έχει ολοκληρωθεί πριν προχωρήσουμε στο επόμενο βήμα.

Όσον αφορά τη μέτρηση της **υστέρησης**, προσφέρονται δύο τρόποι εκτέλεσης, ο στατικός (static execution mode) και ο δυναμικός (dynamic execution mode). Για το τεστ υστέρησης, μόνο η βασική διάταξη που έχει παρουσιαστεί (Σχ. 2.22) χρησιμοποιείται, όπου η πίεση ασκείται μέσω του κινητήρα.



Σχήμα 2. 22: Κύρια διάταξη (αριστερά και κέντρο) και κοντινή όψη του πιεζοηλεκτρικού ενεργοποιητή και του capacitive sensor (δεξιά)

Στον *στατικό* τρόπο εκτέλεσης, το μηχανικό φορτίο ορίζεται πριν τη διέγερση του πιεζοηλεκτρικού στοιχείου και παραμένει σταθερό καθ' όλη τη διάρκεια του τεστ. Η τάση διέγερσης μπορεί να οριστεί είτε σε μια σταθερή τιμή ή σε μια περιοδική κυματομορφή (ημιτονοειδής, τριγωνική, τετραγωνική ή πριονωτή) με παραμέτρους που καθορίζονται από τον χρήστη.

Στον *δυναμικό* τρόπο εκτέλεσης, το μηχανικό φορτίο μεταβάλλεται κατά τη διάρκεια του τεστ, με τη μεταβολή να παραπέμπει σε τριγωνική κυματομορφή μεταξύ μιας ελάχιστης και μιας μέγιστης τιμής φορτίου. Και πάλι, η τάση διέγερσης μπορεί να οριστεί είτε σε μια σταθερή τιμή ή σε μια περιοδική κυματομορφή (ημιτονοειδής, τριγωνική, τετραγωνική ή πριονωτή) με παραμέτρους που καθορίζονται από τον χρήστη. Μια ειδική λειτουργία στον δυναμικό τρόπο εκτέλεσης παρέχει τη δυνατότητα να μεταβάλλεται το μηχανικό φορτίο και η τάση διέγερσης ταυτόχρονα και με την ίδια συχνότητα.

Τέλος, το λογισμικό παρέχει τη δυνατότητα προβολής των γραφικών παραστάσεων των δεδομένων, της αυτόματης μετεπεξεργασίας τους και της αυτόματης παραγωγής αναφορών. Όλες οι λειτουργίες του αναλύονται στο επόμενο κεφάλαιο.

# 3

## *Ανάλυση LabVIEW*

Σκοπός του παρόντος κεφαλαίου είναι η εμπειριστατωμένη ανάλυση του λογισμικού που αναπτύχθηκε σε LabVIEW. Ξεκινώντας σε μια εισαγωγή στις βασικές δομικές έννοιες που χρησιμοποιήθηκαν, συνεχίζουμε με την ανάλυση του κώδικα, εστιάζοντας στις προγραμματιστικές προκλήσεις που προέκυψαν.

### *3.1 Βασικές δομικές έννοιες*

Υπάρχουν πολλές προσεγγίσεις όσον αφορά τον προγραμματισμό σε LabVIEW. Σε αυτή την περίπτωση, η αρχιτεκτονική με State Machine και Functional Global Variables (FGVs) αποτέλεσε ακρογωνιαίο λίθο στην υλοποίηση του προγράμματος και για αυτό το λόγο οι προαναφερθείσες έννοιες αναλύονται παρακάτω.

#### *3.1.1 State Machine*

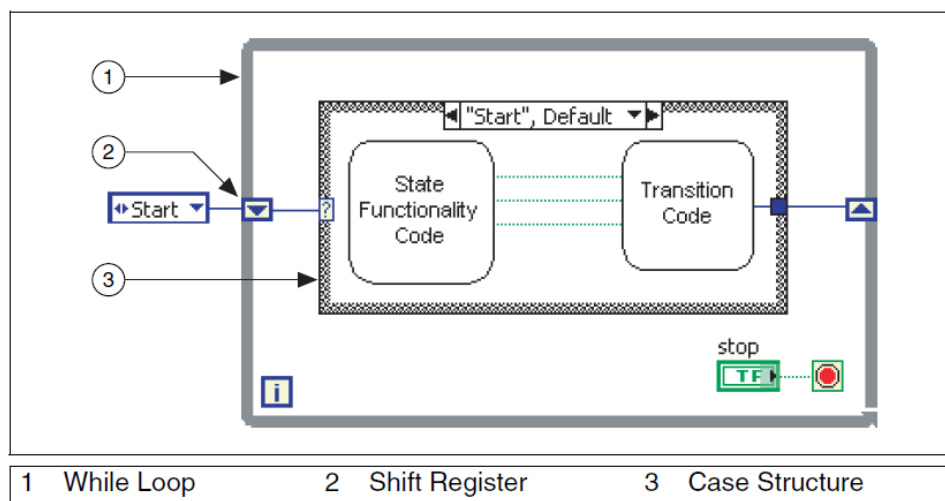
Η σχεδίαση με state machine είναι πολύ διαδεδομένη στο LabVIEW και χρησιμοποιείται για την υλοποίηση κάθε αλγορίθμου που μπορεί να περιγραφεί από ένα διάγραμμα καταστάσεων (state diagram) ή διάγραμμα ροής (flowchart).

Ένα state machine αποτελείται από ένα σετ καταστάσεων και μια συνάρτηση που καθορίζει την επόμενη κατάσταση. Μια κατάσταση μπορεί να οδηγεί σε μία ή πολλές καταστάσεις ή να ολοκληρώνει την ροή επεξεργασίας. Η επόμενη κατάσταση καθορίζεται από input του χρήστη ή in-state υπολογισμούς.

Ένα state machine αποτελείται από τα ακόλουθα στοιχεία:

- While loop – Εκτελεί συνεχώς τις διάφορες καταστάσεις
- Case structure – Περιέχει ένα case για κάθε κατάσταση καθώς και τον κώδικα που εκτελείται σε κάθε κατάσταση
- Shift register – Περιέχει πληροφορίες για τη μετάβαση μεταξύ καταστάσεων
- State functionality code – Υλοποιεί τον κώδικα της κατάστασης
- Transition code – Καθορίζει την επόμενη κατάσταση

Ένα παράδειγμα αρχιτεκτονικής state machine φαίνεται στο Σχ. 3.1.



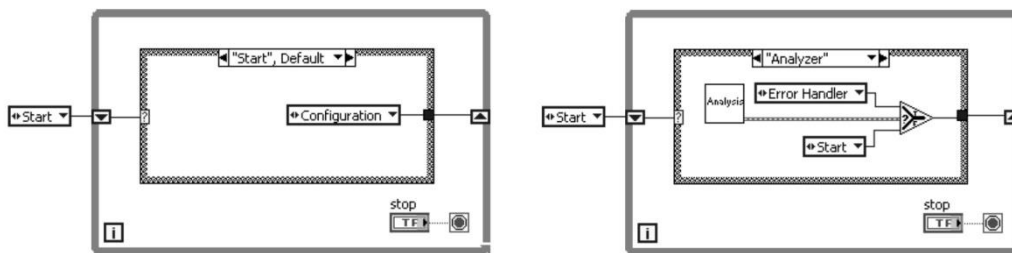
Σχήμα 3. 1: Αρχιτεκτονική state machine

Σε σύγκριση με ένα διάγραμμα καταστάσεων, η ροή σε ένα state machine υλοποιείται από το While Loop, οι διαφορετικές καταστάσεις παρουσιάζονται ως cases στο Case structure, ενώ ο shift register στο While Loop παρακολουθεί την τρέχουσα κατάσταση και μεταλαμπαδεύει την τρέχουσα κατάσταση στο Case structure input.

Όπως φαίνεται στο Σχ. 3.1, η πιο διαδεδομένη μέθοδος ελέγχου της μετάβασης καταστάσεων είναι το enumerated type control (enum). Μια σημαντική λεπτομέρεια είναι ότι πρέπει να είναι type-defined (typedef). Αυτό λύνει τα προβλήματα που προκύπτουν όταν ο χρήστης επιχειρεί να προσθέσει ή διαγράψει μια κατάσταση από το enumerated type control, οδηγώντας σε ρήξη σύνδεσης των καλωδίων που

συνδέονται στα αντίγραφα αυτού του enumerated type control. Με άλλα λόγια, κάνοντας type-define το enumerated type control, όλα τα αντίγραφα του στο πρόγραμμα ενημερώνονται αυτόματα αν μια κατάσταση προστεθεί ή αφαιρεθεί.

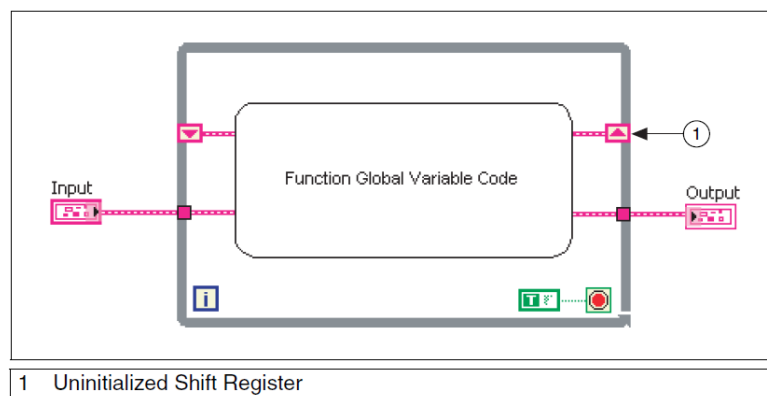
Στο συγκεκριμένο project, η επόμενη κατάσταση ενός state machine ελέγχεται κατά βάση με δύο τρόπους – με τον λεγόμενο “single default transition” και “select function” κώδικα. Ο πρώτος χρησιμοποιείται όταν υπάρχει μόνο μια πιθανή επόμενη κατάσταση (Σχ. 3.2α), ενώ ο δεύτερος όταν υπάρχουν δύο πιθανές καταστάσεις και πρέπει να παρθεί μια απόφαση ανάμεσά τους (Σχ. 3.2β). [58]



Σχήμα 3. 2α (αριστερά): Single default transition και Σχήμα 3. 2β (δεξιά): Δύο πιθανές μεταβάσεις όπου το αποτέλεσμα της έκβασης καθορίζεται από τη συνάρτηση select

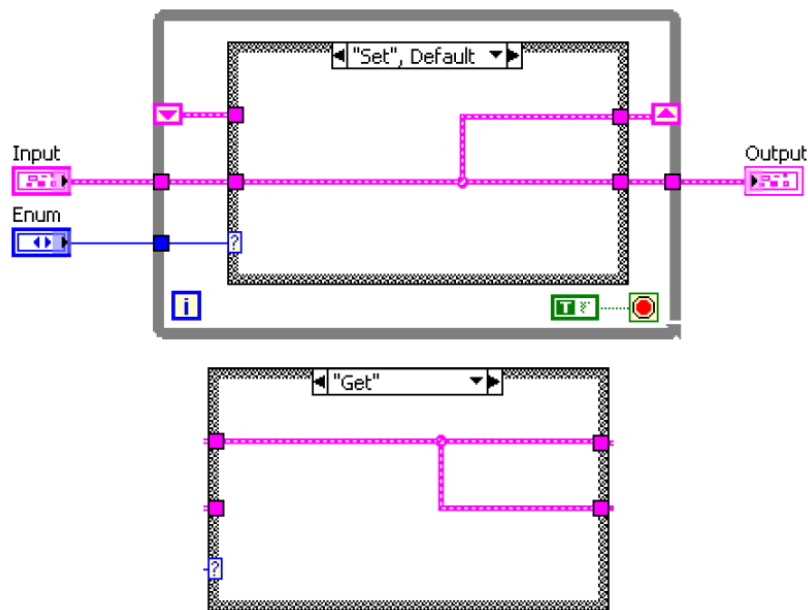
### 3.1.2 Functional Global Variable (FGV)

Τα Functional global variables χρησιμοποιούνται εκτενώς σε αυτό το project καθώς είναι πιο κατάλληλα από τα global variables για τους ακόλουθους λόγους. Ένα functional global variable χρησιμοποιεί uninitialized shift registers σε While loop για να διατηρεί δεδομένα όσο το VI βρίσκεται στη μνήμη ram. Η γενική μορφή ενός Functional Global Variable περιλαμβάνει ένα uninitialized shift register με ένα While loop που εκτελείται μία φορά, όπως φαίνεται στο Σχ. 3.3.



Σχήμα 3. 3: Γενική μορφή Functional Global Variable

Ένα Functional Global Variable συνήθως έχει ένα action input parameter που καθορίζει τη λειτουργία του VI. Το VI χρησιμοποιεί ένα uninitialized shift register για να κρατά το αποτέλεσμα της πράξης. Το πιο κοινό FGV περιλαμβάνει “set” και “get” λειτουργίες (Σχ. 3.4). Όπως φαίνεται, στην κατάσταση “Set”, δεδομένα εισέρχονται στο VI τα οποία ο shift register αποθηκεύει, ενώ στην κατάσταση “Get”, δεδομένα ανακτώνται απ’ τον shift register και προωθούνται στην έξοδο.



Σχήμα 3. 4: FGV με “Set” και “Get” καταστάσεις

Υπάρχουν 3 πλεονεκτήματα στη χρήση Functional Global Variables αντί για Global Variables. Το πρώτο είναι ότι είναι πλήρως προσαρμόσιμα καθώς μπορούν να έχουν όσες καταστάσεις επιθυμούμε (τα FGV μπορούν να έχουν περισσότερες καταστάσεις από τα απλά “Get”/”Set” FGV). Το δεύτερο είναι ότι μπορούν να εξυπηρετήσουν πιο σύνθετες δομές δεδομένων, όπως stack, queue buffer ή complex bundle type, που τα απλά global variables δεν μπορούν. Το τελευταίο και σημαντικότερο είναι ότι δεν είναι re-entrant. Αυτό σημαίνει ότι όταν ένα FGV καλείται από πολλές τοποθεσίες, το ίδιο αντίγραφο του FGV χρησιμοποιείται. Συνεπώς, μόνο μια κλήση προς το FGV μπορεί να συμβαίνει ανά πάσα στιγμή. Αυτό αποτρέπει race conditions που μπορούν να συμβούν όταν 2 διαφορετικές διεργασίες προσπαθούν να έχουν πρόσβαση στο ίδιο FGV (π.χ. μία για να διαβάσει και η άλλη για να γράψει σε αυτό). [59]



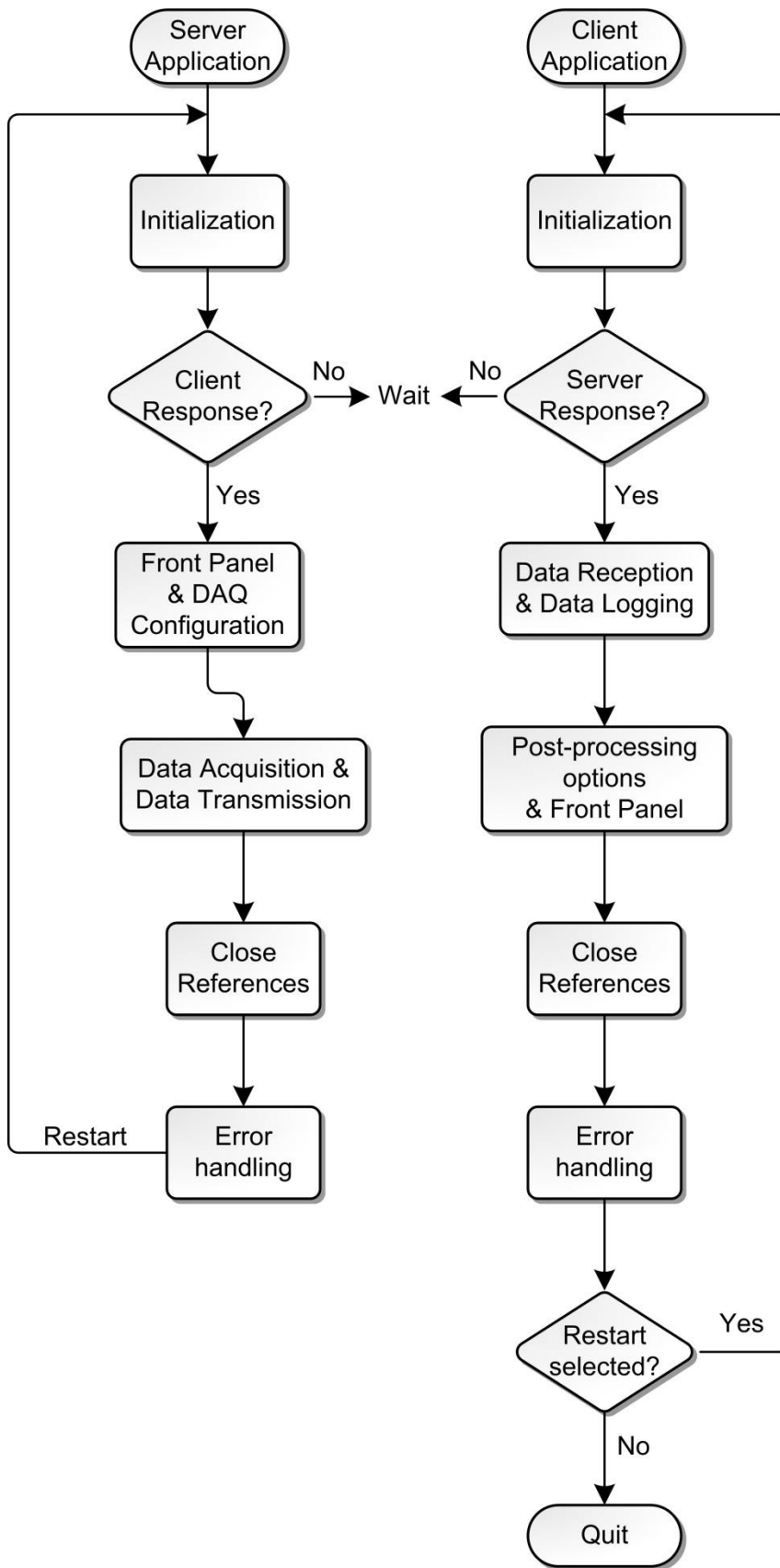
## **3.2 Κύρια ανάλυση**

### **3.2.1 Επισκόπηση**

Το project χωρίζεται σε δύο σκέλη – την εφαρμογή server και την εφαρμογή client. Η εφαρμογή server τρέχει στην NI LabVIEW Real-Time πλευρά του PXI (παραπομπή στην παράγραφο 2.3.4 του 2<sup>ου</sup> κεφαλαίου για λεπτομέρειες), ενώ η εφαρμογή client τρέχει στην Windows XP πλευρά του PXI (ή οποιουδήποτε υπολογιστή μπορεί να εδραιώσει μια σύνδεση μέσω δικτύου με το σύστημα PXI). Αυτό συμβαίνει για να βελτιωθεί η ακρίβεια, αξιοπιστία και συνέπεια των μετρήσεων.

Συγκεκριμένα, η εφαρμογή server είναι υπεύθυνη για τη λειτουργία του κινητήρα και τη λήψη δεδομένων από τους διάφορους αισθητήρες που λειτουργούν σε διαφορετικούς ρυθμούς δειγματοληψίας ο καθένας με ακριβές timing και συγχρονισμό και την αποστολή αυτών των δεδομένων στη εφαρμογή client σε πακέτα, διασφαλίζοντας ότι δεν σημειώνεται απώλεια δεδομένων. Η εφαρμογή client από την άλλη πλευρά είναι υπεύθυνη για την υποδοχή όλων των σταλμένων δεδομένων και την εκτέλεση όλων των απαιτητικών – όσον αφορά τη cpu και τη μνήμη – διεργασιών που δεν απαιτούν ακριβές timing, όπως η σχεδίαση των γραφικών παραστάσεων των δεδομένων, η μετεπεξεργασία τους με Matlab και η παραγωγή των αναφορών.

Το διάγραμμα ροής τόσο της server όσο και της client εφαρμογής φαίνεται στο Σχ. 3.5 , ενώ η ανάλυσή τους ακολουθεί στα επόμενα υποκεφάλαια.



Σχήμα 3. 5: Διάγραμμα ροής των server και client εφαρμογών

### 3.2.2 Εφαρμογή Server

Η εφαρμογή server αποτελείται από ένα state machine 5 καταστάσεων. Όπως φαίνεται στο διάγραμμα ροής του Σχ. 3.5, οι 5 καταστάσεις εκτελούν τα initialization, DAQ configuration και Front Panel selection, data acquisition και transmission, reference closing και error handling και ονομάζονται “init”, “daqConfig”, “runProcesses”, “exit” και “error” αντίστοιχα.

- **Initialization**

Σε αυτό το στάδιο συμβαίνει όλη η απαιτούμενη αρχικοποίηση. Περιλαμβάνει τα subVIs “initFgvs”, “initSharedVariables”, “initFpga”, “initQueues” και “initNetworkStreams”.

- Το subvi “*initFgvs*” αρχικοποιεί όλα τα FGVs που τρέχουν στο real-time σύστημα στις αρχικές τιμές τους.

- Το subvi “*initSharedVariables*” αρχικοποιεί τα shared variables που διευκολύνουν την επικοινωνία server και client. Αυτά είναι η “stopClient” variable, που χρησιμοποιείται για την διακοπή λήψης δεδομένων και logging στην εφαρμογή client, όταν οι μετρήσεις έχουν τελειώσει στην εφαρμογή server, καθώς και η “strainGageIniFile” shared variable, που στέλνει στην εφαρμογή server το ορισμένο από τον χρήστη ini αρχείο στον client computer που έχει τις ενημερωμένες παραμέτρους strain gage, σε περίπτωση που δεν χρησιμοποιηθούν οι default.

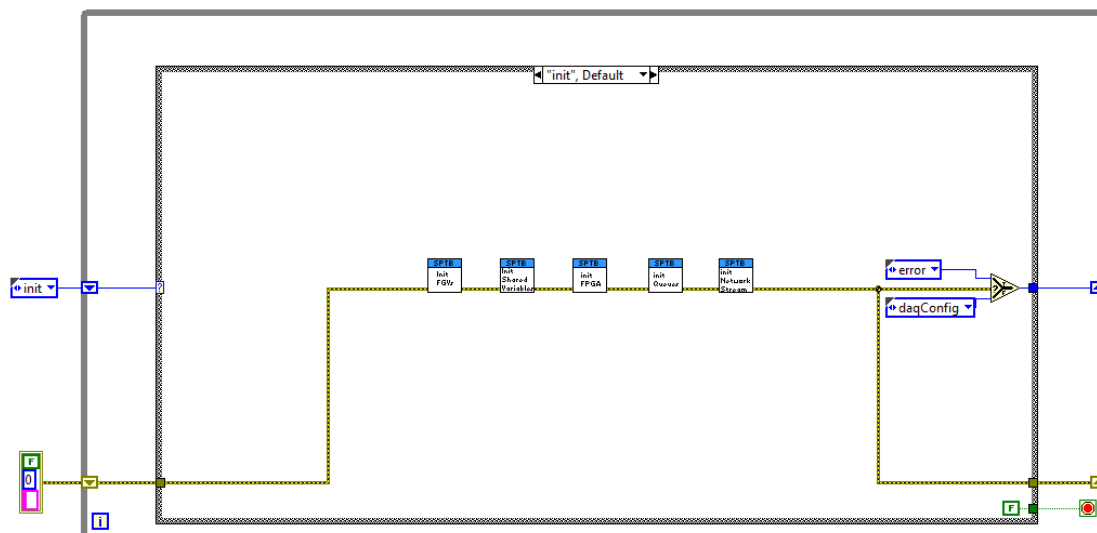
- Το subvi “*initFpga*” αναγνωρίζει τις συνδεδεμένες NI RIO συσκευές και κατεβάζει σε αυτές το fpga bitfile και δημιουργεί και σώζει στο αντίστοιχο FGV το FPGA reference που θα είναι απαραίτητο σε επόμενα στάδια. Προς το παρόν, η μόνη διαθέσιμη NI RIO συσκευή είναι η PXI-7841R αλλά η συγκεκριμένη υλοποίηση επιτρέπει εύκολα την χρήση και άλλων συσκευών στο μέλλον.

- Το subvi “*initQueues*” δημιουργεί τις δομές queue που θα χρησιμοποιηθούν στο στάδιο data acquisition και σώζει τα references τους στα αντίστοιχα FGVs.

- Το subvi “*initNetworkStreams*” δημιουργεί όλα τα network stream writer endpoints και σώζει τα references τους στα αντίστοιχα FGVs. Το subvi τερματίζει όταν τα reader endpoints έχουν δημιουργηθεί στην εφαρμογή client.

Η χρήση των queues και network streams καλύπτεται λεπτομερώς στο υποκεφάλαιο “Διαχείριση Δεδομένων”.

Σε κάθε κατάσταση των state machines τόσο στην server όσο και στην client εφαρμογή, αν συμβεί σφάλμα σε οποιοδήποτε subνι της κατάστασης, το πρόγραμμα μεταβαίνει στην κατάσταση “error”. Αυτό σημαίνει πως το πρόγραμμα σταματά να τρέχει αν συμβεί οποιοδήποτε απροσδόκητο σφάλμα.



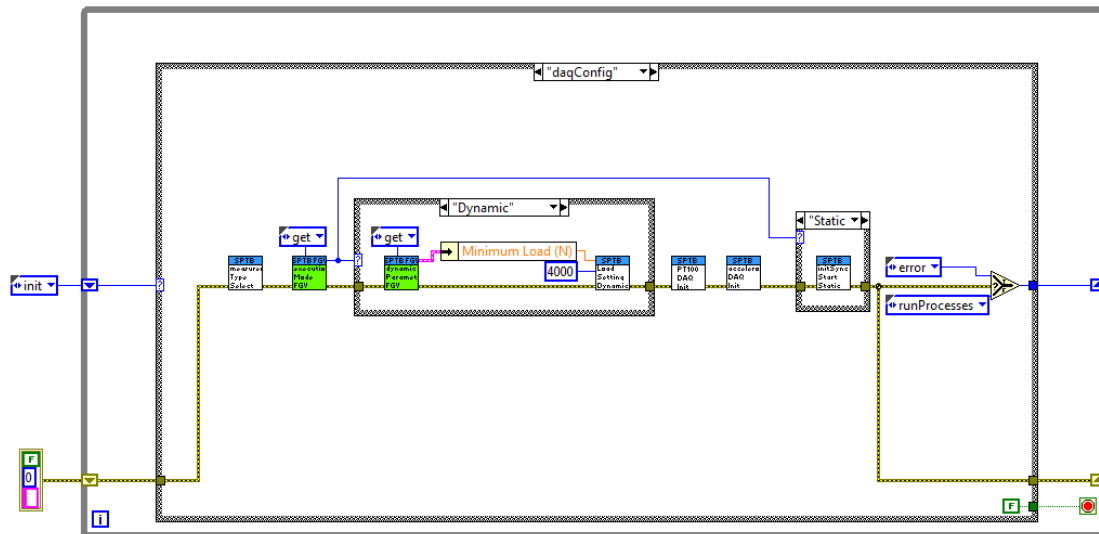
Σχήμα 3. 6: Εφαρμογή server – κατάσταση Initialization

- **DAQ configuration και Front Panel**

Σε αυτό το στάδιο λαμβάνει χώρα η διαμόρφωση λήψης δεδομένων (DAQ configuration) και οι επιλογές στον πίνακα ελέγχου (Front Panel selection). Ειδικά:

- Το subνι “*measurementTypeSelect*” επιτρέπει την επιλογή Creep ή Hysteresis τεστ μέσω του Front Panel. Ανάλογα την επιλογή, καλείται αντίστοιχο subνι που επιτρέπει στους χρήστες να θέσουν τις παραμέτρους των Creep και Hysteresis τεστ.
- Στην περίπτωση που επιλεγθεί dynamic-execution Hysteresis τεστ, ο κινητήρας λειτουργεί ώστε το φορτίο να τεθεί στο minimum επίπεδο.
- Τα PT100 και accelerometer κανάλια δημιουργούνται με τις αντίστοιχες παραμέτρους και τα references τους σώζονται στα αντίστοιχα FGVs (“*PT100DAQInit*” και “*accelerometerDAQInit*” subνις).
- Τέλος, τα “*initSyncStartStatic*” (σε περίπτωση Creep ή static-execution Hysteresis τεστ) και “*initSyncStartDynamic*” (σε περίπτωση dynamic-execution Hysteresis τεστ) subνις δημιουργούν τα υπόλοιπα κανάλια λήψης με τις επιθυμητές παραμέτρους, διασφαλίζοντας τη συγχρονισμένη λειτουργία όλων των modules.

Τα πρώτα δύο σημεία καλύπτονται με λεπτομέρεια στο υποκεφάλαιο “Επιλογές Εκτέλεσης”, ενώ τα επόμενα δύο στο “Συγχρονισμός λήψης δεδομένων”.



Σχήμα 3. 7: Εφαρμογή server – κατάσταση DAQ Configuration και Front Panel

- **Data acquisition και data transmission**

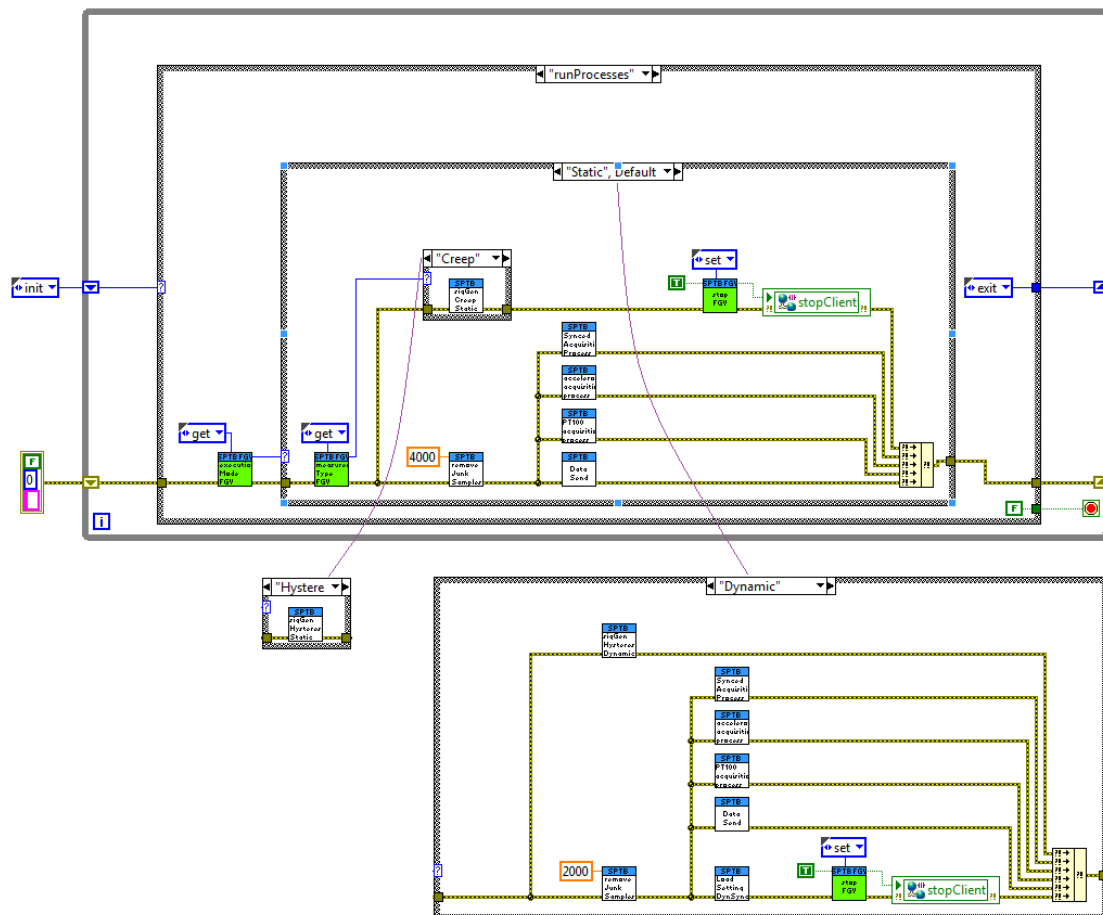
Σε αυτό το στάδιο γίνεται η λήψη δεδομένων των μετρήσεων, τα οποία στέλνονται μέσω των network stream στην εφαρμογή client για logging και επεξεργασία.

Ένα case structure χρησιμοποιείται για τον διαχωρισμό static execution (Creep και static-execution Hysteresis τεστ) και dynamic execution (dynamic-execution Hysteresis test), αλλά και στις δύο περιπτώσεις χρησιμοποιούνται τα “*SyncedAcquisitionProcess*”, “*PT100AcquisitionProcess*” και “*accelerometerAcquisitionProcess*” subvis, που πραγματοποιούν τη λήψη δεδομένων, καθώς και το “*dataSend*” subvi, που στέλνει τα ληφθέντα δεδομένα στον client (περισσότερες πληροφορίες στο υποκεφάλαιο “Διαχείριση Δεδομένων”).

Και τα δύο cases περιλαμβάνουν την αποστολή ενός σήματος διακοπής στην εφαρμογή client μέσω της “stopClient” shared variable όταν το test ολοκληρωθεί. Στην περίπτωση static execution, το σήμα μεταδίδεται όταν το signal generation ολοκληρωθεί, ενώ στην περίπτωση dynamic execution, όταν η κίνηση του κινητήρα που ελέγχεται από το “loadSettingDynamicSync” subvi ολοκληρωθεί.

Η παραγωγή σήματος διέγερσης (signal generation) ελέγχεται από τα subvis (“*signalGenerationHysteresisDynamic*” στην περίπτωση dynamic-execution

Hysteresis τεστ, “*signalGenerationHysteresisStatic*” στην περίπτωση static-execution Hysteresis τεστ και “*signalGenerationCreepStatic*” στην περίπτωση Creep test), ενώ το “*removeJunkSamples*” subνι διασφαλίζει ότι η παραγωγή σήματος διέγερσης είναι συγχρονισμένη με τη λήψη δεδομένων από τους αισθητήρες (περισσότερες πληροφορίες στο υποκεφάλαιο “Συγχρονισμός λήψης δεδομένων”).



Σχήμα 3. 8: Εφαρμογή server – Data acquisition και data transmission state με παραλλαγές case

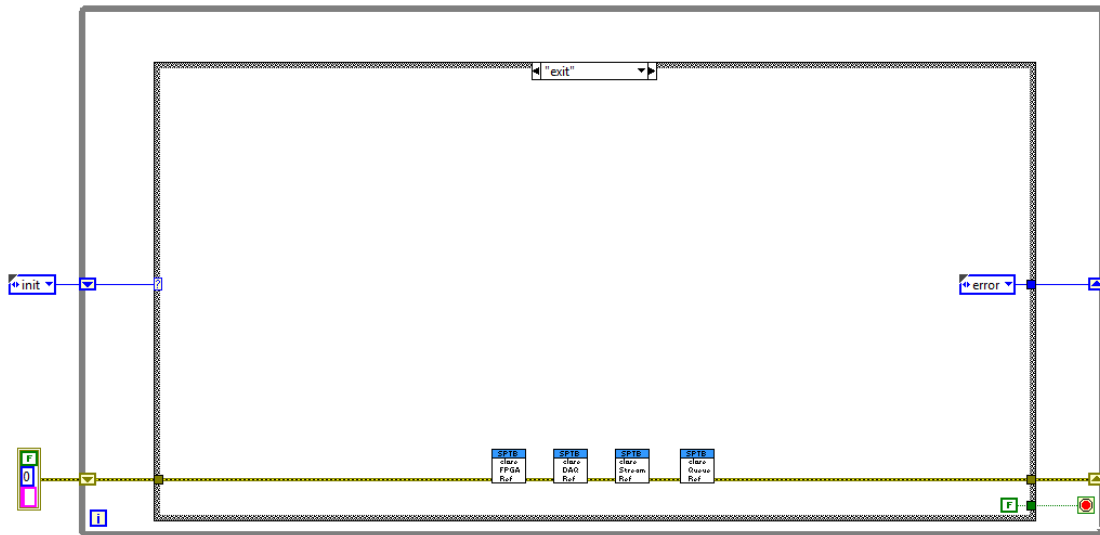
- **Close references**

Σε αυτό το στάδιο κλείνονται όλα τα references. Συγκεκριμένα:

- Το subνι “*closeFPGARef*” κλείνει το FPGA reference.
- Το subνι “*closeDAQRef*” καθαρίζει τα data acquisition και signal generation tasks.
- Το subνι “*closeStreamRef*” καταστρέφει τα network stream writer endpoints. Αξίζει να αναφερθεί ότι πριν καταστραφούν καλείται το “*Flush Stream*” subνι, το οποίο μεταφέρει όλα τα δεδομένα που είναι ακόμα διαθέσιμα για ανάγνωση στο

reader endpoint (client). Αυτό διασφαλίζει ότι το writer endpoint buffer είναι άδειο πριν καταστραφεί και κανένα δεδομένο δεν απορρίπτεται πριν διαβαστεί στον client.

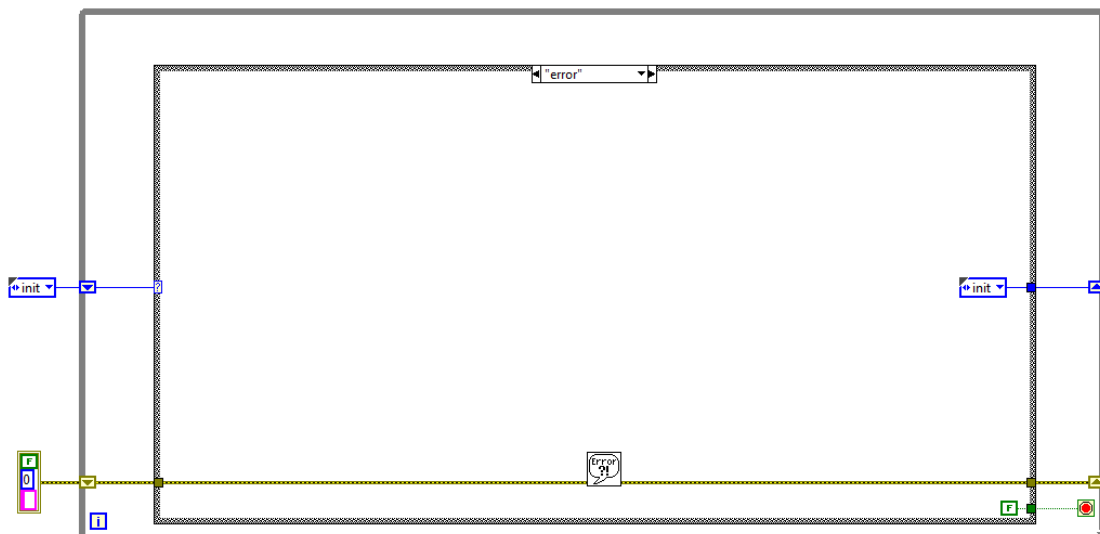
- ο Το subνι “*closeQueueRef*” ελευθερώνει τα references των διάφορων queues.



Σχήμα 3. 9: Εφαρμογή server – Close references κατάσταση

- **Error**

Το στάδιο τρέχει το “*Simple Error Handler*”, που σε περίπτωση σφάλματος εμφανίζει τον κωδικό σφάλματος και την περιγραφή. Όπως αναφέρθηκε, όταν συμβεί ένα σφάλμα, γίνεται μετάβαση στην κατάσταση “error” και η εκτέλεση διακόπτεται.



Σχήμα 3. 10: Εφαρμογή server – Error κατάσταση

### 3.2.3 Εφαρμογή Client

Η client εφαρμογή, όπως και η server, αποτελείται από ένα state machine 5 καταστάσεων. Σύμφωνα με το διάγραμμα ροής του Σχ. 3.5, αυτές οι 5 καταστάσεις διενεργούν τα initialization, data reception και logging, post-processing options και Front Panel selection, reference closing και error handling και ονομάζονται “init”, “dataLogging”, “postOptions”, “exit” και “error” αντίστοιχα.

- **Initialization**

Όπως και στην εφαρμογή server, σε αυτό το στάδιο λαμβάνει χώρα η απαραίτητη αρχικοποίηση. Συγκεκριμένα:

- ο Το subvi “*initFGVsClient*” αρχικοποιεί όλα τα FGVs που τρέχουν στον client στις αρχικές τιμές τους.

- ο Το subvi “*initStrainGageConfigClient*” παρακινεί το χρήστη να διαλέξει τη χρήση είτε των default είτε των καθορισμένων από το χρήστη DAQ ρυθμίσεων για το strain gage. Στη 2<sup>η</sup> περίπτωση, καλείται κατάλληλο subvi που διαβάζει το ini<sup>9</sup> αρχείο που επέλεξε ο χρήστης και στέλνει τις παραμέτρους στον server μέσω της “*strainGageIniFile*” shared variable. Αξίζει να σημειωθεί πως αν το ίδιο subvi καλούνταν από την server εφαρμογή, τότε το file directory που περιείχε το ini αρχείο θα αναφέρονταν στο real-time λειτουργικό. Καθώς όμως το real-time λειτουργικό δεν έχει δικό του γραφικό περιβάλλον, ο μόνος δυνατός τρόπος να μεταφερθεί το ini αρχείο θα ήταν μέσω κάποιου FTP service, όπως το FileZilla. Συνεπώς, είναι φανερό για ποιο λόγο επιλέχθηκε η υλοποίηση όπου το ini αρχείο ανεβάζεται από την client πλευρά (Windows XP πλευρά του PXI ή οποιοσδήποτε υπολογιστής μπορεί να

---

<sup>9</sup> Το format INI είναι ένα ανεπίσημο πρότυπο για configuration files με μια βασική δομή που αποτελείται από “section” και “properties”. Το key (ή property) είναι το βασικό στοιχείο ενός INI αρχείου και κάθε key έχει ένα όνομα και μια τιμή, που χωρίζονται με το σύμβολο ίσον (=). Το όνομα εμφανίζεται στα αριστερά του ίσον (π.χ.: όνομα=τιμή). Τα keys μπορούν να ομαδοποιηθούν σε sections. Το όνομα του section εμφανίζεται σε μια γραμμή μόνο του με αγκύλες (π.χ.: [section]). Όλα τα keys μετά τη δήλωση του section συσχετίζονται με αυτό το section. Τα section και key δεν είναι case sensitive. [60]

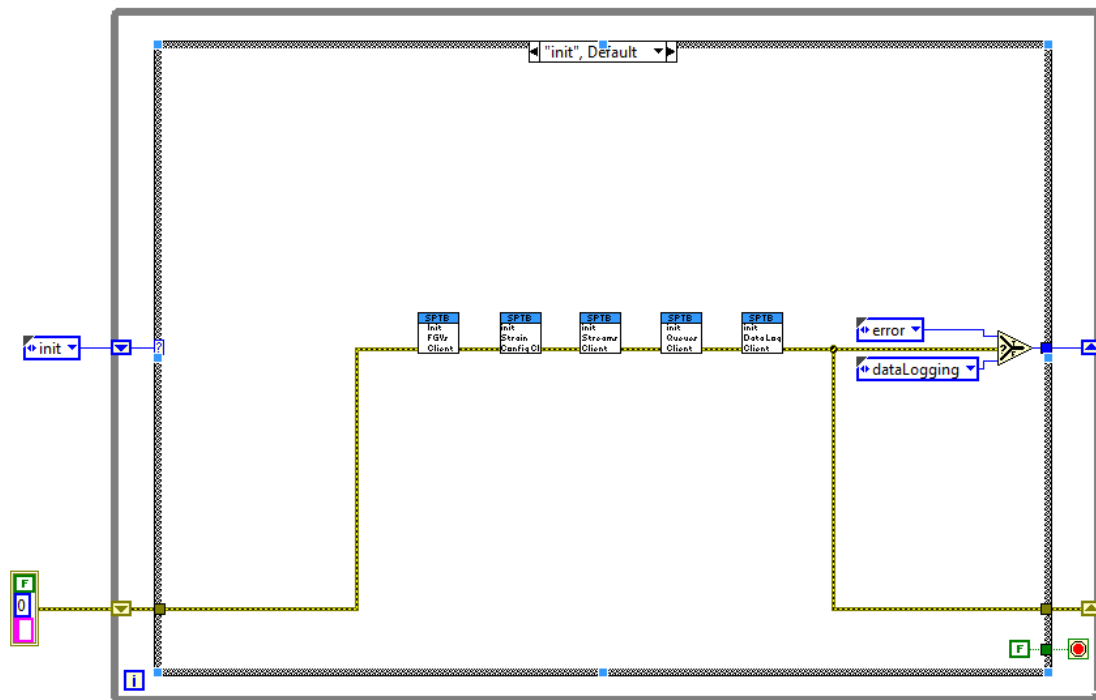


εδραιώσει μια σύνδεση μέσω δικτύου με το σύστημα PXI), με τις τιμές να αποστέλλονται στον server μέσω μιας shared variable.

- ο Το subvi “*initNetworkStreamsClient*” δημιουργεί όλα τα network stream reader endpoints και σώζει τα references τους στα αντίστοιχα FGVs. Το subvi ολοκληρώνει όταν τα writer endpoints της server εφαρμογής αναγνωρίζονται.

- ο Το subvi “*initQueuesClient*” δημιουργεί τις δομές queue που θα χρησιμοποιηθούν στο στάδιο λήψης των πακέτων δεδομένων και σώζει τις αναφορές τους στα αντίστοιχα FGVs.

- ο Το subvi “*initDataLoggingClient*” δημιουργεί (αν δεν υπάρχουν) ή αντικαθιστά (αν υπάρχουν) τα tsv<sup>10</sup> αρχεία που χρησιμοποιούνται για την αποθήκευση των ληφθέντων δεδομένων σε ένα προκαθορισμένο directory του client.



Σχήμα 3. 11: Εφαρμογή client – Initialization κατάσταση

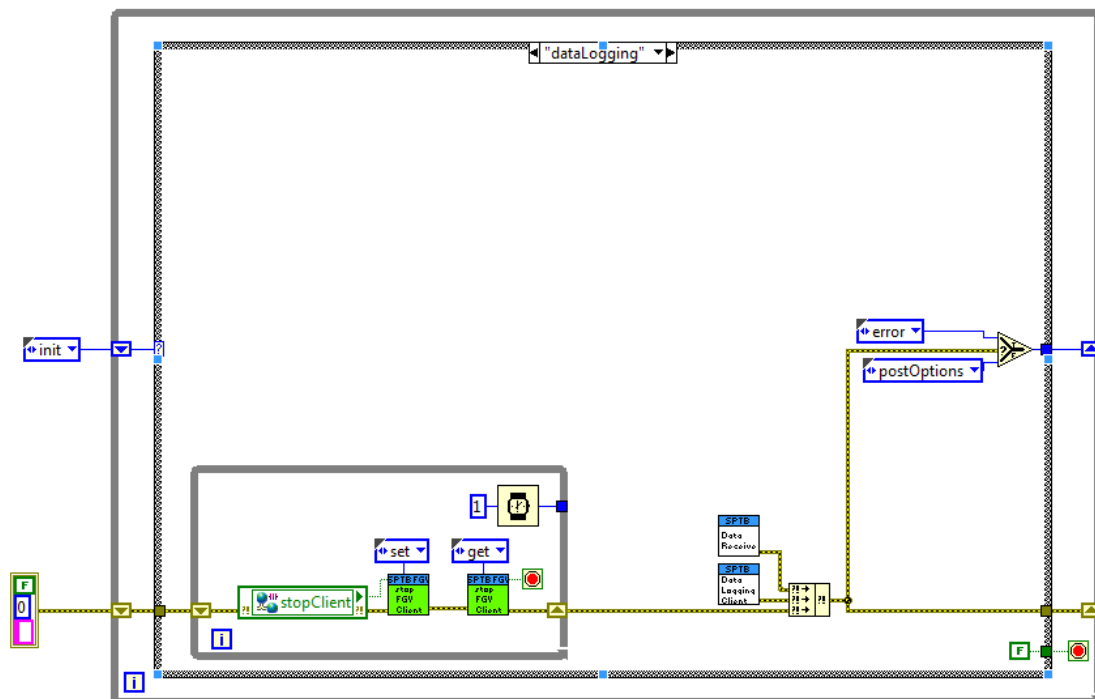
<sup>10</sup> Το TSV αντιπροσωπεύει το “tab-separated values”. Το TSV είναι μια εναλλακτική στο σύνηθες comma-separated values (CSV) format και η διαφορά είναι ότι το delimiter είναι ένας χαρακτήρας tab αντί για ένα κόμμα. [61]

- **Data reception και data logging**

Σε αυτό το στάδιο λαμβάνει χώρα η υποδοχή των δεδομένων που στέλνονται από τον server και η αποθήκευσή τους στο σκληρό δίσκο (logging). Συγκεκριμένα:

- Το subνι “*dataReceive*” διαβάζει όλα τα πακέτα δεδομένων από το network stream και τα τοποθετεί στα κατάλληλα queues.
- Το subνι “*dataLoggingProcessClient*” αφαιρεί στοιχεία από τα προαναφερθέντα queues και τα επισυνάπτει στα αντίστοιχα tsv αρχεία.
- Το στάδιο ολοκληρώνεται όταν το σήμα διακοπής από τον server ληφθεί μέσω της “*stopClient*” shared variable.

Περισσότερες λεπτομέρειες για τα πρώτα δύο σημεία παρατίθενται στο υποκεφάλαιο “*Διαχείριση Δεδομένων*”.



Σχήμα 3. 12: Εφαρμογή client – Κατάσταση data reception και data logging

- **Post-processing options και Front Panel**

Όταν ολοκληρωθεί η υποδοχή δεδομένων και το logging, παρουσιάζεται στον χρήστη ένα Front Panel με τις παρακάτω επιλογές:

- Copy data files: Το “*copyDataFilesClient*” subνι αντιγράφει τα tsv files σε μια τοποθεσία του σκληρού δίσκου του client που καθορίζεται από τον χρήστη. Αυτό

είναι χρήσιμο καθώς αν τα αρχεία δεν αντιγραφούν, στο επόμενο τρέξιμο του προγράμματος, τα προηγούμενα tsv αρχεία θα αντικατασταθούν από τα νέα.

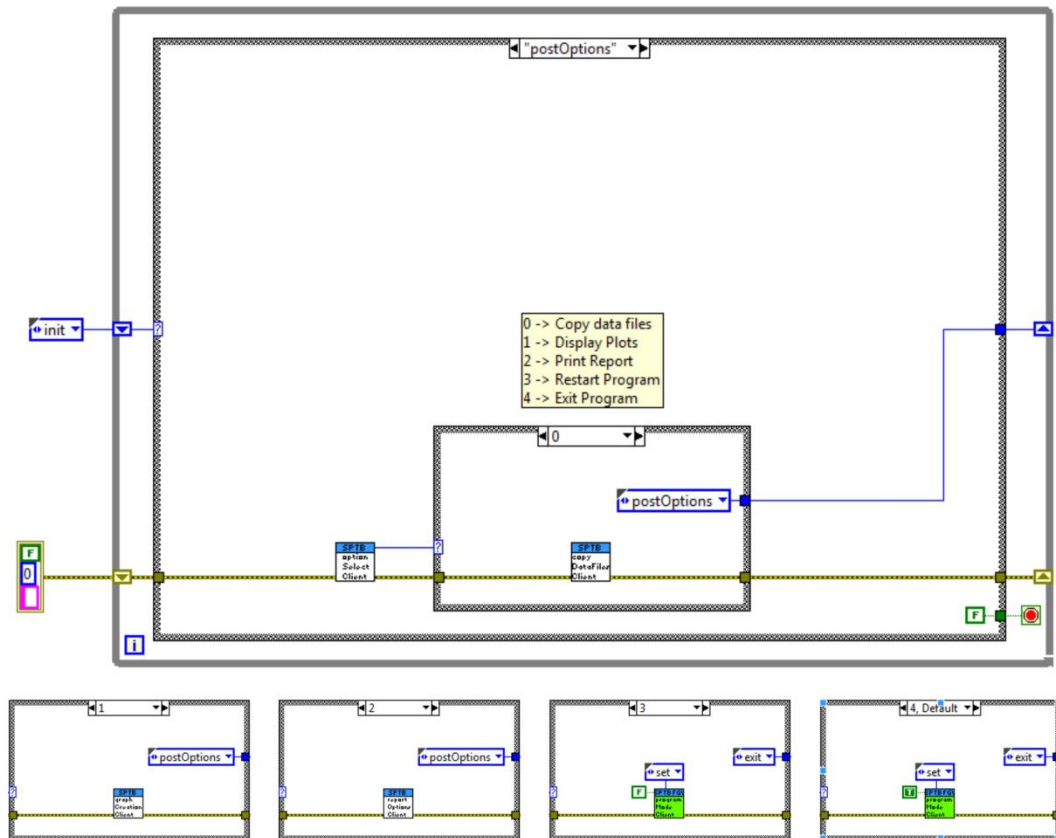
- Display plots: Το “*graphCreationClient*” subvi διαβάζει τα δεδομένα από τα αποθηκευμένα tsv αρχεία και σχεδιάζει τις γραφικές παραστάσεις. Υπάρχουν ξεχωριστά tabs για κάθε γραφική για τα strain gage, load cell, capacitive sensor, current, signal generator, PT100 και τα δύο accelerometers. Υπάρχουν επίσης δύο επιπλέον tabs για τον γρήγορο μετασχηματισμό Fourier - FFT<sup>11</sup> (Fast Fourier Transform) – των δεδομένων των δύο επιταχυνσιόμετρων. Τέλος υπάρχει ένα ξεχωριστό tab όπου τα δεδομένα των strain gage, load cell, capacitive sensor, current και signal generator αναπαρίστανται στην ίδια γραφική παράσταση, αφότου κατάλληλα subvis έχουν κάνει down-sampling στα δεδομένα ώστε να έχουν ίσο μήκος.

- Print Report: Το “*reportOptionsClient*” subvi παρουσιάζει στον χρήστη ένα Front Panel όπου μπορεί να παράγει μια αναφορά αφού συμπληρώσει τα απαραίτητα πεδία. Ο χρήστης μπορεί να επιλέξει να παράγει μια αναφορά Hysteresis (που τρέχει το Hysteresis post-processing Matlab script και παράγει το διάγραμμα Hysteresis Loop), Creep (που τρέχει το Creep post-processing Matlab script και παράγει το διάγραμμα Creep) ή Leakage Current ή οποιονδήποτε συνδυασμό αυτών. Περισσότερες πληροφορίες παρέχονται στο υποκεφάλαιο “Αυτόματη δημιουργία αναφοράς”.

- Restart Program
- Exit Program

---

<sup>11</sup> Ο Fast Fourier Transform είναι ένας αλγόριθμος γρήγορου υπολογισμού του διακριτού μετασχηματισμού Fourier (DFT), που μετατρέπει από πεδίο χρόνου σε πεδίο συχνότητας. [62]

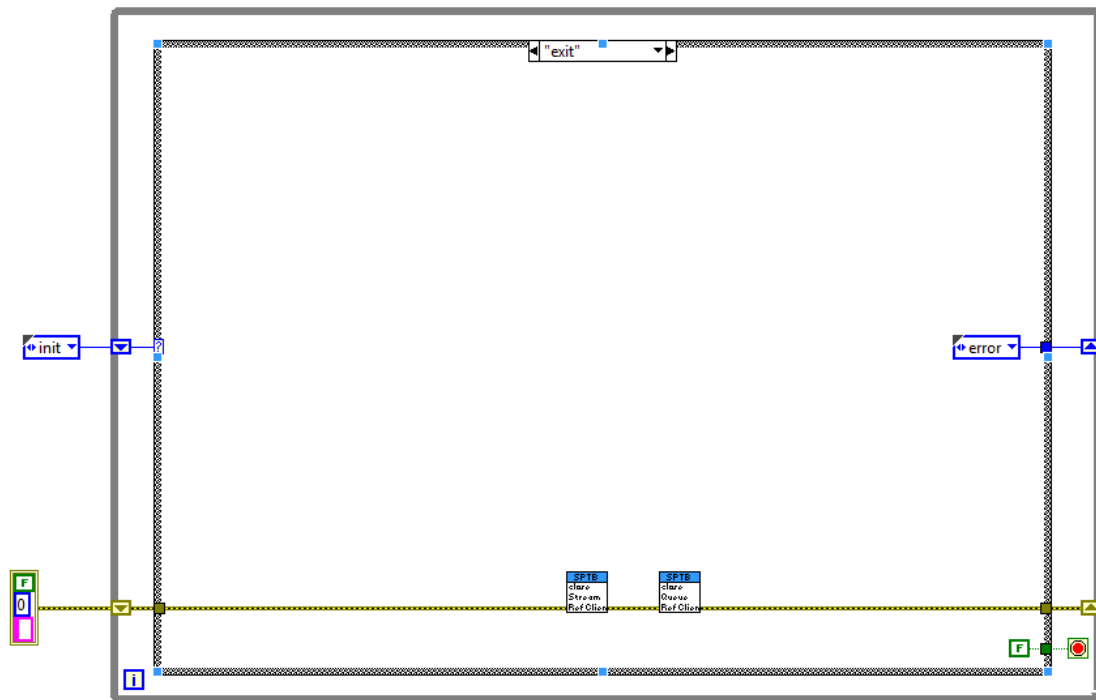


Σχήμα 3. 13: Εφαρμογή client: Post-processing options και Front Panel case με παραλλαγές case

- *Close references*

Σε αυτό το στάδιο κλείνονται όλα τα references. Συγκεκριμένα:

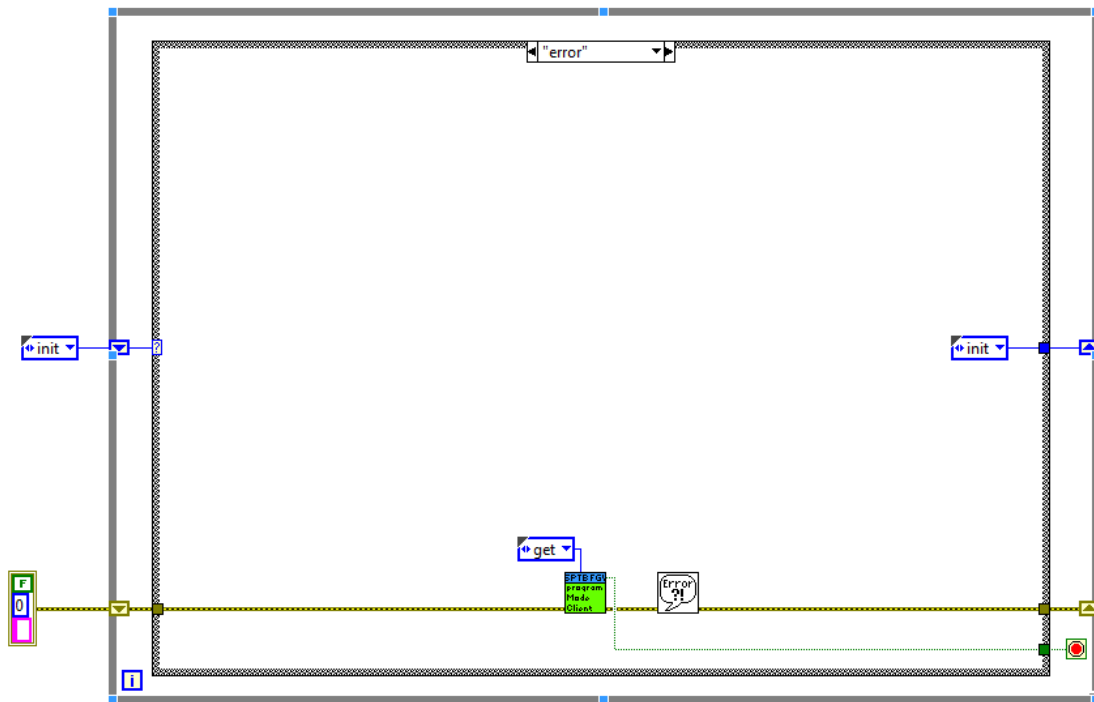
- Το subvi “*closeStreamRefClient*” καταστρέφει τα network stream reader endpoints.
- Το subvi “*closeQueueRefClient*” απελευθερώνει τα references στα διάφορα queues.



Σχήμα 3. 14: Εφαρμογή client – Close references κατάσταση

- **Error**

Όπως και στην server εφαρμογή, στο στάδιο αυτό τρέχει το “*Simple Error Handler*”, που σε περίπτωση σφάλματος εμφανίζει τον κωδικό σφάλματος και την περιγραφή. Όπως αναφέρθηκε, όταν συμβεί ένα σφάλμα σε κάποιο subνι, γίνεται μετάβαση στην κατάσταση “error” και η εκτέλεση διακόπτεται. Ανάλογα την επιλογή που έγινε ανάμεσα στα “Restart Program” και “Exit Program” στο στάδιο “post-processing options και Front-Panel”, το πρόγραμμα είτε σταματά είτε επανεκκινεί.



Σχήμα 3. 15: Εφαρμογή client – Error κατάσταση

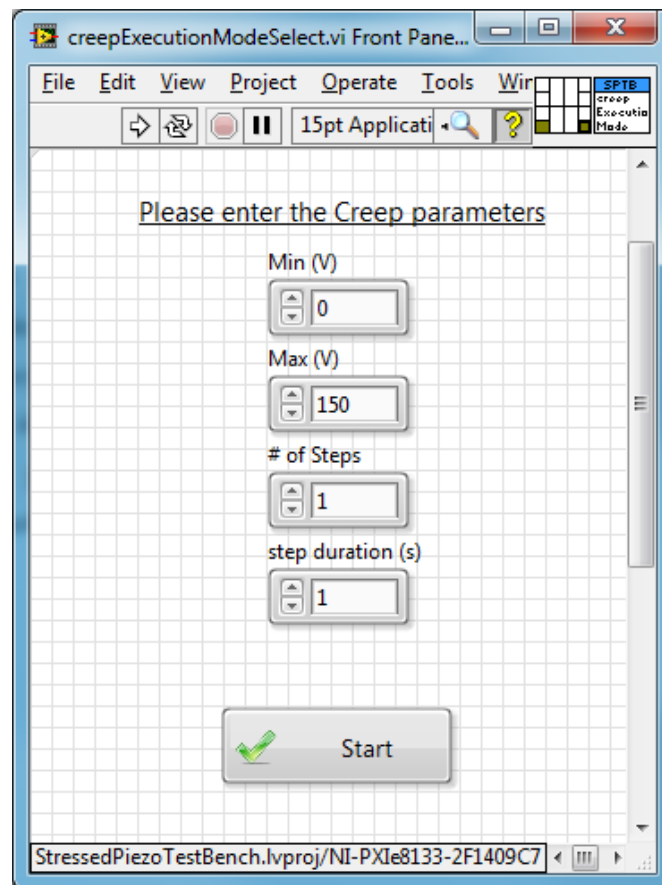
### 3.3 Επιλογές εκτέλεσης

Το πρόγραμμα προσφέρει 3 διακριτά τεστ προς το παρόν. Αυτά είναι τα Creep, static-execution Hysteresis και dynamic-execution Hysteresis τεστ. Η διαφορά ανάμεσα στα static-execution και dynamic-execution Hysteresis τεστ είναι ότι στο πρώτο το φορτίο παραμένει σταθερό καθ' όλη τη διάρκεια του τεστ ενώ στο δεύτερο μεταβάλλεται. Η modular φύση της συγκεκριμένης υλοποίησης (χρήση φωλιασμένων subvnis και αρχιτεκτονικής case structure και state machine) επιτρέπει την εύκολη επέκταση λειτουργικότητας αν είναι επιθυμητό στο μέλλον.

#### 3.3.1 Creep τεστ

Όπως αναφέρεται στο υποκεφάλαιο 2.4 “Λειτουργική ανάλυση”, σε ένα Creep τεστ το πιεζοηλεκτρικό στοιχείο αφήνεται χωρίς φορτίο και χρησιμοποιείται η βοηθητική διάταξη του Σχ. 2.20 για ευκολία. Όσον αφορά το λογισμικό, αυτό σημαίνει ότι σε ένα creep τεστ δεν υπάρχει ανάγκη λειτουργίας του κινητήρα και η μεγαλύτερη πρόκληση είναι η υλοποίηση της επιθυμητής μορφής κυματομορφής διέγερσης του πιεζοηλεκτρικού στοιχείου, ένα παράδειγμα της οποίας φαίνεται στο Σχ. 2.21.

Ο πίνακας ελέγχου (Front Panel) ενός Creep test φαίνεται στο Σχ. 3.16. Οι παράμετροι είναι minimum και maximum τάση διέγερσης, ο αριθμός βημάτων μεταξύ minimum και maximum τάσης (π.χ. στο Σχ. 2.21 είναι 5) και η διάρκεια βήματος σε δευτερόλεπτα. Κατάλληλα subvis αξιοποιούν τις παραμέτρους και δημιουργούν την επιθυμητή κυματομορφή εξόδου, που τροφοδοτεί το πιεζοηλεκτρικό στοιχείο μέσω του slot 9/output 0 του PXI και μέσω του Cedrat LA75B. Όπως περιγράφηκε στο υποκεφάλαιο 2.3.3 “Αισθητήρες και περιφερειακές συσκευές”, το Cedrat LA75B έχει παράγοντα κέρδους 20. Αυτό φυσικά συνυπολογίζεται και ο χρήστης εισάγει τις επιθυμητές τάσεις εξόδου στον πίνακα ελέγχου. Αυτές προγραμματιστικά διαιρούνται με το 20 πριν διοχετευθούν στο Cedrat LA75B.



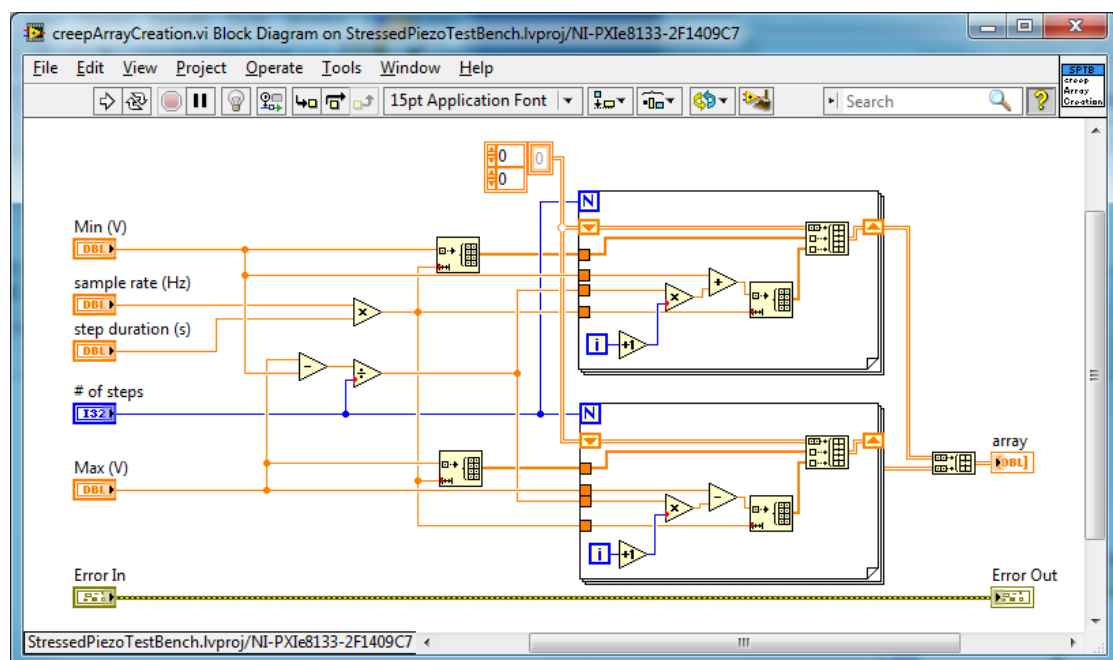
Σχήμα 3. 16: Creep test Front Panel

Η δημιουργία της επιθυμητής κυματομορφής διέγερσης είναι ένα πολύπλοκο θέμα που αναλαμβάνει το “*creepArrayCreation*” subvi (Σχ. 3.17), το οποίο δουλεύει ως εξής: Χρησιμοποιώντας δύο for loops, δημιουργεί δύο ξεχωριστές ακολουθίες που ενώνονται στο τέλος. Η πρώτη ακολουθία είναι η αύξουσα που κινείται από τη

minimum στη maximum τάση. Ξεκινώντας από τη minimum τάση, προσθέτει στον δημιουργημένο πίνακα όσα σημεία με τιμή minimum τάσης όσα χρειάζονται για να επιτευχθεί η ορισμένη διάρκεια βήματος σε δευτερόλεπτα. Ο αριθμός σημείων υπολογίζεται πολλαπλασιάζοντας τη διάρκεια βήματος με το ρυθμό δειγματοληψίας του slot 9 output module (όπως ορίστηκε στο DAQ configuration stage που επεξηγείται αργότερα). Κατόπιν, προσθέτει τον ίδιο αριθμό σημείων αλλά αυτή τη φορά με τιμή του πρώτου επιπέδου πάνω από την minimum τάση. Τα επίπεδα τάσης υπολογίζονται διαιρώντας τη διαφορά μεταξύ maximum και minimum τάσης με τον αριθμό βημάτων και προσθέτοντας το αποτέλεσμα όσες φορές όσες ο αριθμός βημάτων στη minimum τάση μέχρι να επιτευχθεί η maximum (π.χ., αν η minimum τάση είναι 4V, η maximum 10V και ο αριθμός βημάτων 3, τότε τα επίπεδα τάσης είναι 6V, 8V και 10V). Η εναλλαγή μεταξύ minimum τάσης και των διαφόρων επιπέδων τάσης συνεχίζεται μέχρι να επιτευχθεί το maximum επίπεδο τάσης.

Η δεύτερη ακολουθία δημιουργείται με παρόμοιο τρόπο και είναι μια εναλλαγή μεταξύ maximum τάσης και φθινουσών επιπέδων τάσης μέχρι η minimum τάση να επιτευχθεί, όπου κάθε στάδιο διαρκεί όσο η ορισμένη διάρκεια βήματος.

Οι δύο πίνακες στη συνέχεια ενώνονται και το επιθυμητό αποτέλεσμα επιτυγχάνεται (ένα παράδειγμα του οποίου φαίνεται στο Σχ. 2.21).



Σχήμα 3. 17: Δημιουργία Creep κυματομορφής εξόδου



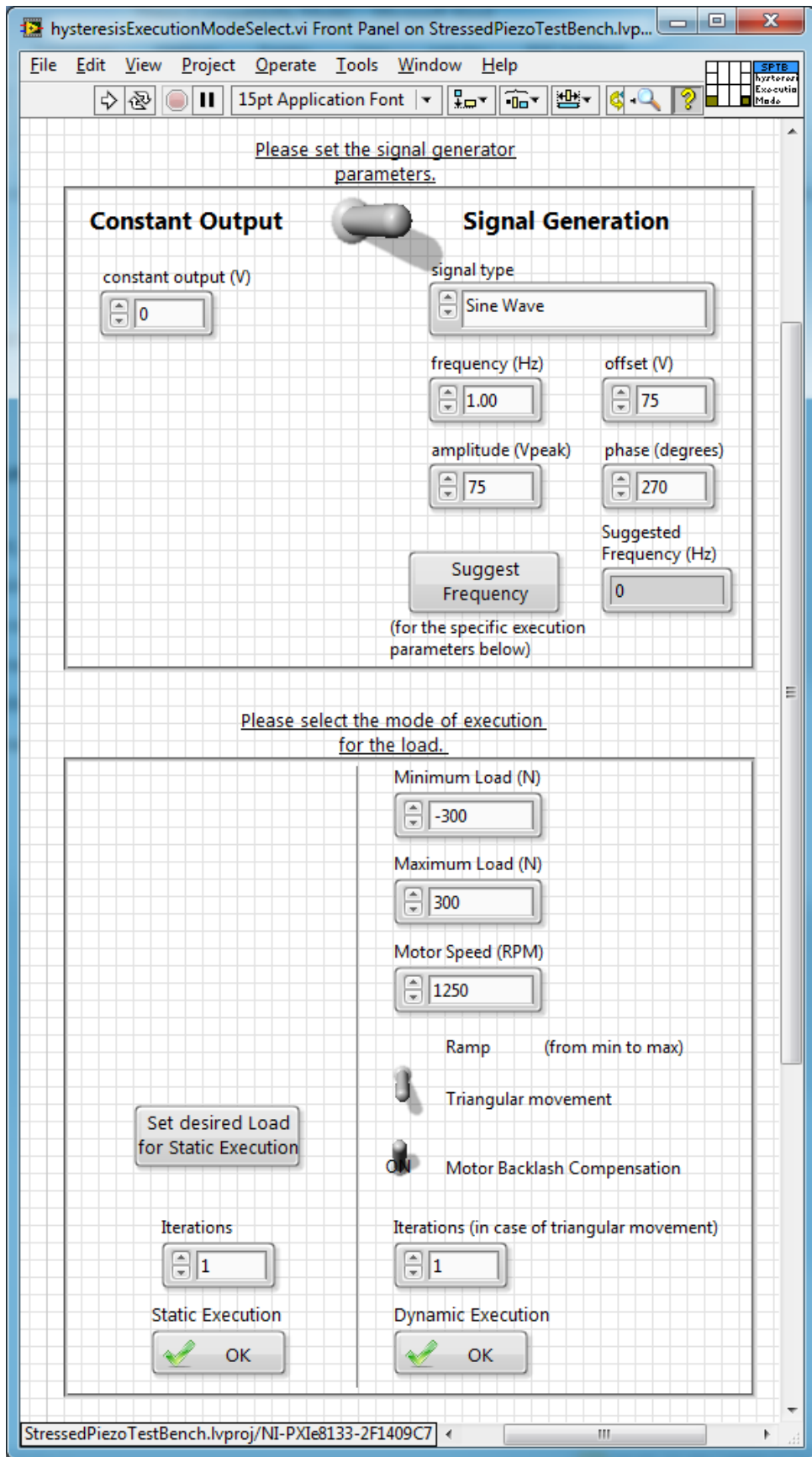
### 3.3.2 *Static-execution Hysteresis* τεστ

Το static-execution Hysteresis τεστ, είναι ένα Hysteresis τεστ όπου το φορτίο παραμένει σταθερό καθ' όλη τη διάρκεια του τεστ. Φυσικά το φορτίο μπορεί να ρυθμιστεί σε οποιοδήποτε επιθυμητό επίπεδο πριν ξεκινήσει η διέγερση του πιεζοηλεκτρικού στοιχείου. Όσον αφορά την έξοδο, μπορεί να είναι είτε μια σταθερή τάση ή μια περιοδική κυματομορφή.

Ο πίνακας ελέγχου (Front Panel) τόσο για το static όσο και για το dynamic execution Hysteresis τεστ είναι ο ίδιος και φαίνεται στο Σχ. 3.18. Το πάνω πλαίσιο περιέχει της παραμέτρους του signal generator και είναι το ίδιο και στις δύο περιπτώσεις, ενώ το κάτω πλαίσιο χωρίζεται σε δύο μέρη, με το αριστερό να αναφέρεται στο static execution και το δεξί στο dynamic.

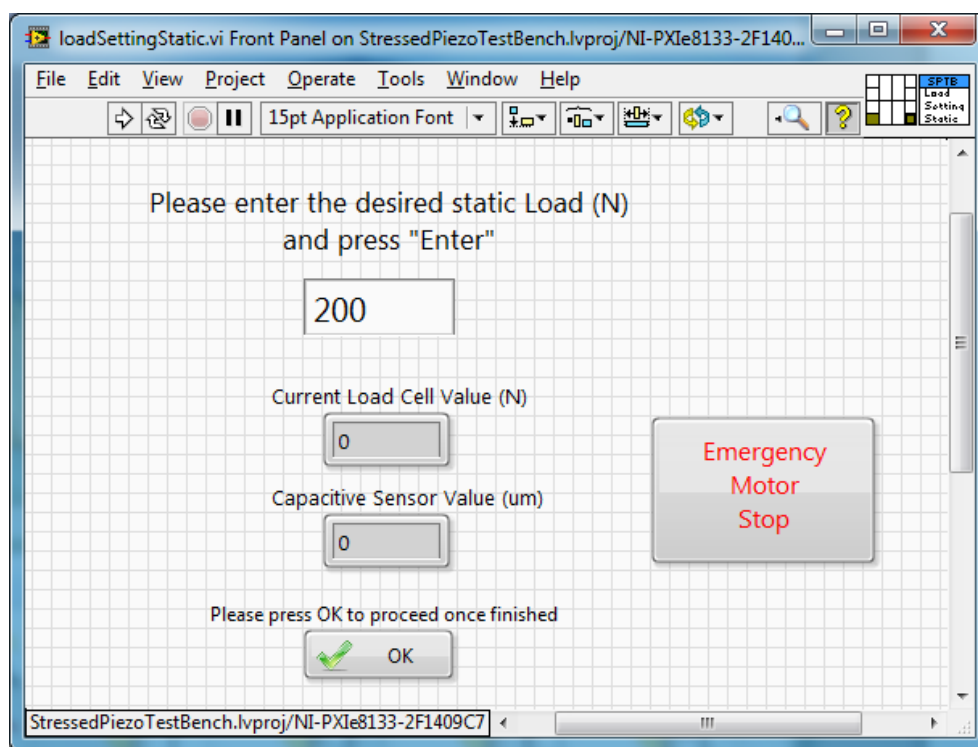
Όσον αφορά τις παραμέτρους του signal generator, αν επιλεγεί το “**Constant Output**”, τότε η μόνη παράμετρος είναι η *επιθυμητή* έξοδος σε voltage (και πάλι, ο παράγοντας κέρδους 20 του Cedrat LA75B λαμβάνεται εσωτερικά υπόψη). Αν επιλεγεί το “**Signal Generation**”, τότε ο χρήστης επιλέγει ανάμεσα σε μία ημιτονοειδή (Sine), τριγωνική (Triangle), τετραγωνική (Square) ή πριονωτή (Sawtooth) κυματομορφή και μετά ορίζει την συχνότητα (Hz), το offset (V), το πλάτος (σε peak voltage, όχι peak-to-peak) και τη φάση (μοίρες). Στην περίπτωση της τετραγωνικής κυματομορφής, το duty cycle είναι προκαθορισμένο στο 50%. Το κουμπί “Suggest Frequency” και ο δείκτης δίπλα του είναι χρήσιμα στο dynamic execution και θα αναλυθούν στην επόμενη ενότητα.

Όσον αφορά τις παραμέτρους του static-execution στην αριστερή πλευρά του κάτω πλαισίου, το κουμπί “Set desired Load” επιτρέπει στον χρήστη να θέσει την επιθυμητή στάθμη φορτίου, ενώ το πεδίο “Iterations” ουσιαστικά ελέγχει τη διάρκεια του τεστ. *Κάθε iteration ισοδυναμεί με 1 δευτερόλεπτο*. Έτσι αν π.χ. επιλεγεί constant output 150V με 10 iterations, το τεστ θα διαρκέσει 10 δευτερόλεπτα στα οποία το πιεζοηλεκτρικό στοιχείο θα τροφοδοτείται με 150V και οι αισθητήρες θα λαμβάνουν δεδομένα. Αν επιλεγεί ένα sine wave με 2Hz, 75V offset, 75V peak, 270° φάση και 10 iterations, το τεστ θα διαρκέσει 10 δευτερόλεπτα κατά τα οποία η έξοδος θα είναι 20 ημιτονοειδή κύματα μεταξύ 0V και 150V και οι αισθητήρες θα λαμβάνουν δεδομένα. Το test ξεκινά όταν πατηθεί το static-execution “OK” κουμπί.



Σχήμα 3. 18: Hysteresis test Front Panel

Το κουμπί “Set desired Load” πυροδοτεί το subvi “loadSettingStatic” που φορτώνει ένα ξεχωριστό πίνακα ελέγχου που φαίνεται στο Σχ. 3.19. Αυτό το subvi συγκρίνει συνεχώς την επιθυμητή τιμή φορτίου με την τρέχουσα και όταν επιτευχθεί σταματά τη λειτουργία του κινητήρα. Στην πραγματικότητα, ένα φωλιασμένο subvi μέσα στο “loadSettingStatic” ελέγχει αν η επιθυμητή τιμή φορτίου είναι ψηλότερα ή χαμηλότερα από την τρέχουσα και αντίστοιχα λειτουργεί τον κινητήρα δεξιόστροφα ή αριστερόστροφα με μια προκαθορισμένη χαμηλή ταχύτητα. Για να λειτουργήσει ο κινητήρας πρέπει να απενεργοποιηθεί το μηχανικό φρένο, να ενεργοποιηθεί ο σερβομηχανισμός και να έχει οριστεί μια κατάλληλη τιμή για την ταχύτητα περιστροφής (οι πιθανές τιμές είναι μεταξύ  $-32,768$  και  $32,767$  όπως εξηγήθηκε στο υποκεφάλαιο 2.3.4). Όταν η επιθυμητή τιμή φορτίου επιτευχθεί, η ταχύτητα περιστροφής μηδενίζεται, ο σερβομηχανισμός απενεργοποιείται και το μηχανικό φρένο ενεργοποιείται. Το κουμπί “Emergency Motor Stop” διακόπτει τη λειτουργία του κινητήρα οποιαδήποτε στιγμή και συνεπώς μπορεί να αποτρέψει ζημιές σε περίπτωση δυσλειτουργίας (π.χ. αν το load cell σταματήσει να λειτουργεί). Όταν ολοκληρωθεί η ρύθμιση του φορτίου, το κουμπί “OK” κλείνει το subvi.



Σχήμα 3. 19: Static-execution load setting Front Panel

### 3.3.3 *Dynamic-execution Hysteresis* τεστ

Το dynamic-execution Hysteresis τεστ διαφέρει από το static από την άποψη ότι το φορτίο μεταβάλλεται κατά τη διάρκεια του τεστ. Μια χαρακτηριστική διαφορά είναι επίσης είναι ότι η διάρκεια του τεστ είναι σχετική. Αυτό συμβαίνει γιατί η συνθήκη τερματισμού δεν είναι η επίτευξη συγκεκριμένων iterations όπως στο static execution, αλλά η ολοκλήρωση της επιθυμητής μεταβολής στο φορτίο. Παρότι η ταχύτητα του κινητήρα μπορεί να ελεγχθεί με ακρίβεια, ιδιότητες όπως το motor backlash (όπως ορίστηκε στο 2.3.2) σημαίνουν ότι η διάρκεια δεν είναι ντετερμινιστικό μέγεθος.

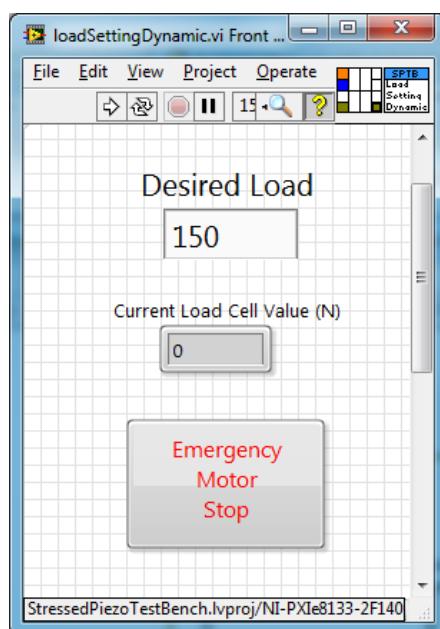
Ο πίνακας ελέγχου του dynamic-execution Hysteresis τεστ είναι, όπως αναφέρθηκε, ο ίδιος με του static-execution και φαίνεται στο Σχ. 3.18. Οι παράμετροι του dynamic-execution βρίσκονται στη δεξιά πλευρά του κάτω πλαισίου και το τεστ ξεκινά όταν πατηθεί το πλήκτρο dynamic-execution “OK”.

Η χρήστης επιλέγει ανάμεσα σε “Ramp” και “Triangular movement” μορφή για το φορτίο. Η διαφορά είναι ότι στο “Ramp”, το φορτίο πηγαίνει από τη minimum τιμή στη maximum με την ορισμένη ταχύτητα και το τεστ σταματά, ενώ στο “Triangular movement”, το φορτίο πηγαίνει από τη minimum τιμή στη maximum και ξανά πίσω στη minimum με την ορισμένη ταχύτητα και αυτό συμβαίνει όσες φορές ορίζεται στο πεδίο “Iterations”.

Όταν πατηθεί το κουμπί dynamic-execution “OK” και πριν ενεργοποιηθεί η έξοδος και οι αισθητήρες, ο κινητήρας λειτουργεί μέχρι να επιτευχθεί η minimum τιμή φορτίου (“loadSettingDynamic” subv1, Σχ. 3.20. Σε περίπτωση ανάγκης ο χρήστης μπορεί να σταματήσει τον κινητήρα με το κουμπί “Emergency Motor Stop”).

Αξίζει να αναφερθεί ότι το backlash εξαρτάται από την κατεύθυνση κίνησης του κινητήρα. Είναι αδύνατο να εξαλειφθεί το backlash ωστόσο μπορούμε να κάνουμε το τεστ με τέτοιο τρόπο ώστε να είναι προβλέψιμο. Αυτό κάνει ο διακόπτης “Motor Backlash Compensation”. Διασφαλίζει ότι η minimum τιμή φορτίου επιτυγχάνεται από μια υψηλότερη τιμή φορτίου. Αν π.χ. το αρχικό φορτίο είναι 200N, το επιθυμητό minimum 300N και ο διακόπτης “Motor Backlash Compensation” ενεργοποιημένος, το φορτίο θα αυξηθεί αρχικά από τα 200N στα 350N (είναι προκαθορισμένο να πηγαίνει 50N υψηλότερα από την επιθυμητή) πριν υποχωρήσει στα 300N. Αν ήταν

απενεργοποιημένος, το φορτίο θα αυξανόταν από τα 200N στα 300N και θα σταματούσε. Αν η αρχική τιμή φορτίου είναι υψηλότερα από την επιθυμητή minimum, τότε ανεξαρτήτως θέσεως του διακόπτη, το φορτίο θα μειωθεί απευθείας στην επιθυμητή minimum τιμή. Αυτή η λειτουργία δεν είναι απαραίτητη στο “Ramp” αλλά είναι αναγκαία στο “Triangular movement” καθώς κάνει το backlash παρόμοιο σε κάθε iteration. Αν ήταν απενεργοποιημένη και η αρχική τιμή φορτίου ήταν χαμηλότερα από την επιθυμητή minimum, τότε η minimum τιμή φορτίου θα επιτυγχανόταν από χαμηλότερη τιμή φορτίου, το τεστ θα ξεκινούσε, το φορτίο θα έφτανε τη maximum τιμή και θα υποχωρούσε στη minimum, μόνο που αυτή τη φορά η minimum τιμή φορτίου θα επιτυγχανόταν από υψηλότερη τιμή φορτίου και το backlash θα ήταν διαφορετικό.



Σχήμα 3. 20: Dynamic-execution load setting Front Panel

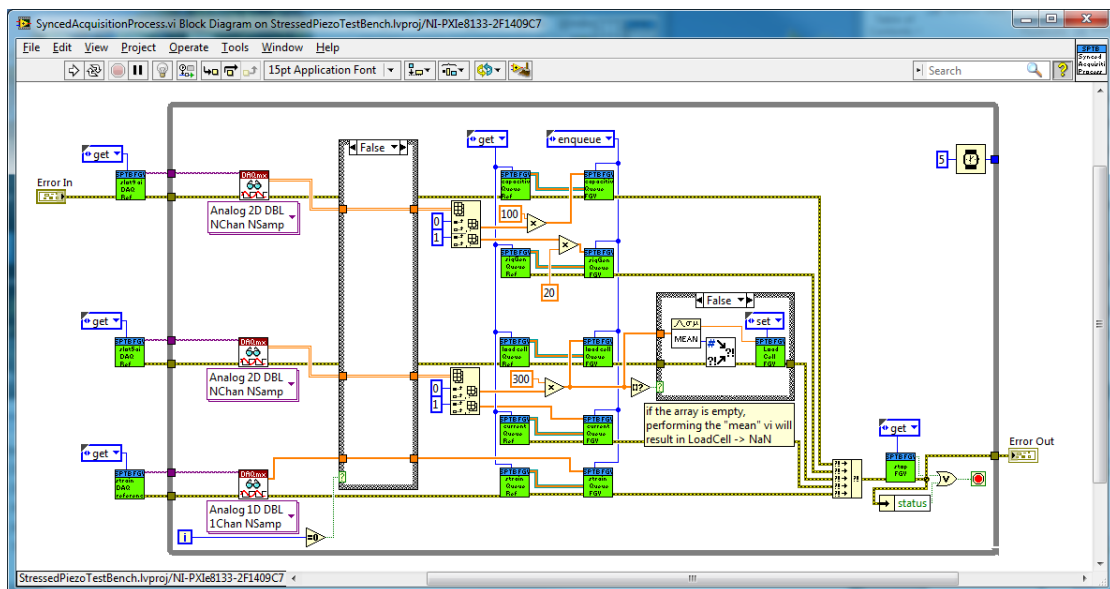
Οι παράμετροι του signal generator λειτουργούν όπως εξηγήθηκε προηγουμένως αλλά μια σημαντική προσθήκη είναι το κουμπί “Suggest Frequency” και η αντίστοιχη ένδειξη. Αυτή η λειτουργία είναι χρήσιμη στην περίπτωση “Triangular Movement” και όταν πατηθεί το κουμπί, πυροδοτείται το “*suggestFrequency*” subvi, που εκτελεί την κίνηση κινητήρα με τις παραμέτρους εκτέλεσης του κάτω πλαισίου αλλά με την έξοδο και τους αισθητήρες απενεργοποιημένους. Μετρά το χρόνο σε δευτερόλεπτα που χρειάζεται για να ολοκληρωθεί η κίνηση και τον διαιρεί με τον αριθμό iterations.

Το αντίστροφο του αποτελέσματος είναι η συχνότητα της τριγωνικής κίνησης του φορτίου και παρουσιάζεται στην ένδειξη. Έτσι, ο χρήστης μπορεί να θέσει το signal generator ως τριγωνική κυματομορφή με την ίδια συχνότητα, ώστε το φορτίο και η έξοδος να ακολουθούν την ίδια τριγωνική μορφή με την ίδια συχνότητα. Όπως αναφέρθηκε, εξαιτίας της σχετικής φύσης του backlash, είναι πιο ακριβές να εκτελέσουμε μια συγκεκριμένη μεταβολή φορτίου και να μετρήσουμε τη διάρκειά της παρά να προσπαθήσουμε να την υπολογίσουμε με μαθηματικό τρόπο.

### 3.4 Διαχείριση δεδομένων

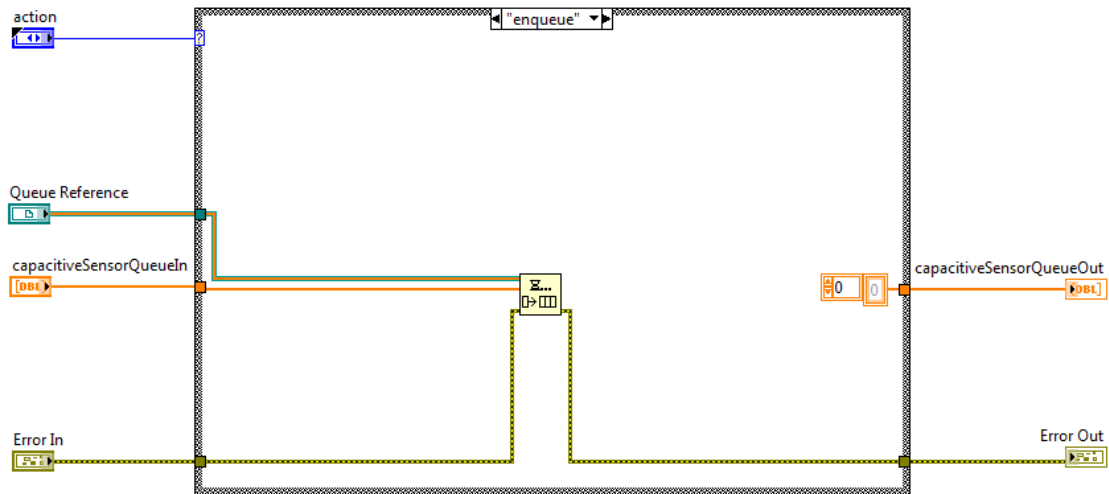
Όπως αναφέρθηκε, η διαχείριση των δεδομένων επιτυγχάνεται με τη χρήση queues, network streams και tsν αρχείων. Για να γίνουν πιο κατανοητοί οι μηχανισμοί, κρίνεται σκόπιμο να μελετηθεί η ροή των δεδομένων από την αρχή ως το τέλος.

Στην εφαρμογή server, τα δεδομένα που συλλέγονται από τα “DAQmx Read” vis στο στάδιο “data acquisition και data transmission” τοποθετούνται στα αντίστοιχα queues με τη βοήθεια κατάλληλων FGVs. Συγκεκριμένα, τα δεδομένα από τα capacitive sensor, signal generator feedback, load cell, current και strain gage συλλέγονται και τοποθετούνται στα queues στο “SyncedAcquisitionProcess” subvi (Σχ. 3.21) και ο τύπος τους είναι *1-dimensional array of double (1-d array of DBL)*.



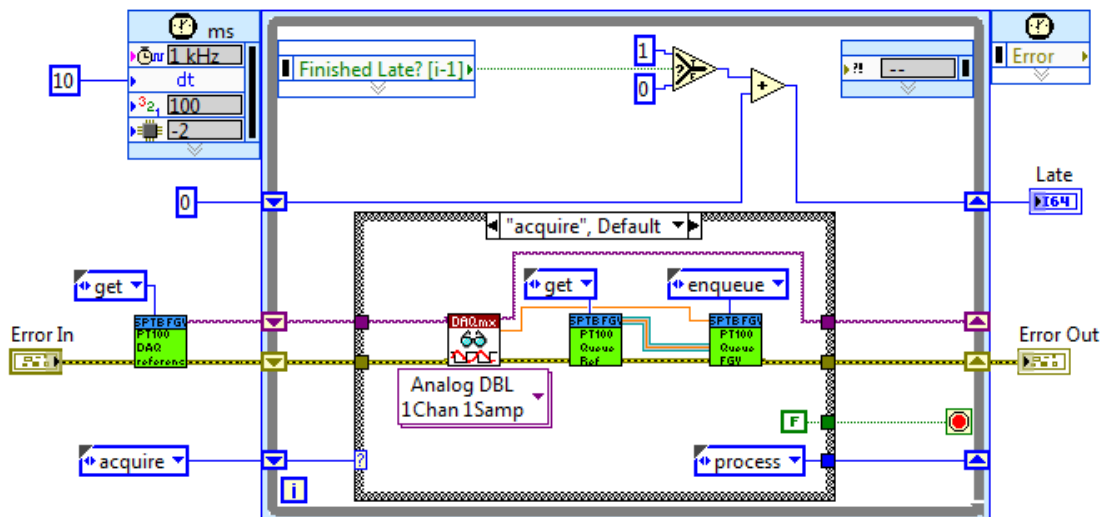
Σχήμα 3. 21: “SyncedAcquisitionProcess” subvi

Όλα τα data FGVs στο project έχουν δύο καταστάσεις – **enqueue** και **dequeue**. Παρότι αλλάζει ο τύπος δεδομένων για τα διάφορα FGVs, η κατάσταση enqueue είναι παρόμοια σε όλα και απλά περιλαμβάνει μια “*Enqueue Element*” ρουτίνα, ένα παράδειγμα της οποίας φαίνεται στο Σχ. 3.22, από το capacitive sensor FGV.



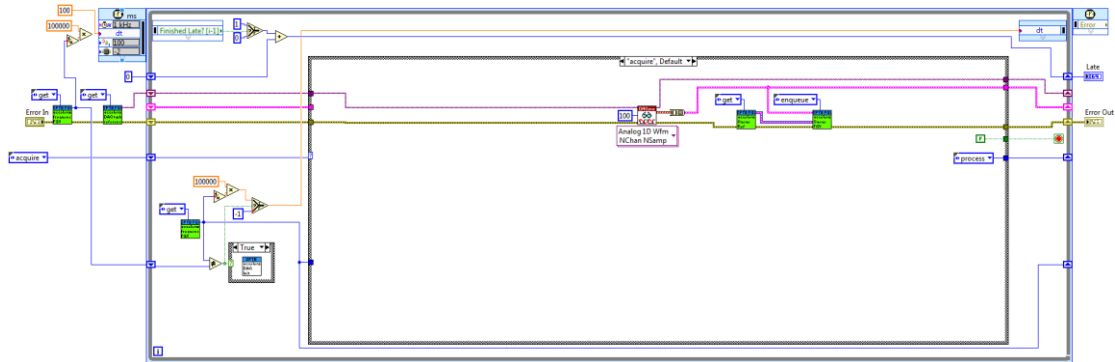
Σχήμα 3. 22: “Enqueue” κατάσταση των data FGVs στην εφαρμογή server

Δεδομένα από το PT100 τύπου *double* συλλέγονται στο “*PT100AcquisitionProcess*” subνι και τοποθετούνται στο queue του αντίστοιχου FGV (Σχ. 3.23)

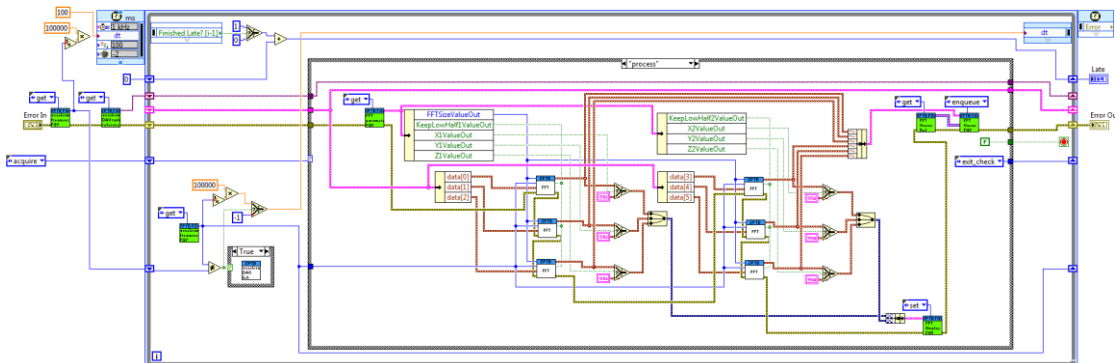


Σχήμα 3. 23: “PT100AcquisitionProcess” subvi

Τέλος, δεδομένα των δύο επιταχυνσιόμετρων συλλέγονται από το “*accelerometerAcquisitionProcess*” subvi. Ο τύπος τους είναι *cluster of 6 waveforms of double (cluster of 6 wvf(DBL))* και τοποθετούνται στο queue του αντίστοιχου FGV (Σχ. 3.24). Αυτό το subvi έχει ακόμα μια κατάσταση που ακολουθεί τη συλλογή δεδομένων και εκτελεί το Fast Fourier Transform (FFT) και τοποθετεί τα δεδομένα στο αντίστοιχο FGV (Σχ. 3.25).



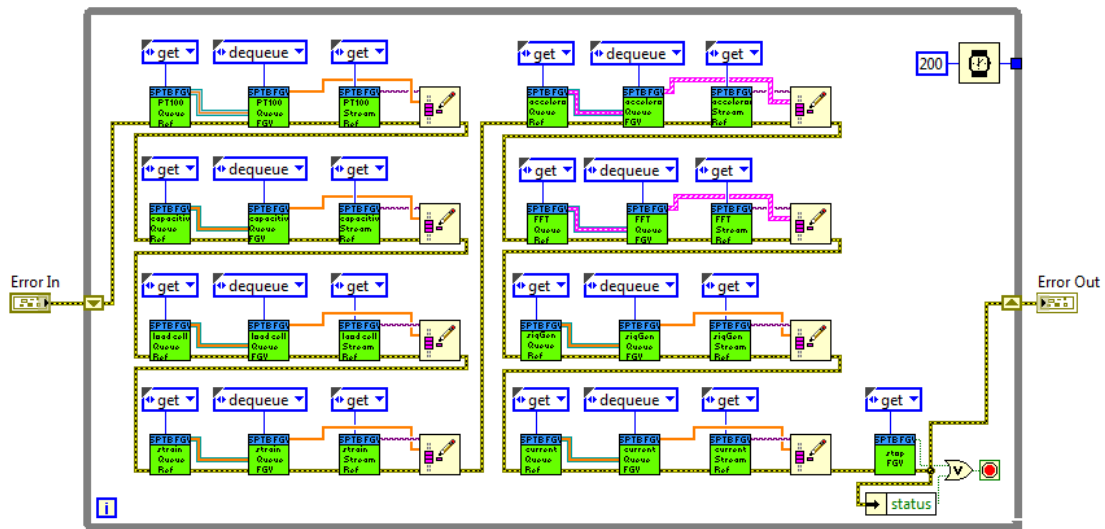
Σχήμα 3. 24: “*accelerometerAcquisitionProcess*” subvi



Σχήμα 3. 25: “*accelerometerAcquisitionProcess*” subvi - FFT

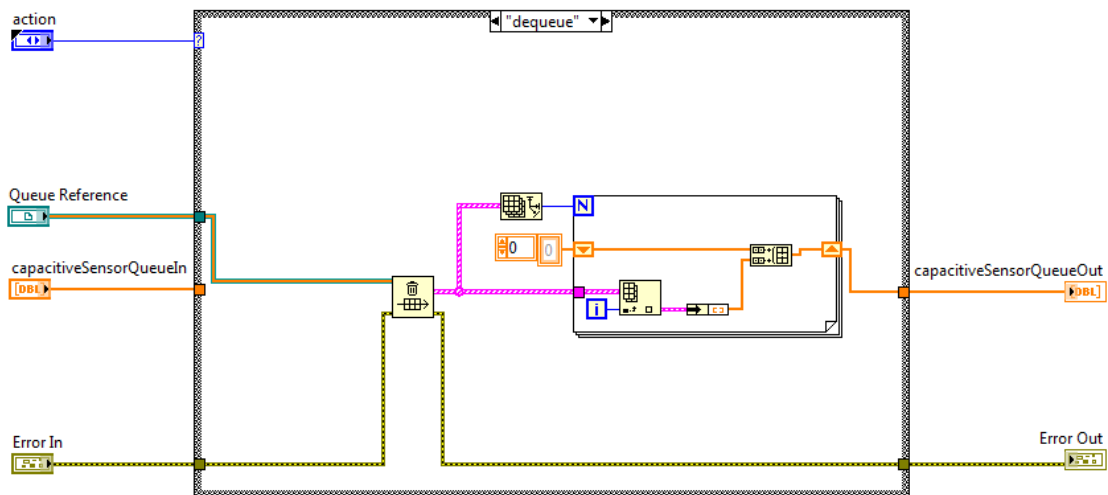
Τα δεδομένα που συλλέγονται τοποθετούνται συνέχεια στα queues και ανά τακτά χρονικά διαστήματα αποστέλλονται στην εφαρμογή client. Η λογική είναι ότι τα δεδομένα συλλέγονται χωρίς διακοπή ενώ μια παράλληλη διεργασία αποστέλλει πακέτα κάθε τόσο. Ο ρυθμός αποστολής είναι χαμηλότερος του ρυθμού συλλογής. Η αποστολή συμβαίνει στο “*dataSend*” subvi που βρίσκεται στο ίδιο στάδιο με τη συλλογή δεδομένων στην εφαρμογή server (Σχ. 3.26). Τα δεδομένα αφαιρούνται από τα queues με την “*dequeue*” κατάσταση των data FGVs και αποστέλλονται μέσω των network stream στον client με τη ρουτίνα “*Write Single Element to Stream*”.





Σχήμα 3. 26: Αποστολή δεδομένων μέσω network stream

Ενώ η κατάσταση “enqueue” ήταν παρόμοια για όλα τα data FGVs, η “dequeue” είναι πιο πολύπλοκη καθώς περιέχει τη δημιουργία των πακέτων δεδομένων. Η κατάσταση “dequeue” των capacitive sensor, signal generator feedback, load cell, current και strain gage απεικονίζεται στο Σχ. 3.27.

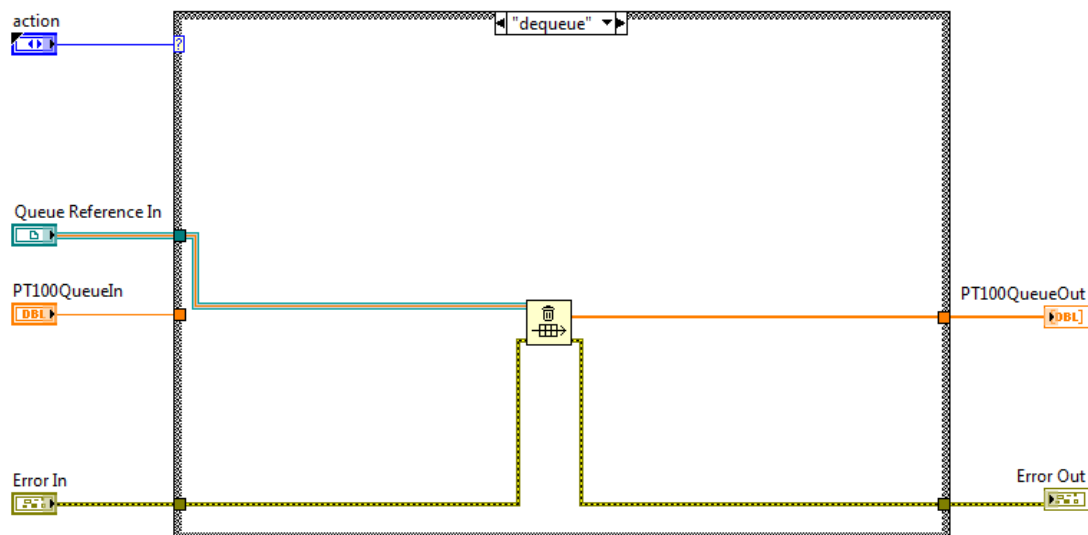


Σχήμα 3. 27: “Dequeue” κατάσταση για τα περισσότερα data FGVs στην εφαρμογή server

Αρχικά, όλα τα διαθέσιμα στοιχεία του queue αφαιρούνται με τη λειτουργία “Flush Queue”. Ωστόσο αυτό δεν είναι αρκετό καθώς ο επιστρεφόμενος τύπος δεδομένων είναι 1-dimensional array of cluster of 1 element of 1-dimensional array of double.

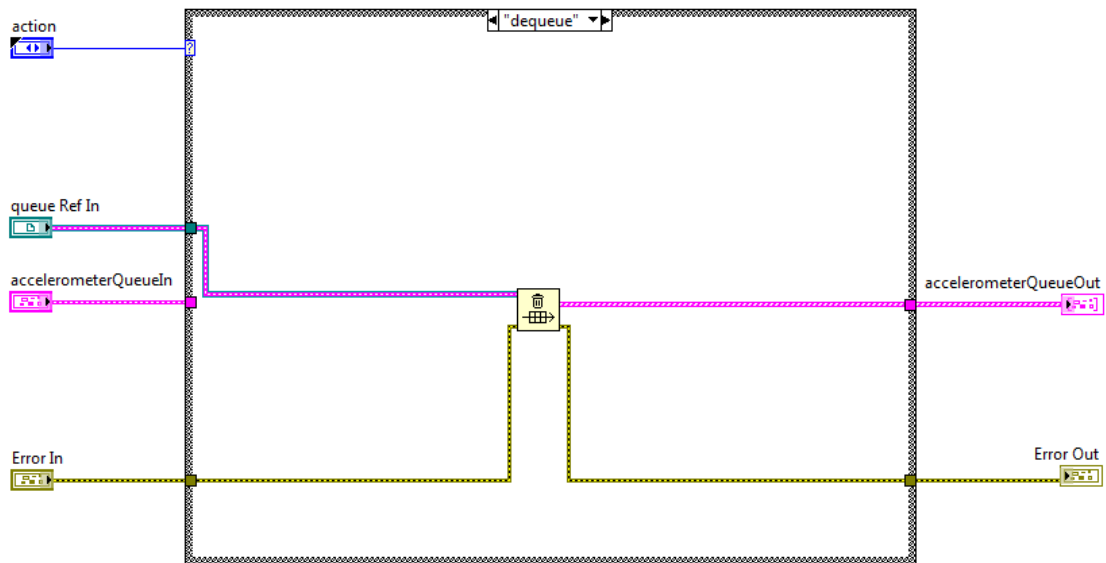
Αυτό συμβαίνει γιατί όπως προαναφέρθηκε κάθε στοιχείο του queue είναι τύπου *1-dimensional array of double*. Εμείς επιθυμούμε να καταλήξουμε με δεδομένα τύπου *1-dimensional array of double* όπου όλα τα στοιχεία queue array έχουν τοποθετηθεί σε ένα μοναδικό array το ένα μετά το άλλο. Για να επιτευχθεί αυτό, αναπτύχθηκε το κομμάτι κώδικα δίπλα στη λειτουργία “Flush Queue” και το αποτέλεσμα είναι *1-dimensional array* που περιλαμβάνει όλα τα στοιχεία του queue τοποθετημένα το ένα μετά το άλλο.

Η κατάσταση είναι πιο απλή όσον αφορά το PT100 FGV. Τα στοιχεία queue του PT100 FGV είναι τύπου *double*, όπως αναφέρθηκε. Χρησιμοποιώντας τη ρουτίνα “Flush Queue”, το αποτέλεσμα είναι το επιθυμητό *1-dimensional array of double* που περιέχει όλα τα στοιχεία του queue τοποθετημένα το ένα μετά το άλλο.



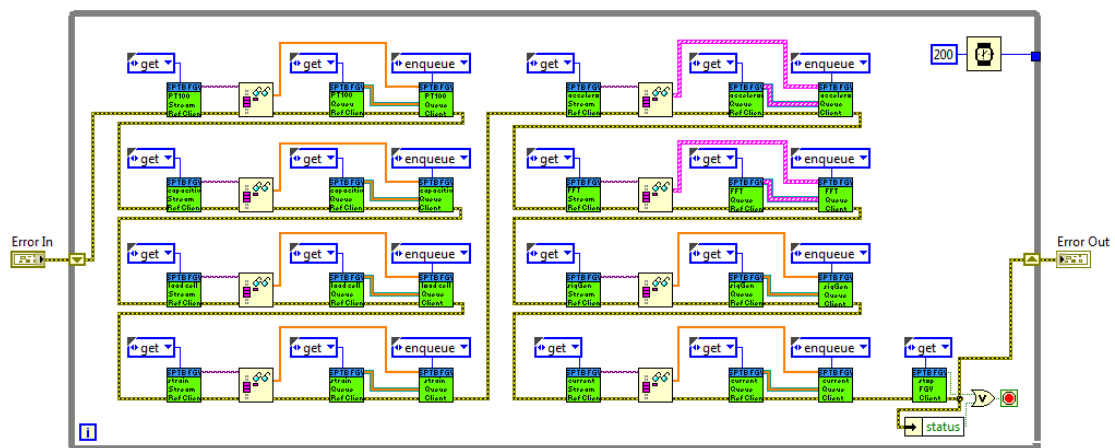
Σχήμα 3. 28: PT100 “Dequeue” κατάσταση στην εφαρμογή server

Το ίδιο ισχύει και για τα FGVs των επιταχυνσίόμετρων και FFT παρά το γεγονός ότι ο τύπος δεδομένων είναι διαφορετικός. Συγκεκριμένα, ο τύπος των στοιχείων των queues και των δύο FGVs είναι όπως αναφέρθηκε *cluster of 6 waveforms of double*. Χρησιμοποιώντας τη ρουτίνα “Flush Queue” καταλήγουμε με το επιθυμητό *1-dimensional array of cluster of 6 waveforms of double*. Η κατάσταση “dequeue” απεικονίζεται στο Σχ. 3.29.



Σχήμα 3. 29: “Dequeue” FGV κατάσταση των επιταχυνσιόμετρων και FFT στη server εφαρμογή

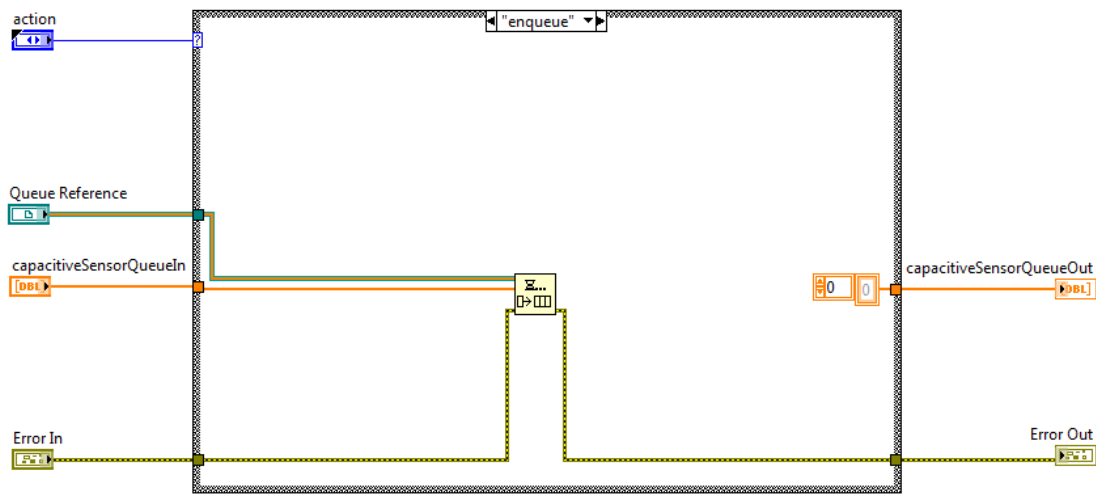
Τα δεδομένα που στέλνονται μέσω network stream λαμβάνονται από την client εφαρμογή στο “dataReceive” subνι (Σχ. 3.30). Στη συνέχεια τοποθετούνται στα queues του client με τη βοήθεια κατάλληλων FGVs. Τα FGVs αυτά έχουν και πάλι 2 καταστάσεις – “enqueue” και “dequeue” και αναλύονται παρακάτω.



Σχήμα 3. 30: Λήψη δεδομένων στη client εφαρμογή

Η κατάσταση “enqueue” είναι παρόμοια για όλα τα data FGVs και περιλαμβάνει μια “Enqueue Element” ρουτίνα. Ο τύπος δεδομένων διαφέρει στα FGVs και στα capacitive sensor, signal generator feedback, load cell, current, strain gage και PT100 είναι *1-dimensional array of double*, ενώ στα επιταχυνσιόμετρα και FFT είναι *1-*

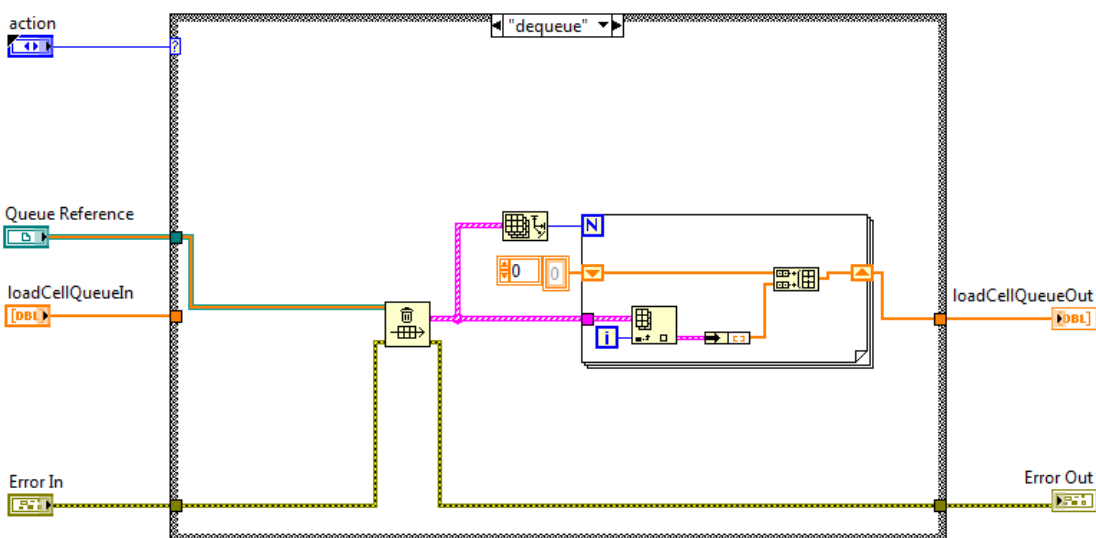
dimensional array of cluster of 6 waveforms of double. Ένα παράδειγμα κατάστασης “enqueue” από το capacitive sensor FGV φαίνεται στο Σχ. 3.31.



Σχήμα 3. 31: “Enqueue” κατάσταση των data FGVs στην client εφαρμογή

Η ιδέα στην client εφαρμογή είναι ότι τα νέα στοιχεία που λαμβάνονται μέσω network stream τοποθετούνται ανά τακτά χρονικά διαστήματα σε queues ενώ ένα παράλληλο VI γράφει τα δεδομένα στα tsv αρχεία. Για το τελευταίο βήμα εκτελείται ένα “dequeue” στα FGVs και τα δεδομένα που εξάγονται γράφονται στα tsv αρχεία.

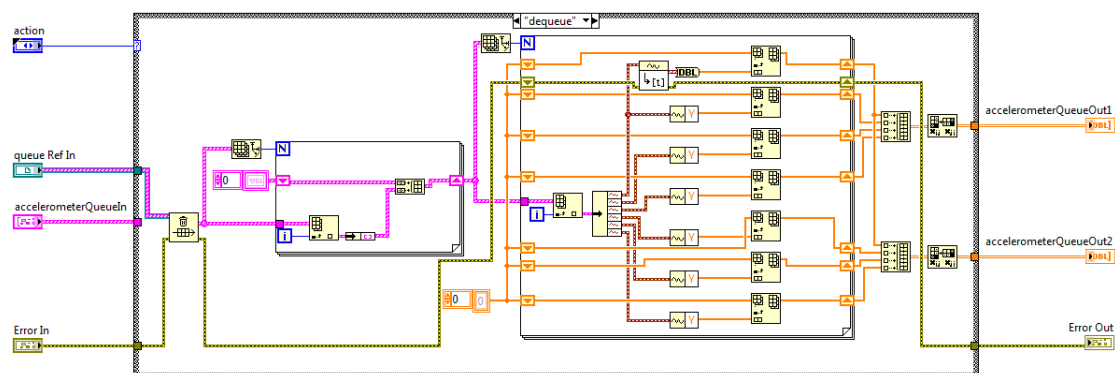
Η “dequeue” κατάσταση είναι παρόμοια για τα capacitive sensor, signal generator feedback, load cell, current, strain gage και PT100 FGVs και φαίνεται στο Σχ. 3.32.



Σχήμα 3. 32: “Dequeue” κατάσταση για τα περισσότερα data FGVs στην client εφαρμογή

Ο κώδικας είναι παρόμοιος με την “dequeue” κατάσταση πολλών FGVs στην server εφαρμογή που αναλύθηκε παραπάνω και το αποτέλεσμα είναι *1-dimensional array of double* που περιέχει όλα τα 1-dimensional array στοιχεία του queue τοποθετημένα το ένα μετά το άλλο σε ένα ενιαίο πίνακα.

Η κατάσταση είναι διαφορετική όσον αφορά την “dequeue” κατάσταση των FGVs των επιταχυνσιόμετρων και FFT, που φαίνεται στο Σχ. 3.33.

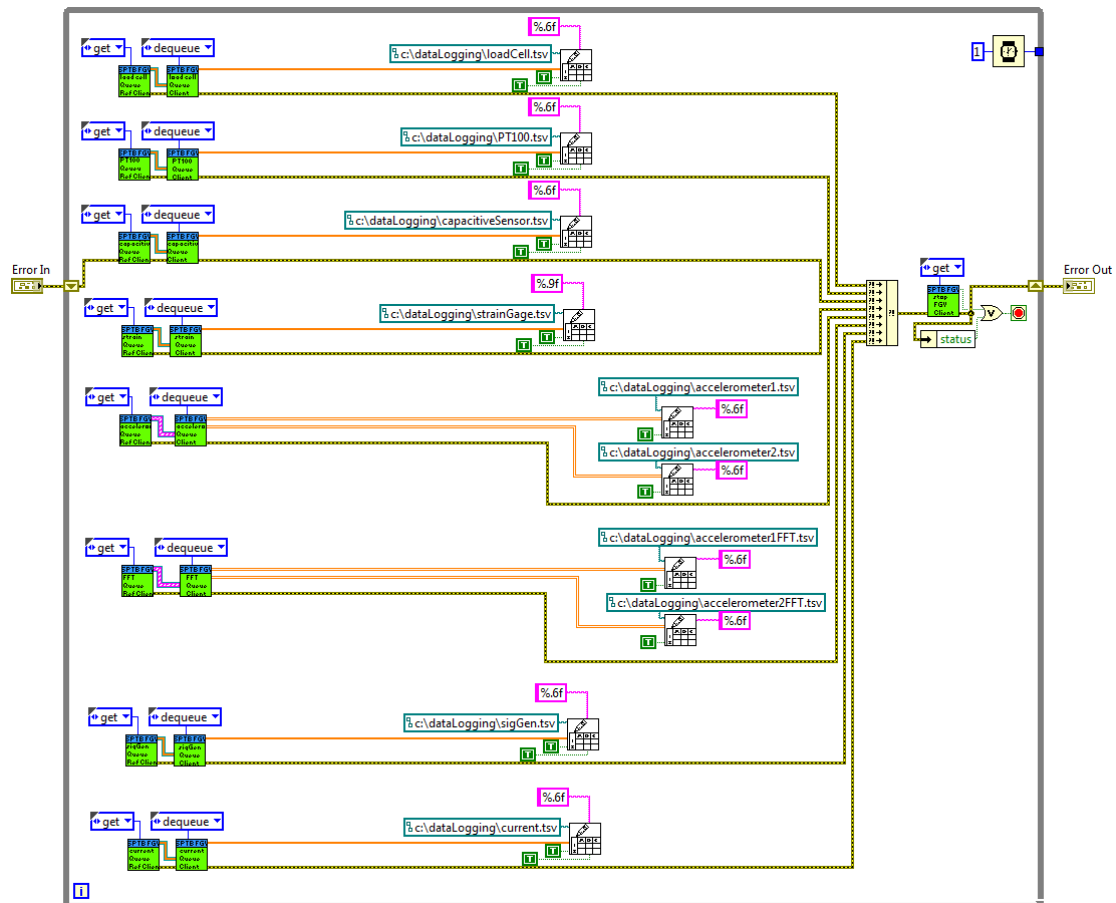


Σχήμα 3.33: “Dequeue” κατάσταση FGV επιταχυνσιόμετρων και FFT στην client εφαρμογή

Ο παραπάνω κώδικας αφαιρεί όλα τα στοιχεία από τα queues τύπου *1-dimensional array of cluster of 6 waveforms of double* και τα τοποθετεί το ένα μετά το άλλο σε ένα νέο 1-dimensional array (of clusters of 6 waveforms of double). Ένα for loop, που τρέχει όσες φορές όσες το μέγεθος του πίνακα, χωρίζει κάθε cluster στα 6 waveforms (X,Y,Z άξονες για τα επιταχυνσιόμετρα 1 και 2 αντίστοιχα) και δημιουργεί ένα νέο πίνακα για καθένα από αυτά που περιέχει τα αντίστοιχα δεδομένα από όλα τα clusters τοποθετημένα το ένα μετά το άλλο. Τελικά, οι τρεις άξονες κάθε επιταχυνσιόμετρου μαζί με το χρόνο (για FGV επιταχυνσιόμετρου) ή τη συχνότητα (για FFT FGV) συσκευάζονται ώστε να δημιουργήσουν δύο ξεχωριστούς *2-dimensional arrays of double* 4 στηλών (χρόνος ή συχνότητα, X, Y, Z). Με άλλα λόγια, το τελικό αποτέλεσμα είναι δύο ξεχωριστοί 2-d arrays (ένας για κάθε επιταχυνσιόμετρο) καθένας με τα στοιχεία X, Y, Z και χρόνο ή συχνότητα όλων των διαθέσιμων δεδομένων στο queue.

Αφότου τα δεδομένα αφαιρεθούν από το queue στο “dataLoggingProcessClient” subvi (Σχ. 3.34), αποθηκεύονται στα αντίστοιχα tsv αρχεία με τη ρουτίνα “Write To Spreadsheet File”. Τα tsv έχουν ένα προκαθορισμένο directory και η data logging

διαδικασία γίνεται παράλληλα και όχι σειριακά με τη λήψη δεδομένων καθώς γενικά η εγγραφή στο σκληρό δίσκος είναι μια χρονοβόρα διαδικασία. Με τη συγκεκριμένη υλοποίηση, η λήψη δεδομένων συμβαίνει αδιάκοπα ανά τακτά χρονικά διαστήματα και το data logging συμβαίνει όταν οι αναγκαίοι υπολογιστικοί πόροι είναι διαθέσιμοι, γράφοντας όλα τα διαθέσιμα δεδομένα τη συγκεκριμένη στιγμή στα tsv αρχεία.



Σχήμα 3. 34: Data logging στην client εφαρμογή

Όταν ολοκληρωθεί η λήψη και το logging των δεδομένων, τα tsv αρχεία μπορούν να αντιγραφούν σε νέα τοποθεσία στο σκληρό δίσκο, χρησιμοποιηθούν για τη σχεδίαση των γραφικών παραστάσεων ή για την παραγωγή των αναφορών με τη βοήθεια των Matlab scripts. Οι δύο τελευταίες περιπτώσεις καλύπτονται με λεπτομέρεια στις επόμενες ενότητες.

### 3.5 Συγχρονισμός λήψης δεδομένων

Ο συγχρονισμός των συσκευών λήψης δεδομένων στο PXI αποδείχθηκε μια εκ των μεγαλύτερων προκλήσεων. Ήταν απαραίτητο να διασφαλιστεί ότι η λήψη είναι συγχρονισμένη, καθώς διαφορετικά η διαδικασία χαρακτηρισμού δεν θα ήταν αξιόπιστη και λάθος συμπεράσματα μπορεί να προέκυπταν.

Οι συσκευές που απαραίτητα πρέπει να είναι συγχρονισμένες μεταξύ τους είναι τα slots 5 (PXI-4461), 7 (PXIe-4330) και 9 (PXI-4461). Το slot 5 λαμβάνει μετρήσεις από τα load cell και current sensing board, το slot 7 από το strain gage ενώ το slot 9 από το capacitive sensor και signal generator feedback καθώς και παράγει την έξοδο. Όλες αυτές επιστρατεύουν delta-sigma analog-to-digital converters, γεγονός που επιβάλλει περιορισμούς στη διαδικασία συγχρονισμού. Οι συσκευές στα slot 2 (PXIe-4357), που λαμβάνει μετρήσεις από το PT100, και slot 3 (PXIe-4496), που λαμβάνει μετρήσεις από τα δύο επιταχυνσιόμετρα, δεν απαιτούν το ίδιο επίπεδο ακρίβειας στο συγχρονισμό και αυτός είναι ο λόγος που τα tasks τους δημιουργούνται ξεχωριστά στα “PT100DAQInit” και “accelerometerDAQInit” subvis.

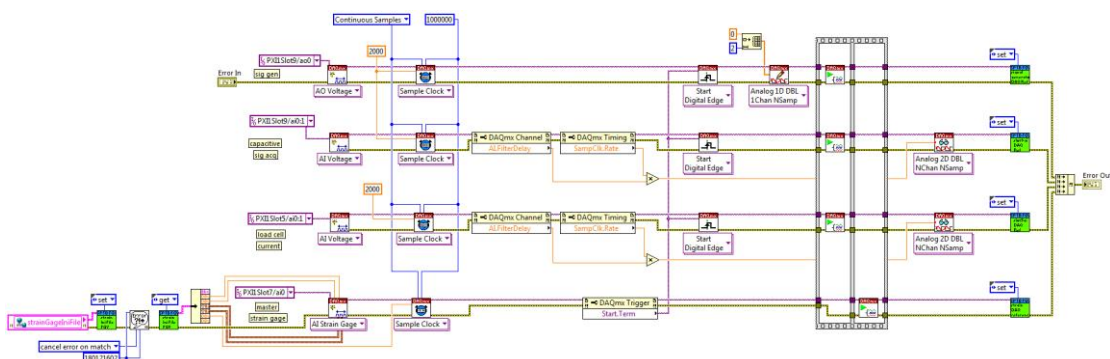
Ο συγχρονισμός των slots 5, 7 και 9 επιτυγχάνεται στο “initSyncStartStatic” για τεστ Creep και static-execution Hysteresis και στο “initSyncStartDynamic” για τεστ dynamic-execution Hysteresis αντίστοιχα. Η υλοποίηση αυτή επιτρέπει συγχρονισμό πολλαπλών DSA καρτών με διαφορετικό ρυθμό δειγματοληψίας η κάθε μία.

Και στα 2 προαναφερθέντα subvis, το PXIe-4330 του slot 7 ορίστηκε ως η master συσκευή. Τα slots 5 και 9 ορίστηκαν ως slave. Οι συσκευές συγχρονίζονται με τη μέθοδο reference clock synchronization. Και η master και οι slave συσκευές κλειδώνουν τα analog-to-digital converters και digital-to-analog converters τους στο ρυθμό ενός shared reference clock στο πίσω μέρος του PXIe chassis. Για την επίτευξη του συγχρονισμού, τα slave slots δέχονται ένα σήμα που παράγεται από τη master συσκευή ως το start trigger τους. Αυτό το ερέθισμα εξάγεται από τη master συσκευή όταν το task της ξεκινά. Για να γίνει αντιληπτό από τις slave συσκευές όμως, είναι απαραίτητο τα tasks τους να έχουν αρχίσει πριν από αυτό της master συσκευής. Αυτό επιτυγχάνεται με τη δομή Flat Sequence που υπάρχει και στα δύο subvis.

Επιπλέον, εξαιτίας της φύσης των καρτών PXI-4461 (στα slots 5 και 9), είναι απαραίτητο να αφαιρεθεί η καθυστέρηση που εισάγεται από το ψηφιακό φίλτρο. Αυτές οι κάρτες χρησιμοποιούν ένα ψηφιακό φίλτρο για να αφαιρέσουν συχνότητες

πάνω από τη συχνότητα Nyquist, το οποίο όμως εισάγει καθυστέρηση επειδή απαιτείται ένας ελάχιστος αριθμός δειγμάτων ώστε να λειτουργήσει αποτελεσματικά. Για να επιτευχθεί αυτό, εκτελούμε μία “dump” ανάγνωση δεδομένων με το που ξεκινήσουν τα tasks των συσκευών, κατά την οποία διαβάζουμε όσα στοιχεία όσα το γινόμενο της καθυστέρησης (*AI.FilterDelay* DAQmx Channel property node) με το ρυθμό δειγματοληψίας του καναλιού (*SampClk.Rate* DAQmx Timing property node). Επίσης πριν ξεκινήσει το task στο slot 9, πρέπει να εκτελεστεί μια “dump” εγγραφή. Αυτό συμβαίνει γιατί εξ’ ορισμού ένα task εξόδου πρέπει να παράγει κάτι. Η λύση είναι η εγγραφή ενός μικρού μηδενικού πίνακα πριν την αρχή του task. Ο ελάχιστος αριθμός σημείων εξαρτάται από το ρυθμό δειγματοληψίας – στην περίπτωσή μας (2kS/s) απαιτούνται 34 δείγματα – αλλά επειδή εξ’ ορισμού η επιλογή Analog Output Regeneration<sup>12</sup> είναι ενεργοποιημένη, ο πίνακας είναι μόνο 2 στοιχείων.

Μέχρι αυτό το σημείο, τα δύο subvis είναι ίδια. Η διαφορά έγκειται στις παραμέτρους του task εξόδου. Στην περίπτωση του dynamic-execution (dynamic-execution Hysteresis τεστ), όπως αναφέρθηκε, η συνθήκη τερματισμού είναι η ολοκλήρωσης της επιθυμητής μεταβολής στο φορτίο. Η έξοδος λοιπόν δεν έχει προκαθορισμένη διάρκεια ή iterations και συνεχίζει να παράγει την επιλεγμένη έξοδο μέχρι το φορτίο να ολοκληρώσει την επιθυμητή μεταβολή. Αυτό διευκολύνει την επιλογή παραμέτρων του task εξόδου, καθώς ένα Continuous task με αρκετά μεγάλο buffer επαρκεί. Το “*initSyncStartDynamic*” subvi απεικονίζεται στο Σχ. 3.35.

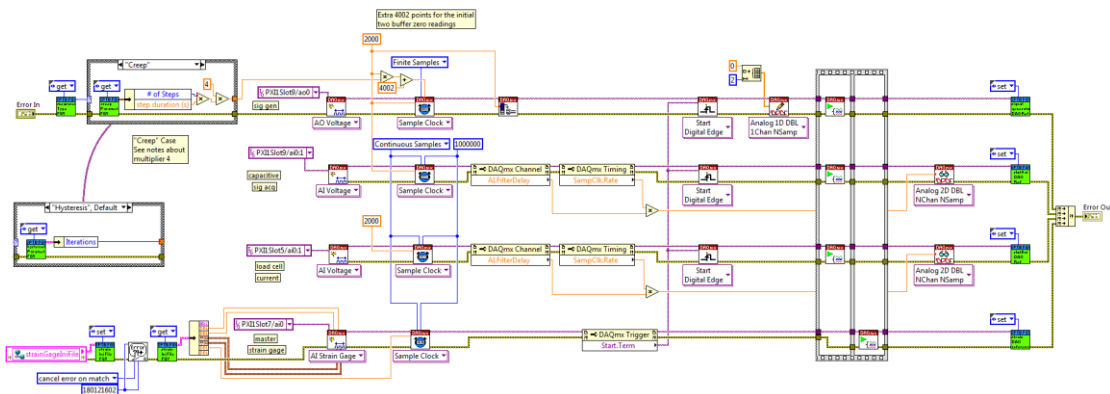


Σχήμα 3. 35: “*initSyncStartDynamic*” subvi

<sup>12</sup> Το regeneration αναφέρεται στην δυνατότητα της συσκευής να 'αναπαράγει' δείγματα που παρήγαγε προηγουμένως. Όταν είναι ενεργοποιημένη, δεδομένα που έχουν γραφεί στον buffer ή τη FIFO επαναχρησιμοποιούνται. Έτσι, μπορεί να επιτευχθεί συνεχής έξοδος χωρίς το LabVIEW να γράφει συνέχεια νέα δεδομένα στον buffer.



Η περίπτωση static-execution (Creep or static-execution Hysteresis τεστ) είναι πιο πολύπλοκη. Η συνθήκη τερματισμού του τεστ σε αυτή την περίπτωση είναι η παραγωγή της επιθυμητής εξόδου. Άρα το task εξόδου πρέπει να οριστεί ως “Finite Samples” και ο αριθμός των στοιχείων που θα παραχθούν πρέπει να καθοριστεί ακριβώς. Ο υπολογισμός των στοιχείων διαφέρει ανάλογα με τον τύπο του τεστ (Creep ή static-execution Hysteresis). Στην περίπτωση του static-execution Hysteresis τεστ, ο αριθμός αυτός είναι το γινόμενο του πεδίου the “Iterations” από τον αντίστοιχο πίνακα ελέγχου (Σχ. 3.18, κάτω-αριστερή γωνία) με το ρυθμό δειγματοληψίας. Αυτή η περίπτωση φαίνεται στο “initSyncStartStatic” subvi (Σχ. 3.36) για την κατάσταση “Hysteresis” του case structure.



Σχήμα 3. 36: “initSyncStartStatic” subvi

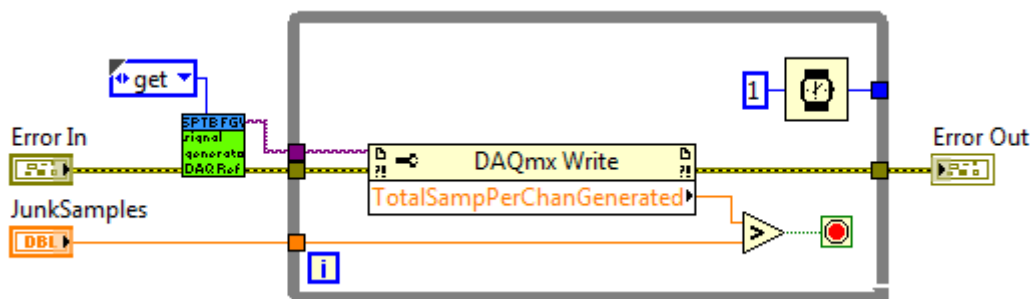
Στη περίπτωση του creep (Σχ. 3.36, κατάσταση “Creep” του case structure), ο αριθμός σημείων που πρέπει να παραχθούν είναι το γινόμενο των πεδίων “# of steps” και “step duration” του αντίστοιχου πίνακα ελέγχου του Creep τεστ που φαίνεται στο Σχ. 3.16. Το αποτέλεσμα πολλαπλασιάζεται επί 4 εξαιτίας της μορφής του creep σήματος διέγερσης (το αποτέλεσμα διπλασιάζεται λόγω της εναλλαγής μεταξύ σταθερής τιμής και μεταβαλλόμενου επιπέδου τάσης και ξαναδιπλασιάζεται λόγω της αύξουσας και φθίνουσας φύσης του σήματος). Το αποτέλεσμα πολλαπλασιάζεται με το ρυθμό δειγματοληψίας και έτσι προκύπτει ο αριθμός σημείων.

Και στις δύο περιπτώσεις, το μέγεθος του buffer εξόδου ορίζεται στην ίδια τιμή με το ρυθμό δειγματοληψίας (2000) και ο αριθμός των σημείων που θα παραχθούν

αυξάνεται κατά ένα μέγεθος λίγο μεγαλύτερο από δύο buffers (4002 σημεία στην περίπτωση μας) για λόγους που θα γίνουν εμφανείς στην επόμενη παράγραφο.

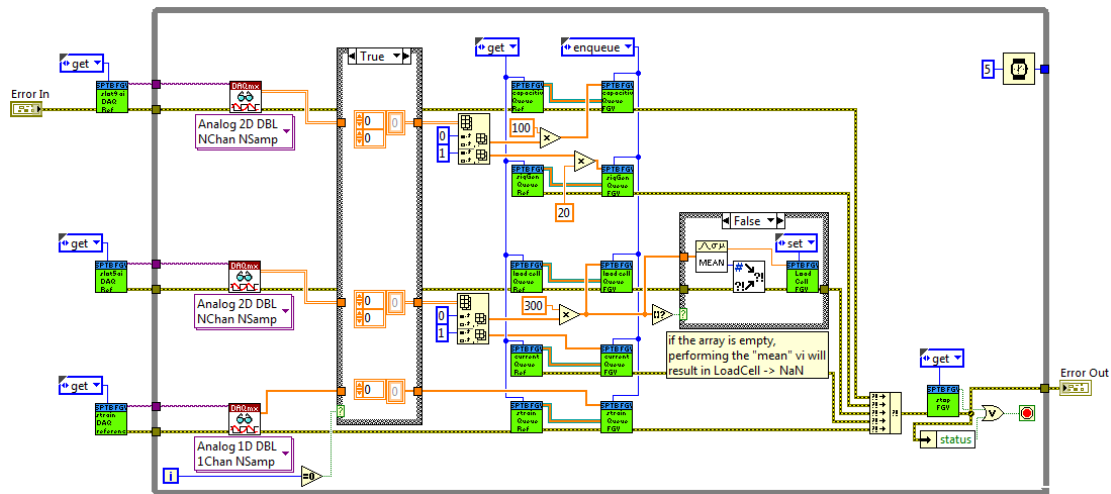
Μια σημαντική ιδιότητα της delta-sigma PXI-4461 κάρτας που χρησιμοποιείται για την έξοδο είναι ότι εξαιτίας του oversampling που συμβαίνει λόγω των delta-sigma converters, η έξοδος δεν μπορεί να μεταβληθεί ακαριαία από την αρχική μηδενική τιμή της όπως ορίστηκε πριν την έναρξη του task. Χρειάζεται δύο sampling περιόδους, που σημαίνει 4000 σημεία, στην περίπτωση Finite Samples (Creep και static-execution Hysteresis τεστ) και μια sampling περίοδο, που σημαίνει 2000 σημεία, στην περίπτωση Continuous samples (dynamic-execution Hysteresis τεστ) ώστε η έξοδος να παράγει τις νέες τιμές. Έτσι, για στατική εκτέλεση, χρειάζεται 2 δευτερόλεπτα από την έναρξη του signal generation μέχρι η έξοδος να παράγει τις τιμές. Μέχρι τότε, η έξοδος παράγει την τιμή 0. Το ίδιο ισχύει και για δυναμική εκτέλεση με τη διαφορά ότι η καθυστέρηση είναι 1 δευτερόλεπτο.

Η συμπεριφορά αυτή δεν αποτελεί πρόβλημα, αλλά καθώς η έξοδος και η ανάγνωση από τους αισθητήρες τρέχουν παράλληλα και ξεκινούν την ίδια στιγμή, τα δεδομένα των πρώτων δευτερολέπτων είναι άχρηστα. Η λύση είναι να μην ξεκινήσει η συλλογή (“SyncedAcquisitionProcess” subvi) και η αποστολή δεδομένων μέσω network stream (“dataSend” subvi, Σχ. 3.26) μέχρι τα δεδομένα που γράφτηκαν στην έξοδο (TotalSampPerChanGenerated DAQmx Write property node) να είναι 4000 για στατική εκτέλεση και 2000 για δυναμική εκτέλεση. Αυτή την εργασία αναλαμβάνει το “removeJunkSamples” subvi που φαίνεται στο Σχ. 3.8. Ο κώδικάς του φαίνεται στο Σχ. 3.37. Η τιμή του ορίσματος “JunkSamples” είναι 4000 για στατική εκτέλεση και 2000 για δυναμική.



Σχήμα 3. 37: “removeJunkSamples” subvi

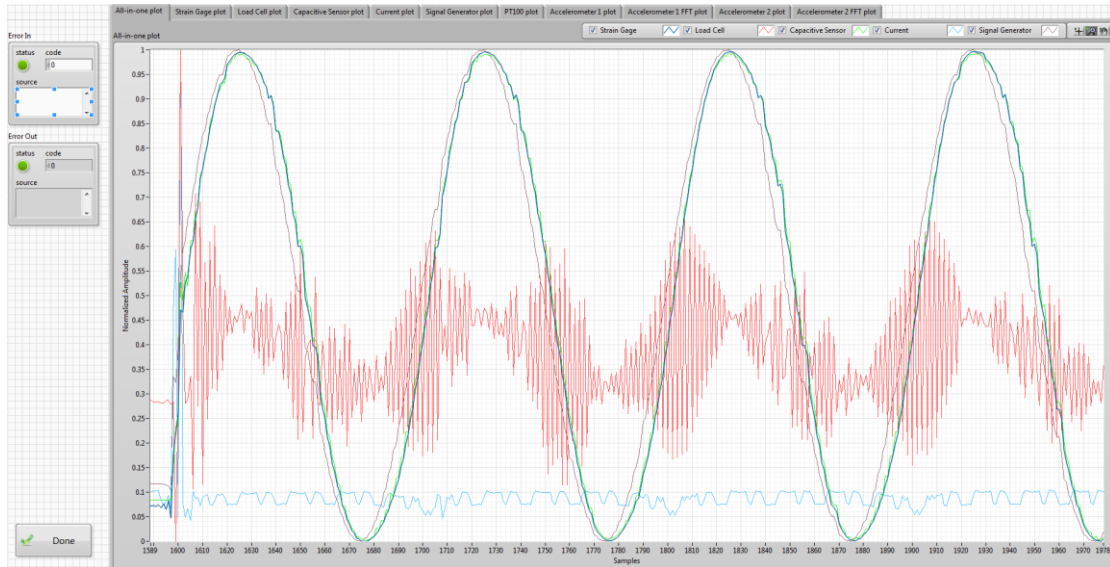
Ακόμα και έτσι, απαιτείται ένα ακόμα βήμα για την ολοκληρωτική απόρριψη των άχρηστων δειγμάτων. Όταν ξεκινάνε τα tasks εισόδου, τα input buffers τους γεμίζουν με δεδομένα. Τα δεδομένα αυτά εξάγονται όποτε διενεργείται μια λειτουργία Read. Έτσι, τα πρώτα “Read” που εκτελούνται για όλες τις συσκευές στο “SyncedAcquisitionProcess” subvi θα έβαζαν στα queues και εν τέλει θα έστελναν άχρηστα δεδομένα. Για να αποφευχθεί αυτό, απορρίπτουμε τα δεδομένα του πρώτου Read και στην θέση τους τοποθετούνται κενοί πίνακες. Αυτή η διεργασία φαίνεται στο Σχ. 3.38. Το επόμενο Read συμβαίνει τόσο γρήγορα που δεν χάνονται πολύτιμα δεδομένα.



Σχήμα 3.38: “SyncedAcquisitionProcess” subvi 1<sup>ο</sup> iteration

### 3.6 Γραφική αναπαράσταση δεδομένων

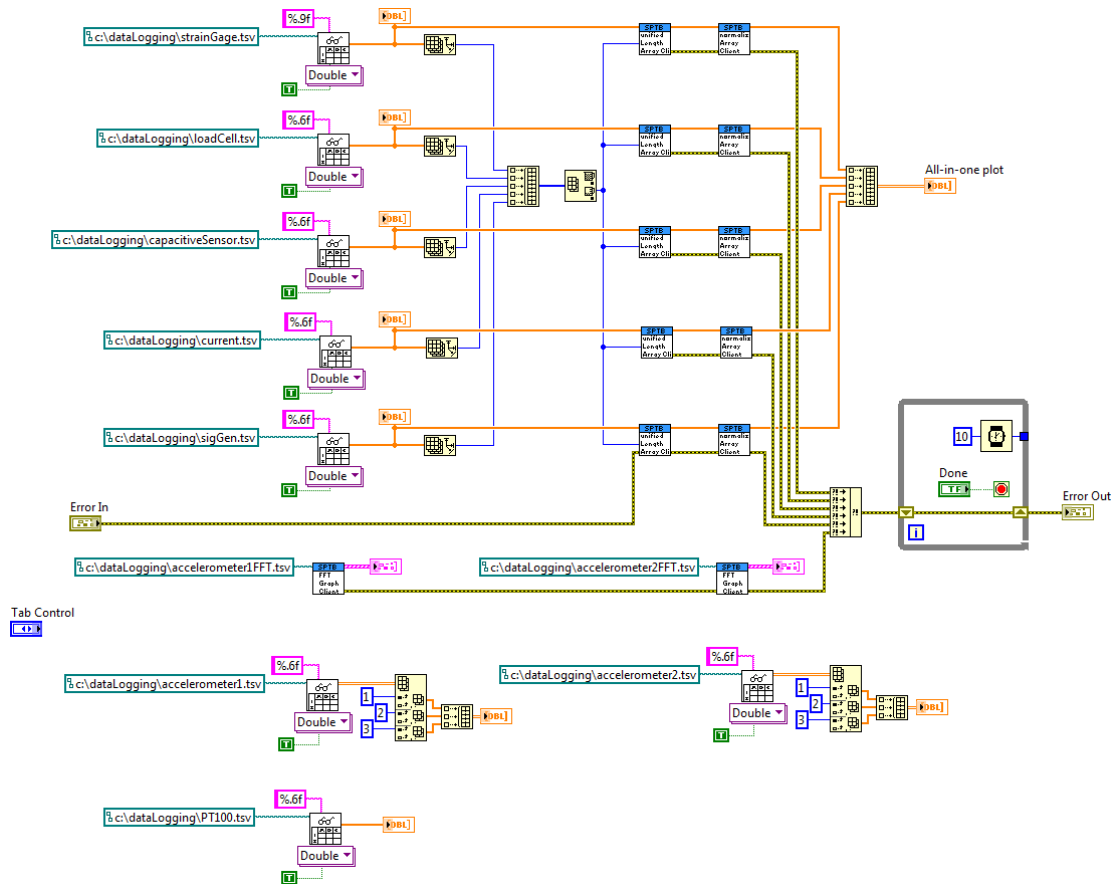
Τα δεδομένα των tsv αρχείων που περιγράφηκαν στο υποκεφάλαιο 3.4 “Διαχείριση Δεδομένων” μπορούν να χρησιμοποιηθούν για την απεικόνιση των δεδομένων. Την εργασία αυτή αναλαμβάνει το “graphCreationClient” στην κατάσταση “Post-processing options και Front Panel” της εφαρμογής client, ο πίνακας ελέγχου του οποίου φαίνεται στο Σχ. 3.39.



Σχήμα 3. 39: “graphCreationClient” subvi

Στο πάνω κομμάτι του πίνακα ελέγχου υπάρχουν tabs για την επιλογή του επιθυμητού διαγράμματος. Οι επιλογές είναι All-in-one (ένα συγκεντρωτικό διάγραμμα που απεικονίζει ταυτόχρονα και σε συγχρονισμό μεταξύ τους τα strain gage, load cell, capacitive sensor, current και signal generator feedback διαγράμματα), strain gage plot, load cell plot, capacitive sensor plot, current plot, signal generator feedback plot, PT100 plot, accelerometer 1 plot, accelerometer 1 FFT plot, accelerometer 2 plot και accelerometer 2 FFT διαγράμματα. Στην κάτω αριστερή γωνία του πίνακα ελέγχου, το κουμπί “Done” τερματίζει το subvi.


Ο κώδικας του subvi φαίνεται στο Σχ. 3.40. Τα διαγράμματα strain gage, load cell, capacitive sensor, current, signal generator feedback και PT100 σχεδιάζονται απευθείας διαβάζοντας τα αντίστοιχα tsv αρχεία. Τα accelerometer 1 και 2 επαφίενται στην ίδια λογική και το τελικό αποτέλεσμα για κάθε επιταχυνσιόμετρο είναι ένα διάγραμμα 3 συνιστωσών – X, Y και Z (που μπορούν να ενεργοποιηθούν και απενεργοποιηθούν για καλύτερη εποπτεία).



Σχήμα 3. 40: Κώδικας “graphCreationClient” subvi

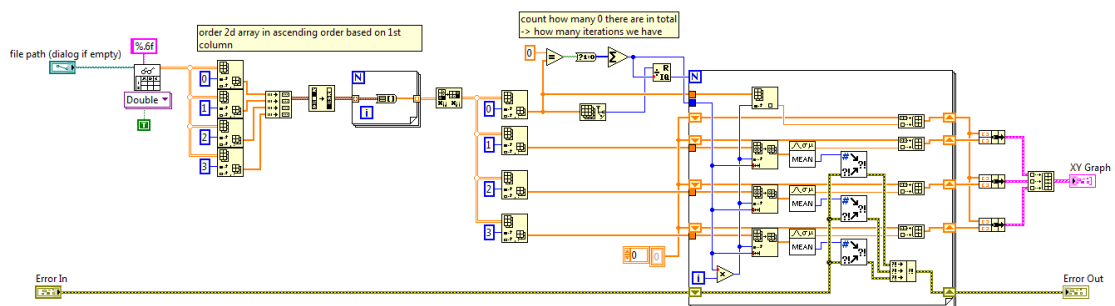
Η σχεδίαση των FFT διαγραμμάτων ήταν πρόκληση καθώς τα δεδομένα έπρεπε να ταξινομηθούν κατ’ αύξουσα συχνότητα. Τα σωσμένα δεδομένα στο FFT tsv αρχείο έχουν τη δομή που κληρονομήθηκε από την “dequeue” κατάσταση – δηλαδή, *1-dimensional array of cluster of 6 waveforms of double* τοποθετημένα το ένα μετά το άλλο σε ένα νέο μονοδιάστατο πίνακα. Πρακτικά η μορφή είναι αυτή της αριστερής πλευράς του Σχ. 3.41. Σε αυτό το σχήμα έχουν απομονωθεί μόνο 4 συχνότητες αλλά η λογική είναι ίδια. Για να αξιοποιηθούν τα δεδομένα, πρέπει να ταξινομηθούν κατ’ αύξουσα συχνότητα (στήλη “F” στο Σχ. 3.41). Το επιθυμητό αποτέλεσμα φαίνεται στη δεξιά πλευρά του σχήματος. Την εργασία αυτή αναλαμβάνει το “FFTGraphClient” subvi (Σχ. 3.42), που ταξινομεί τα δεδομένα κατ’ αύξουσα συχνότητα και υπολογίζει τη μέση τιμή για κάθε συχνότητα του κάθε άξονα (X, Y και Z). Τα αποτελέσματα στη συνέχεια απεικονίζονται στο διάγραμμα FFT το οποίο έχει 3 συνιστώσες – X, Y και Z – που μπορούν να ενεργοποιηθούν και απενεργοποιηθούν για καλύτερη εποπτεία.

f	X	Y	Z
0	0.514107	1.020329	1.135178
9.765625	0.510089	1.010979	1.119166
19.53125	0.498094	0.983271	1.071929
29.29688	0.478305	0.938213	0.995816
0	0.258195	0.49617	0.717426
9.765625	0.259848	0.493711	0.712039
19.53125	0.264513	0.486558	0.695951
29.29688	0.271385	0.47535	0.669416
0	0.301238	0.538313	0.61198
9.765625	0.304762	0.537017	0.611854
19.53125	0.314646	0.533105	0.61114
29.29688	0.329094	0.526503	0.608925



f	X	Y	Z
0	0.514107	1.020329	1.135178
0	0.258195	0.49617	0.717426
0	0.301238	0.538313	0.61198
9.765625	0.510089	1.010979	1.119166
9.765625	0.259848	0.493711	0.712039
9.765625	0.304762	0.537017	0.611854
19.53125	0.498094	0.983271	1.071929
19.53125	0.264513	0.486558	0.695951
19.53125	0.314646	0.533105	0.61114
29.29688	0.478305	0.938213	0.995816
29.29688	0.271385	0.47535	0.669416
29.29688	0.329094	0.526503	0.608925

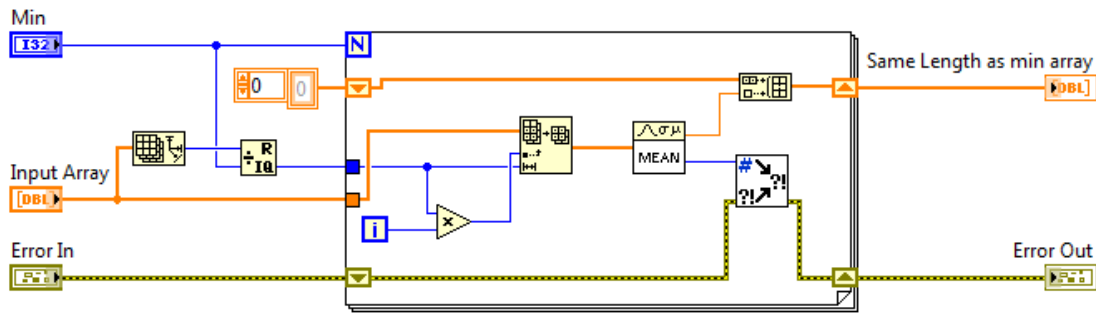
Σχήμα 3. 41: αρχική μορφή δεδομένων FFT (αριστερά) και επιθυμητή (δεξιά)



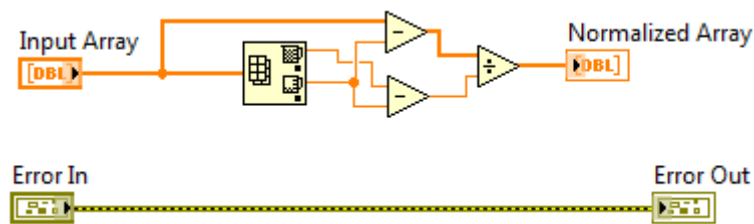
Σχήμα 3. 42: “FFTGraphClient” subvi

Όσον αφορά το All-in-one διάγραμμα, τα subvis “*unifiedLengthArrayClient*” και “*normalizeArrayClient*” διασφαλίζουν την σωστή απεικόνιση.

Το subvi “*unifiedLengthArrayClient*” (Σχ. 3.43) διασφαλίζει ότι όλα τα δεδομένα που εξάγονται από τα διάφορα tsv αρχεία έχουν το ίδιο μήκος (εφαρμόζοντας downsampling υπολογίζοντας τη μέση τιμή των αφαιρούμενων στοιχείων), ενώ το subvi “*normalizeArrayClient*” (Σχ. 3.44) κανονικοποιεί τις τιμές, ώστε το πιθανό εύρος τιμών να είναι μεταξύ 0 και 1 και τα διαγράμματα να συμπίπτουν.



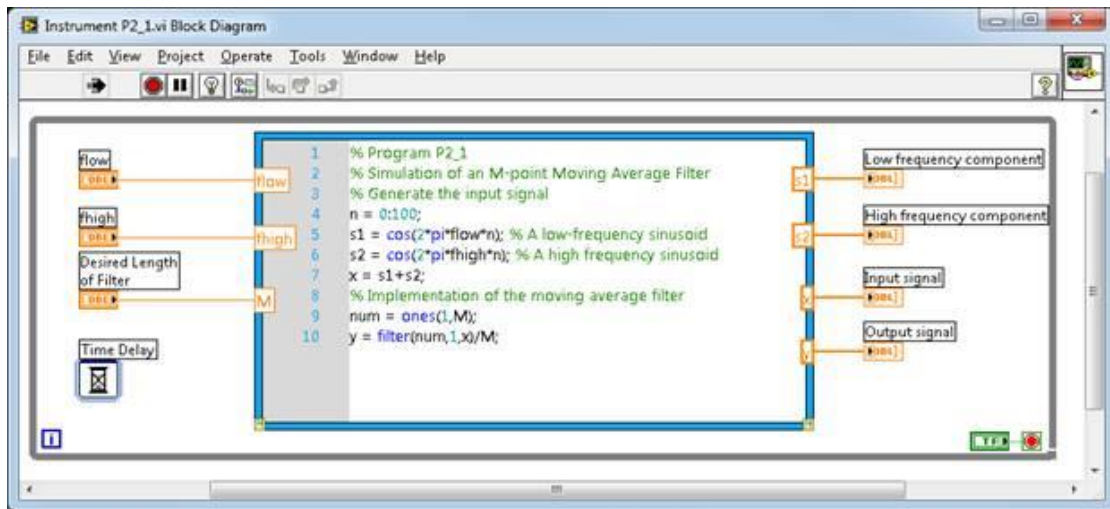
Σχήμα 3. 43: “unifiedLengthArrayClient” subvi



Σχήμα 3. 44: “normalizeArrayClient” subvi

### 3.7 Μετεπεξεργασία Matlab

Η National Instruments δίνει τη δυνατότητα προσθήκης text-based επεξεργασίας και ανάλυσης σημάτων και μαθηματικών υπολογισμών στο γραφικό αναπτυξιακό περιβάλλον του LabVIEW με το επιπρόσθετο module “MathScript RT”. Το MathScript RT μπορεί να τρέχει εντολές που έχουν γραφτεί απευθείας στον κόμβο MathScript ή να τρέξει αρχεία .m. Η ευελιξία που εισάγει είναι μεγάλη καθώς επιτρέπει στους χρήστες να εκτελέσουν σύνθετους μαθηματικούς υπολογισμούς χρησιμοποιώντας υπάρχοντα .m αρχεία και να πραγματοποιήσουν ανάλυση ενώ συλλέγουν δεδομένα. Ένα παράδειγμα του κόμβου MathScript φαίνεται στο Σχ. 3.45, όπου η δήλωση των εισόδων του κόμβου βρίσκεται στην αριστερή πλευρά του πλαισίου, ο κώδικας στη μέση και η δήλωση των εξόδων του κόμβου στην δεξιά πλευρά του πλαισίου. [63]



Σχήμα 3. 45: MathScript κόμβος

Η δυνατότητα να χρησιμοποιηθούν απευθείας Matlab αρχεία είναι ανεκτίμητη στην περίπτωση μας, καθώς επιτρέπει τον αυτόματο υπολογισμό των ιδιοτήτων creep και hysteresis που μας ενδιαφέρουν τρέχοντας τα “creep.m” και “hysteresis.m” Matlab scripts αντίστοιχα (ο κώδικας των οποίων είναι συνημμένος στο Παράρτημα Α).

Συγκεκριμένα, μπορούμε να **σχεδιάσουμε** το **hysteresis loop** και να υπολογίσουμε τα **peak-to-peak displacement**, **hysteresis percentage** και **hysteresis loop area**.

Όσον αφορά το φαινόμενο creep, όπως εξηγήθηκε στο υποκεφάλαιο 1.5.2, η συνάρτηση που το περιγράφει είναι η ακόλουθη:

$$L(t) = L_o \left[ 1 + \gamma \log_{10} \left( \frac{t}{0.1} \right) \right],$$

όπου  $L(t)$  είναι η μετατόπιση του πιεζοηλεκτρικού στοιχείου για σταθερή τάση,  $L_o$  είναι η ονομαστική μετατόπιση 0.1 s μετά την εφαρμογή της τάσης και  $\gamma$  είναι ο παράγοντας ερπυσμού (creep factor) που καθορίζει το ρυθμό του αλγορίθμου.

Η συνάρτηση μπορεί να ξαναγραφτεί ως:

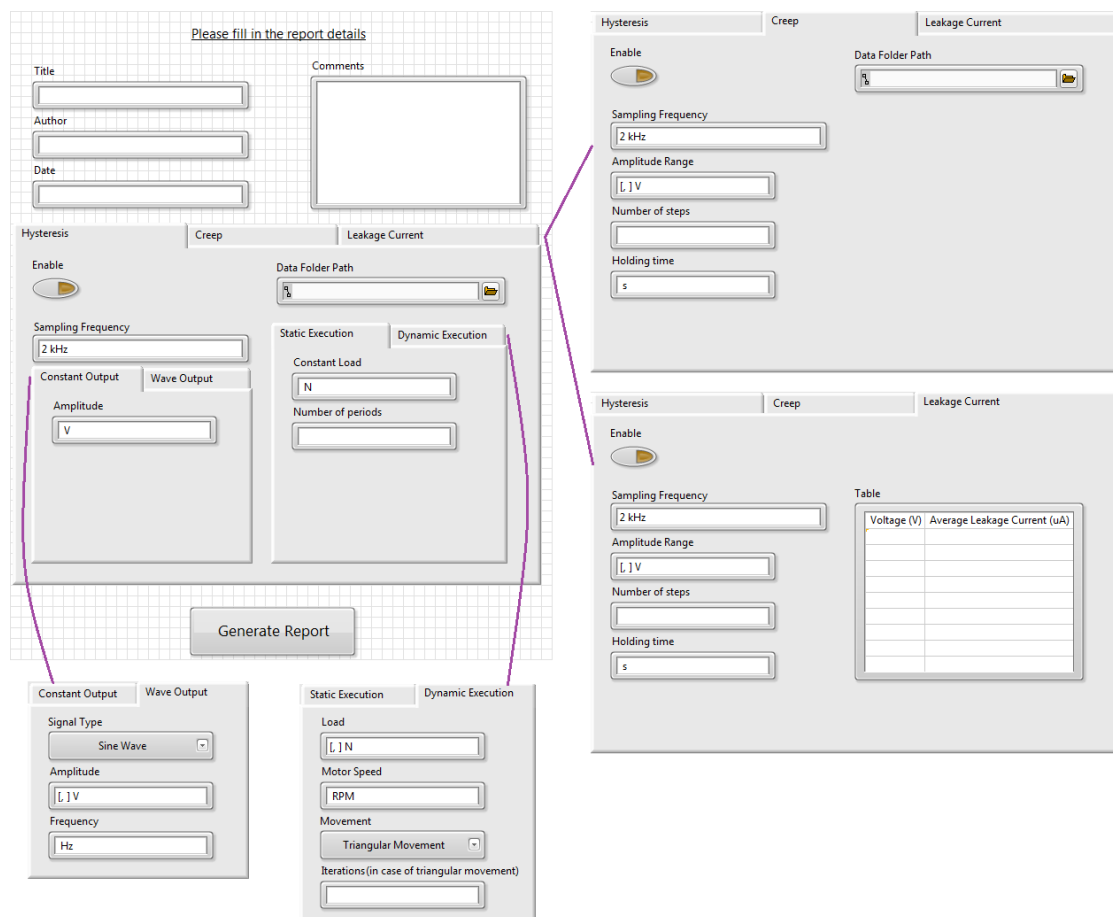
$$L(t) = a + b * \log_{10} \left( \frac{t}{0.1} \right),$$



όπου η σταθερά  $b$  ονομάζεται **σταθερά ερπυσμού (creep coefficient)**. Από το “creep.m” Matlab script, είμαστε σε θέση να την υπολογίσουμε για όλα τα επίπεδα τάσης του creep τεστ τόσο για negative όσο και για positive saturation (αύξοντα και φθίνοντα επίπεδα τάσης αντίστοιχα) και να **σχεδιάσουμε το creep effect** ανά περίπτωση.

### 3.8 Αυτόματη δημιουργία αναφοράς

Όπως αναφέρθηκε, μια επιλογή που προσφέρεται μετά το τέλος του logging των δεδομένων είναι η αυτόματη δημιουργία αναφοράς. Την εργασία αναλαμβάνει το “reportOptionsClient” subνι, το οποίο εμφανίζει στο χρήστη τον πίνακα ελέγχου του Σχ. 3.46. Στο σχήμα φαίνονται όλοι οι υποπίνακες.

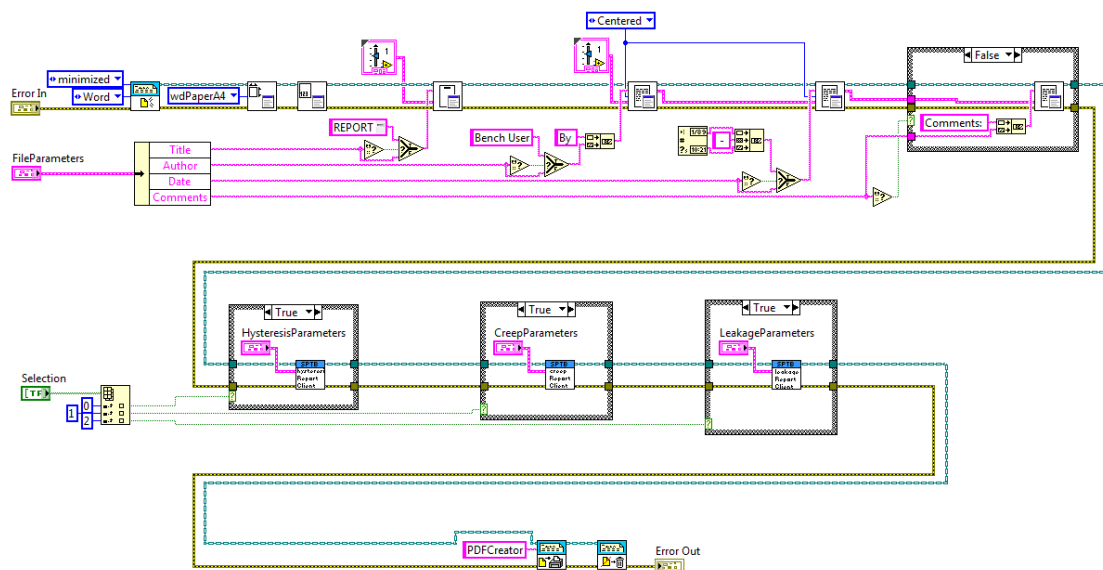


Σχήμα 3. 46: Front Panel του “reportOptionsClient” subνι με όλες τις πιθανές επιλογές

Προς το παρόν, το subνι μπορεί να παράγει Hysteresis, Creep ή Leakage Current αναφορές αλλά η λίστα μπορεί να επεκταθεί στο μέλλον. Ο χρήστης μπορεί να παράγει οποιοδήποτε συνδυασμό των προαναφερθέντων τύπων αναφοράς πιέζοντας το αντίστοιχο “Enable” κουμπί που υπάρχει σε κάθε tab.

Ο χρήστης μπορεί να συμπληρώσει τα πεδία “Title”, “Author” και “Date” της αναφοράς αλλά σε περίπτωση που δεν συμπληρωθούν, οι προεπιλεγμένες τιμές είναι “REPORT on Piezo Characterization”, “Bench User” και η ημερομηνία και ώρα του συστήματος την ώρα δημιουργίας της αναφοράς αντίστοιχα. Το comments box είναι προαιρετικό. Ο χρήστης μπορεί στη συνέχεια να επιλέξει τους επιθυμητούς τύπους αναφοράς και να συμπληρώσει τις παραμέτρους τους.

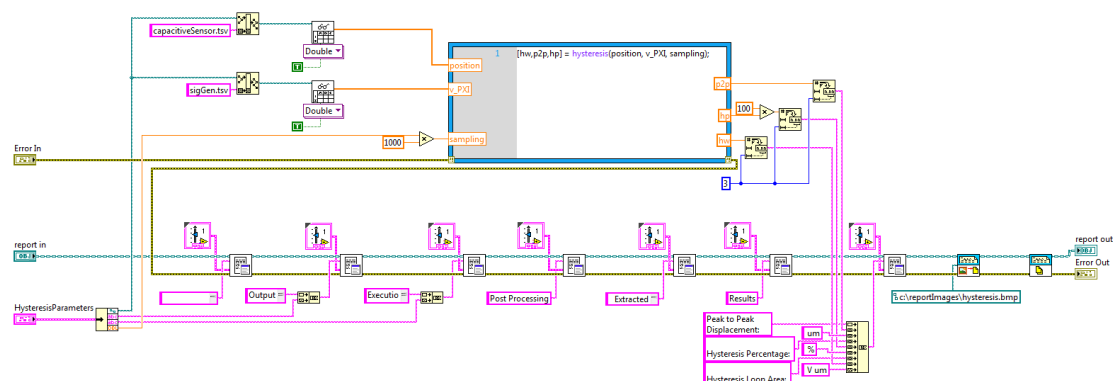
Όταν πατηθεί το κουμπί “Generate Report”, το subνι “*reportGenerationClient*” πυροδοτείται με τις καθορισμένες παραμέτρους. Το subνι φαίνεται στο Σχ. 3.47.



Σχήμα 3. 47: “*reportGenerationClient*” subνι

Το subνι ξεκινά δημιουργώντας μια αναφορά σε Word και τα πεδία Title, Author και Date τοποθετούνται στην κορυφή της. Όπως αναφέρθηκε, υπάρχουν case selectors που φορτώνουν τις προκαθορισμένες τιμές αν τα πεδία δεν συμπληρωθούν στον πίνακα ελέγχου. Ένα άλλο case selector αναγνωρίζει αν το comments box είναι άδειο και σε περίπτωση που δεν είναι, προσθέτει τα σχόλια κάτω από την ημερομηνία. Το ακόλουθο κομμάτι κώδικα αναγνωρίζει ποιοι τύποι αναφοράς ενεργοποιήθηκαν και τρέχει τα αντίστοιχα subνις.

Το subvi που χειρίζεται την Hysteresis αναφορά είναι το “*hysteresisReportClient*” και φαίνεται στο Σχ. 3.48.

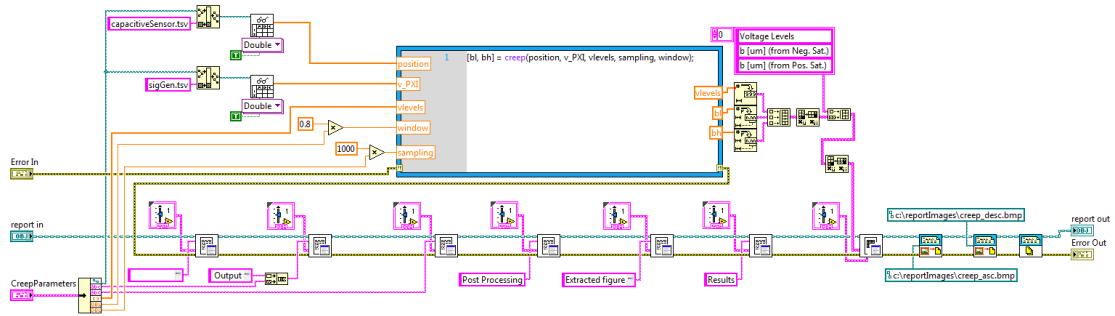


Σχήμα 3. 48: “*hysteresisReportClient*” subvi

Το subvi περιλαμβάνει ένα MathScript κόμβο που τρέχει το Matlab αρχείο “*hysteresis.m*” στα αρχεία δεδομένων που βρίσκονται στον φάκελο που υποδείχθηκε στο πεδίο “Data Folder Path” του πίνακα ελέγχου δημιουργίας αναφοράς (Σχ. 3.46). Τα αποτελέσματα της μετεπεξεργασίας και τα αντίστοιχα διαγράμματα επισυνάπτονται στην αναφορά.

Δύο σημεία άξια αναφοράς είναι ότι και τα δύο αρχεία Matlab, “*hysteresis.m*” και “*creep.m*”, βρίσκονται στο προεπιλεγμένο MathScript directory, το οποίο είναι “*\My Documents\LabVIEW Data*”. Το προεπιλεγμένο MathScript directory μπορεί να αλλάξει από το LabVIEW Tools menu αλλά σε οποιαδήποτε περίπτωση για να είναι οποιοδήποτε Matlab αρχείο προσβάσιμο από το MathScript, πρέπει να είναι τοποθετημένο στο ενδεδειγμένο προκαθορισμένο directory. Το δεύτερο σημείο είναι ότι τα διαγράμματα που δημιουργούνται από το Matlab δεν μπορούν να περαστούν απευθείας στο LabVIEW. Γι’ αυτό το λόγο κατά την εκτέλεση τόσο του “*hysteresis.m*” όσο και του “*creep.m*” τα παραγόμενα διαγράμματα αποθηκεύονται σε μια προκαθορισμένη τοποθεσία του σκληρού δίσκου. Τα διαγράμματα αυτά στη συνέχεια φορτώνονται από την προκαθορισμένη τοποθεσία και επισυνάπτονται στην αναφορά.

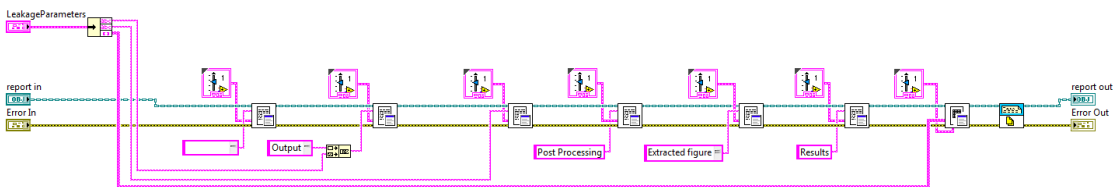
Το ίδιο ισχύει και για την Creep αναφορά, όπου το subvi που χειρίζεται την Creep αναφορά είναι το “*creepReportClient*” και φαίνεται στο Σχ. 3.49.



Σχήμα 3. 49: “*creepReportClient*” subvi

Σε αυτό το subvi, ο MathScript κόμβος τρέχει το Matlab αρχείο “*creep.m*” στα αρχεία δεδομένων που βρίσκονται στον φάκελο που υποδείχθηκε στο πεδίο “Data Folder Path” του πίνακα ελέγχου δημιουργίας αναφοράς που φαίνεται στο Σχ. 3.46. Τα αποτελέσματα της μετεπεξεργασίας και τα αντίστοιχα διαγράμματα επισυνάπτονται στην αναφορά.

Τέλος, την αναφορά Leakage Current χειρίζεται το “*leakageCurrentReportClient*” subvi, που απεικονίζεται στο Σχ. 3.50. Το Leakage Current test είναι ένας τρόπος ελέγχου αν ο πιεζοηλεκτρικός ενεργοποιητής είναι σε καλή λειτουργική κατάσταση ή όχι (ακόμα και αν δεν μπορεί να παρατηρηθεί με γυμνό οφθαλμό). Μετρώντας το ρεύμα διαρροής είμαστε σε θέση να ελέγξουμε χωρίς καταστροφικά αποτελέσματα αν οι ηλεκτρικές ιδιότητες του στοιχείου αντιστοιχούν σε αυτές που ορίζει ο κατασκευαστής. Συνήθως οι κατασκευαστές παρέχουν το κατώφλι ρεύματος διαρροής για διάφορες τιμές τάσης κάτω από το οποίο το στοιχείο θεωρείται ότι λειτουργεί κανονικά.



Σχήμα 3. 50: “*leakageCurrentReportClient*” subvi

Σε αντίθεση με τα Hysteresis και Creep subvis αναφορών, αυτό το subvi δεν τρέχει κάποιο κώδικα Matlab. Το ρεύμα διαρροής προς το παρόν μετριέται με εξωτερικές συσκευές και τα αποτελέσματα του test μπορούν να εισαχθούν στον αντίστοιχο

πίνακα που βρίσκεται στο “Leakage Current” tab του πίνακα ελέγχου δημιουργίας αναφοράς (Σχ. 3.46). Ο πίνακας αυτός στην συνέχεια επισυνάπτεται στην αναφορά.

Τέλος, το τελικό στάδιο του “*reportGenerationClient*” subvi (Σχ. 3.47) είναι η αποθήκευση της αναφοράς στο σκληρό δίσκο. Το LabVIEW δεν προσφέρει κάποιο τρόπο για να σωθεί η αναφορά ως pdf αρχείο. Όπως προαναφέρθηκε, η αναφορά στην περίπτωση μας είναι ένα αρχείο Word και συνεπώς έπρεπε να βρεθεί ένας τρόπος να μετατραπεί αυτόματα σε pdf μέσα από το LabVIEW.

Η λύση που εφαρμόστηκε ήταν η επιστράτευση ενός εικονικού εκτυπωτή (virtual printer). Όπως φαίνεται στον κώδικα του “*reportGenerationClient*” subvi στο Σχ. 3.47, όταν όλα τα επιλεγμένα στοιχεία έχουν επισυναφτεί στην αναφορά, η αναφορά στέλνεται προς εκτύπωση στον εκτυπωτή με το όνομα “PDFCreator”. Όταν η αναφορά τυπωθεί ως pdf αρχείο, απορρίπτεται καθώς δεν συντρέχει λόγος αποθήκευσης του αρχείου Word.

Υπάρχουν πολλοί εικονικοί εκτυπωτές διαθέσιμοι online αλλά σε αυτή την περίπτωση η βασική απαίτηση που καθόρισε την επιλογή ήταν η δυνατότητα να σώζεται η αναφορά ως pdf αυτόματα χωρίς κάποιο pop-up παράθυρο. Οι περισσότεροι εικονικοί εκτυπωτές εμφανίζουν ένα pop-up παράθυρο όπου ο χρήστης καλείται να επιλέξει το directory και να καθορίσει παραμέτρους του εγγράφου. Για να κάνουμε την δημιουργία αναφοράς όσο πιο αυτόματη και αδιάκοπη διαδικασία γίνεται, κρίθηκε απαραίτητος ένας εικονικός εκτυπωτής που θα μπορούσε να σώσει αυτόματα την αναφορά σε μια προκαθορισμένη τοποθεσία.

Ο εικονικός εκτυπωτής που ικανοποιεί αυτές τις απαιτήσεις είναι σε αυτή την περίπτωση ο “PDFCreator”. Πρόκειται για έναν εικονικό εκτυπωτή που είναι διαθέσιμος online δωρεάν και στην περίπτωση μας έχει οριστεί να αποθηκεύει αυτόματα τα παραγόμενα pdf αρχεία σε μια προκαθορισμένη τοποθεσία του σκληρού δίσκου (“*My Documents*”). Ως όνομα του παραγόμενου pdf αρχείου ορίζεται η ημερομηνία και ώρα του συστήματος .



# 4

## *Αποτελέσματα*

Σκοπός του παρόντος κεφαλαίου είναι η παράθεση ενδεικτικών αποτελεσμάτων μιας τυπικής διαδικασίας χαρακτηρισμού πιεζοηλεκτρικού ενεργοποιητή. Τα τεστ που διεξήχθησαν δεν καλύπτουν όλες τις περιπτώσεις ωστόσο επαρκούν για την προβολή της αναφοράς και των αποτελεσμάτων που παρήχθησαν από το λογισμικό.

Στην προκειμένη περίπτωση, πραγματοποιήθηκαν τρία διαφορετικά τεστ στον ίδιο πιεζοηλεκτρικό ενεργοποιητή: ένα τεστ hysteresis, ένα creep και ένα leakage current. Όσον αφορά το πρώτο, πρόκειται για ένα static-execution hysteresis τεστ όπου το φορτίο είναι σταθερό στα 500N και το signal generator παράγει 10 περιόδους ημιτονοειδούς κυματομορφής 1Hz μεταξύ -20V και 150V. Όσον αφορά το creep τεστ, το εύρος τάσης είναι [-20, 150] V και ο αριθμός βημάτων 5 (τα ενδιάμεσα επίπεδα τάσης είναι 14V, 48V, 82V και 116V). Το holding time (διάρκεια βήματος) ορίστηκε στα 5s. Τέλος, το leakage current τεστ πραγματοποιήθηκε σε ένα Keithley 237 High-Voltage Source-Measure Unit. Το ρεύμα διαρροής μετρήθηκε για δύο επίπεδα τάσης - 5V και 150V. Η συχνότητα δειγματοληψίας της συσκευής είναι 1 kHz σύμφωνα με τον κατασκευαστή.

Η αναφορά και τα αποτελέσματα που παρήχθησαν επισυνάπτονται στις επόμενες σελίδες.

# **REPORT on Piezo Characterization**

By Bench User

13/05/2014 - 17:48

Comments: Keithley 237 High-Voltage Source-Measure Unit used for leakage current test.

Current threshold acceptance limit as proposed by the actuator manufacturer: <100uA at 150V, <3.3uA at 5V.

## **Measurement Description (Hysteresis Loop)**

Output Parameters: Sine Wave, Amplitude [-20, 150] V, Frequency 1 Hz

Execution Parameters: Static Execution, Constant Load 500 N, 10 periods, Sampling Frequency 2 kHz

## **Post Processing**

Extracted figures of merit: Peak to Peak Displacement, Hysteresis Percentage, Hysteresis Loop Area.

- o Peak to Peak Displacement: average the cycles and compute the difference between the maximum and minimum displacement values.

- o Hysteresis Percentage: on the averaged loop, compute the maximum difference of the displacement between the ascending and the descending branches for a given input voltage. Normalize it to the peak to peak displacement.

- o Hysteresis Loop Area: on the averaged loop, compute the area within the ascending and descending branches.

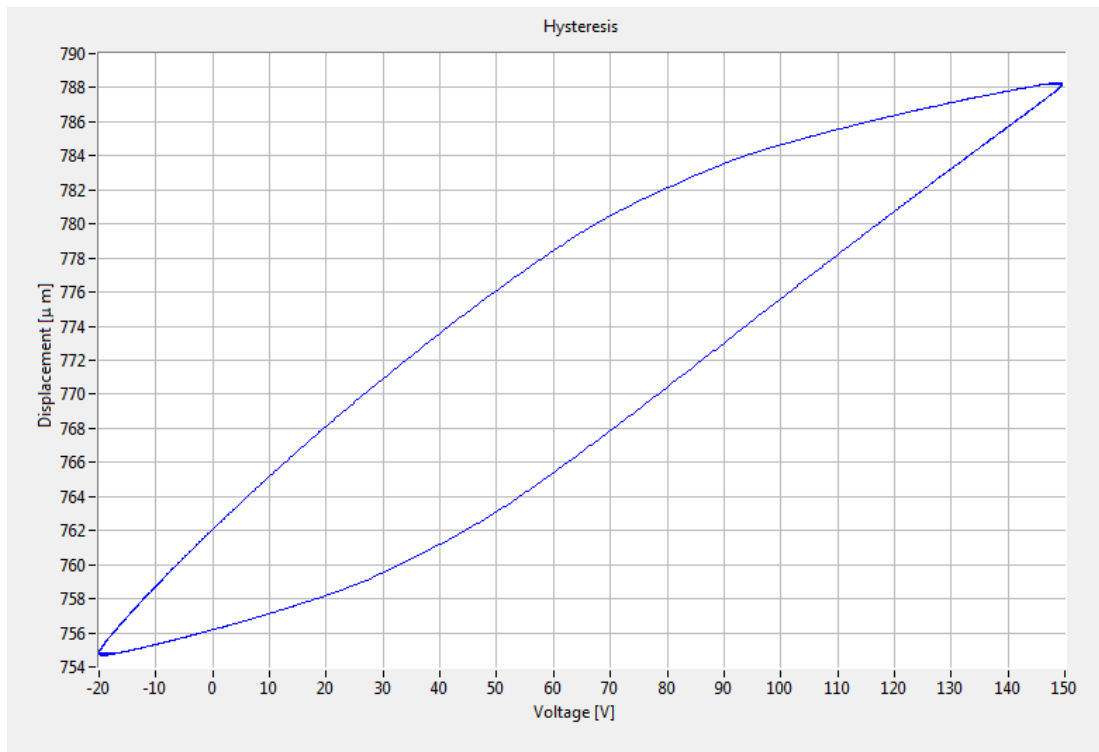
## **Results**

Peak to Peak Displacement: 33.648 um

Hysteresis Percentage: 38.879 %

Hysteresis Loop Area: 13.082 V um





## Measurement Description (Creep)

Output Parameters: Step-like waveform, Two sequences, Amplitude Range [-20, 150] V, 5 steps, Holding time 5 s

Sampling Frequency 2 kHz

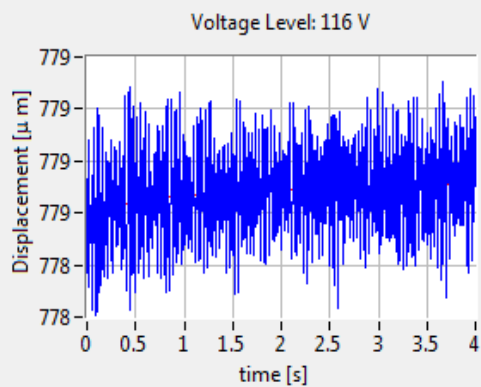
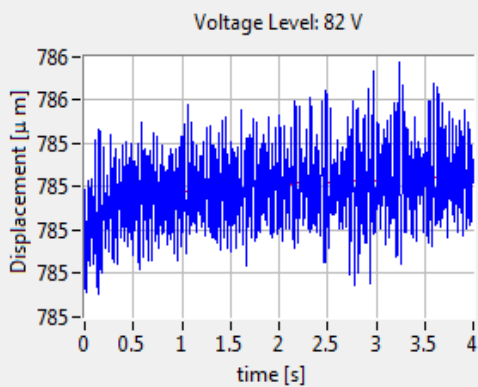
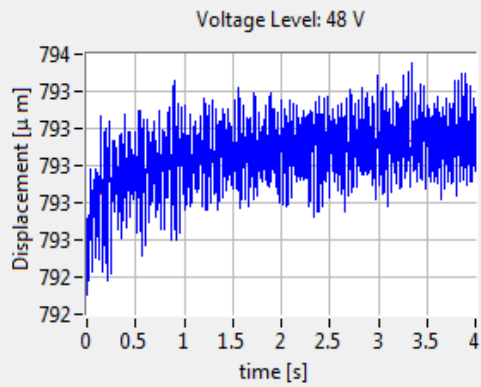
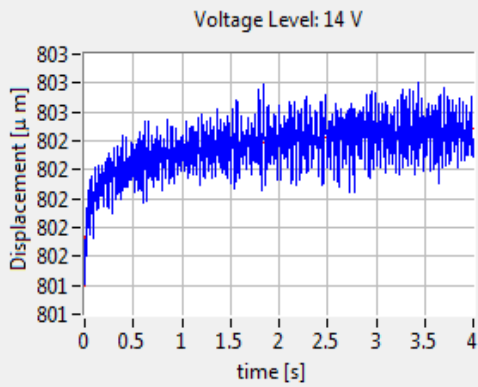
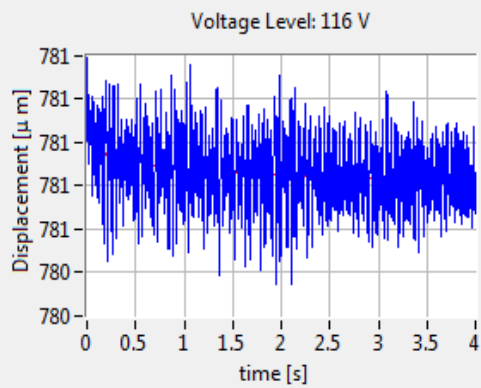
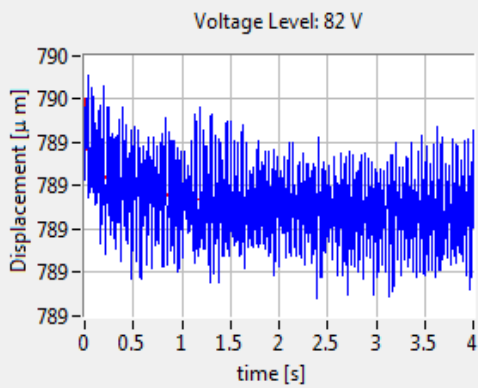
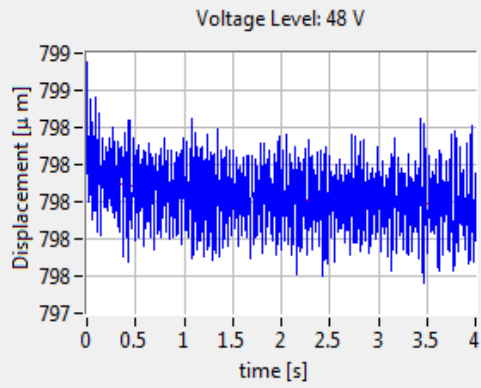
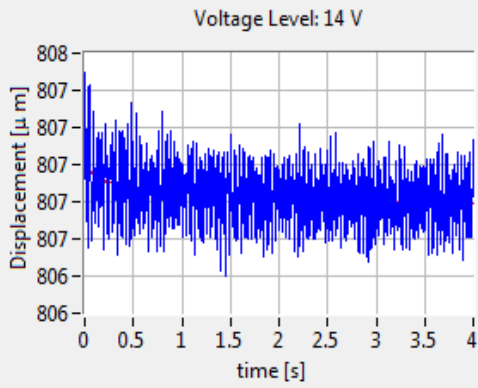
## Post Processing

Extracted figure of merit: Creep Coefficient – b (V).

It is obtained by a linear fit on the displacement acquired waveform over a logarithmic time scale.

## Results

Voltage Levels	14	48	82	116
b [um] (from Neg. Sat.)	-0.039625	-0.048713	-0.059391	-0.041306
b [um] (from Pos. Sat.)	0.151352	0.090981	0.048826	0.033962



## Measurement Description (Leakage Current)

Output Parameters: Amplitude Range 5V, 150V, 2 steps, Holding time 60 s

Sampling Frequency 1 kHz

## Post Processing

Extracted table of merit: Leakage current value.

It is obtained computing the average value of the steady state current during the holding time of the step-like input voltage waveform.

## Results

Voltage	Average Leakage Current
5V	40nA
150V	0.6uA

# 5

## *Επίλογος*

### *5.1 Σύνοψη και συμπεράσματα*

Η πειραματική διάταξη που αναπτύχθηκε και ο κώδικας LabVIEW που τη συνοδεύει αποτελούν ένα πολύ χρήσιμο εργαλείο, που μπορεί να εκτελέσει χαρακτηρισμό πιεζοηλεκτρικών ενεργοποιητών με αξιοπιστία και να μας παρέχει άμεσα αποτελέσματα και αναφορές. Η αξιοπιστία και ακρίβεια του έχουν διαπιστωθεί από τη σύγκριση με καταγεγραμμένα αποτελέσματα διάφορων βιομηχανικών πιεζοηλεκτρικών ενεργοποιητών.

Η πειραματική διάταξη μπορεί να χρησιμοποιηθεί τόσο για να παρέχει αποτελέσματα ανά πάσα στιγμή όσο και για τον επαλήθευση των χαρακτηριστικών που παρέχονται από τους κατασκευαστές πιεζοηλεκτρικών ενεργοποιητών καθώς και για τον επανέλεγχο ενεργοποιητών που έχουν μετρηθεί στο παρελθόν αλλά κάποια παράμετρος έχει μεταβληθεί – για παράδειγμα αφότου έχουν ακτινοβοληθεί, μπορούμε να ελέγξουμε αν τα χαρακτηριστικά τους μεταβάλλονται με αυξανόμενα επίπεδα ακτινοβολήσης.

## **5.2 Μελλοντικές επεκτάσεις**

Το γεγονός ότι το λογισμικό είναι γραμμένο σε LabVIEW σημαίνει ότι μπορεί να προσαρμοστεί εύκολα και γρήγορα σε οποιαδήποτε μελλοντική ανάγκη.

Δύο πιθανές βελτιώσεις που θα επέκτειναν τη λειτουργικότητα της διάταξης είναι οι ακόλουθες: Αρχικά, η διάταξη θα μπορούσε να τροποποιηθεί ώστε να μπορεί να διενεργήσει ένα πλήρες τεστ ρεύματος διαρροής. Αντί να μετράμε το ρεύμα διαρροής με εξωτερική συσκευή και μετά να περνάμε τα αποτελέσματα στο Leakage Current Report Generation tab, όλο το τεστ θα μπορούσε να εκτελείται στη διάταξη. Αν αυτό αποδειχθεί πολύ πολύπλοκο, μια άλλη πιθανότητα θα ήταν να εδραιώσουμε μια σύνδεση με την εξωτερική συσκευή μέτρησης, ώστε τα αποτελέσματα να περνιούνται αυτόματα.

Το ίδιο ισχύει και για το τεστ Εμπέδησης (Impedance test). Η διενέργεια του τεστ θα μπορούσε να προστεθεί ως λειτουργία στη διάταξη ή αν αυτό είναι πολύ πολύπλοκο, η εδραίωση σύνδεσης με τη συσκευή μέτρησης ώστε τα αποτελέσματα να μεταφέρονται αυτόματα θα επαρκούσε.

# 6

## Βιβλιογραφία

- [1] “About CERN.” [Online]. Available: <http://home.web.cern.ch/about>. [Accessed: 04-Apr-2014].
- [2] “CERN,” *Wikipedia, the free encyclopedia*. 03-Apr-2014.
- [3] “Introduction of CERN, LHC and the ATLAS Experiment.” [Online]. Available: <http://zimmer.csufresno.edu/~yogao/ATLAS/CERN-LHC-ATLAS.html>. [Accessed: 04-Apr-2014].
- [4] “Large Electron–Positron Collider,” *Wikipedia, the free encyclopedia*. 02-Apr-2014.
- [5] “CERN LHC guide.” .
- [6] “The Large Hadron Collider.” [Online]. Available: <http://home.web.cern.ch/topics/large-hadron-collider>. [Accessed: 04-Apr-2014].
- [7] E. Koukovini-Platia, “Study of the  $\chi_{b0}$  meson via the decay  $\chi_{b0} \rightarrow J/\psi J/\psi \rightarrow \mu^+\mu^-\mu^+\mu^-$  in the ATLAS experiment,” National Technical University of Athens, School of Applied Mathematical and Physical Sciences, 2010.
- [8] “Large Hadron Collider,” *Wikipedia, the free encyclopedia*. 05-Apr-2014.
- [9] “Higgs boson,” *Wikipedia, the free encyclopedia*. 08-Apr-2014.
- [10] “The LHC: a look inside | Science in School.” [Online]. Available: <http://www.scienceinschool.org/2008/issue10/lhchow>. [Accessed: 08-Apr-2014].
- [11] “Pulling together: Superconducting electromagnets.” [Online]. Available: <http://home.web.cern.ch/about/engineering/pulling-together-superconducting-electromagnets>. [Accessed: 08-Apr-2014].
- [12] “Magnets.” [Online]. Available: <http://lhc-machine-outreach.web.cern.ch/lhc-machine-outreach/components/magnets.htm>. [Accessed: 08-Apr-2014].
- [13] “Radiofrequency cavities.” [Online]. Available: <http://home.web.cern.ch/about/engineering/radiofrequency-cavities>. [Accessed: 08-Apr-2014].
- [14] “LHC beam dumps.” [Online]. Available: <http://lhc-machine-outreach.web.cern.ch/lhc-machine-outreach/components/beam-dump.htm>. [Accessed: 08-Apr-2014].
- [15] “New beam dump for a veteran accelerator.” [Online]. Available: <http://home.web.cern.ch/about/updates/2013/10/new-beam-dump-veteran-accelerator>. [Accessed: 08-Apr-2014].

- [16]“LHC collimation.” [Online]. Available: [http://lhc-collimation-project.web.cern.ch/lhc%2Dcollimation%2Dproject/lhc\\_collimation\\_\\_introduction.htm](http://lhc-collimation-project.web.cern.ch/lhc%2Dcollimation%2Dproject/lhc_collimation__introduction.htm). [Accessed: 08-Apr-2014].
- [17]“Taking a closer look at LHC - LHC.” [Online]. Available: <http://www.lhc-closer.es/1/3/4/0>. [Accessed: 08-Apr-2014].
- [18]“The collimation system: defence against beam loss - CERN Courier.” [Online]. Available: <http://cerncourier.com/cws/article/cern/54384>. [Accessed: 08-Apr-2014].
- [19]T. Weiler, O. Aberle, R. Assmann, R. Chamizo, Y. Kadi, J. Lettry, R. Losito, and S. Redaelli, “LHC collimation system hardware commissioning,” in *IEEE Particle Accelerator Conference, 2007. PAC, 2007*, pp. 1625–1627.
- [20]“UA9.” [Online]. Available: <http://home.web.cern.ch/about/experiments/ua9>. [Accessed: 09-Apr-2014].
- [21]W. Scandale, “UA9 Report for 2013,” Oct. 2013.
- [22]S. Montesano, “Status of crystals for LHC,” presented at the LHC Collimation Working Group, CERN, 17-Feb-2014.
- [23]S. Montesano, “UA9 - Crystal collimation at CERN-SPS,” presented at the ICT, Trento, Italy, 11-Feb-2013.
- [24]“Piezoelectricity,” *Wikipedia, the free encyclopedia*. 07-Apr-2014.
- [25]N. Richardt, “An Introduction to the Piezo World,” Noliac, Denmark, Jun. 2007.
- [26]“Material characteristics | Hysteresis | Creep | Linearity | Thermal properties | Power dissipation.” [Online]. Available: [http://www.noliac.com/Material\\_characteristics\\_-143.aspx](http://www.noliac.com/Material_characteristics_-143.aspx). [Accessed: 10-Apr-2014].
- [27]A. Ledoux, “Theory of Piezoelectric Materials and Their Application in Civil Engineering,” MIT, Massachusetts Institute of Technology, Massachusetts, 2011.
- [28]M. S. Vijaya, *Piezoelectric materials and devices applications in engineering and medical sciences*. Boca Raton, FL: CRC Press, 2013.
- [29]H. Jung and D.-G. Gweon, “Creep characteristics of piezoelectric actuators,” *Rev. Sci. Instrum.*, vol. 71, no. 4, pp. 1896–1900, Apr. 2000.
- [30]“Fundamentals of Piezotechnology.” [Online]. Available: [http://www.piezo.ws/piezoelectric\\_actuator\\_tutorial/Piezo\\_Design\\_part2.php](http://www.piezo.ws/piezoelectric_actuator_tutorial/Piezo_Design_part2.php). [Accessed: 04-Apr-2014].
- [31]L. S. Zhang, Y. B. Liu, C. L. Pan, and Z. H. Feng, “Leakage current characterization and compensation for piezoelectric actuator with charge drive,” *Sens. Actuators Phys.*, vol. 199, pp. 116–122, Sep. 2013.
- [32]“Tutorial: Piezo Actuator Electrical Fundamentals, High Precision Piezo Motion, Nano Positioning Mechanics:” [Online]. Available: <http://www.physikinstrumente.com/en/products/prdetail.php?sortnr=400600.75>. [Accessed: 10-Apr-2014].
- [33]“Electrical impedance,” *Wikipedia, the free encyclopedia*. 06-Apr-2014.
- [34]“LabVIEW,” *Wikipedia, the free encyclopedia*. 05-Apr-2014.
- [35]“Visual programming language,” *Wikipedia, the free encyclopedia*. 10-Apr-2014.
- [36]“What Is LabVIEW? - National Instruments.” [Online]. Available: <http://www.ni.com/newsletter/51141/en/>. [Accessed: 11-Apr-2014].
- [37]“Data flow - LabVIEW Wiki.” [Online]. Available: [http://labviewwiki.org/Data\\_flow](http://labviewwiki.org/Data_flow). [Accessed: 11-Apr-2014].
- [38]“VI - LabVIEW Wiki.” [Online]. Available: <http://labviewwiki.org/VI>. [Accessed: 11-Apr-2014].
- [39]D. Audigier, C. Richard, C. Descamps, M. Troccaz, and L. Eyraud, “PZT uniaxial stress dependence: experimental results,” *Ferroelectrics*, vol. 154, no. 1, pp. 219–224, 1994.
- [40]M. W. Hooker, L. R. Center, and S. A. Wise, *Characterization of multilayer piezoelectric actuators for use in active isolation mounts*. Hampton, Va: National Aeronautics and Space Administration, Langley Research Center, 1997.
- [41]S. Mekid and B. Lim, “Characteristics comparison of piezoelectric actuators at low electric field: analysis of strain and blocking force,” *Smart Mater. Struct.*, vol. 13, no. 5, p. N93, Oct. 2004.



- [42] M. Badescu, S. Sherrit, X. Bao, J. Aldrich, Y. Bar-Cohen, and C. Jones, "Extended life PZT stack test fixture," 2008, pp. 692903–692903–7.
- [43] G. Yang, H. Gou, Y. Ji, Z. Yue, and L. Li, "Influence of uniaxial compressive stress on the ferroelectric behaviors of piezoelectric stack actuators," Mar. 2008.
- [44] M. Mitrovic, G. P. Carman, and F. K. Straub, "Response of piezoelectric stack actuators under combined electro-mechanical loading," *Int. J. Solids Struct.*, vol. 38, no. 24–25, pp. 4357–4374, Jun. 2001.
- [45] R. Yimnirun, S. Ananta, A. Ngamjarujana, and S. Wongsanmai, "Uniaxial stress dependence of ferroelectric properties of xPMN-(1-x)PZT ceramic systems," *Appl. Phys. A*, vol. 81, no. 6, pp. 1227–1231, Nov. 2005.
- [46] D. Zhou, M. Kamlah, and D. Munz, "Effects of uniaxial prestress on the ferroelectric hysteretic response of soft PZT," *J. Eur. Ceram. Soc.*, vol. 25, no. 4, pp. 425–432, Apr. 2005.
- [47] C. S. Lynch, "The effect of uniaxial stress on the electro-mechanical response of 8/65/35 PLZT," *Acta Mater.*, vol. 44, no. 10, pp. 4137–4148, Oct. 1996.
- [48] "Stainless steel," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [49] "Anodizing," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [50] "Series and parallel springs," *Wikipedia, the free encyclopedia*. 23-Mar-2014.
- [51] "Backlash (engineering)," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [52] "Torque limiter," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [53] "Resistance thermometer," *Wikipedia, the free encyclopedia*. 17-Apr-2014.
- [54] "PCI eXtensions for Instrumentation," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [55] "What Is PXI? - National Instruments." [Online]. Available: <http://www.ni.com/pxi/whatis/>. [Accessed: 17-Apr-2014].
- [56] "What is a Real-Time Operating System (RTOS)? - National Instruments." [Online]. Available: <http://www.ni.com/white-paper/3938/en/>. [Accessed: 17-Apr-2014].
- [57] "NI Real-Time Hypervisor for Windows - National Instruments." [Online]. Available: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/207302>. [Accessed: 17-Apr-2014].
- [58] *LabView Basics I Introduction Course Manual Course Software Version 8.0*. National Instruments, 2006.
- [59] *LabView Basics II Development Course Manual Course Software Version 8.0*. National Instruments, 2006.
- [60] "INI file," *Wikipedia, the free encyclopedia*. 03-May-2014.
- [61] "Tab-separated values," *Wikipedia, the free encyclopedia*. 03-May-2014.
- [62] "Fast Fourier transform," *Wikipedia, the free encyclopedia*. 05-May-2014.
- [63] "What is the NI LabVIEW MathScript RT Module? - National Instruments." [Online]. Available: <http://www.ni.com/white-paper/6206/en/>. [Accessed: 08-May-2014].



# Παράρτημα Α

## *Κώδικας Matlab*

### *A1. “Creep.m”*

```
function [bl,bh] = creep(position,v_PXI,vlevels,sampling>window)

v = v_PXI;
x = position;
Voltage_levels = vlevels;
t = 1:max(length(x),length(v));
t = t/sampling;
tolerance = 2;
dt = 0.1;

v_low_sat = v(1:floor(numel(v)/2));
x_low_sat = x(1:floor(numel(v)/2));
time_low_sat = t(1:floor(numel(v)/2));
v_high_sat = v(floor(numel(v)/2)+1:end);
x_high_sat = x(floor(numel(v)/2)+1:end);
time_high_sat = t(floor(numel(v)/2)+1:end);

figure('Position', [0 0 800 800]);
hold on;

for i = 1:numel(Voltage_levels)
    ndx = find(abs(v_low_sat-Voltage_levels(i)) < tolerance);
    ndx = ndx(1);
    onset_time(i) = time_low_sat(ndx);
    ndx = find(time_low_sat > onset_time(i) & time_low_sat <
onset_time(i) + window);

    t_red = time_low_sat(ndx)'-onset_time(i);
    x_red = x_low_sat(ndx);
    p_low_sat(i,:) = polyfit(log(t_red/dt),x_red,1);
```

```

    subplot(ceil(numel(Voltage_levels)/2),2,i);
    hold on;
    plot(t_red,x_red);
    plot(t_red,p_low_sat(i,2)+p_low_sat(i,1)*log(t_red/dt),'r--');
    ylabel('Displacement [\mu m]');
    xlabel('time [s]');
    grid on;
    title(strcat('Voltage Level: ',num2str(Voltage_levels(i)), ' V'));
end

print('c:\reportImages\creep_asc.bmp');
close;

figure('Position', [0 0 800 800]);
hold on;

for i = 1:numel(Voltage_levels)
    ndx = find(abs(v_high_sat-Voltage_levels(i)) < tolerance);
    ndx = ndx(1);
    onset_time(i) = time_high_sat(ndx);
    ndx = find(time_high_sat > onset_time(i) & time_high_sat <
onset_time(i) + window);

    t_red = time_high_sat(ndx)-onset_time(i);
    x_red = x_high_sat(ndx);
    p_high_sat(i,:) = polyfit(log(t_red/dt),x_red,1);

    subplot(ceil(numel(Voltage_levels)/2),2,i);
    hold on;
    plot(t_red,x_red);
    plot(t_red,p_high_sat(i,2)+p_high_sat(i,1)*log(t_red/dt),'r--');
    ylabel('Displacement [\mu m]');
    xlabel('time [s]');
    grid on;
    title(strcat('Voltage Level: ',num2str(Voltage_levels(i)), ' V'));
end

print('c:\reportImages\creep_desc.bmp');
close;

bl = p_low_sat(:,1);
bh = p_high_sat(:,1);

```

## A2. “Hysteresis.m”

```
function [hw, p2p, hp] = hysteresis(position,v_PXI,sampling)

num_samples = min(length(position),length(v_PXI));
N= floor(num_samples/sampling);
ndx=1:1*sampling;
pos_avg = zeros(numel(ndx),1)';
v_PXI_avg = zeros(numel(ndx),1)';

for i = 2:(N)
    ndx = ((i-1)*sampling)+1:((i)*sampling);
    pos_avg = pos_avg + position(ndx);
    v_PXI_avg = v_PXI_avg + v_PXI(ndx);
end

pos_avg = pos_avg/(N-1);
v_PXI_avg = v_PXI_avg/(N-1);
[vmin,ndx] = min(v_PXI_avg);

v_PXI_avg = [v_PXI_avg(ndx+1:end) v_PXI_avg(1:ndx)];
pos_avg = [pos_avg(ndx+1:end) pos_avg(1:ndx)];

[vmin,ndx] = max(v_PXI_avg);
v_PXI_avg_ascending = v_PXI_avg(1:ndx-1);
v_PXI_avg_descending = v_PXI_avg(ndx:end);
pos_avg_ascending = pos_avg(1:ndx-1);
pos_avg_descending = pos_avg(ndx:end);

minv = min(v_PXI_avg_ascending);
maxv = max(v_PXI_avg_descending);
v_resampled = linspace(minv,maxv,1000);

[v_PXI_avg_ascending,I,J] = unique(v_PXI_avg_ascending);
pos_avg_ascending = pos_avg_ascending(I);
[pos_avg_ascending,I,J] = unique(pos_avg_ascending);
v_PXI_avg_ascending = v_PXI_avg_ascending(I);
[v_PXI_avg_descending,I,J] = unique(v_PXI_avg_descending);
pos_avg_descending = pos_avg_descending(I);
[pos_avg_descending,I,J] = unique(pos_avg_descending);
v_PXI_avg_descending = v_PXI_avg_descending(I);

pos_asc_res = interpolateld(v_PXI_avg_ascending, pos_avg_ascending,
v_resampled, 'linear');
pos_desc_res =
interpolateld(v_PXI_avg_descending,pos_avg_descending,v_resampled,
'linear');

[value,ndx]=max(pos_desc_res-pos_asc_res);

hysteresis_width = abs(pos_desc_res(ndx)-pos_asc_res(ndx));
peak2peak = max(pos_avg)-min(pos_avg);
hysteresis_percentage_PXI = hysteresis_width/peak2peak;

disp('Hysteresis Percentage, PXI voltage:');
disp(hysteresis_percentage_PXI);

disp('Hysteresis peak2peak:');
disp(peak2peak);
```

```

areatot=polyarea([v_resampled rot90(v_resampled,2)]*20,-[pos_asc_res
rot90(pos_desc_res,2)]); %V*mu_m
disp('Hysteresis Area:');
disp(areatot);

figure('Position', [0 0 800 600]);
plot([v_PXI_avg_ascending v_PXI_avg_descending(end:-
1:1)], [pos_avg_ascending pos_avg_descending(end:-1:1)]);
xlabel('Voltage [V]');
ylabel('Displacement [\mu m]');
grid on;
title('Hysteresis');
print('c:\reportImages\hysteresis.bmp');
close

hw = hysteresis_width;
p2p = peak2peak;
hp = hysteresis_percentage_PXI;

```

# Παράρτημα Β

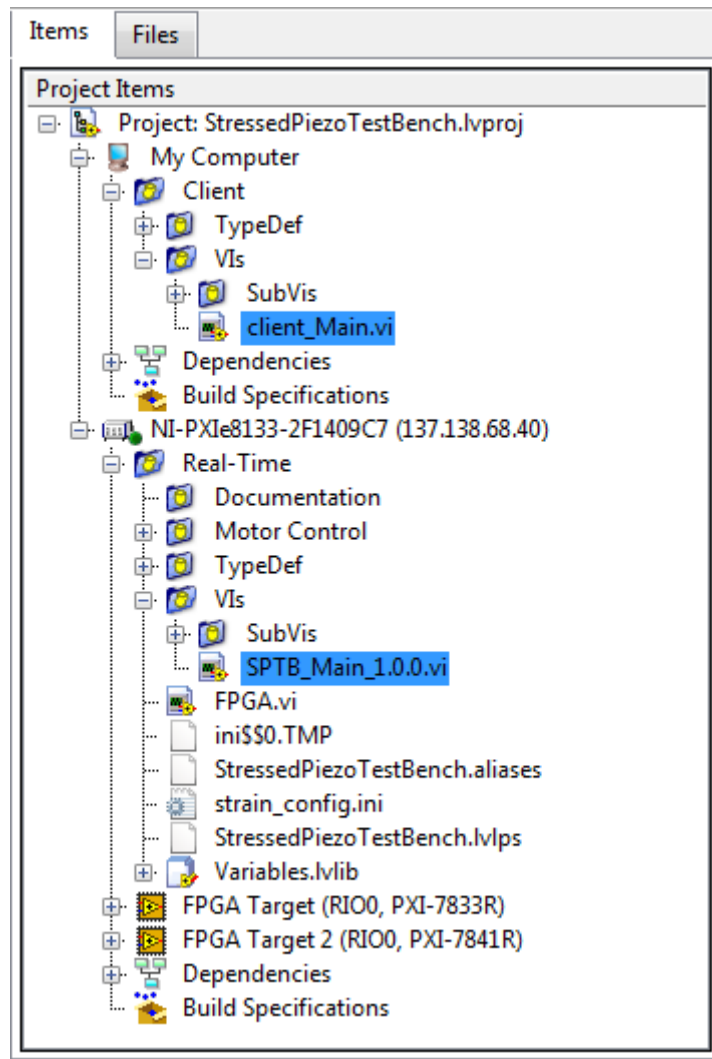
## *Οδηγίες για το LabVIEW Project*

Το παρόν έργο εκπονήθηκε με το λογισμικό National Instrument's LabVIEW έκδοση 2011 Service Pack 1.

Το αρχείο του LabVIEW Project, το οποίο ανοίγει όλο το project, βρίσκεται στο φάκελο “Real-Time” και ονομάζεται “StressedPiezoTestbench.lvproj”.

Μέσα στο project, η server εφαρμογή, που ονομάζεται “SPTB\_Main\_1.0.0.vi”, βρίσκεται στο “NI-PXIE8133-F1409C7\Real-Time\VIs\”. Η client εφαρμογή, που ονομάζεται “client\_Main.vi”, βρίσκεται στο “My Computer\Client\VIs\”. Και τα δύο vis είναι υπογραμμισμένα στο σχήμα που ακολουθεί.

Τα αρχεία Matlab (“creep.m” and “hysteresis.m”), όπως αναφέρθηκε, πρέπει να είναι τοποθετημένα στο προεπιλεγμένο MathScript directory ώστε να λειτουργούν σωστά. Βρίσκονται στο “Support Files\Matlab scripts”.



Τέλος, έχουν υπάρξει αναφορές ασυμβατότητας του National Instruments ‘Report Generation Toolkit For Microsoft Office’ version 11.0.0 με εκδόσεις των Microsoft Office νεότερες των Office 2003. Συγκεκριμένα, το “Append Table to Report.vi” δεν λειτουργεί όπως θα έπρεπε και το πρόβλημα είναι ότι οι τίτλοι των στηλών (column labels) τυπώνονται στη θέση των δεδομένων στον πίνακα. Αυτό συμβαίνει γιατί οι σύγχρονες εκδόσεις Word χειρίζονται τα cell separators όταν επικολλούνται τα περιεχόμενα ενός πίνακα με διαφορετικό τρόπο απ’ ότι παλαιότερα. Το report generation toolkit χρησιμοποιεί line feed (LF, \n) για να κάνει διαχωρισμό μεταξύ στηλών και γραμμών, γेमίζοντας ένα κελί κάθε φορά, από αριστερά προς τα δεξιά, και συνεχίζοντας στην επόμενη σειρά. Ωστόσο, οι νεότερες εκδόσεις Word προσδοκούν ένα χαρακτήρα tab (\t) για να διαχωρίσουν κάθε στήλη και μετά ένα line feed για να συνεχίσουν στην επόμενη σειρά.



Μια λύση σε αυτό το πρόβλημα είναι η αντικατάσταση των αρχικών αρχείων “Word\_Update\_Table.vi” και “Word\_Insert\_Table.vi” με τις τροποποιημένες εκδόσεις τους που βρίσκονται στο “Support Files\Word Array Fix\”. Τα πρωτότυπα αρχεία βρίσκονται στο “\National Instruments\LabVIEW 2011\vi.lib\addons\\_office\\_wordsub.llb” μέσα στο installation directory (συνήθως “C:\Program Files\”).









NATIONAL TECHNICAL UNIVERSITY of ATHENS  
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING  
DIVISION OF COMMUNICATION, ELECTRONIC AND INFORMATION ENGINEERING

**Development of a test-bench control for piezoelectric  
actuators' characterization**

DIPLOMA THESIS

of

**IOANNIS A. KOUKOVINIS PLATIAS**

**Supervisor:** Elias Koukoutsis  
Assistant Professor, NTUA

Athens, June 2014

This page intentionally left blank.



NATIONAL TECHNICAL UNIVERSITY OF ATHENS  
SCHOOL OF ELECTRICAL AND  
COMPUTER ENGINEERING  
DIVISION OF COMMUNICATION, ELECTRONIC  
AND INFORMATION ENGINEERING

## **Development of a test-bench control for piezoelectric actuators' characterization**

**DIPLOMA THESIS**

of

**IOANNIS A. KOUKOVINIS PLATIAS**

**Supervisor:** Elias Koukoutsis  
Assistant Professor, NTUA

Approved by the examining committee on the 11<sup>th</sup> of May 2014.

.....  
Elias Koukoutsis  
Assistant Professor

.....  
Konstantinos Papaodyssefs  
Associate Professor

.....  
George Cambourakis  
Associate Professor

Athens, June 2014

.....

**IOANNIS A. KOUKOVINIS PLATIAS**

Graduate of School of Electrical and Computer Engineering, NTUA

Copyright © Ioannis A. Koukovinis Platias, 2014.  
All rights reserved.

No part of this thesis may be reproduced, in any form or by any means, without permission in writing from the author.

Any opinions and conclusions expressed herein are solely those of the author and should not be construed as representing the official opinions or policy of the National Technical University of Athens



## Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την ανάπτυξη πειραματικής διάταξης για τον χαρακτηρισμό πιεζοηλεκτρικών υλικών. Η ολοένα και αυξανόμενη ζήτηση για μια άμεση και αξιόπιστη διαδικασία χαρακτηρισμού πιεζοηλεκτρικών υλικών οδήγησε στην ανάγκη ανάπτυξης μιας πειραματικής διάταξης, η οποία θα μπορεί να πραγματοποιήσει μετρήσεις υστέρησης (hysteresis) και ερπυσμού (creep) των πιεζοηλεκτρικών υλικών μεταξύ των λειτουργιών της. Η διάταξη που αναπτύχθηκε θα χρησιμοποιηθεί μεταξύ άλλων στα πλαίσια του UA9 Project, το οποίο είναι μια διακρατική σύμπραξη που σκοπό έχει την βελτίωση του συστήματος collimation του LHC (Large Hadron Collider – Μεγάλος Επιταχυντής Αδρονίων), του μεγαλύτερου και ισχυρότερου επιταχυντή σωματιδίων παγκοσμίως που βρίσκεται στο CERN.

Το λογισμικό για την διάταξη αναπτύχθηκε στο περιβάλλον οπτικού προγραμματισμού LabVIEW της National Instruments ενώ τα scripts για την μεταγενέστερη επεξεργασία των δεδομένων που συλλέχθηκαν σε Matlab. Όλες οι σημαντικές πτυχές της σχεδίασης του λογισμικού αναλύονται με λεπτομέρεια, μεταξύ των οποίων η σχεδιαστική αρχιτεκτονική, η διαχείριση, απεικόνιση και μετεπεξεργασία των δεδομένων, ο συγχρονισμός της διαδικασίας λήψης δεδομένων, όλες οι επιλογές εκτέλεσης καθώς και η αυτόματη παραγωγή αναφοράς.

**Λέξεις Κλειδιά:** Πιεζοηλεκτρικά υλικά, LabVIEW, Matlab, υστέρηση, ερπυσμός, UA9, CERN, LHC, Μεγάλος Επιταχυντής Αδρονίων

This page intentionally left blank.

## **Abstract**

The present diploma thesis deals with the development of a test-bench control for piezoelectric actuators characterization. The growing demand for a fast and reliable procedure for piezoelectric actuators' characterization has subsequently led to the need for development of a test-bench structure that can perform hysteresis and creep measurements among its other functions. The developed test bench will be used to provide results that will be utilized within the UA9 project, a collaborative initiative that has undertaken the task of improving the collimation system, a critical component of the Large Hadron Collider (LHC), the world's largest and most powerful particle accelerator at CERN.

The software part of the project is written in National Instruments' LabVIEW visual programming development environment and the scripts for the post-processing of the collected data in Matlab. All of the important software design aspects are covered in detail and those include among others the design architecture, the data management, representation and post-processing, the synchronization of the data acquisition, all the execution options and the automatic report generation.

**Keywords:** Piezoelectric, actuator, LabVIEW, Matlab, hysteresis, creep, UA9, CERN, LHC

This page intentionally left blank.

## **Acknowledgments**

I would like to thank my professor, Elias Koukoutsis, for his ongoing support and encouragement to pursue working at CERN. I would also like to express my sincerest gratitude to my supervisor at CERN, Mario Di Castro, for his trust and invaluable guidance. I wish to express my appreciation to all my colleagues at CERN, without whose support this thesis could not have been accomplished.

Lastly, I would like to thank my family and friends for their love and support throughout all these years.

This page intentionally left blank.

## Table of contents

<b>1</b>	<b>Introduction .....</b>	<b>19</b>
1.1	Introduction to CERN .....	19
1.2	Introduction to the LHC .....	20
1.2.1	<i>Purpose of the LHC</i> .....	22
1.2.2	<i>LHC Layout</i> .....	23
1.3	The LHC Collimation System.....	26
1.4	UA9 Project .....	28
1.5	Introduction to Piezoelectric Materials .....	31
1.5.1	<i>Hysteresis effect</i> .....	35
1.5.2	<i>Creep effect</i> .....	36
1.5.3	<i>Other considerations</i> .....	37
1.6	Introduction to LabVIEW .....	38
<b>2</b>	<b>Structure &amp; Hardware.....</b>	<b>41</b>
2.1	Introduction to the Test bench structure.....	41
2.2	State of the art .....	43
2.3	Technical Specifications .....	46
2.3.1	<i>Mechanical structure</i> .....	46
2.3.2	<i>Motor assembly</i> .....	46
2.3.3	<i>Sensors and auxiliary devices</i> .....	48
2.3.4	<i>PXI system</i> .....	51
2.4	Functional analysis.....	58
<b>3</b>	<b>LabVIEW Analysis .....</b>	<b>61</b>
3.1	Basic building concepts .....	61
3.1.1	<i>State Machine</i> .....	61
3.1.2	<i>Functional Global Variable (FGV)</i> .....	63
3.2	Main Analysis .....	65
3.2.1	<i>Overview</i> .....	65
3.2.2	<i>Server application</i> .....	67
3.2.3	<i>Client application</i> .....	72
3.3	Execution Options.....	78
3.3.1	<i>Creep test</i> .....	78
3.3.2	<i>Static-execution Hysteresis test</i> .....	81
3.3.3	<i>Dynamic-execution Hysteresis test</i> .....	84

3.4	Data management.....	86
3.5	DAQ Synchronization.....	94
3.6	Graph creation.....	99
3.7	Matlab Post-Processing.....	103
3.8	Report Generation.....	105
<b>4</b>	<b>Results .....</b>	<b>111</b>
<b>5</b>	<b>Conclusion and Future Work.....</b>	<b>117</b>
5.1	Conclusion .....	117
5.2	Future Work .....	117
<b>6</b>	<b>Bibliography .....</b>	<b>119</b>
<b>Appendix A</b>	<b>Matlab Code .....</b>	<b>123</b>
<b>Appendix B</b>	<b>LabVIEW Project Instructions.....</b>	<b>127</b>



## List of Figures

Figure 1. 1: Location of the underground LHC tunnel.....	20
Figure 1. 2: The accelerator complex at CERN.....	21
Figure 1. 3: Overall view of the LHC experiments .....	24
Figure 1. 4: Overview of the LHC layout.....	26
Figure 1. 5: Collimator base-support assembly .....	27
Figure 1. 6: An LHC collimator with the vacuum tank open, left, and a "beam's eye-view" of the collimator aperture, right. ....	27
Figure 1. 7: Crystal collimation process.....	29
Figure 1. 8: Piezoelectric actuated goniometer .....	31
Figure 1. 9: Piezoelectric materials' behaviour .....	32
Figure 1. 10: Linear actuator .....	33
Figure 1. 11: Bender actuator .....	34
Figure 1. 12: Typical hysteresis plot .....	34
Figure 1. 13: Full hysteresis curve .....	36
Figure 1. 14: Creep pattern over time.....	37
Figure 1. 15: Typical VI example.....	39
Figure 2. 1: Main test bench structure .....	42
Figure 2. 2: Peripheral devices- National Instruments PXI chassis and modules (bottom-left), motor servo amplifier and mechanical brake power supply (top-left) and stack of linear amplifier for piezo actuators and capacitive measurement modules (right).....	43
Figure 2. 3: Setup for single PZT stack (top-left) and setup for two cylindrical PZT stacks (top-right and bottom) .....	44
Figure 2. 4: Instron 8516 servo-hydraulic fatigue system .....	45
Figure 2. 5a (left) and Figure 2. 5b (right).....	45
Figure 2. 6: Motor servo amplifier (left) and motor (right).....	47
Figure 2. 7: Neugart PLPE 70 front view (left), rear view (middle) and innards (right).....	47
Figure 2. 8: R+W SK2.....	48
Figure 2. 9: Fogale MCC-10 attached to MC900 (left) and the modules at hand (right) .....	49
Figure 2. 10: Cedrat LA75B and LC75B .....	49
Figure 2. 11: Althen SG-KP-12E-420 (left), Tedea Huntleigh 615 (middle) and Althen ALF256 (right) .....	50

Figure 2. 12: PCB Piezotronics 356B18 accelerometer .....	50
Figure 2. 13: Images of a populated and an unpopulated PXIe-1078 chassis .....	52
Figure 2. 14: NI PXIe-8133 .....	53
Figure 2. 15: PXIe-4357 (top-left), TB-4357 (top-right), TB-4357 connected to PXIe-4357 (bottom-left) and TB-4357 innards (bottom-right).....	54
Figure 2. 16: PXIe-4496 .....	55
Figure 2. 17: PXI-4461 .....	55
Figure 2. 18: PXI-7841R (left) and SCB-68 connector block (right).....	56
Figure 2. 19: Relay board and external power supply used to control the motor brake .....	57
Figure 2. 20: Auxiliary structure for creep measurements (left) and close-up of piezo actuator and capacitive sensor in place (right) .....	58
Figure 2. 21: Piezo excitation pattern for creep measurement .....	59
Figure 2. 22: Primary structure (left and center) and close-up of piezo actuator and capacitive sensor (right).....	59
Figure 3. 1: State machine architecture .....	62
Figure 3. 2a (left): Single default transition and Figure 3. 2b (right): Two possible transitions where the outcome is defined by the select function.....	63
Figure 3. 3: Functional Global Variable general form .....	63
Figure 3. 4: FGV with “Set” and “Get” states.....	64
Figure 3. 5: Flowchart of server and client applications .....	66
Figure 3. 6: Server application – Initialization state.....	68
Figure 3. 7: Server application – DAQ Configuration and Front Panel state.....	69
Figure 3. 8: Server application – Data acquisition and data transmission state with case variants .....	70
Figure 3. 9: Server application – Close references state.....	71
Figure 3. 10: Server application – Error state.....	71
Figure 3. 11: Client application – Initialization state .....	73
Figure 3. 12: Client application – Data reception and data logging state.....	74
Figure 3. 13: Client application: Post-processing options and Front Panel state with case variants .....	76
Figure 3. 14: Client application – Close references state .....	77
Figure 3. 15: Client application – Error state .....	78
Figure 3. 16: Creep test Front Panel.....	79
Figure 3. 17: Creep output pattern creation.....	80
Figure 3. 18: Hysteresis test Front Panel.....	82
Figure 3. 19: Static-execution load setting Front Panel.....	83

Figure 3. 20: Dynamic-execution load setting Front Panel .....	85
Figure 3. 21: “ <i>SyncedAcquisitionProcess</i> ” subvi .....	86
Figure 3. 22: “Enqueue” state of data FGVs in server application.....	87
Figure 3. 23: “ <i>PT100AcquisitionProcess</i> ” subvi.....	87
Figure 3. 24: “ <i>accelerometerAcquisitionProcess</i> ” subvi.....	88
Figure 3. 25: “ <i>accelerometerAcquisitionProcess</i> ” subvi - FFT.....	88
Figure 3. 26: Data transmission over network stream .....	89
Figure 3. 27: “Dequeue” state for most of the data FGVs in server application .....	89
Figure 3. 28: PT100 “Dequeue” state in server application .....	90
Figure 3. 29: Accelerometer and FFT “Dequeue” FGV state in server application.....	91
Figure 3. 30: Data reception on the client application.....	91
Figure 3. 31: “Enqueue” state of data FGVs in client application.....	92
Figure 3. 32: “Dequeue” state for most of the data FGVs in client application .....	92
Figure 3. 33: Accelerometer and FFT “Dequeue” FGV state in client application.....	93
Figure 3. 34: Data logging in client application .....	94
Figure 3. 35: “ <i>initSyncStartDynamic</i> ” subvi .....	96
Figure 3. 36: “ <i>initSyncStartStatic</i> ” subvi.....	97
Figure 3. 37: “ <i>removeJunkSamples</i> ” subvi .....	98
Figure 3. 38: “ <i>SyncedAcquisitionProcess</i> ” subvi 1 <sup>st</sup> iteration .....	99
Figure 3. 39: “ <i>graphCreationClient</i> ” subvi.....	100
Figure 3. 40: “ <i>graphCreationClient</i> ” subvi code .....	101
Figure 3. 41: initial FFT data structure (left) and desired structure (right) .....	102
Figure 3. 42: “ <i>FFTGraphClient</i> ” subvi.....	102
Figure 3. 43: “ <i>unifiedLengthArrayClient</i> ” subvi .....	103
Figure 3. 44: “ <i>normalizeArrayClient</i> ” subvi .....	103
Figure 3. 45: MathScript node.....	104
Figure 3. 46: Front Panel of “ <i>reportOptionsClient</i> ” subvi with all options shown.....	105
Figure 3. 47: “ <i>reportGenerationClient</i> ” subvi .....	106
Figure 3. 48: “ <i>hysteresisReportClient</i> ” subvi.....	107
Figure 3. 49: “ <i>creepReportClient</i> ” subvi.....	107
Figure 3. 50: “ <i>leakageCurrentReportClient</i> ” subvi.....	108



## *Preface*

The work presented has been carried out in collaboration with the European Organization for Nuclear Research (CERN).



# 1

## *Introduction*

The purpose of this chapter is to introduce readers to the concepts dealt in the present thesis.

Starting off, an introduction to CERN is made, followed by a description of the Large Hadron Collider (LHC) and its structure and basic components. Subsequently, we focus on one of LHC's vital parts, the Collimation system, providing details about its function, as well as future trends. Specifically, we discuss about the proposed crystal collimation UA9 project, which will improve the functionality and efficiency of the collimation system and involves the utilization of piezoelectric elements. A small introduction to piezoelectric materials science, focusing mainly on the properties herein studied, is presented, as well as a brief introduction to the LabVIEW software by National Instruments, which allowed the work of this thesis to be carried through.

### *1.1 Introduction to CERN*

CERN, the European Organization for Nuclear Research, is considered one of the world's largest and most respected laboratories for scientific research and is located in the northwest suburbs of Geneva on the Franco-Swiss border. It currently has twenty

European member states, as well as Israel, which is the first (and currently only) non-European country granted full membership.

Established in 1954 and evolving ever since, CERN has been at the forefront of science research and several important discoveries have taken place in it, leading among others to two Nobel physics prizes and the invention of the World Wide Web.

Currently, more than 10000 scientists and 600 institutions from more than 100 countries participate in the activities of CERN, making it one of the most multicultural and diverse environments. [1][2][3]

## ***1.2 Introduction to the LHC***

The LHC (Large Hadron Collider) is a circular accelerator with a circumference of 27 km, located 175 m beneath the Franco-Swiss border near Geneva, built by the European Organization for Nuclear Research (CERN).



**Figure 1. 1: Location of the underground LHC tunnel**



The LHC is located in the same tunnel as the previous accelerator at CERN, the Large Electron Positron (LEP), the experimental results of which helped to establish the Standard Model's<sup>1</sup> validity through its years of operation from 1989 to 2000. [4]

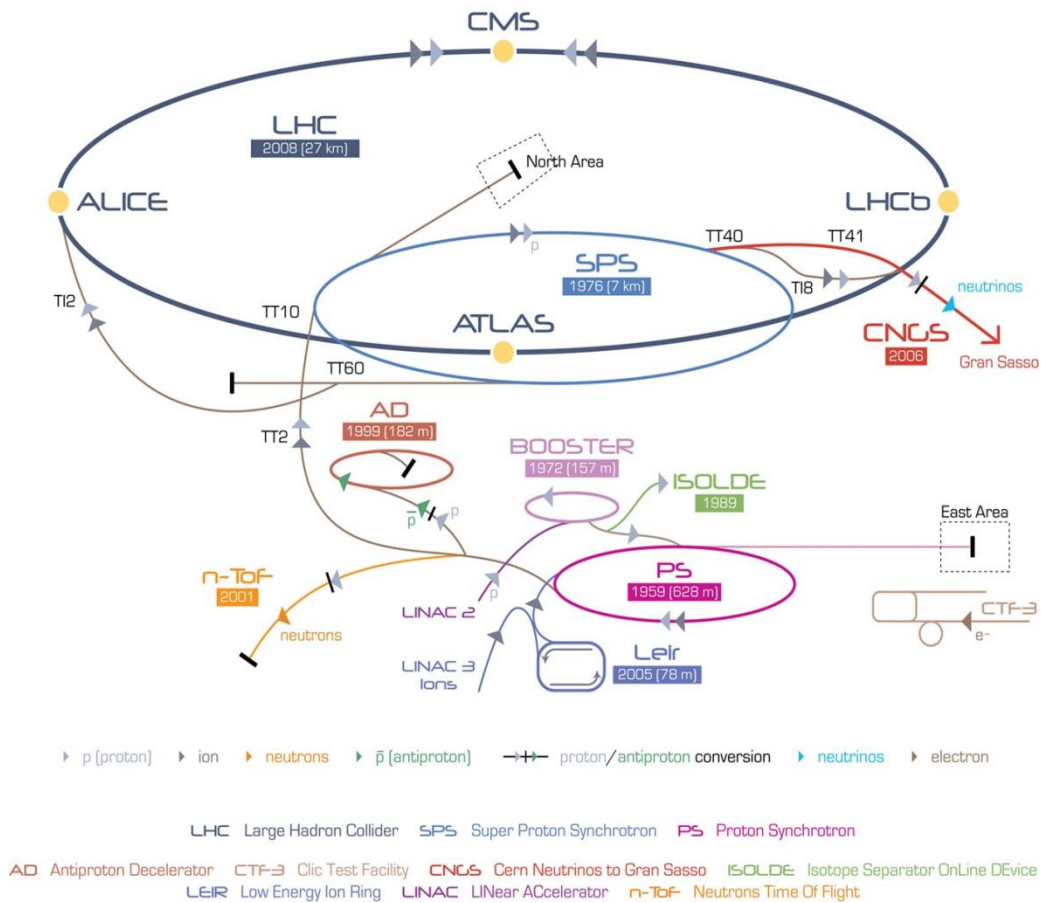


Figure 1. 2: The accelerator complex at CERN

As soon as the particles are produced from the ionization of hydrogen, linear accelerators (LINAC), boost their energy before being injected into the 630 m circumference Proton Synchrotron (PS). The protons are then injected into the 7 km circumference Super Proton Synchrotron (SPS) which accelerates them up to 450

<sup>1</sup> The Standard Model (SM) is a group of theories that describe the fundamental particles and their interactions and comprise quantum electrodynamics (QED) and the theory of quantum chromodynamics (QCD). Despite its experimental success, the SM is not yet considered as a complete theory because it does not include gravitation, dark matter or dark energy.

GeV. The LHC then accelerates the particles to 7 TeV<sup>2</sup>. Two proton beams, with energy of 7 TeV each collide, leading to a center-of-mass energy of 14 TeV per collision.

Each beam contains 2808 bunches with a minimum distance of 7 m at maximum capacity. Each bunch contains about  $1.15 \times 10^{11}$  protons with a length of about 11 cm and a diameter of about 400  $\mu\text{m}$ . Close to the interaction points, the bunches are compressed until their length is 7.5 cm and their diameter 16  $\mu\text{m}$  in order to increase the probability of collision. Twenty proton-proton collisions per bunch are expected during a high luminosity run, but given the collision frequency of 40 MHz, the total number of collisions is over 800 million per second. Luminosity, being equal to the number of particles passing through a cross section of the beam per unit time, reaches its peak value of  $10^{34} \text{ cm}^{-2} \text{ s}^{-1}$  after two lower phases,  $5.0 \cdot 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$  and  $2 \cdot 10^{33} \text{ cm}^{-2} \text{ s}^{-1}$ .

In order to deflect the proton beams in the LHC ring, 1232 superconducting (SC) dipole magnets of 14.3 m length each are used and 858 SC quadrupole magnets in order to focus them. For the low temperatures needed to induce superconductivity in the magnets, liquid helium is used. The cryogenics used to supply the helium constitute the largest refrigerated system in the world, operating at 1.9 °K. The LHC is regarded as the most complex scientific experiment in history. [5][6][7][8]

### ***1.2.1 Purpose of the LHC***

One of the most important goals of the LHC was to prove the existence of the Higgs boson. Initially theorised in 1964, the Higgs boson is an elementary particle and its existence was announced at CERN on the 4<sup>th</sup> of July 2012. The importance of this discovery lies in the confirmation of the Higgs field, which is crucial to the Standard Model and other theories within particle physics. With the Standard Model, physicists have been able to describe the fundamental particles and the interactions between them. The Higgs boson, a massive scalar elementary particle, explains why the other

---

<sup>2</sup> TeV is an energy unit used in particle physics. 1 TeV is about the energy of motion of a flying mosquito. What makes the LHC so extraordinary is that it squeezes energy into a space about a million times smaller than a mosquito.

elementary particles are massive and in electroweak theory it generates the masses of the leptons (electron, muon, tau) and quarks. [9]

Data from all detectors will also be analysed for the observation of extra dimensions. String theory predicts that additional spatial dimensions exist and they can maybe be observed at very high energies.

### *1.2.2 LHC Layout*

The LHC is composed of several components which accomplish distinguished tasks. Although vast in number, the main components of the LHC can be summarized in the following fashion.

The key elements of course are the **detectors**. Four particle detectors have been constructed and housed in huge underground caverns, corresponding to the four collision points of the two beams. These are ATLAS (A Toroidal LHC ApparatuS), CMS (Compact Muon Solenoid), ALICE (An LHC Ion Collision Experiment) and LHCb (LHC beauty). ATLAS and CMS are designed to study as broad a variety of physics as possible and are considered to be general-purpose experiments. ALICE is dedicated to heavy ions collisions and it recreates conditions similar to those just after the Big Bang in order to study and analyse the properties of the quark-gluon plasma while the LHCb is dedicated to B-physics phenomena – in other words, it focuses on B-hadron decays which should give us better insight into the differences between matter and anti-matter as well as possibly providing indirect evidence for SM-violating processes. In addition there are also the TOTEM (Total Cross Section, Elastic Scattering and Diffraction Dissociation at the LHC), LHCf (Large Hadron Collider forward) and MoEDAL (the Monopole and Exotics Detector At the LHC). [10]

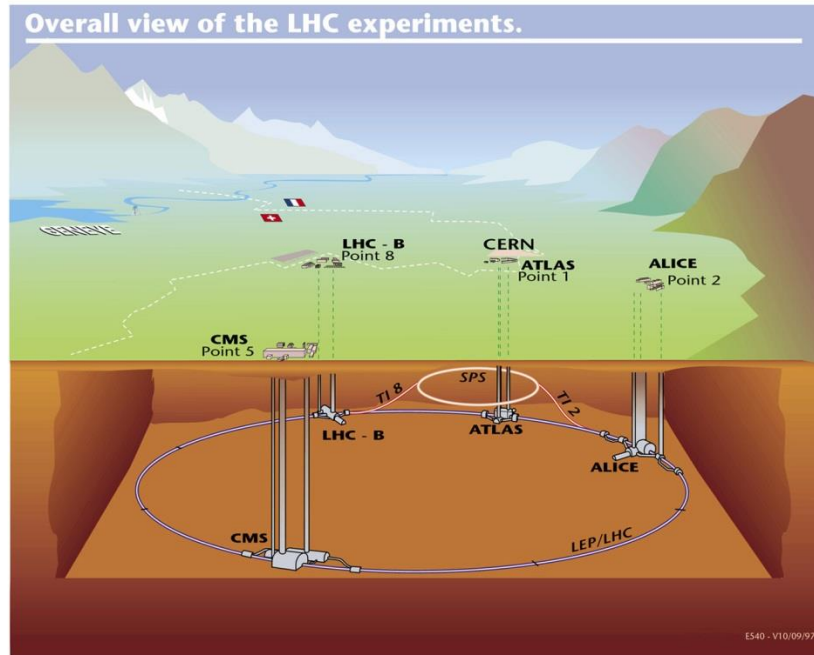


Figure 1. 3: Overall view of the LHC experiments

Other important elements in the function of the LHC are the **superconducting magnets**. They are responsible for keeping the beams stable and precisely aligned. Specifically **dipole magnets** are some of the most complex parts of the LHC and are used to bend the trajectory of the particles. This is absolutely necessary as LHC is a *circular accelerator*, which means that the beam circulates many times in closed orbit before reaching its peak energy level, as opposed to a *linear accelerator*, where charged particles travel along a straight trajectory and go through a number of accelerating structures. There are 1232 main dipoles installed in the LHC, each 14.3 metres long and weighing in at 35 tonnes.

Other vital parts are the **quadrupole magnets**. Those help to keep the particles in a tight beam; in other words, they focus the beam. The idea behind this is that when particles are bunched together, they are more likely to collide in greater numbers when they reach the LHC detectors. In addition to the dipole and quadrupole magnets, there are also smaller sextupole, octupole and decapole magnets for correcting small imperfections in the magnetic field at the extremities of the dipoles. [11][12]

Of great importance are also the **radiofrequency (RF) cavities**. An RF cavity is a metallic chamber that contains an electromagnetic field and its primary purpose is to accelerate charged particles. The RF cavity is constructed to a specific shape and size

so that electromagnetic waves become resonant and build up inside the cavity. The result is that charged particles passing through the cavity feel the overall force and direction of the resulting electromagnetic field, which transfers energy to push them forwards along the accelerator. One important detail is that the field in an RF cavity is made to oscillate (change direction) at a given frequency, so timing the arrival of particles is critical. [13]

Another critical element of the LHC is the **beam dump system**. Beams do not circulate inside accelerators indefinitely. As particles collide with the sides of the beam pipe or with each other, the beams ‘degrade’ and so they become less likely to give collisions of interest. Accelerator physicists can choose to ‘dump’ the beams, removing them from the accelerator and sending them to be safely absorbed at a ‘beam dump’ – usually a radiation-shielded block deep underground. To understand the necessity of the beam dump system, the nominal LHC beam contains an unprecedented stored energy of 350 MJ, contained in 2808 bunches with a beam sigma of the order of 0.3 mm. The extremely high destructive power of such a beam imposes an external dump, where the beam must be extracted completely from the LHC, diluted to reduce the peak energy density and then absorbed in a dedicated system. [14][15]

Last but not least is the LHC **collimation system**. The high luminosity performance of the LHC relies on storing, accelerating, and colliding beams with unprecedented intensities. The transverse energy density of the nominal beam is 1000 times higher than previously achieved in proton storage rings. Tiny fractions of the stored beam suffice to quench<sup>3</sup> a super-conducting LHC magnet or even to destroy parts of the accelerators. The LHC collimation system provides protection to the accelerator against unavoidable regular and irregular beam losses and achieves the necessary beam halo cleaning. [16]

---

<sup>3</sup> “Quenching” occurs when any part of a superconducting cable of a magnet goes from the superconducting to the normal resistive state. It occurs when the critical temperature, the critical current or the critical field is by-passed, or in case of movement of the superconductor by several mm (friction and heat dissipation), cooling failures or beam loss.

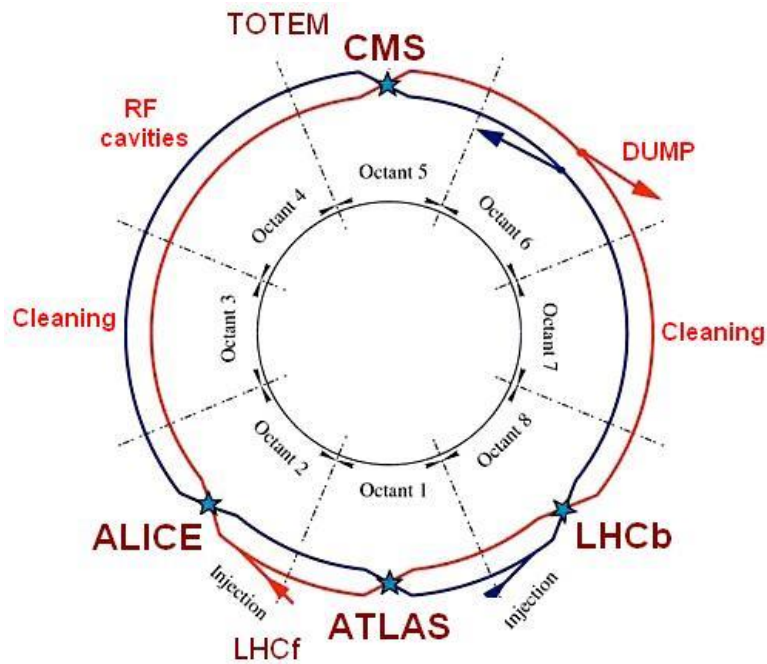


Figure 1. 4: Overview of the LHC layout

### 1.3 The LHC Collimation System

As briefly explained in the previous paragraph, the LHC collimation system constitutes the LHC’s defence against unavoidable beam losses. Contrary to an ideal experiment, real-world experience has shown that certain processes lead to beam losses. The manipulations needed to prepare the beams for collision, such as injection, the energy ramp and “squeeze”, as well as the collisions themselves, all involve unavoidable beam losses. And as it is easily understood, those losses become greater as the beam current and luminosity are increased. What is more, due to the LHC’s superconducting environment, an efficient beam-loss cleaning to avoid “quenches” from uncontrolled losses is crucial. [17]

The main purpose of the collimation system is to clean away the beam halo while maintaining losses at sensitive locations below safe limits. The current system ensures that peak losses below 0.01% of the energy lost from the beam is deposited in the cold magnets.

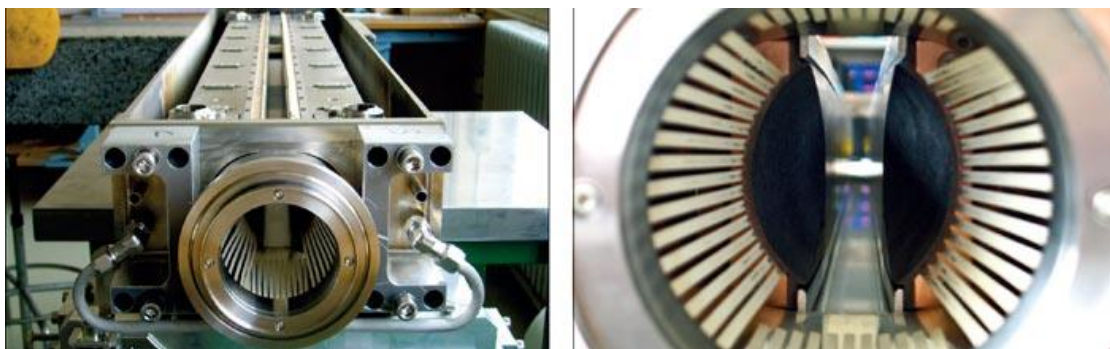
The fully installed collimator consists of five main parts: (1) the base-support, which serves as stand of the collimator in the tunnel, (2) the alignment plate, which allows the adjustment of the collimator and (3) the lower plugin, which provides cooling

water and electrical connections to the collimator. The aforementioned parts constitute the base-support assembly, which can be seen in Fig. 1.5.



**Figure 1. 5: Collimator base-support assembly**

The rest of the collimation system consists of (4) the vacuum tank with the two jaws and motors for moving the jaws and (5) the upper support assembly, which hosts all needed parts for the installation in the tunnel on a base-support assembly. The discrimination between base-support and upper-support assembly serves to quickly install and exchange a collimator in case of damage.



**Figure 1. 6: An LHC collimator with the vacuum tank open, left, and a "beam's eye-view" of the collimator aperture, right.**

The main tank consists of two jaws that define a slit for the beam, effectively constraining the beam halo from both sides. These jaws are enclosed in a vacuum tank that can be rotated in the transverse plane to intercept the halo, whether it is

horizontal, vertical or skew. Precise sensors are used to monitor the jaws' position and collimators gaps. There are also temperature sensors mounted on the jaws. All the aforementioned are connected to the beam-interlock system and trigger a beam dump if potentially dangerous conditions are detected.

The LHC collimation system utilizes a multi-stage cleaning process. Primary, secondary and tertiary collimators and absorbers are utilized to reduce halo particles to tolerable levels. Robust carbon-based and non-robust but high-absorption metallic materials are used for different purposes.

The LHC collimation system is the largest and most advanced cleaning system ever built for a particle accelerator. It consists of 84 two-sided movable collimators of various designs and materials, which together with injection protection collimators translate into a total of 396 degrees-of-freedom.

At the LHC's top energy, a beam size of less than 200  $\mu\text{m}$  translates into the necessity for the collimators to act as high-precision devices. The inherent complexity of the current collimators, as well as the desire to reach even higher levels of halo cleaning efficiency, led subsequently to the UA9 project. [18][19]

## ***1.4 UA9 Project***

The UA9 collaboration, which consists of several universities and research institutions from France, Italy, Russia and the United Kingdom, set out to investigate how tiny bent crystal could improve how beams are collimated in large colliders, such as the LHC.

What makes bent crystals unique and useful in the collimation process is that the planes in crystalline solids can constrain the directions that charged particles take as they pass through. This "channelling" property of crystals can be used to steer particle beams. After being deflected by the crystal, halo particles are intercepted by existed secondary collimators that will be used as absorbers.



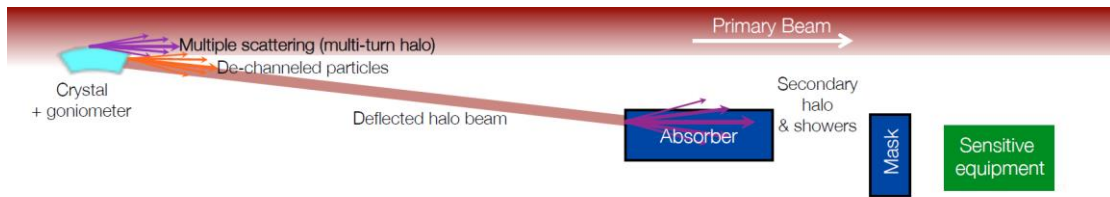


Figure 1. 7: Crystal collimation process

As explained previously, current collimators are massive and the absorbers are very close to the beam, making alignment very crucial. The utilization of crystals comes to solve these issues, as a crystal used as a primary collimator can deflect halo particles coherently at large angles and direct them into a secondary collimator-absorber. In this way, the massive collimator-absorber can be placed at an increased distance from the beam, thus reducing the complexity of the system and at the same time increasing its efficiency.

The first tests of this idea were carried out in 2009, using beams from the SPS (Super Proton Synchrotron) and they have been largely successful. Unpublished results of tracking simulations on the proposed LHC collimation system, indeed, predict a cleaning efficiency 5 to 10 times better than with the present collimation system. However, adapting this idea to a high-energy collider like the LHC poses a few challenges. In particular, in steady conditions a bent crystal can deposit up to 0.5 MW power in a small spot on the collimator-absorber, which means that the collimator-absorber should be able to withstand this power for several seconds without damage. Furthermore, the growth rate of the beam halo is so slow that the first impacts on the crystal occur in a region exceeding a few atomic layers. This translates into the requirement to have a flat surface parallel to the crystal planes with unprecedented tolerance. Lastly, it is true that the higher the particle energy, the lower the angular acceptance for channelling. That calls for alignment mechanisms with high angular accuracy and in this context relies the necessity to construct a structure to characterize piezoelectric elements.

At this point it is necessary to present some details of the crystal collimator structure designed for the LHC proposed by the UA9 collaboration. The whole design revolves around a goniometer suitable for orienting and positioning the crystal with an angular accuracy of the order of  $1 \mu\text{rad}$  required for the LHC operation. The proposed solution consists of two stages, a mechanical linear stage and a piezoelectric rotational stage

that guarantee a linear variation of the angular orientation of the crystal also during the dynamical transients of angular adjustments. A layout has been proposed that introduces the minimum number of extra devices in the standard LHC layout, taking advantage of the existing collimators and beam instrumentation.

So far, the rotational movement is achieved by transforming a linear movement with a lever arm, using hybrid stepping motors for the motors and LVDTs for monitoring the linear positioning of the axes. This configuration has mainly fulfilled the specifications in steady conditions, however, an overshoot of the linear and angular positioning has been observed under specific conditions.

To overcome the angular accuracy issue, the piezoelectric actuated goniometer shown in Fig. 1.8 has been proposed. The aforementioned solution guarantees the following functionality:

- Move the crystal linearly and perpendicular to the beam with a stroke of 60 mm and a linear resolution of 5  $\mu\text{m}$ .
- Rotate the crystal through a total yaw angular range of  $\pm 10$  mrad with an angular resolution of 0.1  $\mu\text{rad}$  and angular accuracy of  $\pm 1$   $\mu\text{rad}$  over a linear stroke of 10 mm from the beam axis.
- Perform steps in the yaw angle with an overshoot of  $< 10\%$  and a settling time to within 99% of the final value of 20 ms.
- Function correctly after a bake out at a temperature of 250  $^{\circ}\text{C}$ .
- Function correctly up to a total accumulated radiation dose of 10 MGy.
- Be easily installed using the LHC Collimator quick plug support system.
- Be transparent to normal LHC operation.

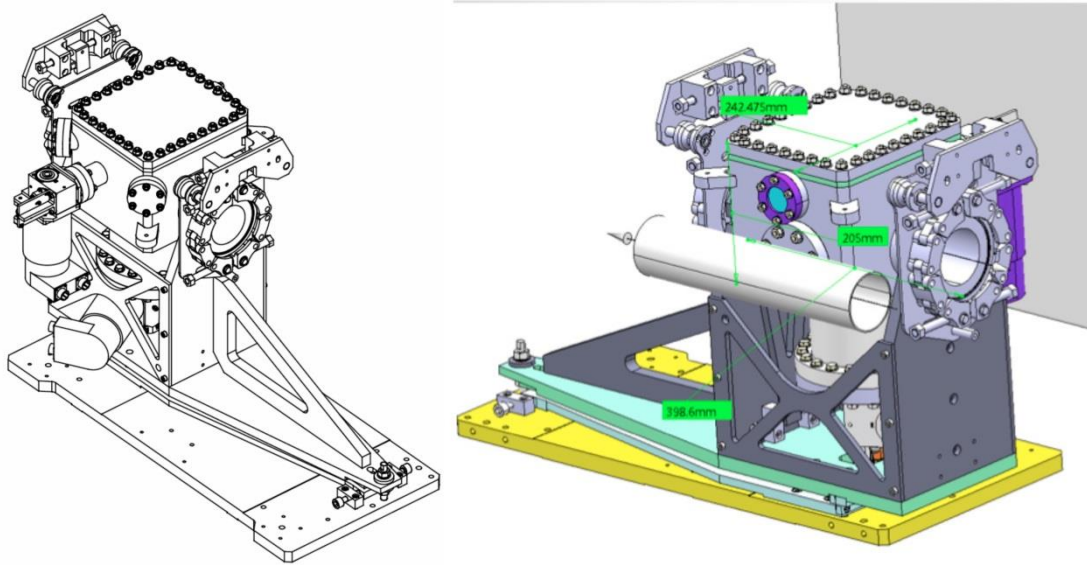


Figure 1. 8: Piezoelectric actuated goniometer

So, all in all, the design incorporates a piezoelectric actuated rotational stage mounted on a high precision linear axis, which together allow the rotational and linear specifications to be met.

In that respect, it was deemed necessary to develop a piezoelectric material characterization structure which would allow us to characterize piezoelectric elements, verify existing measurements and examine complex issues, e.g., how the functionality of piezoelectric elements is affected by accumulated radiation. [20][21][22][23]

## ***1.5 Introduction to Piezoelectric Materials***

The name piezo derives from the Greek work “πιέζειν”, which means “to press”. Piezoelectric materials possess two inverse abilities: the ability to develop electrical charge on their surface when mechanical stress is exerted on them and the ability to develop mechanical stress when an electric charge is included. These properties can be visualised in Fig. 1.9.

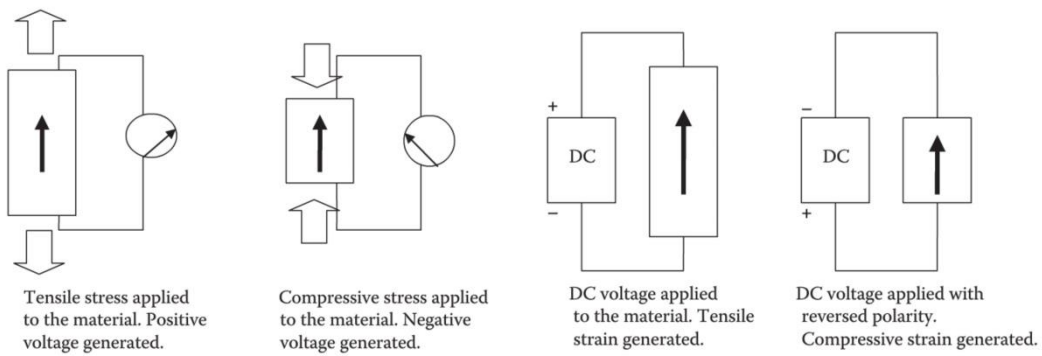


Figure 1. 9: Piezoelectric materials' behaviour

There are natural piezoelectric materials like quartz and tourmaline but the piezoelectric effect is very low in them and the material has to be cut into specific shapes. Artificial crystals such as Rochelle salt and lithium sulphate can acquire the piezoelectric effect by industrial processes but it requires a long processing time. This leads us to ceramic materials like Lead Zirconate Titanate (PZT)<sup>4</sup>, which do not possess any piezoelectric virtues by nature, but they can acquire them through a procedure called *poling*. There are several advantages in ceramic piezos like their easy manufacturing as well as their large flexibility in design and shape considerations.

The piezoelectric effect was first discovered by Jacques and Pierre Curie in 1880 and ever since piezo materials have been utilized in a variety of applications. Such are actuators, sensors, high voltage and power sources, piezoelectric motors, noise and vibration control, the IT industry (quartz clocks used to generate clock pulses or in Hard Disk Drive designs), the musical instrument industry (acoustic guitar pickups and microphones) and the medical industry (pressure and heartbeat monitoring, ultrasonic imaging among others).

As stated earlier, the UA9 project focuses on the utilization of piezoelectric actuators in the goniometer design and as such it is appropriate to provide further information about them.

The indirect piezoelectric effect, in which an electrical input changes the dimensions of a piezoelectric material, has resulted in the development of many piezoelectric actuators for varied applications. Their advantage is that they don't require complex designs and they can generate considerable forces with quick response for low-voltage

<sup>4</sup> PZT is the chemical formula, where P stands for "Plombus" which is Lead.

input. Three are the main types of piezoelectric actuators: the linear, stacked and bending actuators.

The **linear actuators** have a linear strain which is due to the uniform and equal distribution of the induced electric charge on their surface. Linear actuators generate a linear strain proportional to the applied voltage. As illustrated in Fig. 1.10, the deformation takes place in both the X and Y directions.

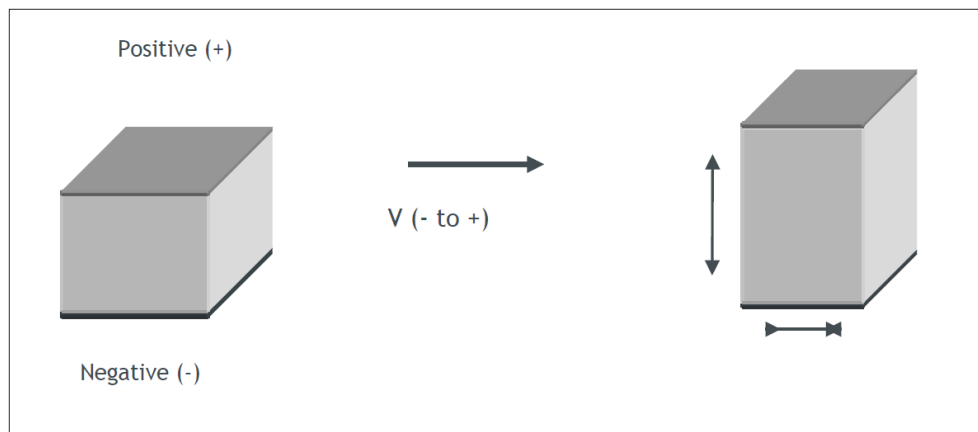


Figure 1. 10: Linear actuator

**Stacked multilayer actuators** on the other hand, are merely linear actuators glued one on another. The advantage of stacked actuators is that the force of the actuator will be the same as for linear actuators, but the displacement will be the sum of each linear actuator that is stacked. For example, if there are 10 linear actuators stacked, the displacement of the stacked actuator will be 10 times the displacement of a single linear.

Lastly, **bender actuators**, as the name implies, give a bending strain proportional to the applied voltage. Bender actuators have two layers poled in opposing directions. In Fig. 1.11 such an actuator can be seen. The blue line denotes the line between the two layers and is actually the grounding point. So, when voltage is applied, due to the opposing polarity of the two layers, one of them expands and the other compresses.

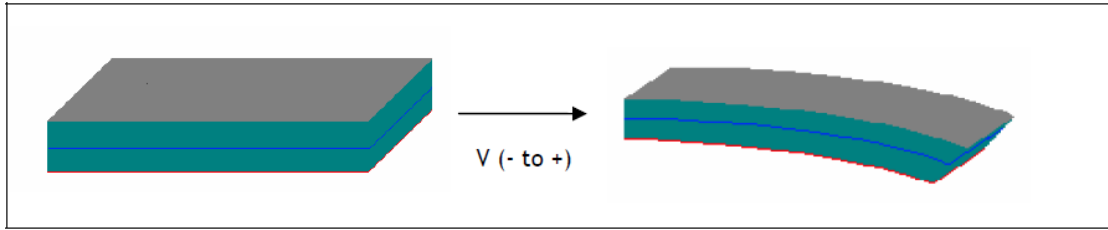


Figure 1. 11: Bender actuator

The exploitation of piezoelectric materials offers many advantages, however one must consider and cater for the unique effects they exhibit – especially the hysteresis and creep effect. All piezoelectric materials exhibit a mechanical hysteresis as the strain does not follow the same track upon charging and discharging. The hysteresis is expressed as a percentage and represents the maximum strain divided by the maximum difference between the two tracks (as seen for example in a typical hysteresis plot in Fig. 1.12). The mechanical hysteresis depends on the type of ceramics and can vary from 4% to 20%. On the other hand, the creep effect translates into the material continuing to expand for some time upon charging. Correspondingly, the material doesn't immediately return to the initial strain level upon discharging. Creep is usually measured in seconds and represents the time necessary for reaching the final strain value.

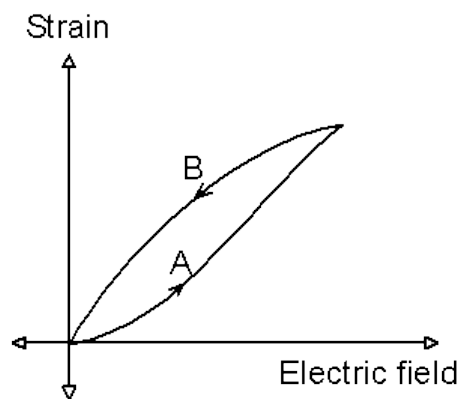


Figure 1. 12: Typical hysteresis plot

At this point it is advisable to provide further details about the aforementioned.  
 [24][25][26]

### 1.5.1 Hysteresis effect

The nature of piezoelectric materials is closely linked to the significant quantity of electric dipoles within these materials. A dipole is a vector, so it has a direction and a value in accordance with the electrical charges around. These dipoles tend to have the same direction when next to each other, and they altogether form regions called “Weiss domains”. The domains are generally randomly oriented but they can be aligned using the aforementioned process of poling, which is a process by which a strong electric field is applied across the material.

With that in mind, a step-by-step approach to the hysteresis effect would be the following. Initially, when there is no applied voltage, the ferroelectric domains are all randomly oriented and so the polarization is zero (point O). As the field is increased, the domains get oriented in the direction of the field and the polarization increases linearly at first. This is represented by the curve OA. As the field further increases, more and more domains get oriented, the curve becomes nonlinear and ultimately when all the domains get oriented, the polarization attains its maximum value (point B). The polarization at this point (denoted as  $P_o$  in Fig. 1.13) represents portion OE and is called *saturation polarization*. From that point, if the electric field is gradually decreased, the polarization decreases but the curve is not retraced. The polarization decrease is rather slow, meaning that the polarization lags behind the electric field, a process that is presented by portion BD. When the field reaches zero, there remains a finite polarization called the *remnant polarization*, represented by portion OE (and denoted as  $P_R$  in Fig. 1.13). In order to make it disappear, an electric field in the reverse direction has to be applied. The polarization becomes zero (point F) at an electric field of  $-E_c$ , called the *coercive field*. If the field is further increased in the reverse direction beyond  $E_c$ , the domains get oriented in the direction of the field and the polarization increases as the field increases, in the new reversed direction. The polarization reaches the maximum value, saturation polarization  $-P_s$ , in the reverse direction at point G. If the field now returns back to zero, the curve traces the path GH and there will be a remnant polarization  $-P_R$  (point H). If the field increases from zero in the positive direction, the remnant polarization disappears when the field reaches the value  $+E_c$ . Further increase in the field will trace the path  $E_cB$ , closing the loop. The closed loop is called the *hysteresis curve*. [27][28]

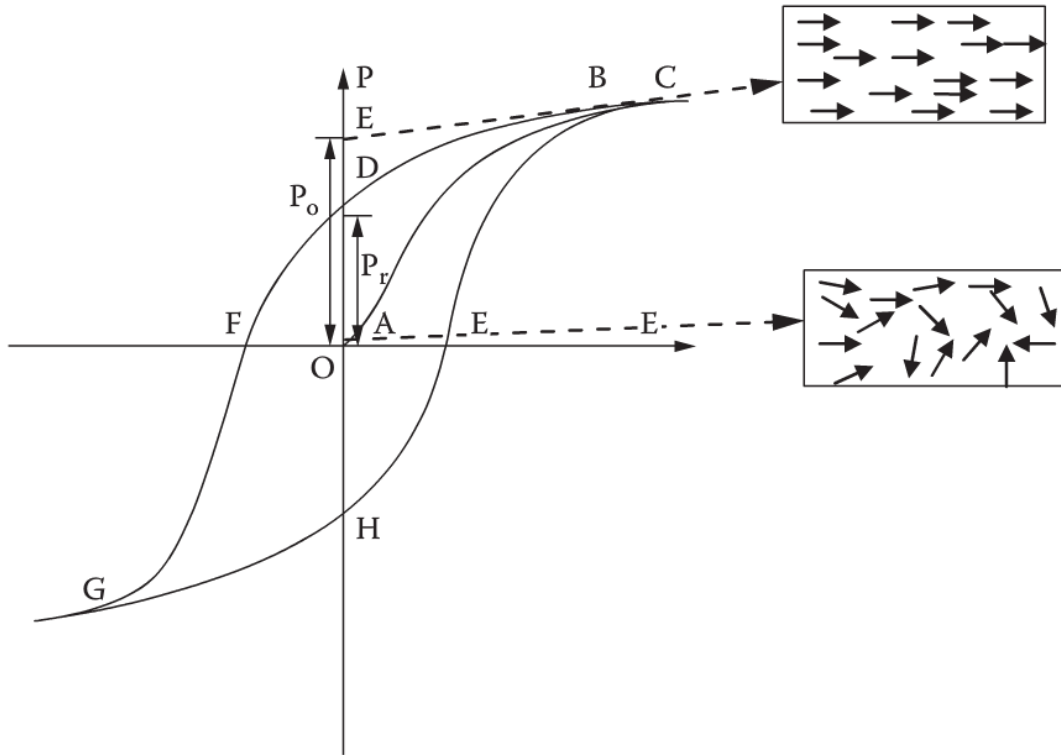


Figure 1.13: Full hysteresis curve

### 1.5.2 Creep effect

The same material properties responsible for hysteresis also cause the creep effect. As stated earlier, creep is a change in displacement with time without any accompanying change in the control voltage. If the operating voltage of a piezo actuator is changed, the remnant polarization continues to change, manifesting itself in a slow change of position. The rate of creep decreases logarithmically with time. Fig. 1.14 illustrates a typical response for a step input voltage. Initially, when applying any specified input voltage, the actuator shows its step response with a dynamic transient behaviour within a few milliseconds followed by the creep response, which as illustrated is a much slower drift response than the previous dynamic one. The equation that describes the creep effect is the following:

$$L(t) = L_o \left[ 1 + \gamma \log_{10} \left( \frac{t}{0.1} \right) \right],$$



where  $L(t)$  is the actuator's displacement for any fixed input voltage,  $L_o$  is a nominal constant displacement value which is the displacement of 0.1 s after applying the input voltage and  $\gamma$  is the creep factor that determines the rate of the logarithm. [29][30]

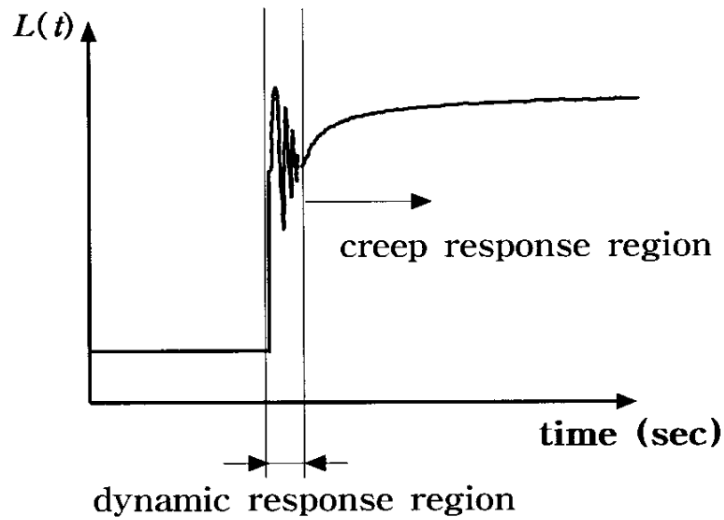


Figure 1. 14: Creep pattern over time

### 1.5.3 Other considerations

Nevertheless, the hysteresis and creep effects are not the only aspects in need of attention when designing a piezo-actuated system. Other aspects of the piezoelectric materials must be taken into consideration, such as the leakage current, the impedance and the ferroelectric transition temperature.

Regarding the leakage current, when operated below the resonant frequency, a piezo actuator behaves as a capacitor. The capacitance of the actuator is dependant of the area and thickness of the ceramic, as well as its material properties. For stacked actuators, the capacitance also depends on the number of layers. The actual electrical equivalent of a piezoelectric actuator is a parallel connection of capacitor  $C_{act}$  and resistor  $R_{act}$ . According to Ohm's law, if  $V$  is the applied voltage, the leakage current is equal to:

$$I_l = \frac{V}{R_{act}}$$

As with every electrical circuit, the electrical impedance should also be taken under consideration. The electrical impedance is the measure of the opposition that a circuit presents to a current when a voltage is applied. It is represented as  $Z$  and it is the complex ratio of the voltage to the current. Thus, the impedance can be represented as a complex number with the following formula:

$$Z = R + jX,$$

where the real part,  $R$ , represents the resistance and the imaginary part,  $X$ , represents the reactance.

Lastly, another important design consideration is the ferroelectric transition temperature. This represents the temperature beyond which ferroelectric materials lose their spontaneous polarization. [31][32][33]

## ***1.6 Introduction to LabVIEW***

LabVIEW is the abbreviation for “Laboratory Virtual Engineering Workbench” and is a platform developed by National Instruments. LabVIEW is one of the most established visual programming environments. The graphical language utilized is called “G” and is a dataflow programming language.

LabVIEW dates back to the 1980’s as it was originally released for the Apple Macintosh in 1986. Since that first version, it has evolved and adapted to the IT world’s needs and is currently offered for Windows, Linux or Mac OS X, with up-to-date version being 13.

What separates visual programming languages to traditional ones is that the user can create programs by manipulating program elements graphically rather than textually. Execution is determined by the structure of a graphical block diagram, the LabVIEW source code. The philosophy of dataflow is that the passage of data through nodes within the program determines the order of execution of the functions of the program. That means that a node will be scheduled for execution only when all its inputs contain valid data. That node will be executed as soon as a processor becomes available. After the node is executed, values on its output wires become available and flow towards the nodes that they in-turn are connected to. It can be easily understood that the programmer is able to control the order that the nodes execute by chaining

together nodes that have compatible inputs and outputs. G is inherently capable of parallel execution and multi-processing and multi-threading hardware is automatically exploited by the built-in scheduler.

The basic building blocks of programs written in LabVIEW are called VIs<sup>5</sup>. They can be thought of as similar to functions or subroutines in other programming languages. A VI includes the Front Panel, which contains all Controls and Indicators, the Block Diagram, which is the layout of nodes and wiring, the VI icon and the Connector Pane, which specifies the VI's input and output connection terminals. A typical example is presented in Fig. 1.15. [34][35][36][37][38]

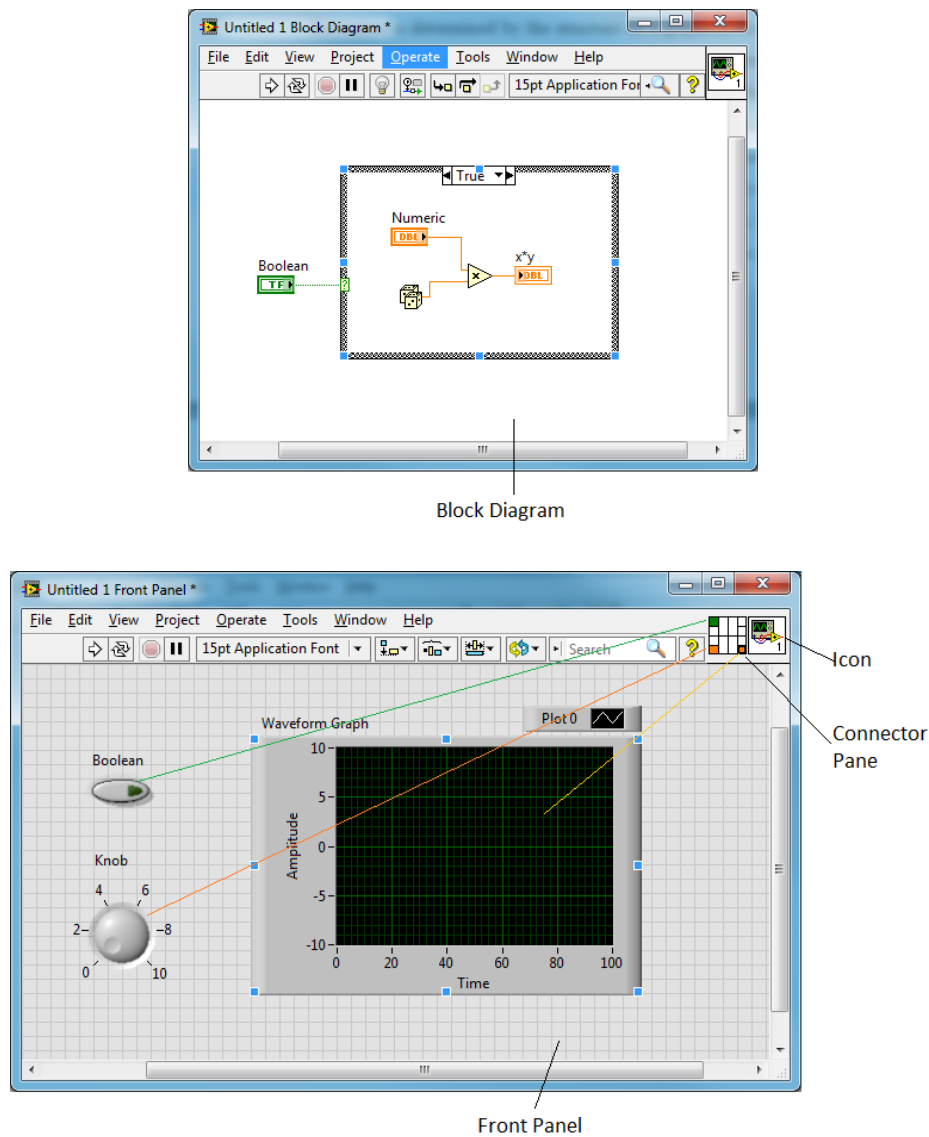


Figure 1. 15: Typical VI example

<sup>5</sup> VI stands for **V**irtual **I**nstrument



# 2

## *Structure & Hardware*

The purpose of this chapter is to introduce readers to the concept of the piezo test bench that was developed, as well as the hardware that was necessary to allow the project to come into fruition.

The chapter starts with a reference to the purpose of the test bench, as well as a short description of it and is followed by the state of the art of similar piezoelectric measurement test benches. The next section is dedicated to the technical aspects of the test bench and utilized hardware. Finally, the functions of the test bench are described in detail.

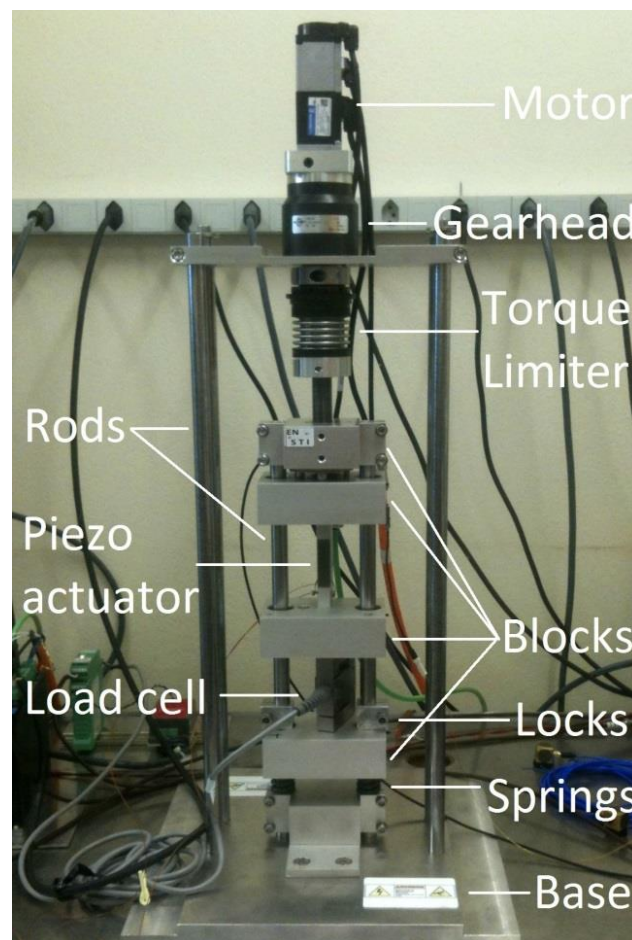
### *2.1 Introduction to the Test bench structure*

As mentioned in the first chapter, the development of the goniometer proposed by the UA9 collaboration requires the development of a system that can provide detailed analysis of the characteristics of piezoelectric materials.

For that purpose, the development of a specialised system dedicated to piezoelectric actuators' characterization, customized to the requirements at hand and easily adaptable to future needs was deemed necessary.

In summary, the system consists of five distinct parts: the *mechanical structure*, the *motor assembly*, the *sensors and auxiliary devices*, the *PXI chassis and modules* and the *software* developed in LabVIEW. The last part is covered thoroughly in the 3<sup>rd</sup> chapter, while the first four are covered in detail later in this chapter.

In brief, the main test bench structure consists of the mechanical structure (the base, the rods, the low-friction blocks, the locks and the springs), part of the motor assembly (the servo motor, the gearhead and the torque limiter) and a few mounted sensors (load cell, capacitive sensor). The current setup is illustrated in Fig. 2.1



**Figure 2. 1: Main test bench structure**

The peripheral devices of the structure mainly consist of the National Instruments PXI chassis and modules, which provide the interface between the software and the sensor readings, piezo excitation and motor movement, the motor servo amplifier, which drives the motor and translates the commands generated by the PXI to actual motor movement, the linear amplifier for the piezo excitation, which transforms a low-level

signal into a higher-level that can excite the piezo actuator and the rest of the sensors and auxiliary devices (the capacitive measurement system, which translates the capacitive sensor readings into voltage, the current board, the load cell amplifier, the temperature sensor and the accelerometers). A few of them are illustrated in Fig. 2.2.



Figure 2. 2: Peripheral devices- National Instruments PXI chassis and modules (bottom-left), motor servo amplifier and mechanical brake power supply (top-left) and stack of linear amplifier for piezo actuators and capacitive measurement modules (right)

## 2.2 State of the art

Over the past decades, the need to study piezoelectric materials has unavoidably led to the development of piezo characterization setups.

The most commonly found setups allow for a fixed load to be applied to the piezo element. In this way, scientists can study the behaviour of piezo elements in predefined stress levels under varying conditions of electrical excitation of the piezo element. Those setups are usually called “pre-stress” setups and can be found in "*PZT uniaxial stress dependence: experimental results*" by Audigier et al. [39],

"Characterization of multilayer piezoelectric actuators for use in active isolation mounts" by Wise [40] and "Characteristics comparison of piezoelectric actuators at low electric field: analysis of strain and blocking force" by Mekid and Lim [41] among others. The force is usually applied manually via a bolt, a clamp or a similar mechanism. Such two noteworthy compact setups that use a bolt are presented in "Extended Life PZT Stack Test Fixture" by Badescu et al. [42] and are illustrated in Fig. 2.3. The first one is adapted to a PZT stack and the second to two PZT cylindrical stacks.

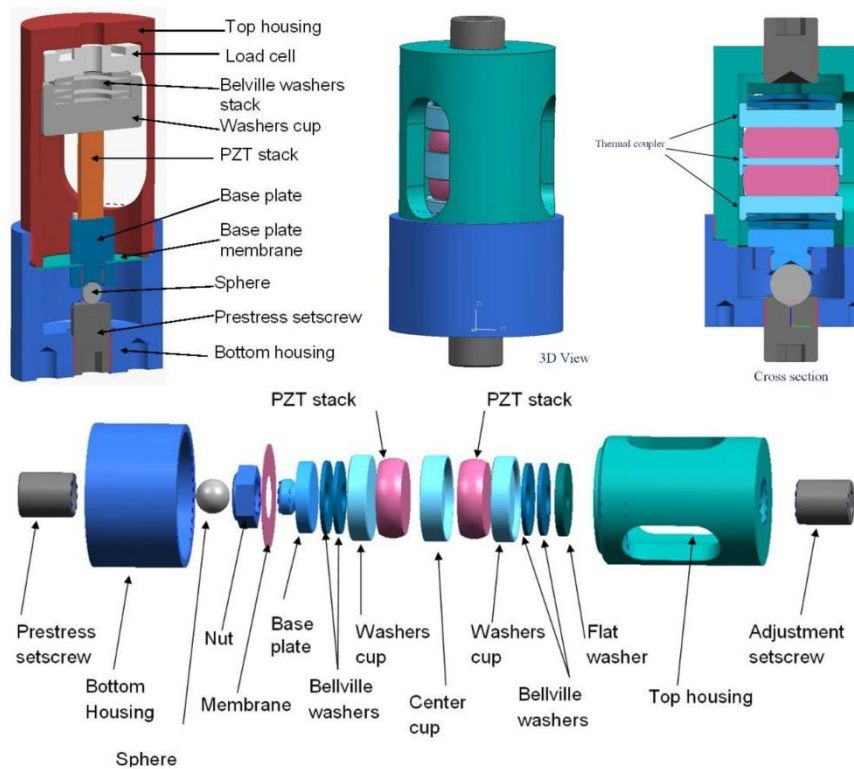


Figure 2. 3: Setup for single PZT stack (top-left) and setup for two cylindrical PZT stacks (top-right and bottom)

An advance of this idea is the use of a motor to apply pressure. Such a setup can be found in "Influence of uniaxial compressive stress on the ferroelectric behaviors of piezoelectric stack actuators" by Yang Gang et al. [43].

A different approach to pressure application is the utilization of specialised servo-hydraulic test frames. Such an approach can be found in "Response of piezoelectric stack actuators under combined electro-mechanical loading" by Mitrovic et al. [44]



(Fig. 2.4.) and “Uniaxial stress dependence of ferroelectric properties of  $x$ PMN-(1- $x$ )PZT ceramic systems” by Yimnirun et al. [45].



Figure 2. 4: Instron 8516 servo-hydraulic fatigue system

Finally, a more sophisticated approach to the aforementioned method involves the immersion of the piezo actuator into a suitable bath to prevent high-voltage arcing during electric loading. In "Effects of uniaxial prestress on the ferroelectric hysteretic response of soft PZT" by Zhou et al. [46] a fluorinert electric liquid (FC-40, 3M) bath was used (Fig. 2.5a), while in “The effect of uniaxial stress on the electro-mechanical response of 8/65/35 PLZT” by Lynch [47] a silicone oil bath was used (Fig. 2.5b).

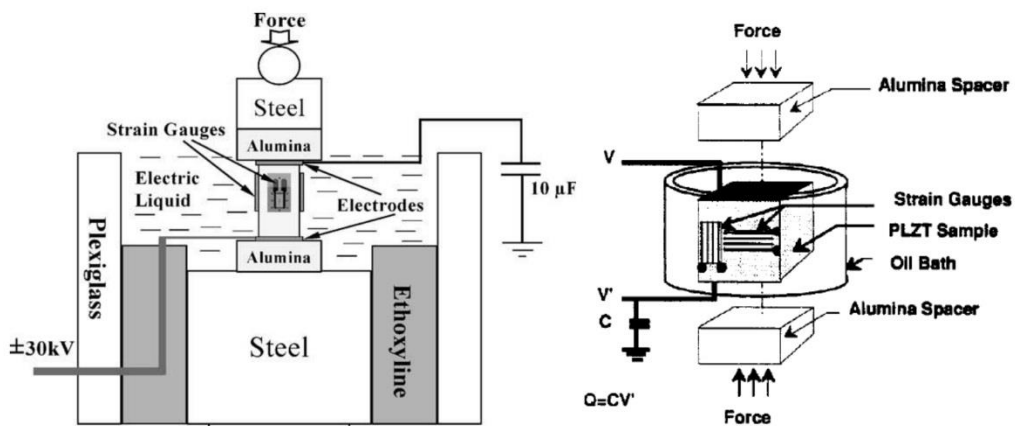


Figure 2. 5a (left) and Figure 2. 5b (right)

## 2.3 Technical Specifications

### 2.3.1 Mechanical structure

Starting with the mechanical aspect of the structure, all the metallic parts are made of an alloy of *stainless steel Grade 316* and *anodized aluminium*. That includes the **base**, the **rods**, the **blocks**, the **locks**, as well as any other metallic part of the structure. Stainless steel is generally resistant to acids but Grade 316 offers greater defence against sulphuric, phosphoric and fatty acid, aldehydes, amines and fats, as well as superior corrosion resistance. The anodized aluminium exhibits increased corrosion resistance compared to regular one. [48][49]

The **springs** are made of *iron* and have a spring constant of  $k = 78 \text{ N/mm}$ . The fact that in our setup we have two identical springs in parallel means that the equivalent spring constant of the system is  $k_{eq} = 2 \cdot k = 156 \text{ N/mm}$ .<sup>6</sup>

### 2.3.2 Motor assembly

Moving on, the motor, the gearhead, the torque limiter and the motor driver comprise the motor assembly. The servo **motor** is model *SANMOTION R R2AA04010FCP00M6* from *Sanyo Denki*. It is a 200 V AC servo motor rated at 100 W with a motor flange size of 40 mm<sup>2</sup>. It has 3000 RPM rated speed, 6000 RPM maximum rotation speed, 0.318 Nm rated and continuous stall torque and 1.18 Nm peak stall torque and it features a 24 V DC brake, a battery backup method absolute encoder, UL/CE approval and reduction ratio of 1/3.

The motor is driven by a **motor servo amplifier** made by *Sanyo Denki*, model number *RS1A01AA*. It is a single-axis analog/pulse input type servo amplifier, powered by AC200 to 230V, and is rated at 15A. It features a wire-saving incremental encoder and battery backup method absolute encoder and a NPN selectable output.

The motor and the motor servo amplifier are illustrated in Fig. 2.6.

---

<sup>6</sup> In general, a system of two springs in parallel, whose spring constants are  $k_1$  and  $k_2$  exhibits an equivalent spring constant of  $k_{eq} = k_1 + k_2$ . [50]



Figure 2. 6: Motor servo amplifier (left) and motor (right)

The shaft of the DC motor connects to a **gearhead** or **gearbox**, as it is also known. The output torque of the DC motor itself is generally low and for that reason a gearhead is utilized to increase the output torque and reduce RPM. The specific gearhead is manufactured by *Neugart* and is model *PLPE 70* (Fig. 2.7) with a ratio of  $i = 64$ <sup>7</sup> and low backlash.<sup>8</sup> In this case, that means that the maximum speed of 6000 RPM of the motor is constrained to a maximum value of  $n_{out} = 6000/64 = 93.75$  RPM, but at the same time the output torque, which is what is necessary to apply pressure to the piezo element, is greatly increased. Specifically, the rated torque with the addition of the gearhead is equal to  $\tau = \tau_{motor} \cdot i = 0.318 \cdot 64 = 20.352$  Nm.



Figure 2. 7: Neugart PLPE 70 front view (left), rear view (middle) and innards (right)

<sup>7</sup> The ratio is defined as  $i = n_{in}/n_{out}$ , where  $n$  is speed measured in RPM

<sup>8</sup> Backlash is defined as the clearance or lost motion in a mechanism caused by gaps between the parts [51]

The last part of the motor assembly, the **torque limiter**, also known as **overload clutch**, is a device that automatically protects mechanical equipment, or whatever is connected to it, from damage by mechanical overload by physically disconnecting the drive from the load. [52] The specific torque limiter is manufactured by *R+W* and is model *SK2/0030/087/W* (Fig. 2.8). It offers precise overload protection, is absolutely backlash-free and provides disengagement within 1-3 milliseconds.



Figure 2. 8: R+W SK2

### 2.3.3 Sensors and auxiliary devices

Of great importance in the piezoelectric actuator characterization process are the readings from the capacitive sensor. In order to get those readings, a capacitive sensor in conjunction with a capacitive measurement module is necessary.

The **capacitive sensor** is manufactured by *Fogale* and is model *MCC-10*. The measurement range of the sensor is from 0 to 1.0 mm with a resolution of 0.4 nm RMS/Hz<sup>1/2</sup>. It features metal-resin technology, is very strong and can withstand temperatures up to 200 °C.

The **capacitive measurement modules** are made also by *Fogale* and their model number is *MC900*. There are currently 3 identical modules housed in two separate 19" racks, each with a suitable power supply module. The MC900 has a power output range of 0-10 V, which, taking into account the 0-1.0 mm measurement range of the capacitive sensor, translates into a ratio of 100  $\mu\text{m}/\text{V}$ .



Figure 2. 9: Fogale MCC-10 attached to MC900 (left) and the modules at hand (right)

Piezo actuators generally can withstand a lot of voltage - for example, the application of 150 V to a piezo actuator is not at all uncommon and it is actually necessary in order to achieve the biggest displacement that the actuator can produce. For that purpose, it is necessary to utilize a **voltage amplifier**, which is required to be linear within its range. In this case, the amplifier utilized to drive the piezo actuators is a *Cedrat LA75B*, accompanied by its corresponding power supply module, *LC75B* (both illustrated in Fig. 2.10). Specifically, the LA75B is designed to drive capacitive loads like piezoelectric actuators with extremely low noise. It is a high-power amplifier, which can perform amplifying operations in the -20/150 V range, and is integrated in a rack along with the AC/DC converter LC75B board. The LC75B AC/DC module is designed to produce stabilised DC voltages, which are necessary to supply the amplifier. The maximum output current of the LA75B is 360 mA. The voltage gain that the amplifier provides is 20 and that means that if the input to the amplifier is 1 V, then the amplifier will drive the piezo actuator with 20 V.

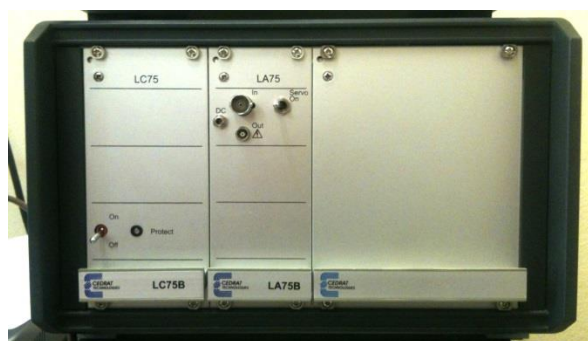


Figure 2. 10: Cedrat LA75B and LC75B

Similar to the case of capacitive sensor readings, in order to measure the load cell, both a load cell and a load cell measurement module are necessary.

There are currently two different **load cells** utilized depending on the needs. One, the Althen *ALF256DBR0KN* is rated at 5 kN, while the other, *Tedea Huntleigh Model No. 615*, is rated at 300 kg. The Althen can be used only in compression, while the Tedea in both tension and compression.

The **load cell measurement module** utilized in this case is the *Althen SG-KP-12E-420*. This unit connects to the load cell and provides an analog output ranging from 0 to 10 V. So, in the case of the Althen load cell rated at 5 kN, this means that the ratio is  $500 \text{ N/V}$ . In the case of the Tedea Huntleigh load cell rated at 300 kg, the ratio is  $30 \text{ kg/V} \approx 294.2 \text{ N/V}$ .



Figure 2. 11: Althen SG-KP-12E-420 (left), Tedea Huntleigh 615 (middle) and Althen ALF256 (right)

There are two identical **accelerometers** used to measure the environmental vibrations of the setup. They are manufactured by *PCB Piezotronics* and their model number is *356B18*. Those are triaxial, high sensitivity, ceramic shear ICP® accelerometers, with a sensitivity of 1000 mV/g or 102 mV/(m/s<sup>2</sup>) in the SI system, 0.5 to 3k Hz frequency range and 4-pin connector.



Figure 2. 12: PCB Piezotronics 356B18 accelerometer

The **current** is measured with an in-house sensing board which utilizes an ACPL-C87B integrated circuit (IC). The ACPL-C87B is chosen when high precision is required and it features advanced sigma-delta modulation technology and  $\pm 0.5\%$  High Gain Accuracy. It has 0 to 2 V nominal input range and 1 GOhm input impedance. The board connects to the negative lead of a piezo actuator and provides a direct conversion of current to voltage. That means that if the current at the board input is 1 mA, then the output of the board will be 1 mV.

Lastly, for **temperature** readings a *PT100* resistance temperature detector (RTD) is utilized. The RTD is a sensor used to measure temperature by correlating the resistance of the RTD element with temperature. The PT100 is one of the most commonly used RTDs and is made from pure platinum (hence the 'PT' prefix) and has a nominal resistance of 100 Ohms at 0 °C (hence the '100' suffix). As all RTDs, the PT100 has a predictable change in resistance as the temperature changes and it is this predictable change that is used to determine temperature. The sensitivity of a standard PT100 is a nominal 0.385 Ohm/°C. [53]

#### **2.3.4 PXI system**

The last but probably most important item of the peripheral devices is the **National Instruments PXI system**. PXI stands for PCI eXtensions for Instrumentation and is a modular instrumentation platform originally introduced in 1997 by National Instruments (NI). It serves as a basis for building electronic test equipment, automation systems and modular laboratory instruments in science among others and the fact that it is modular permits great flexibility. With PXI, one can select the modules from a number of vendors and integrate them into a single PXI system, with over 1150 module types being available in 2006. To ensure standards compliance and system interoperability, PXI uses PCI-based technology and an industry standard governed by the PXI Systems Alliance (PXISA). At some point around 2004, the PXI Systems Alliance passed specification PXI-5, which specifies how to integrate PCI Express performance into PXI while preserving backward compatibility. This led subsequently to the introduction of the PXI Express hybrid slots, which deliver support for both PCI and PCI Express. [54]

The PXI system in this case consists of a PXI-ready chassis, a main controller and the extension modules. The chassis is the backbone of the PXI system as it provides the power, cooling and communication buses of PCI and PCI Express for the main controller and the various extension modules. The main controller is a high-performance embedded controller that eliminates the need for an external PC, as in this way a complete system is contained within the PXI chassis. Embedded controllers include standard features such as an integrated CPU, hard drive, memory, Ethernet, video, serial, USB, and other peripherals. Lastly, extension modules can be added to a system to increase functionality by providing for example inputs and outputs that can be used in conjunction with several sensors. [55]

The **chassis** utilized in this project is the *NI PXIe-1078*. This is a 9-slot, 3U PXI Express chassis with the first slot dedicated to the main controller and the rest available for use with extension modules. Specifically, it offers 5 hybrid slots (slots 5-9) and 3 PXI Express slots (slots 2-4). The chassis includes built-in timing and synchronization features including 10 and 100 MHz reference clocks, as well as the PXI trigger bus. In Fig. 2.13, it can be seen both populated and unpopulated.



Figure 2. 13: Images of a populated and an unpopulated PXIe-1078 chassis

The **main controller** fitted in the first slot of the chassis is the *NI PXIe-8133*. It is equipped with a high-performance quad-core Intel Core i7-820QM that operates in 1.73 GHz base frequency and 3.06 GHz in single-core Turbo Boost mode, 2GB of DDR3 1333MHz ram memory and a 200GB hard disk. It also features two 10/100/1000BASE-TX (Gigabit) Ethernet, 4 Hi-Speed USB, ExpressCard/34, GPIB, serial, DVI and external trigger connections. It is equipped with a dual-boot operating



system of either Microsoft Windows XP or NI LabVIEW Real-Time. The difference is that the NI LabVIEW Real-Time is a real-time operating system, which means that it is designed to run applications with very precise timing and a high degree of reliability. This is especially important in a measurement system like in this case and its necessity will be thoroughly explained in the next chapter. Operating systems like Windows are designed to maintain user responsiveness with many programs and services running (ensuring "fairness"), while real-time operating systems are designed to run critical applications reliably and with precise timing (paying attention to the programmer's priorities). The NI Real-Time Hypervisor uses virtualization technology to run NI LabVIEW Real-Time in parallel with Windows XP, thus making use of real-time processing and at the same time Windows XP services and graphical interface. [56][57] The PXIe-8133 is illustrated in Fig. 2.14.



Figure 2. 14: NI PXIe-8133

Lastly, the **extension modules** used can be summarized in the following table:

Slot Number	Module	Utilization
2	PXIe-4357 + TB-4357	PT100 temperature sensor
3	PXIe-4496	Accelerometer sensors
5	PXI-4461	Load Cell and current sensors
7	PXIe-4330 + TB-4330	Strain gage
8	PXI-7841R + SCB-68	FPGA module for motor control
9	PXI-4461	Capacitive sensor, piezo excitation signal (generation and feedback acquisition)

- The *PXIe-4357* input module provides integrated data acquisition and signal conditioning for resistance temperature detector (RTD) measurements. It offers 20 channels and features five 24-bit delta-sigma analog-to-digital converters with anti-aliasing and low-pass filters for superior accuracy, along with sampling rates up to 100 S/s per channel in high-speed mode or 1 S/s with typical measurement accuracies of 0.09 °C in high-resolution mode. It supports 2-, 3- or 4-wire RTDs. The additional *TB-4357* is a front mount terminal block with screw terminal connectivity for the NI *PXIe-4357*. This assembly is illustrated in Fig. 2.15 and it is used to get readings from a PT100 temperature sensor.



Figure 2. 15: PXIe-4357 (top-left), TB-4357 (top-right), TB-4357 connected to PXIe-4357 (bottom-left) and TB-4357 innards (bottom-right)

- The *PXIe-4496* is a high-accuracy data acquisition (DAQ) module specifically designed for high-channel-count sound and vibration applications. It offers 16 simultaneously sampled analog inputs at up to 204.8 kS/s, with 24-bit resolution ADCs with 114 dB dynamic range. It uses a method A/D conversion known as delta-sigma modulation. With this technology, input signals are oversampled or sampled at many times the chosen data rate and then applied to a digital filter, resulting in excellent phase linearity while maintaining low noise and distortion. It is used to acquire reading from the two accelerometers and is illustrated in Fig. 2.16.



Figure 2. 16: PXIe-4496

- The *PXI-4461* is a high-accuracy data acquisition module. It features two simultaneously updated analog outputs at up to 204.8 kS/s, software-configurable AC/DC coupling and IEPE conditioning, 24-bit resolution ADCs and DACs with 118 dB dynamic range, variable anti-aliasing and anti-imaging filters, six gain settings for input ranges from  $\pm 316$  mV to  $\pm 42.4$  V and two simultaneously sampled analog inputs at up to 204.8 kS/s. Similar to the PXIe-4461 it adopts delta-sigma modulation in both its inputs and outputs. There are two of these modules in the specific setup, with the first one being used for the load cell and current readings and the other for the capacitive sensor readings and the generation and feedback acquisition of the piezo excitation signal. It is illustrated in Fig. 2.17.



Figure 2. 17: PXI-4461

- The *PXIe-4330* features eight channels with 24-bit analog-to-digital converters (ADCs) per channel and 25 kS/s sampling rate. To remove noise, each module offers anti-aliasing and digital filters per channel. Each channel also features an

independent programmable 0.625 to 10 V excitation voltage. The PXIe-4330 is used in conjunction with the NI *TB-4330* front-mounting terminal block, which facilitates connectivity. In our setup the PXIe-4330/TB-4330 combination is used to acquire strain gage readings. The illustration is similar to Fig. 2.15.

- Lastly, the *PXI-7841R* is a multifunction reconfigurable I/O (RIO) module featuring a user-programmable FPGA chip for on-board processing and flexible I/O operation. All analog and digital functionality can be configured using NI LabVIEW and the LabVIEW FPGA Module. It features 8 analog inputs with independent sampling rates up to 200 kHz, 16-bit resolution and  $\pm 10$  V range, 8 analog outputs with independent update rates up to 1 MHz, 16-bit resolution and  $\pm 10$  V output range and 96 digital lines configurable as inputs, outputs, counters, or custom logic at rates up to 40 MHz. The PXI-7841R is used in conjunction with NI *SCB-68* connector block, which provides screw terminals for easy I/O connections. The PXI-7841R/SCB-68 combo is used for the control of the motor and is illustrated in Fig. 2.18. In particular, it provides three separate controls: it operates the servo and the mechanical brake mechanisms of the motor via enabling and disabling two digital outputs and it also controls the rotation speed of the motor by varying an analog output which is fed to the motor driver.



Figure 2. 18: PXI-7841R (left) and SCB-68 connector block (right)

Regarding the servo function, a digital output from the 7841-R/SCB-68 is fed directly to the corresponding input of the RS1A01AA servo driver. The motor does not work without the servo function being enabled, so having this feature separate from the motor speed adjustment is useful to avoid unintentional motor function.

Regarding the mechanical brake, like the servo function, it is controlled by a digital output of the 7841-R/SCB-68. In this case, however, the digital output voltage and current were not sufficient to operate the brake directly. According to the motor manufacturer's specifications, the mechanical brake requires a 24 V DC input voltage with 270 mA current. In order to overcome this obstacle, the utilization of a relay board was deemed necessary. The relay board serves as an intermediary between the 7841-R/SCB-68 and the motor servo amplifier. The relay is controlled by the digital output of the 7841-R/SCB-68 and when energised it feeds 24 V DC to the brake inputs of the motor amplifier. That voltage comes from an external 24 V DC rated at 2300 mA power supply. The relay board along with the external 24 V power supply can be seen in Fig. 2.19.



**Figure 2. 19: Relay board and external power supply used to control the motor brake**

The motor speed is controlled by an analog output of the 7841-R/SCB-68. The analog output has a range of -10 V to +10 V with a 16-bit resolution. A 16-bit integer can store  $2^{16}$  (or 65,536) distinct values. In an unsigned representation, these values are the integers between 0 and 65,535 and using two's complement, the possible values range from -32,768 to 32,767. That means that when the desired output is -10 V, the value passed to the analog output should be -32,768 and when the desired output is +10 V, the value should be 32,767, with all in-between values varying linearly. This necessity will be further explained in chapter 3.

## 2.4 Functional analysis

The setup is able to perform both creep and hysteresis measurements with a variety of options.

In order to perform the **creep** measurement, the piezoelectric element needs to be free of load – that means that no mechanical load is applied on top of the piezo actuator. Although the creep measurement could be carried out with the primary structure, an auxiliary structure (Fig. 2.20) is preferred as it provides easier and faster mounting and dismounting of the piezo actuators.

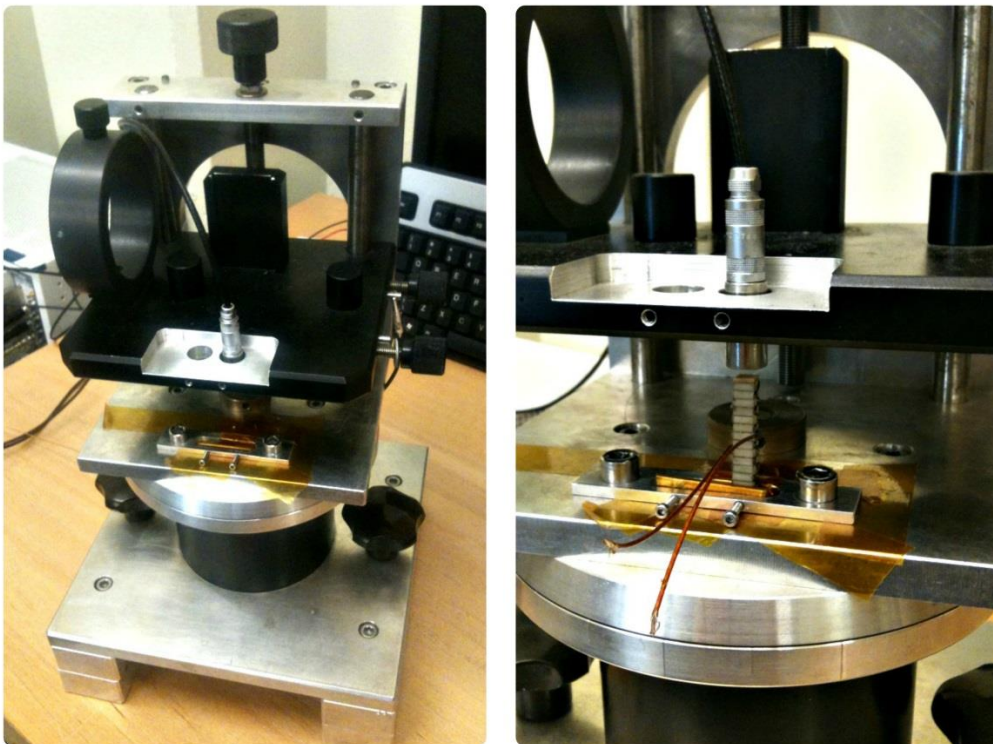


Figure 2. 20: Auxiliary structure for creep measurements (left) and close-up of piezo actuator and capacitive sensor in place (right)

The aforementioned auxiliary structure works by fastening the base of the piezo actuator with its top being left free. The capacitive sensor is then placed above the top of the piezo actuator and when voltage is applied to it, the distance between the top of the piezo and the capacitive sensor varies and that change is captured by the capacitive sensor. In order to perform a creep test, the piezo excitation follows a similar pattern as in Fig. 2.21.

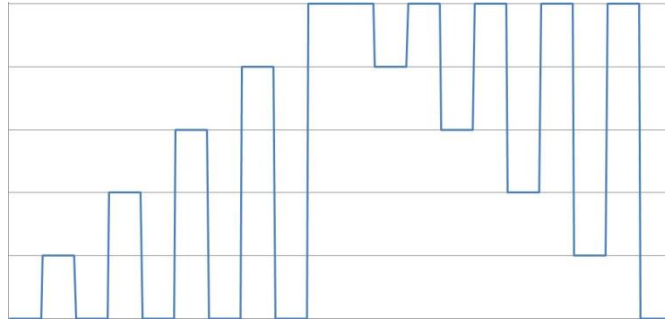


Figure 2. 21: Piezo excitation pattern for creep measurement

The voltage alternates between the minimum value and an increasing value at each step until reaching the maximum desired voltage. The process is then repeated in the reverse order – the voltage alternates between the maximum value and a decreasing value at each step until reaching the minimum voltage. The number of steps, the minimum and maximum voltage and the duration of each step are all parameters that can be set before the execution of the creep measurement. In general, the duration of each step is relatively big, so that the piezo displacement has stabilized before advancing to the next step.

Regarding the **hysteresis** measurement, both a static and a dynamic execution mode are offered. For the hysteresis test, only the primary structure can be used (Fig. 2.22), where pressure is applied via the motor.

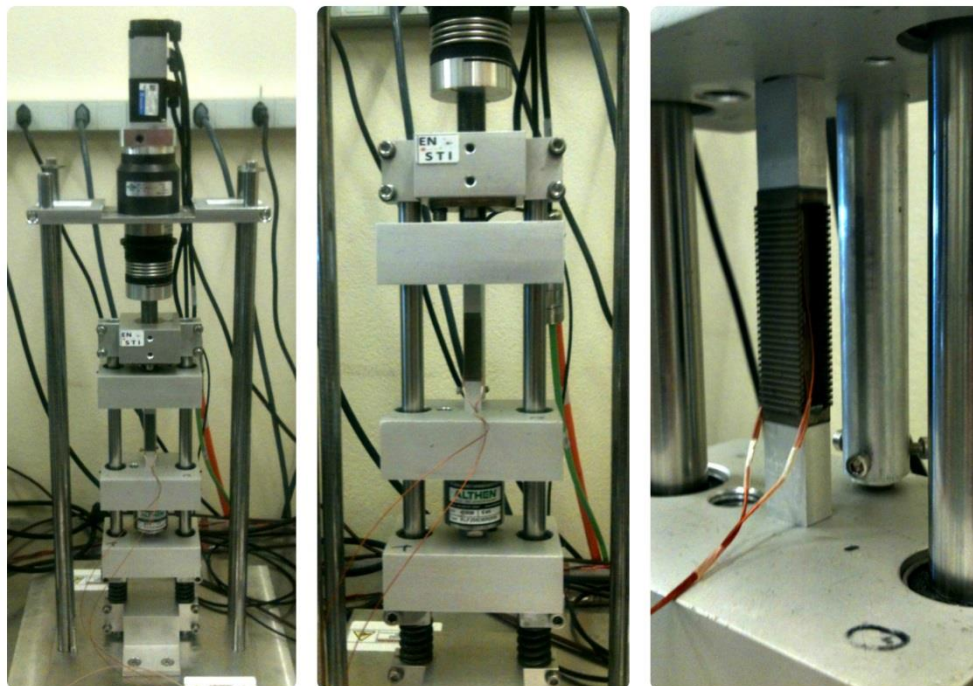


Figure 2. 22: Primary structure (left and center) and close-up of piezo actuator and capacitive sensor (right)

In the *static* execution mode, the mechanical load is pre-set before exciting the piezo and remains constant throughout the duration of the test. The piezo excitation voltage can be set to a constant value or a periodic signal (sine, triangular, square or sawtooth wave) with user-configured parameters.

In the *dynamic* execution mode, the mechanical load varies throughout the duration of the test, with the motion resembling a triangular wave between a minimum and a maximum load value. Again, the piezo excitation voltage can be set to a constant value or a periodic signal (sine, triangular, square or sawtooth wave) with user-configured parameters. A special function in the dynamic execution mode provides the ability to vary the mechanical load and the excitation voltage at the same time and with the same frequency.

Finally, the software provides the ability to display the data plots, automatically post-process the data and automatically generate reports. All the functions will be discussed in detail in the next chapter.



# 3

## *LabVIEW Analysis*

The purpose of this chapter is to provide a detailed analysis of the project software developed in LabVIEW. Starting with an introduction to the basic building concepts used in the specific implementation, we then move to an overview of the program structure, focusing progressively on the main programming challenges faced.

### *3.1 Basic building concepts*

There are several building approaches when it comes to LabVIEW programming. In this case, the State Machine architecture and the Functional Global Variables (FGVs) constitute key elements of the program and for that reason they are analysed in advance.

#### *3.1.1 State Machine*

The state machine design pattern is very common in LabVIEW and is used to implement any algorithm that can be explicitly described by a state diagram or a flowchart.

A state machine consists of a set of states and a transition function that maps to the next state. Each state can lead to one or multiple states or end the process flow. The next state is determined by user input or in-state calculation.

A state machine consists of the following elements:

- While loop – Continually executes the various states
- Case structure – Contains a case for each state and the code to execute for each state
- Shift register – Contains state transition information
- State functionality code – Implements the function of the state
- Transition code – Determines the next state in the sequence

An example of state machine architecture is illustrated in Fig. 3.1.

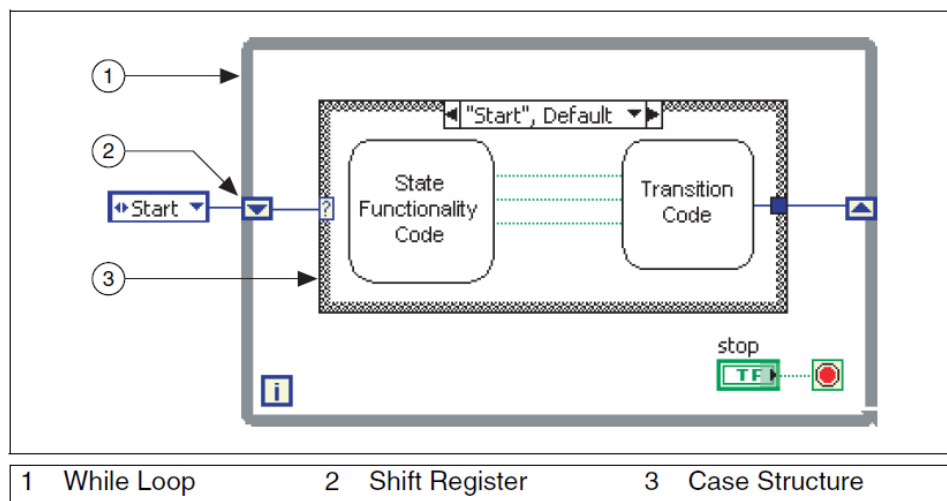


Figure 3. 1: State machine architecture

Compared to a state diagram, the flow in a state machine is implemented by the While Loop, the individual states are represented by cases in the Case structure, while the shift register on the While Loop keeps track of the current state and communicates the current state to the Case structure input.

As illustrated in Fig. 3.1, the most popular method for controlling the state transition is the enumerated type control (enum). One important detail to remember though is that the enumerated control needs to be type-defined (typedef). This solves the problems that arise when the user attempts to add or delete a state from the enumerated type control, leading to a break of the remaining wires that are connected

to the copies of this enumerated type control. In other words, by type-defining the enumerated type control, all the enumerated type control copies are automatically updated if a state is added or removed.

In the specific project, the next state of a state machine is controlled mainly in two ways – by what is called a “single default transition” and a “select function” transition code. The first one is used when there is only one possible state that occurs next (Fig. 3.2a), while the second when a decision on a transition between two possible states needs to be made (Fig. 3.2b). [58]

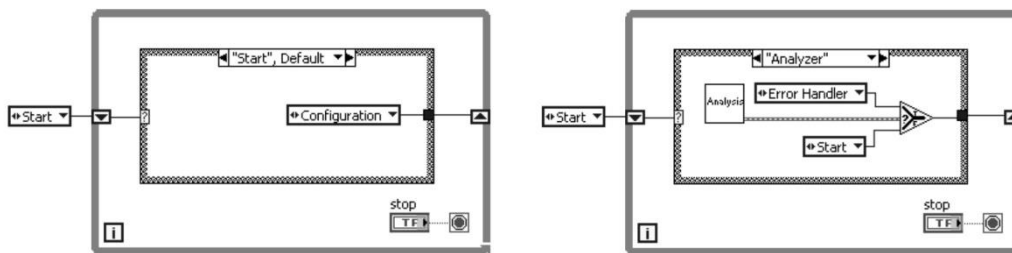


Figure 3. 2a (left): Single default transition and Figure 3. 2b (right): Two possible transitions where the outcome is defined by the select function

### 3.1.2 Functional Global Variable (FGV)

Functional global variables are used extensively in this project as they are more suited than global variables for a variety of reasons that will be explained. A functional global variable makes use of uninitialized shift registers in a While loop to hold data as long as the VI never goes out of memory. The general form of a Functional Global Variable includes an uninitialized shift register with a single iteration While loop, as illustrated in Fig. 3.3.

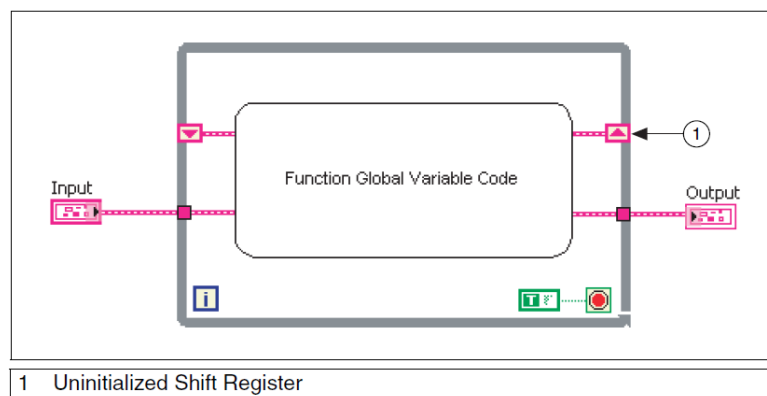


Figure 3. 3: Functional Global Variable general form

A Functional Global Variable usually has an action input parameter that specifies which task the VI performs. The VI uses the uninitialized shift register to hold the result of the operation. The most common FGV includes “set” and “get” functionality (Fig. 3.4). As shown, in the “Set” state, data passes into the VI and the shift register stores that data, while in the “Get” state, the data is retrieved from the shift register and is forwarded to the output.

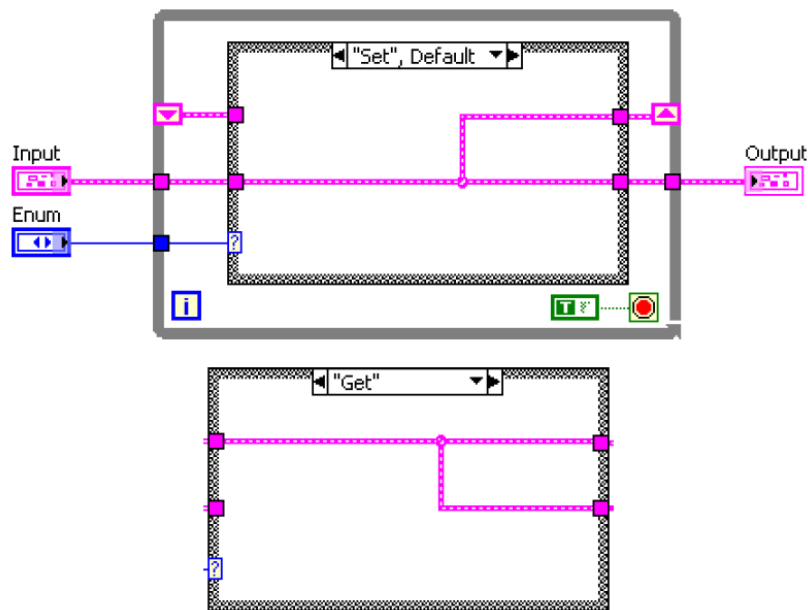


Figure 3. 4: FGV with “Set” and “Get” states

There are three main advantages in using Functional Global Variables instead of Global Variables. The first one is that they are fully customisable in the sense that they can have as many states as desired (the functionality of the FGV is in no way limited to the popular two-state “Get”/”Set” FGV). The second advantage is that they can implement more complex data structures, such as a stack, a queue buffer or a complex bundle type, that normal global variables cannot. The last and most important advantage is that they are not re-entrant. This means that when a FGV is called from multiple locations, the same copy of the FGV is used. Therefore, only one call to the FGV can occur at a time. This prevents race conditions which can occur when two different processes try to access the same resource (for example one trying to read it and the other trying to write to it). [59]

## ***3.2 Main Analysis***

### ***3.2.1 Overview***

The project is split into two parts – the server application and the client application. The server application runs on the NI LabVIEW Real-Time side of the PXI (please refer to section 2.3.4 of chapter 2 for details), while the client application runs on the Windows XP side of the PXI (or any other personal computer that can establish a network connection with the PXI system). This is done in order to improve the accuracy, reliability and consistency of the acquired data.

Specifically, the server application is responsible for operating the motor and acquiring the data of all different sensors operating at distinct sampling rates with precise timing and in a synchronized manner and to send the acquired data to the client application in packages, ensuring that no data loss occurs on the way. The client application on the other hand is responsible for receiving all of the transmitted data and performing all cpu- and memory-intensive tasks that don't require sharp timings, like drawing the data plots, running the post-processing Matlab scripts and generating the reports.

The flowchart of both the server and client applications can be seen in Fig. 3.5 and an analysis of both applications is provided in the following subchapters.

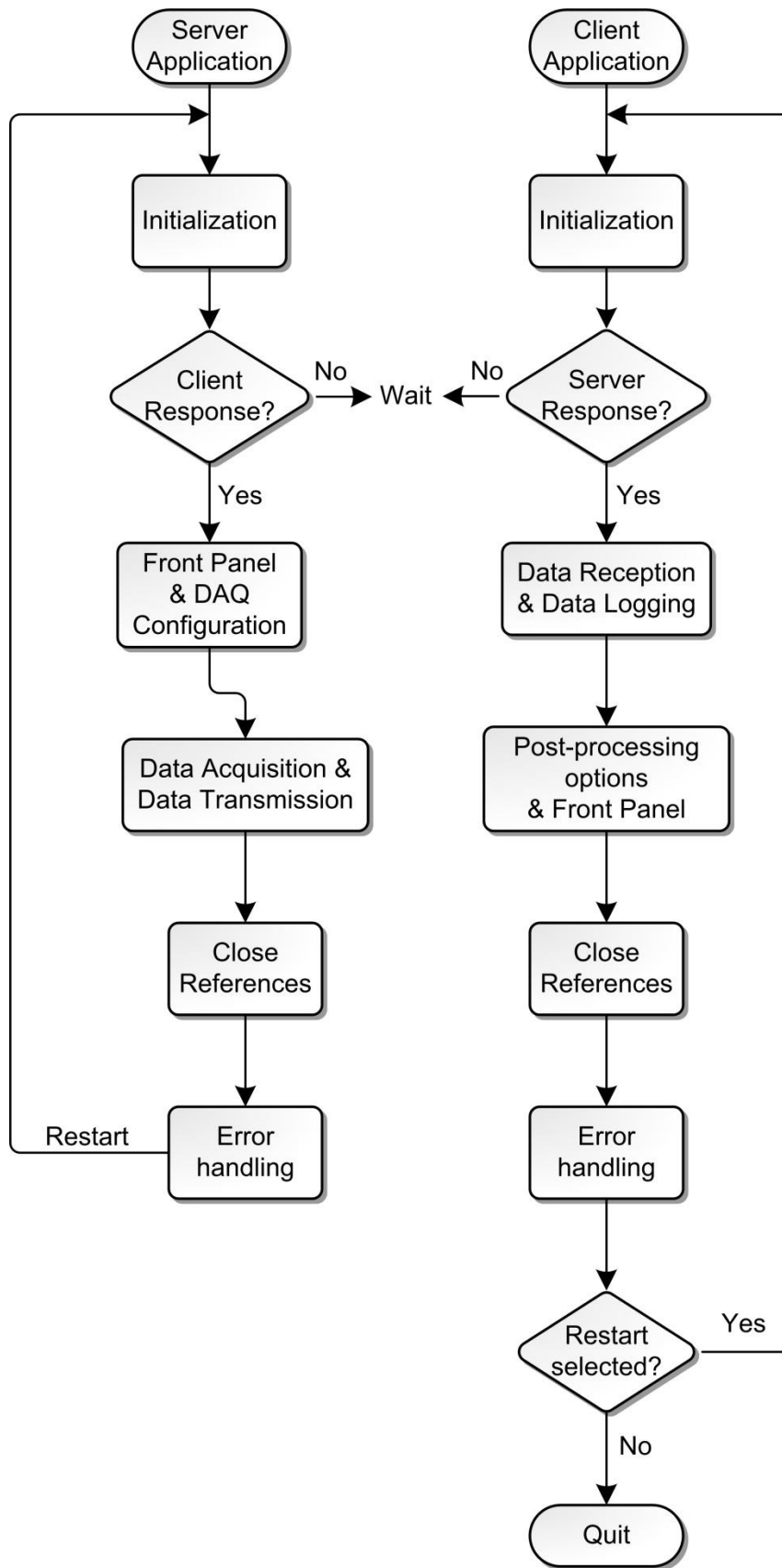


Figure 3. 5: Flowchart of server and client applications

### 3.2.2 *Server application*

The server application consists of a state machine with 5 distinct states. As shown in the flowchart of Fig. 3.5, the 5 states conduct the initialization, the DAQ configuration and Front Panel selection, the data acquisition and transmission, the reference closing and the error handling and are named “init”, “daqConfig”, “runProcesses”, “exit” and “error” respectively.

- ***Initialization***

This is the stage where all the necessary initialization takes place. It includes the subVIs named “initFgvs”, “initSharedVariables”, “initFpga”, “initQueues” and “initNetworkStreams”.

- Subvi “*initFgvs*” is responsible for initializing all FGVs that run in the real-time system to their initial values.

- Subvi “*initSharedVariables*” initializes the shared variables which are used to facilitate server and client communication. Those include the “stopClient” variable, which is used to stop the data reception and logging in the client application, once the data acquisition has stopped in the server application, as well as the “strainGageIniFile” shared variable, which is used to send to the server application the user-appointed ini file in the client computer that contains the updated strain gage parameters, in case the user desires to use settings other than default.

- Subvi “*initFpga*” recognizes the connected NI RIO devices and downloads to them the appointed fpga bitfile and creates and saves to the respective FGV the FPGA reference that will be necessary in next stages. For the time being, the only NI RIO device used is the PXI-7841R but this implementation makes the utilization of different modules in the future very straightforward.

- Subvi “*initQueues*” creates the queue structures that will be used in the data acquisition stage and saves their references to the respective FGVs.

- Subvi “*initNetworkStreams*” creates all network stream writer endpoints and saves their references to the respective FGVs. The subvi finishes once the reader endpoints have been created in the client application.

The utilization of the queues and network streams is covered in detail in the “Data Management” subchapter.

As in every state of the state machines found in both the server and client application, if an error occurs in any of the subvis of that state, the program jumps to the “error” state instead of the next state. This stops the programs from running if an unexpected situation occurs.

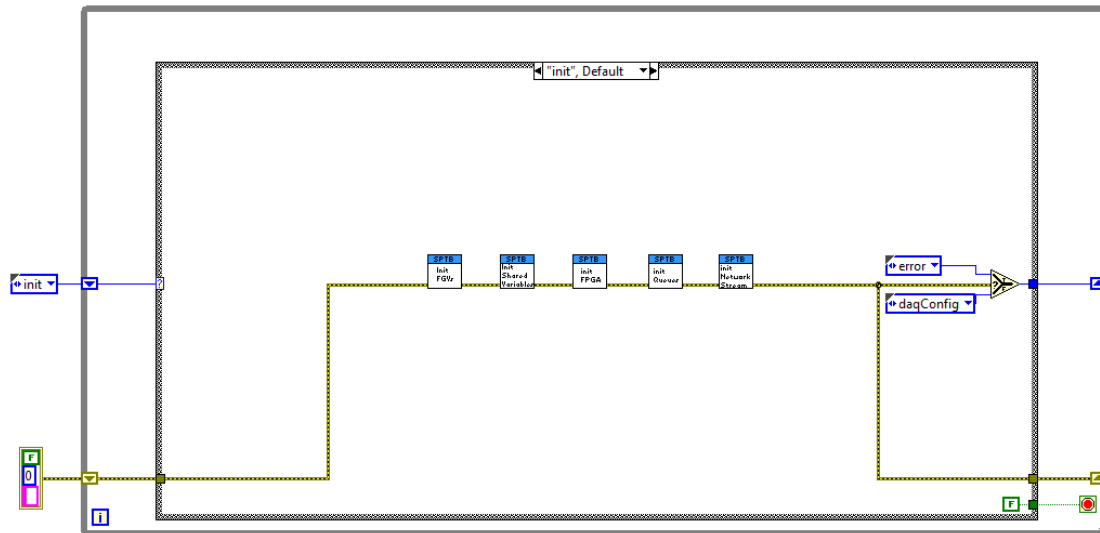


Figure 3. 6: Server application – Initialization state

- **DAQ configuration and Front Panel**

In this stage the DAQ configuration and the Front Panel selection take place. In particular:

- Subvi “*measurementTypeSelect*” allows users to select either a Creep or a Hysteresis test via a Front Panel. Depending on the selection, another subvi is called which allows users to set the desired parameters for the Creep or Hysteresis tests.
- In case a dynamic-execution Hysteresis test is selected, the motor is operated so that the load is set to the minimum setting.
- The PT100 and accelerometer channels are created along with their respective parameters and their references are saved in the respective FGVs (“*PT100DAQInit*” and “*accelerometerDAQInit*” subvis).
- Lastly, the “*initSyncStartStatic*” (in case of a Creep or a static-execution Hysteresis test) and “*initSyncStartDynamic*” (in case of a dynamic-execution Hysteresis test) subvis create the rest of the acquisition channels, passing the desired parameters and ensuring the synchronized function of all modules.



The first two bullet points will be covered in detail in subchapter “Execution Options”, while the latter two in subchapter “DAQ Synchronization”.

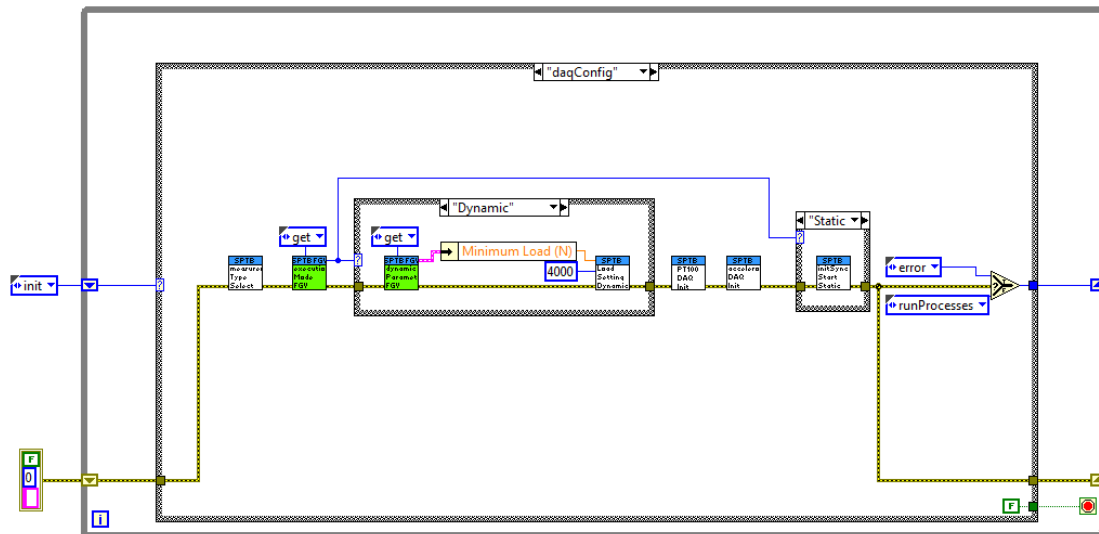


Figure 3. 7: Server application – DAQ Configuration and Front Panel state

- **Data acquisition and data transmission**

This is the stage where data acquisition occurs and the data is sent over the established network stream to the client application for logging and further processing.

A case structure is used to differentiate between static execution (Creep and static-execution Hysteresis tests) and dynamic execution (dynamic-execution Hysteresis test), but both cases utilize the “*SyncedAcquisitionProcess*”, “*PT100AcquisitionProcess*” and “*accelerometerAcquisitionProcess*” subvis, which conduct the data acquisition, as well as the “*dataSend*” subvi, which transmits the acquired data to the client (more details are provided in subchapter “Data Management”).

Both cases also include the transmission of a stop signal to the client application via the “stopClient” shared variable when the test is complete. In case of static execution, the stop signal is transmitted when the signal generator completes its task, while in case of dynamic execution, it is transmitted once the motor movement controlled by “*loadSettingDynamicSync*” subvi completes its task.

The piezo excitation signal generation is controlled by the respective subvis (“*signalGenerationHysteresisDynamic*” in case of dynamic-execution Hysteresis test,

“*signalGenerationHysteresisStatic*” in case of static-execution Hysteresis test and “*signalGenerationCreepStatic*” in case of Creep test), while the “*removeJunkSamples*” subvi ensures that the piezo excitation signal generation is synchronized with the data acquisition from the sensors (more details can be found in subchapter “DAQ Synchronization”).

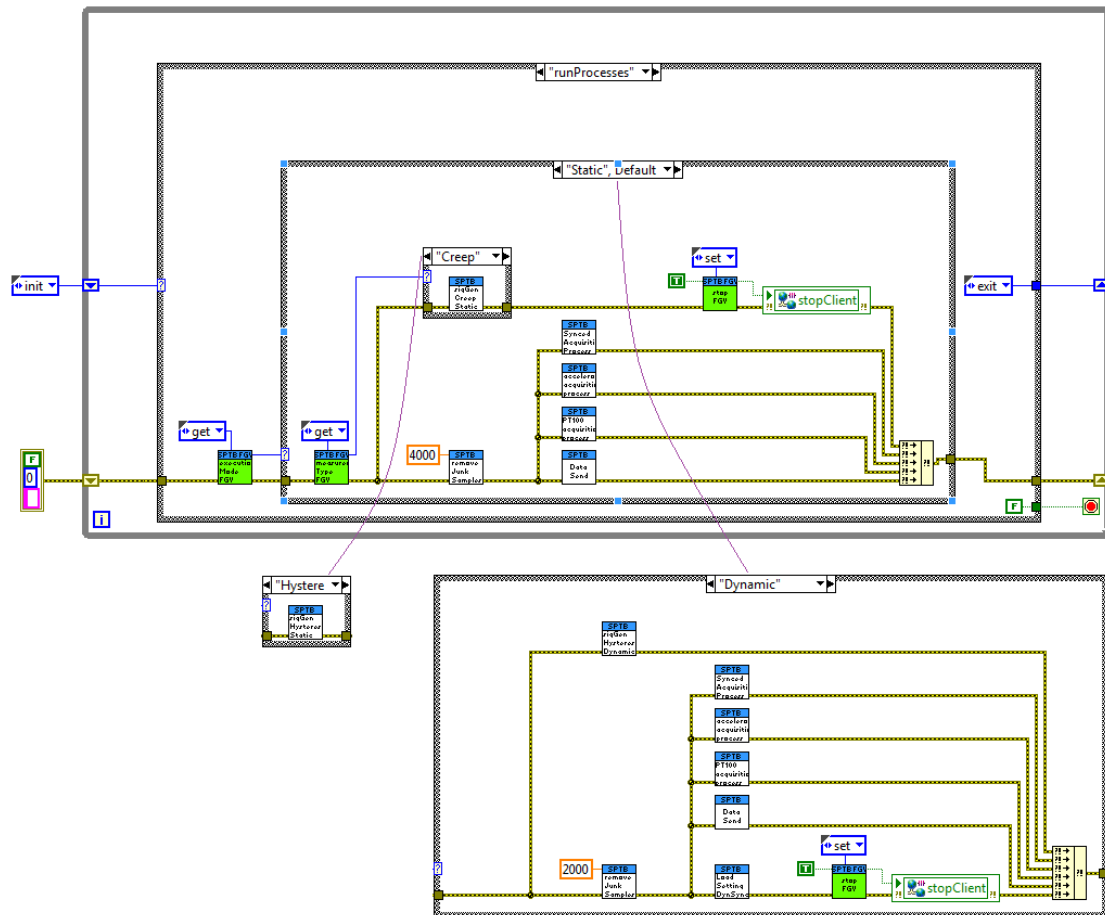


Figure 3. 8: Server application – Data acquisition and data transmission state with case variants

- **Close references**

In this stage all references are closed. In particular:

- Subvi “*closeFPGARef*” is responsible for closing the FPGA reference.
- Subvi “*closeDAQRef*” clears all the data acquisition and signal generation tasks.
- Subvi “*closeStreamRef*” destroys the network stream writer endpoints. It is worth pointing out that before destroying the endpoints, the “*Flush Stream*” subvi is called, which transfers all data still available for reading to the reader endpoint

(client). This ensures that the writer endpoint buffer is empty before it is destroyed and no data is left unread.

- Subvi “*closeQueueRef*” releases the references to the various queues.

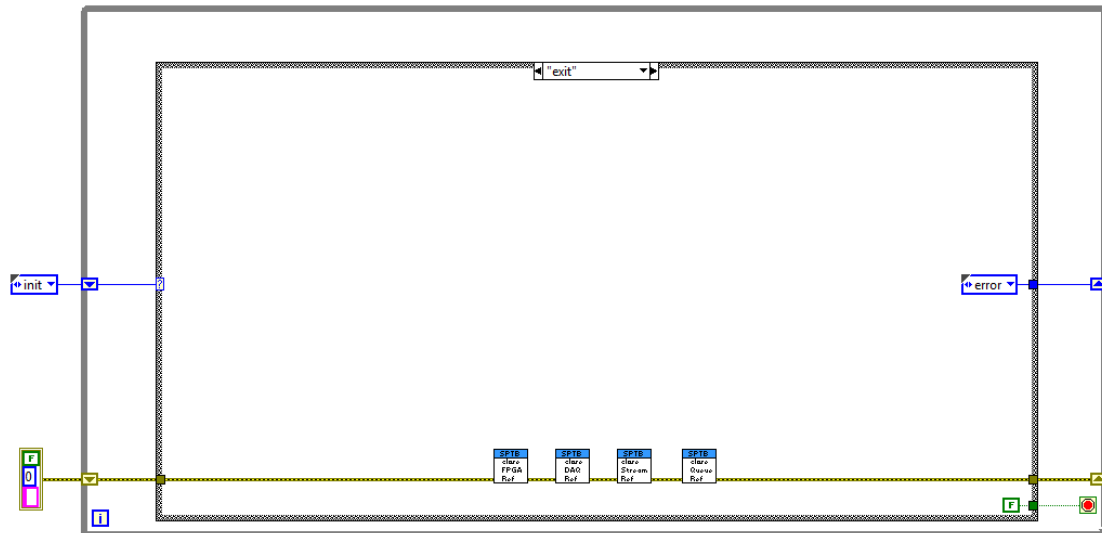


Figure 3. 9: Server application – Close references state

- **Error**

This stage runs the “*Simple Error Handler*”, which in case of error presents the error code and description. As noted, if an error occurs in a subvi of a previous stage, the state automatically switches to the “error” state and the execution is stopped.

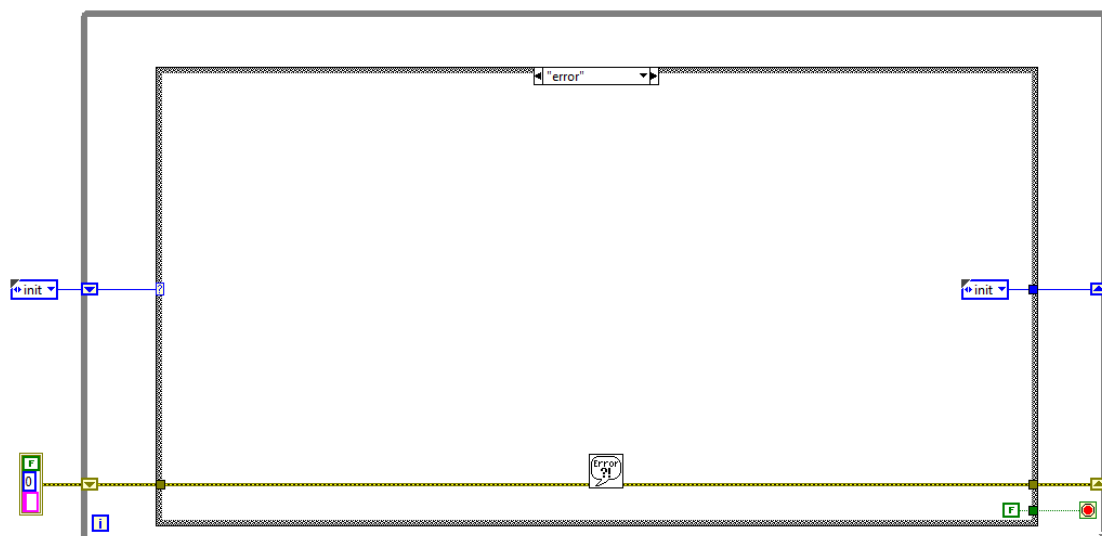


Figure 3. 10: Server application – Error state

### 3.2.3 Client application

The client application, like the server application, consists of a state machine with 5 distinct states. According to the flowchart of Fig. 3.5, these 5 states conduct the initialization, the data reception and logging, the post-processing options and Front Panel selection, the reference closing and the error handling and are named “init”, “dataLogging”, “postOptions”, “exit” and “error” respectively.

- **Initialization**

As in the server application, in this stage all the necessary initialization takes place. In particular:

- Subvi “*initFGVsClient*” initializes all FGVs that run in the client to their initial values.

- Subvi “*initStrainGageConfigClient*” prompts the user to select the utilization of either the default or user-configured DAQ settings for the strain gage. In case the user selects to use custom settings, another subvi is called which reads the user-uploaded ini<sup>9</sup> file and then sends the parameters to the server via the “*strainGageIniFile*” shared variable. At this point it is worth noting that if the same subvi was called from the server application, then the file directory which contained the user-uploaded ini file would refer to the real-time operating system. Since the real-time operating system does not have a GUI however, the only possible way to transfer the user ini file would be through an FTP service, like FileZilla. With that in mind, it becomes obvious why the implementation where the user ini file is uploaded from the client side (Windows XP side of PXI or any other personal computer that can

---

<sup>9</sup> The INI file format is an informal standard for configuration files with a basic structure composed of “section” and “properties”. The key (or property) is the basic element of an INI file and every key has a *name* and a *value*, delimited by an equals sign (=). The name appears to the left of the equals sign (for example: name=value). Keys can be grouped into sections. The section name appears on a line by itself, in square brackets (for example: [section]). All keys after the section declaration are associated with that section. Section and key names are not case sensitive. [60]

establish a network connection with the PXI system) with its values sent to the server via a shared variable was chosen.

- Subvi “*initNetworkStreamsClient*” creates all network stream reader endpoints and saves their references to the respective FGVs. The subvi finishes once the writer endpoints of the server application are recognized.

- Subvi “*initQueuesClient*” creates the queue structures that will be used in the data reception stage and saves their references to the respective FGVs.

- Subvi “*initDataLoggingClient*” creates (if not present) or replaces (if present) the tsv<sup>10</sup> files that will be used to save the received data on a pre-assigned directory of the hard disk of the client.

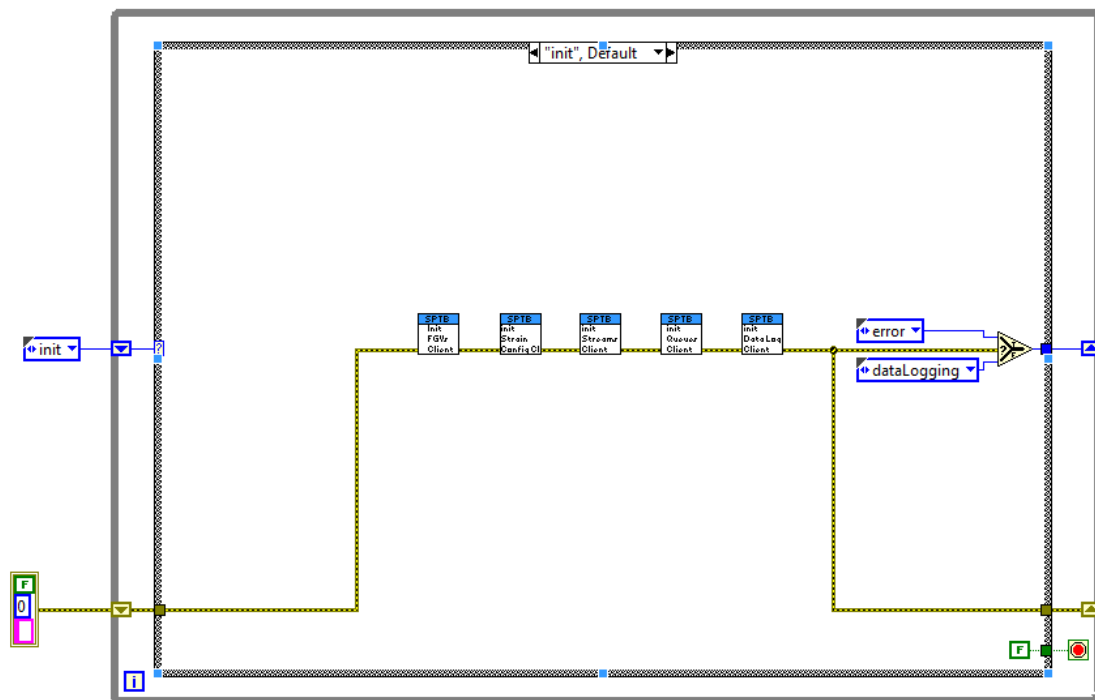


Figure 3. 11: Client application – Initialization state

- ***Data reception and data logging***

In this stage the data reception and data saving to the hard disk (logging) take place. In detail:

<sup>10</sup> TSV stands for tab-separated values. TSV is an alternative to the common comma-separated values (CSV) format and the difference is that the delimiter is a tab stop character instead of a comma. [61]

- Subvi “*dataReceive*” reads data packages from the network stream and places them in queues.
- Subvi “*dataLoggingProcessClient*” takes elements from the aforementioned queues and appends them to the respective tsv files.
- The stage concludes once the stop signal from the server is received via the “*stopClient*” shared variable.

More details about the first two bullet points are provided in subchapter “Data Management”.

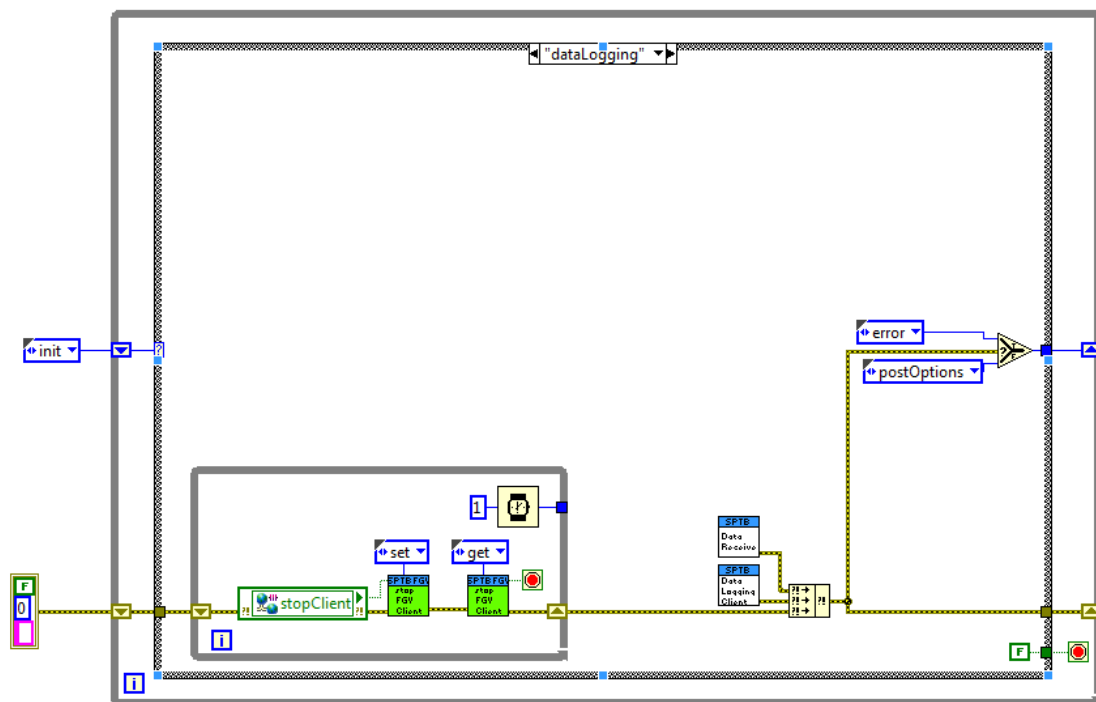


Figure 3. 12: Client application – Data reception and data logging state

- **Post-processing options and Front Panel**

After the data reception and logging is complete, the user is presented with a Front Panel with the following options:

- Copy data files: The “*copyDataFilesClient*” subvi copies the tsv files to a user-appointed location on the hard disk drive of the client. This is useful since if the files are not copied, in the next run of the program, the tsv files in the pre-assigned location will be replaced with the new data.

- Display plots: The “*graphCreationClient*” subvi reads the saved data from the tsv files and draws the plots. There are separate tabs for the plots of the strain gage, the load cell, the capacitive sensor, the current, the signal generator, the PT100 and the two accelerometers. There are also two separate tabs for the FFT<sup>11</sup> (Fast Fourier Transform) of the two accelerometers’ data. There is finally a separate tab where the strain gage, load cell, capacitive sensor, current and signal generator can be plotted in the same graph, after appropriate subvis have down-sampled the various data to the same amount of data points.

- Print Report: The “*reportOptionsClient*” subvi presents the user with a Front Panel where he can select to generate a report upon filling in the required details. The user can select to generate a Hysteresis (which runs the Hysteresis post-processing Matlab script and generates the Hysteresis Loop plot), Creep (which runs the Creep post-processing Matlab script and generates the Creep plot) or Leakage Current report or any combination of them. More information will be provided in the “Report Generation” subchapter.

- Restart Program
- Exit Program

---

<sup>11</sup> A Fast Fourier Transform is an algorithm to rapidly compute the discrete Fourier transform (DFT), which converts time to frequency. [62]

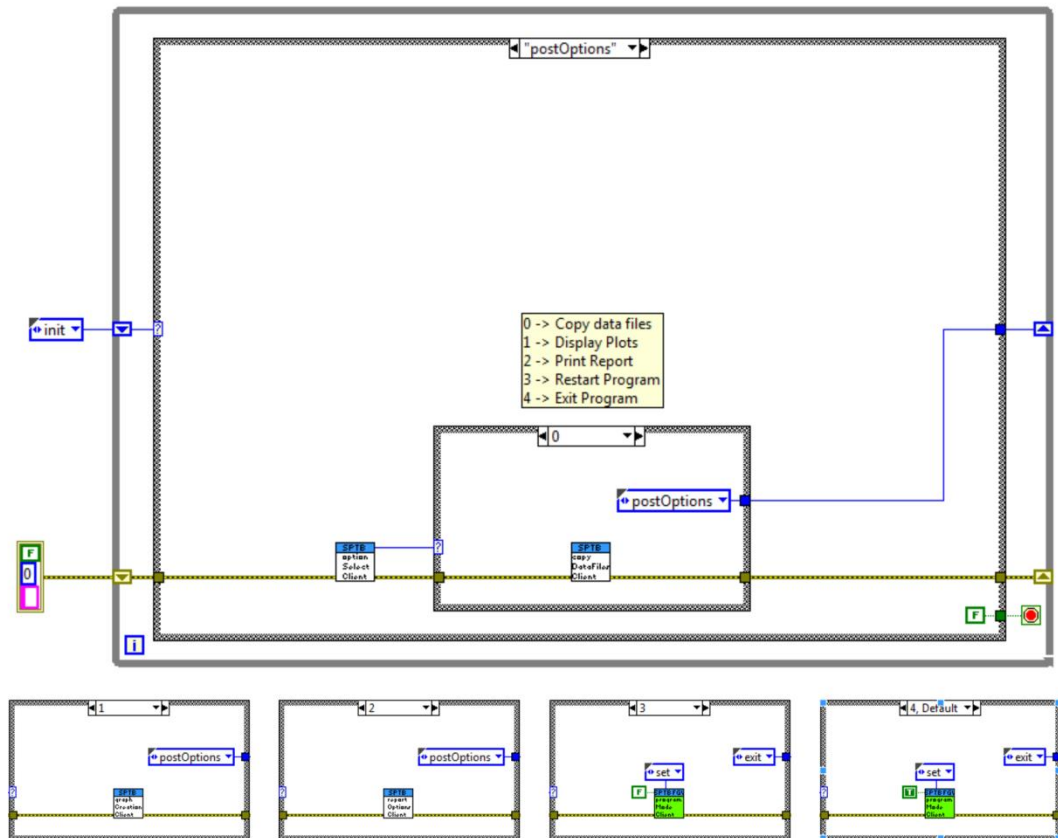


Figure 3. 13: Client application: Post-processing options and Front Panel state with case variants

- *Close references*

In this stage all references are closed. In particular:

- Subvi “closeStreamRefClient” destroys the network stream reader endpoints.
- Subvi “closeQueueRefClient” releases the references to the various queues.



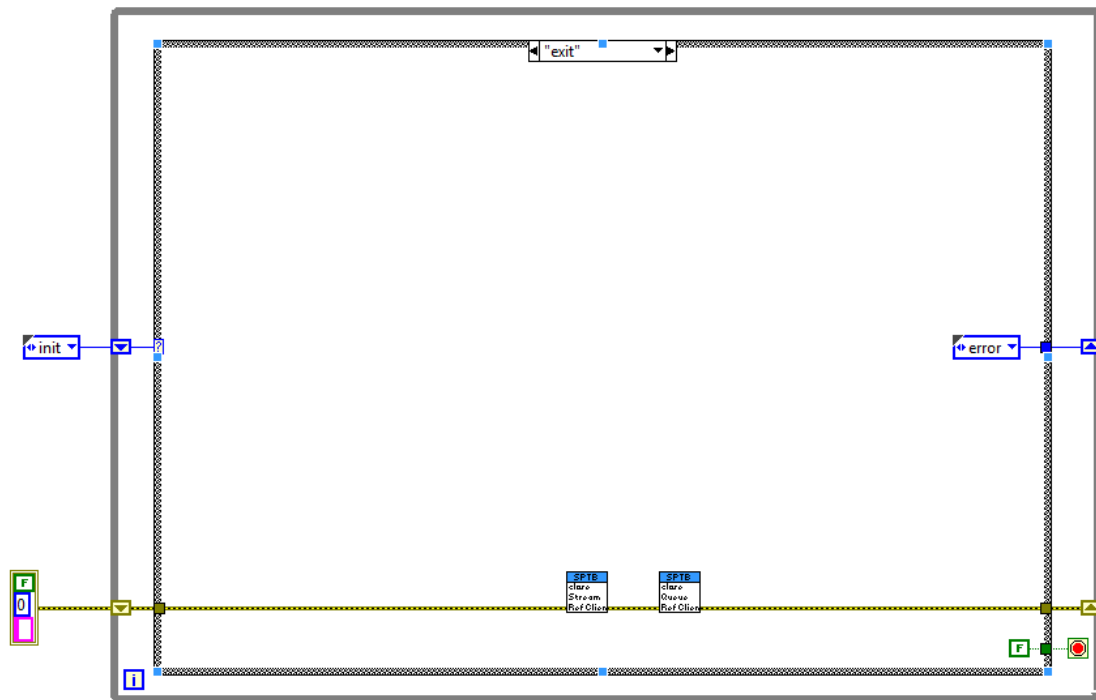


Figure 3. 14: Client application – Close references state

- **Error**

As in the server application, this stage runs the “*Simple Error Handler*”, which in case of error presents the code error and description. As previously noted, if an error occurs in a subvi of a previous stage, the state automatically switches to the “error” state and the execution is stopped. Depending on the choice made between “Restart Program” and “Exit Program” in the “post-processing options and Front-Panel” state, the program either stops or restarts.

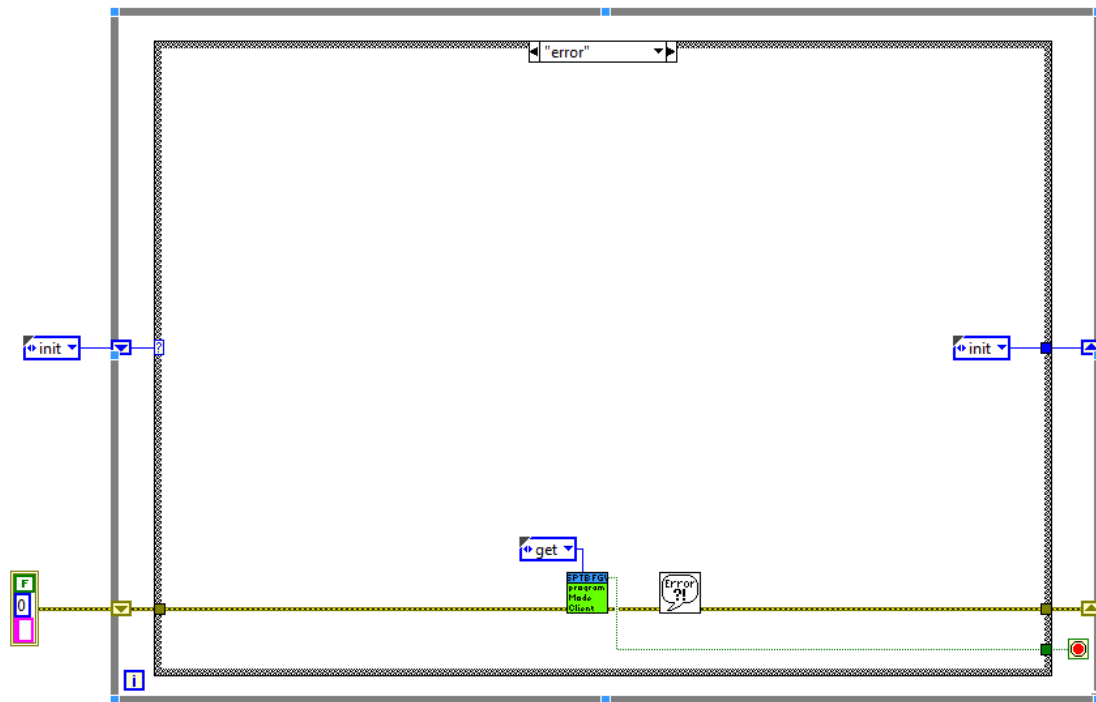


Figure 3. 15: Client application – Error state

### 3.3 Execution Options

The program offers 3 distinct tests for the time being. Those are Creep test, static-execution Hysteresis test and dynamic-execution Hysteresis test. The difference between a static-execution and a dynamic-execution Hysteresis test is that in the former the load remains constant throughout the test while in the latter it is varying. The modular nature of the specific implementation (use of nested subvis, case structures and state machine architecture) allows for a straightforward expansion of the functions if desired in the future.

#### 3.3.1 Creep test

As described in subchapter 2.4 “Functional Analysis”, in a Creep test the piezo element is left free of load and the auxiliary structure of Fig. 2.20 is used for convenience. Software-wise this means that in a creep test there is no need for motor movement and the biggest challenge lies in the implementation of the desired piezo excitation pattern, an example of which is illustrated in Fig. 2.21.

The Front Panel of a Creep test can be seen in Fig. 3.16. The parameters are minimum and maximum excitation voltage, the number of steps between minimum and maximum voltage (the number of steps of Fig. 2.21 for example is 5) and step duration in seconds. Appropriate subvis then take these parameters and create the desired output pattern, which is fed to the piezo actuator from slot 9/output 0 of the PXI and through the Cedrat LA75B. As described in subchapter 2.3.3 “Sensors and auxiliary devices”, the Cedrat LA75B has a gain factor of 20. This is of course taken into account and the user inserts the *desired* output voltages into the Front Panel. Those are then internally divided by 20 before fed to the Cedrat LA75B.

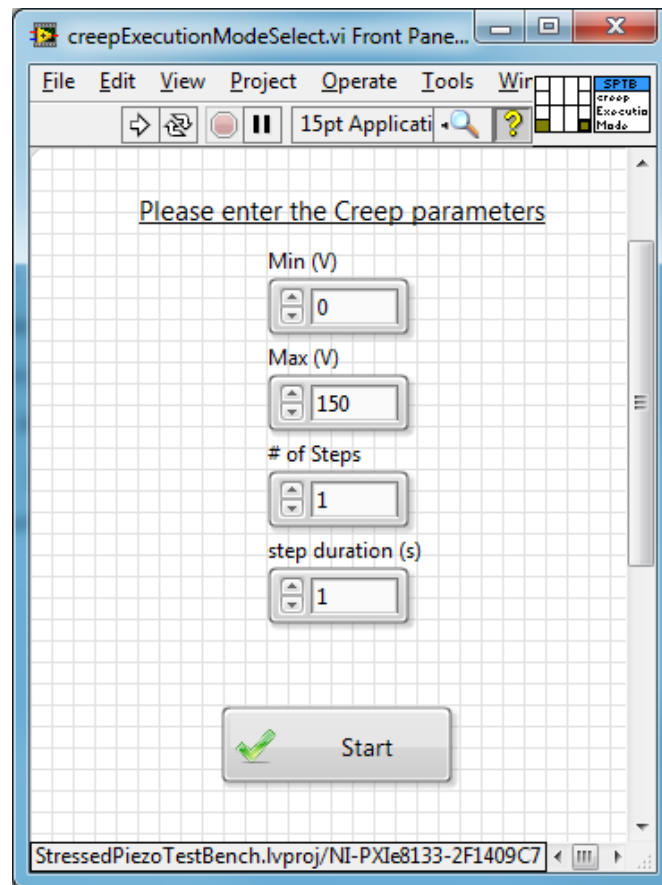


Figure 3. 16: Creep test Front Panel

Creating the desired creep output is a task on its own. This task is undertaken by “*creepArrayCreation*” subvi (illustrated in Fig. 3.17) and works in the following way. Using two for loops, it creates two separate sequences which are appended in the end. The first sequence is the rising sequence moving from the minimum to the maximum desired voltage. Starting with the minimum voltage, it adds to the created array as

many points with the value of minimum voltage as are needed to achieve the defined step duration in seconds. The number of points is calculated by multiplying the step duration in seconds with the sampling rate of the slot 9 output module (which is set in the DAQ configuration stage and is explained later). After that, it adds the same number of points but this time their value is that of the first level above minimum voltage. The voltage levels are calculated by dividing the difference between maximum and minimum voltage with the number of steps and then adding the result as many times as the number of steps to the minimum voltage until reaching the maximum (for example, if the minimum voltage is 4V, the maximum 10V and the number of steps is 3, then the voltage levels are 6V, 8V and 10V). The alternation between minimum voltage and different voltage levels is continued until the maximum voltage level is reached.

The second sequence works in a similar way and creates an alternation between the maximum voltage and decreasing levels of voltage until the minimum voltage is reached, each lasting the defined step duration in seconds.

The two arrays are then joined and the desired result is achieved (an example of which is seen in Fig. 2.21).

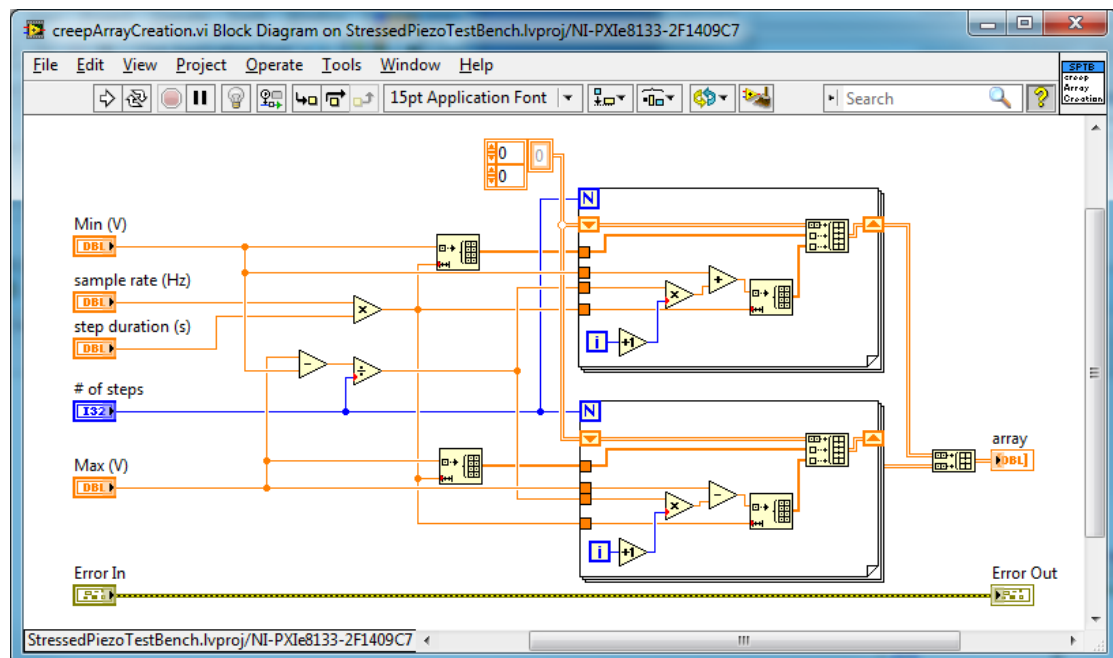


Figure 3. 17: Creep output pattern creation

### 3.3.2 *Static-execution Hysteresis test*

The static-execution Hysteresis test refers to a Hysteresis test where the load remains constant throughout the duration of the test. The load can be set to any desired level before the output exciting the piezo actuator is enabled though. Regarding the output, it can be either a constant voltage or a wave signal.

The Front Panel for both the static and the dynamic execution Hysteresis tests is the same and is illustrated in Fig. 3.18. The upper frame contains the signal generator parameters and is the same for both, while the lower frame is divided in two parts, with the left referring to the static execution and the right referring to the dynamic.

Regarding the signal generator parameters, if the “**Constant Output**” is selected, then the only parameter to set is the *desired* output in voltage (again, the Cedrat LA75B gain factor of 20 is internally taken into account). If the “**Signal Generation**” mode is selected, then the user can select among a Sine, Triangle, Square or Sawtooth wave and then set the frequency (Hz), offset (V), amplitude (measured in peak voltage, not peak-to-peak) and phase (degrees). In case of Square wave, the duty cycle is pre-set at 50%. The button labelled “Suggest Frequency” and the indicator next to it are useful in the dynamic execution and will be analysed in the following section.

Regarding the static-execution parameters found in the left side of the lower frame, the button labelled “Set desired Load” allows the user to set the desired load level, while the “Iterations” control effectively controls the duration of the test. *Each iteration equals one second*. That means that if for example a constant output of 150V with 10 iterations is selected, the test will last 10 seconds before stopping. During these 10 seconds, the piezo will be constantly fed 150V and the sensors will be acquiring data. If for example a sine wave with 2Hz, 75V offset, 75V peak, 270° phase and 10 iterations is selected, the test will last 10 seconds during which the output will be 20 sinusoidal waves from 0V to 150V and the sensors will be acquiring data. The test starts once the static-execution “OK” button is pressed.

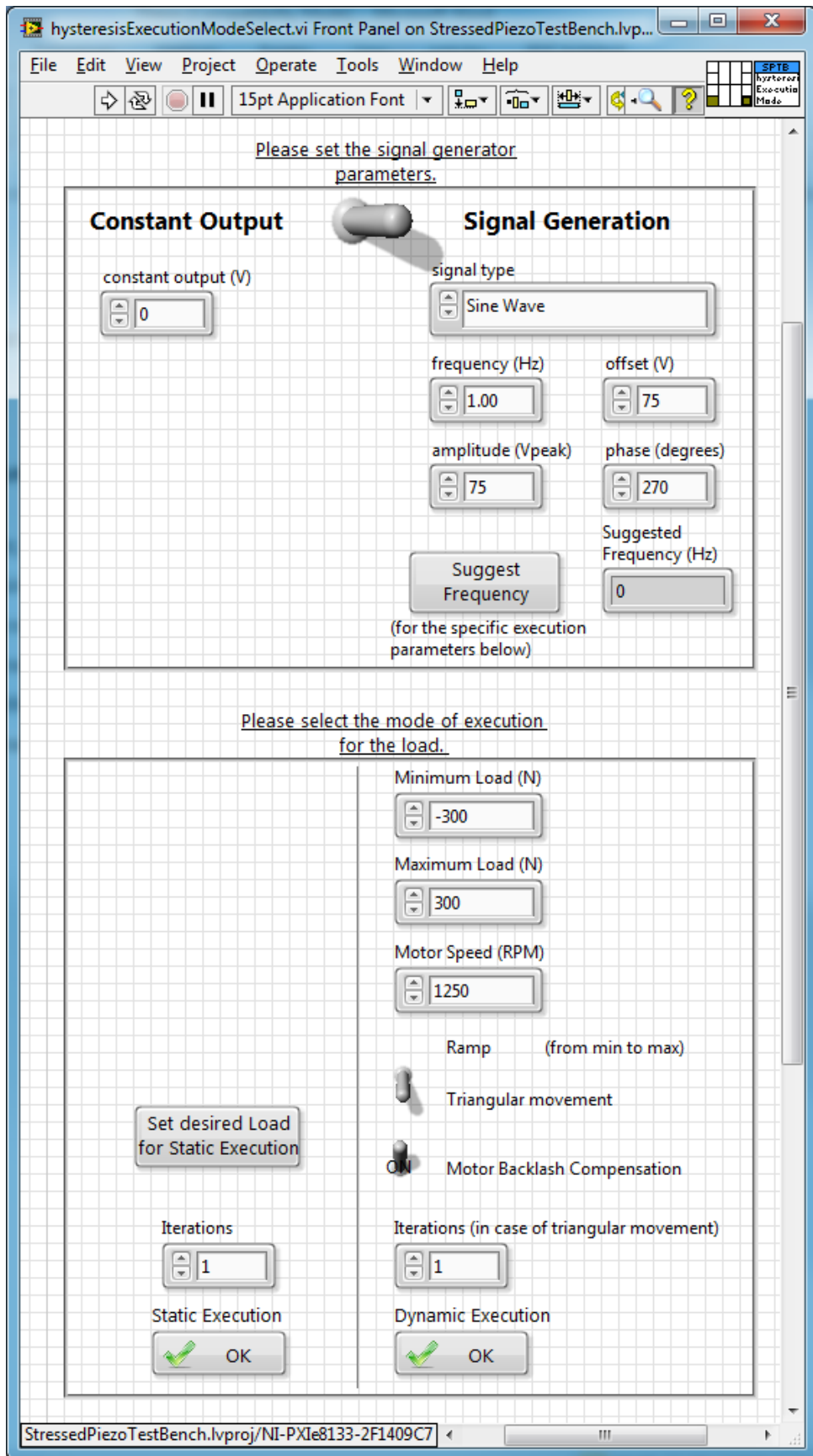


Figure 3. 18: Hysteresis test Front Panel

Button “Set desired Load” triggers subvi “*loadSettingStatic*” which loads a separate Front Panel that can be seen in Fig. 3.19. This subvi compares constantly the desired load value with the current load value and when the desired value is reached the motor stops operating. In fact, a nested subvi inside “*loadSettingStatic*” checks whether the desired load value is higher or lower than the current load value and respectively turns the motor clockwise or counter-clockwise with a predefined low speed. The sequence to operate the motor includes disabling the mechanical brake, enabling the servo function and entering an appropriate motor speed value (possible values range from  $-32,768$  to  $32,767$  as explained in subchapter 2.3.4). When the desired load setting is reached, the motor speed value is set to zero, the servo function is disabled and the mechanical brake is enabled. The button labelled “Emergency Motor Stop” stops the motor function at any time and can prevent damages in case of malfunction (for example if the load cell stops functioning). When all load adjustments are complete, the “OK” button closes the subvi.

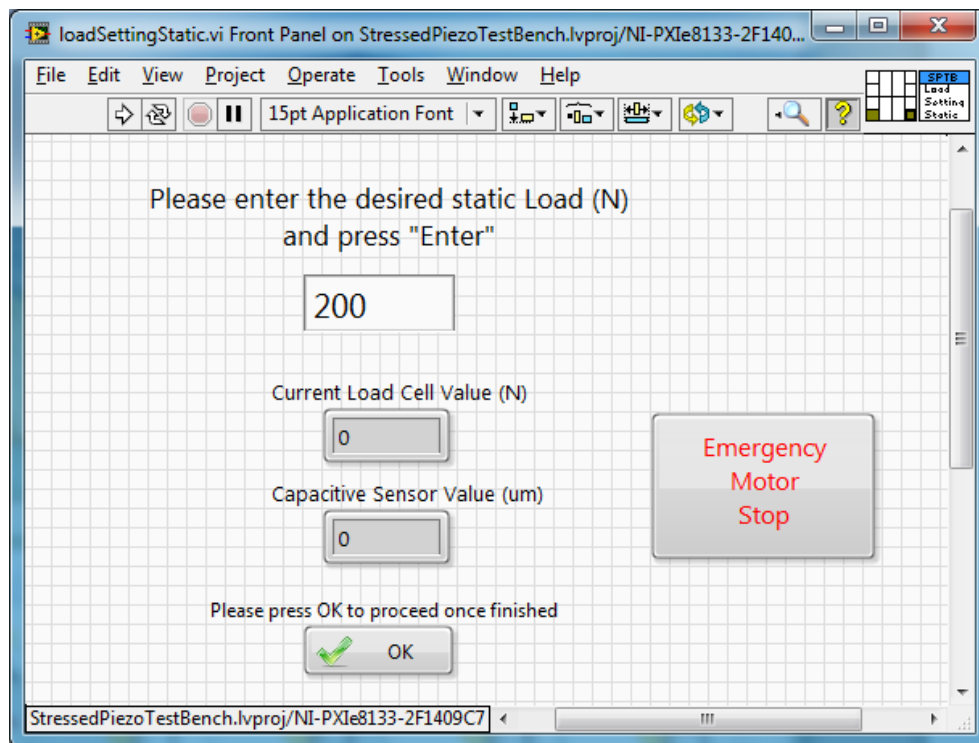


Figure 3. 19: Static-execution load setting Front Panel

### 3.3.3 *Dynamic-execution Hysteresis test*

The dynamic-execution Hysteresis test differs from the static in the sense that the load is varying throughout the duration of the test. One key difference is also that the duration of the test is not explicitly defined but is somewhat relative. This happens because the termination condition is not reaching a specific number of iterations as in the static execution, but rather completing the desired change in load. Although the motor speed can be precisely controlled, properties like the motor backlash (defined in subchapter 2.3.2) prevent the test duration from being absolutely deterministic.

The Front Panel of the dynamic-execution Hysteresis test is, as previously described, the same as the static-execution and is illustrated in Fig. 3.18. The dynamic-execution parameters are found in the right side of the lower frame and the test starts once the dynamic-execution “OK” button is pressed.

The user can select either a “Ramp” or a “Triangular movement” pattern for the load. The difference is that in the “Ramp” setting, the load goes from the minimum value to the maximum value with the defined motor speed and then the test ends, while in the “Triangular movement” setting, the load goes from the minimum value to the maximum value and back to the minimum value with the defined motor speed and this happens as many times as indicated in the “Iterations” field.

When the dynamic-execution “OK” button is pressed and before the piezo excitation output and the sensors are enabled, the motor operates until the minimum load value is reached (*loadSettingDynamic* subvi, Fig. 3.20. While in operation, the user can stop the motor function in case of emergency with the “Emergency Motor Stop” button).

At this point it is crucial to note that the backlash depends on the direction of the motor movement. It is impossible to eliminate the motor backlash but it is possible to conduct the tests in such a manner that the backlash is in essence predictable. This is what the “Motor Backlash Compensation” switch does. When enabled, it ensures that the minimum load value is reached from a higher load value. So, if for example the initial load value is 200N, the desired load minimum value is 300N and the “Motor Backlash Compensation” is enabled, the load will rise from the initial 200N to 350N (it is pre-set to go 50N higher than the desired value) before falling back to the desired 300N. If in the same scenario the “Motor Backlash Compensation” was disabled, the



load would rise from the initial 200N directly to the desired 300N and stop. If the initial load value is higher than the desired minimum load value, then regardless of whether the “Motor Backlash Compensation” is enabled or not, the load will fall directly to the desired minimum load value. This function is not necessary in case of the “Ramp” pattern but it is indispensable in the “Triangular movement” pattern as it makes the backlash identical in each iteration. If it was disabled and the initial load value was lower than the desired minimum load value, then the minimum load value would be reached from a lower value, the test would start, the load would reach the maximum value and then fall back to the minimum value, but this time the minimum load value would be reached from a higher load value and the backlash would be different.

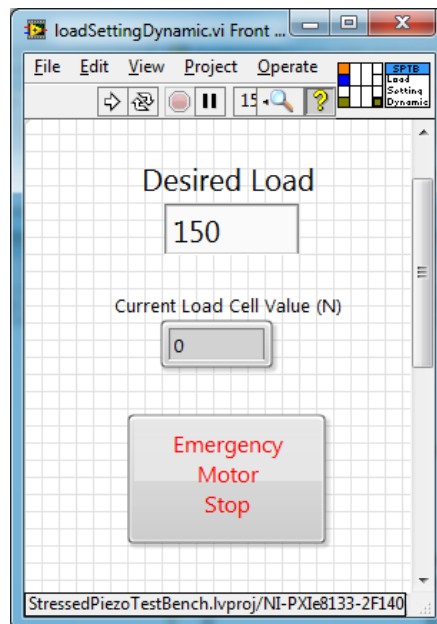


Figure 3. 20: Dynamic-execution load setting Front Panel

The signal generator parameters work as previously explained but a key difference is the addition of the “Suggest Frequency” button and indicator. This function is useful in case of the “Triangular Movement” pattern and when pressed, this button triggers the “*suggestFrequency*” subvi, which executes the motor movement with the execution parameters of the lower frame but with the output and the sensors disabled. It measures the time in seconds that is needed for the movement to be completed and divides it with the number of iterations. The reciprocal of this result is the frequency

of the triangular movement of the load and is presented in the indicator. This allows the user to set the signal generator to a triangular wave of the same frequency, so that the load and the output follow the same triangular pattern with the same frequency. As stated, because of the relative nature of the backlash, it is more accurate to perform a specific load movement and measure its duration than trying to calculate it mathematically.

### 3.4 Data management

As previously mentioned, data management is achieved through the utilization of queues, network streams and tsv files. In order to fully understand the mechanisms behind data management, it is necessary to present the course of data from start to end.

In the server application, the data acquired from the “*DAQmx Read*” vis in the “data acquisition and data transmission” state are enqueued in the respective queues with the help of suitable FGVs. Specifically, the data from the capacitive sensor, the signal generator feedback, the load cell, the current and the strain gage are being read and enqueued in the “*SyncedAcquisitionProcess*” subvi (Fig. 3.21) and their type is *1-dimensional array of double (1-d array of DBL)*.

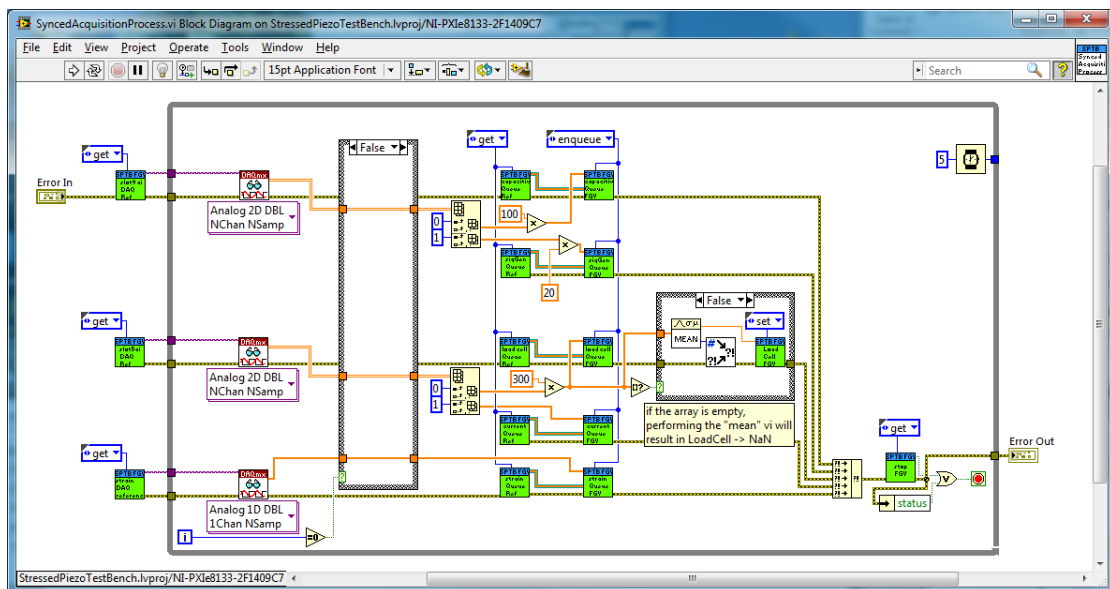


Figure 3. 21: “*SyncedAcquisitionProcess*” subvi

All data FGVs used in this project have two distinct states – **enqueue** and **dequeue**. Although the data type varies among the different FGVs, the enqueue state is similar for all of them and simply includes an “*Enqueue Element*” function, an example of which can be seen in Fig. 3.22, taken from the capacitive sensor FGV.

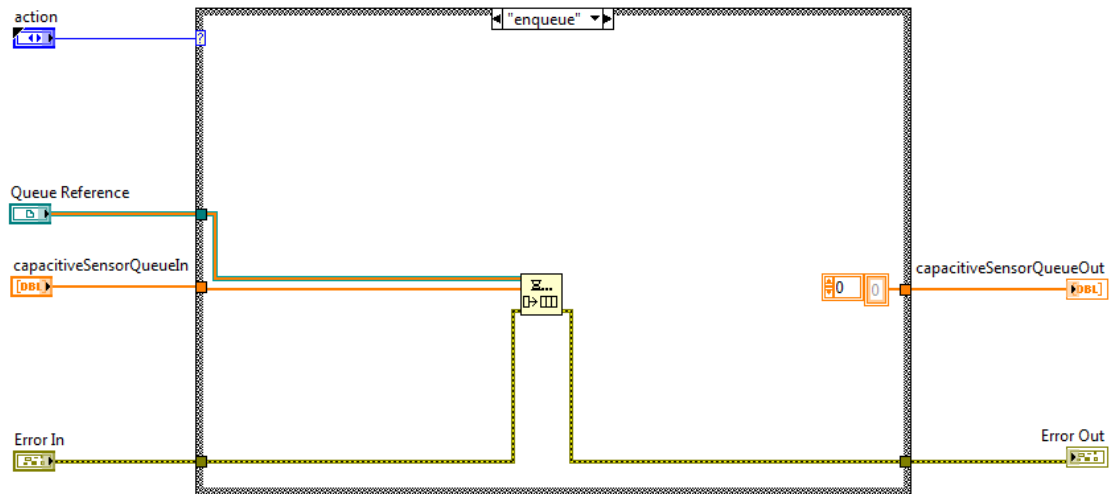


Figure 3.22: “Enqueue” state of data FGVs in server application

Data from the PT100 sensor is acquired in the “*PT100AcquisitionProcess*” subvi and that data of type *double* is enqueued to the respective FGV (Fig. 3.23)

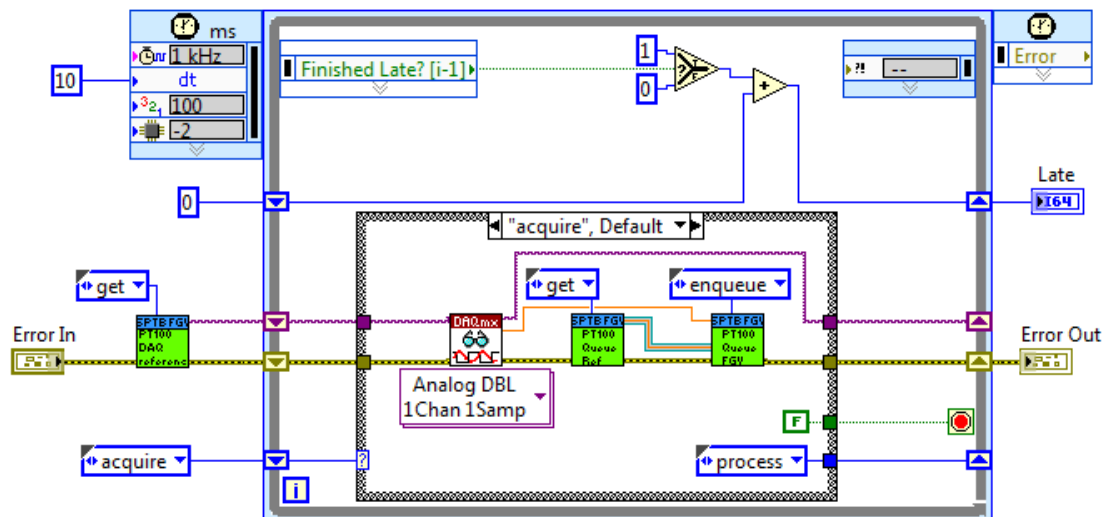


Figure 3.23: “PT100AcquisitionProcess” subvi

Lastly, data from the two accelerometers is acquired in the “*accelerometerAcquisitionProcess*” subvi. The data type is *cluster of 6 waveforms of double (cluster of 6 wvf(DBL))* and again the data is enqueued in the respective FGV (Fig. 3.24). This subvi has another state that follows the data acquisition and performs the Fast Fourier Transform (FFT) and enqueues the data to the respective FGV (Fig. 3.25).

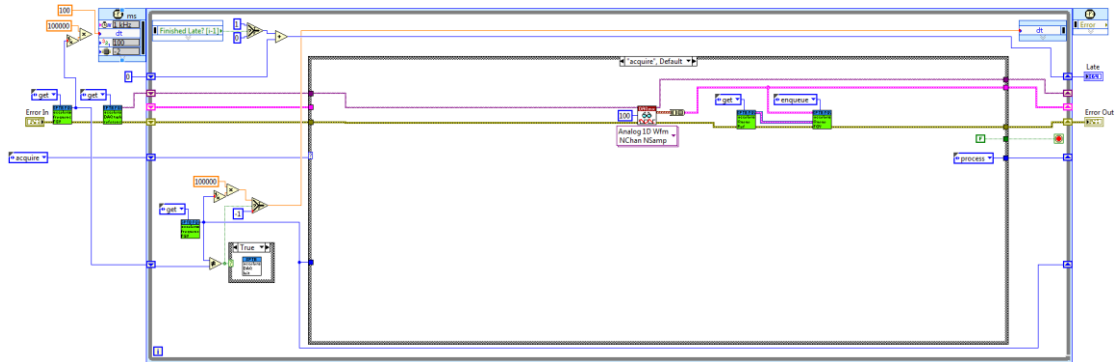


Figure 3.24: “*accelerometerAcquisitionProcess*” subvi

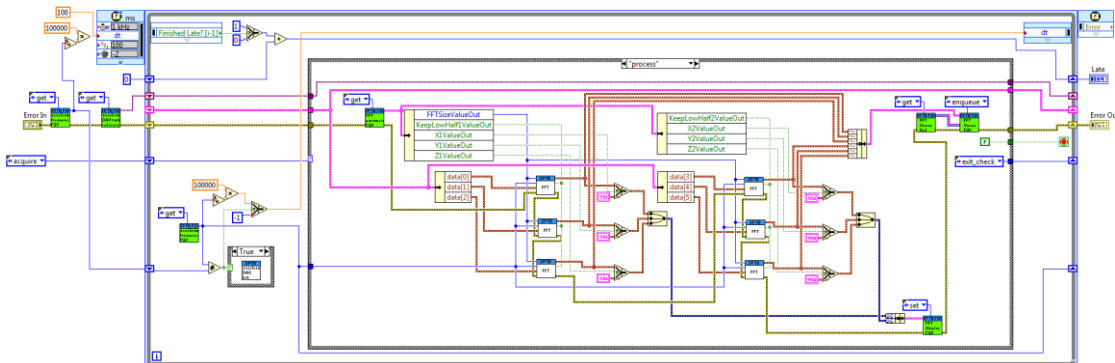


Figure 3.25: “*accelerometerAcquisitionProcess*” subvi - FFT

The data acquired is constantly added to the queues and at regular intervals it is transmitted to the client application. The logic here is that data is acquired without interruption and a process in parallel transmits cumulative packages from time to time. It is easy to understand that the transmission rate is much lower than the reading rate. The data transmission is undertaken by the “*dataSend*” subvi found in the same state as the data acquisition in the server application (Fig. 3.26). The data is extracted from the queues with the “*dequeue*” state of the data FGVs and is then transmitted via the network stream to the client with the “*Write Single Element to Stream*” function.

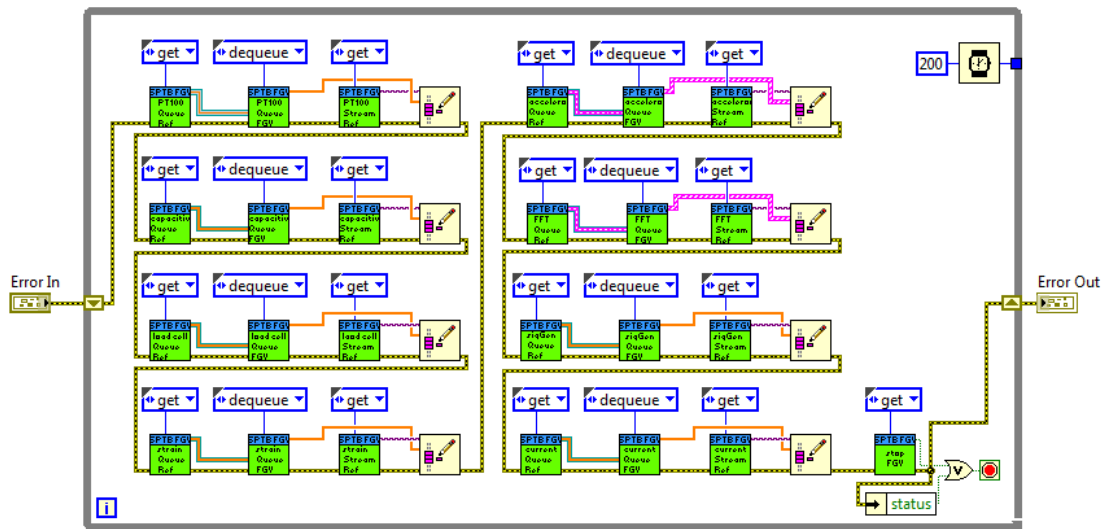


Figure 3.26: Data transmission over network stream

Contrary to the “enqueue” state which was similar for all data FGVs, the “dequeue” state is more complex as it involves the creation of the data packages. The “dequeue” state for the capacitive sensor, the signal generator feedback, the load cell, the current and the strain gage is illustrated in Fig. 3.27.

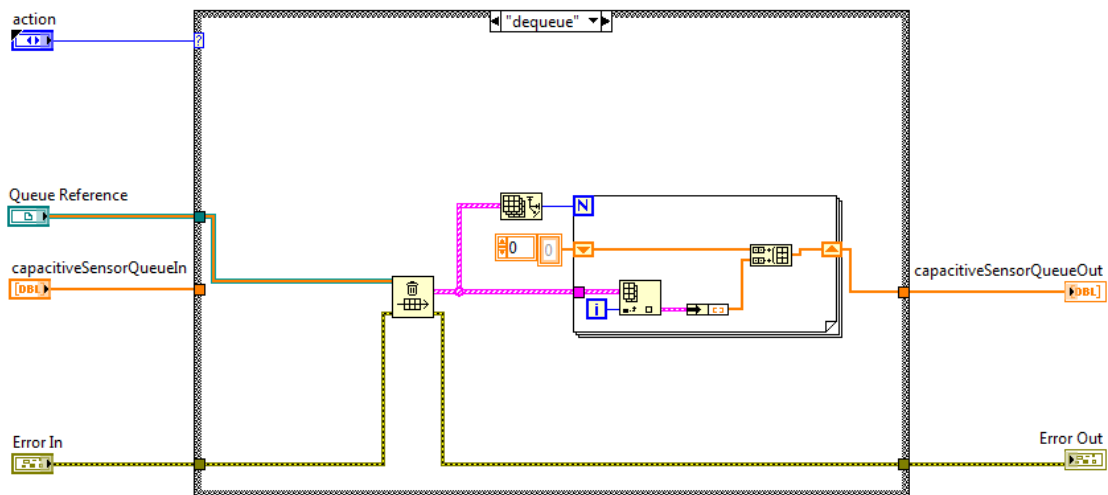


Figure 3.27: “Dequeue” state for most of the data FGVs in server application

At first, all the available queue elements are extracted with the “Flush Queue” function. This, however, is not enough as the returned data type is 1-dimensional array of cluster of 1 element of 1-dimensional array of double. This happens because

each element of the queue is, as mentioned, of type *1-dimensional array of double*. What we want is to end up with just 1-dimensional array of double where all queue array elements are placed in the same array one after another. To achieve this, the code segment next to the “Flush Queue” function was developed and the result is a 1-dimensional array containing all the available queue elements placed one after the other.

The situation is more straightforward when it comes to the PT100 FGV. The queue elements for the PT100 FGV are of type *double*, as previously mentioned. Using the “Flush Queue” function, the result is the desired *1-dimensional array of double* containing all queue entries one after the other.

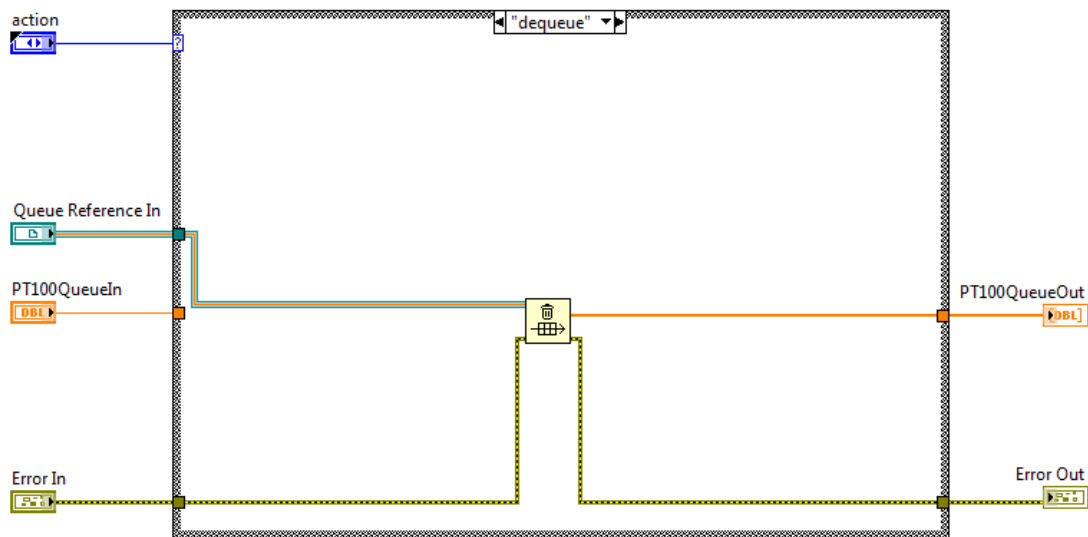


Figure 3. 28: PT100 “Dequeue” state in server application

The same applies to the accelerometer and FFT FGVs although the data type is different. Specifically, the queue elements of both the accelerometer and FFT FGVs are as mentioned of type *cluster of 6 waveforms of double*. By using the “Flush Queue” function we end up with the desired *1-dimensional array of cluster of 6 waveforms of double*. The “dequeue” state can be seen in Fig. 3.29.

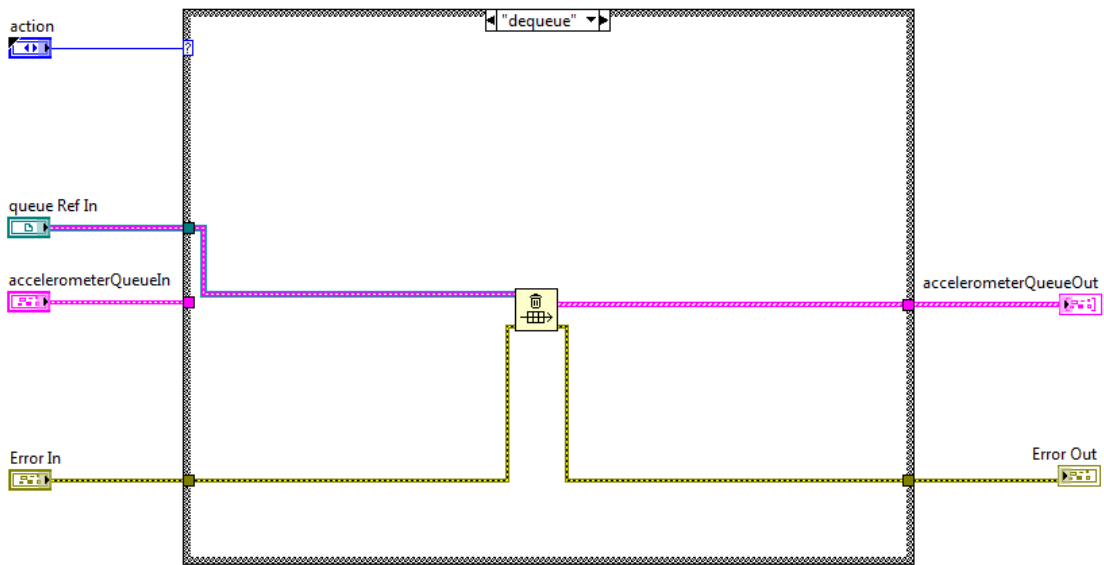


Figure 3.29: Accelerometer and FFT “Dequeue” FGV state in server application

The data sent over the network stream are received from the client application in the “dataReceive” subvi (Fig. 3.30). The data are enqueued in queues that exist in the client side with the help of suitable FGVs. The FGVs have again two states – “enqueue” and “dequeue” and their content will be analysed shortly.

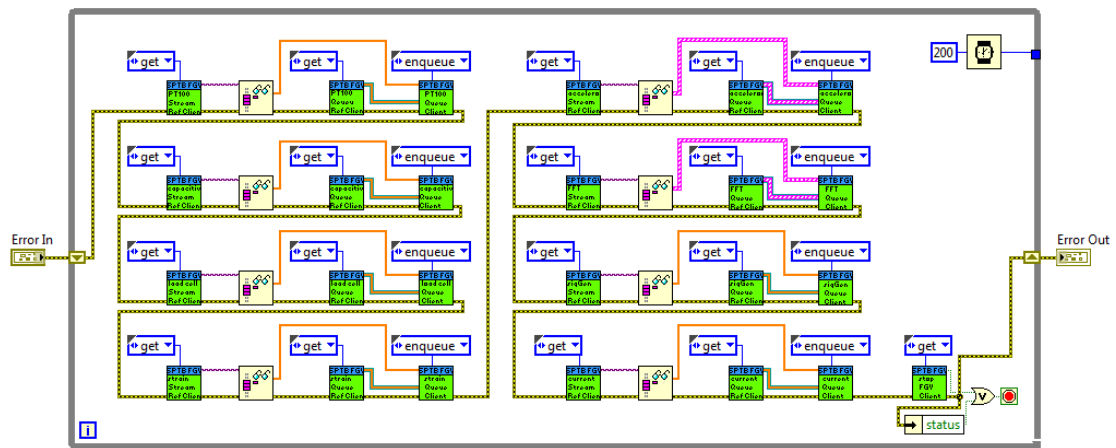


Figure 3.30: Data reception on the client application

The “enqueue” state is similar for all data FGVs and simply includes an “*Enqueue Element*” function. The data type varies among FGVs with the capacitive sensor, signal generator feedback, load cell, current, strain gage and PT100 featuring a *1-dimensional array of double*, while the accelerometer and FFT a *1-dimensional array*

of cluster of 6 waveforms of double. An example of the “enqueue” state is taken from the capacitive sensor FGV and can be seen in Fig. 3.31.

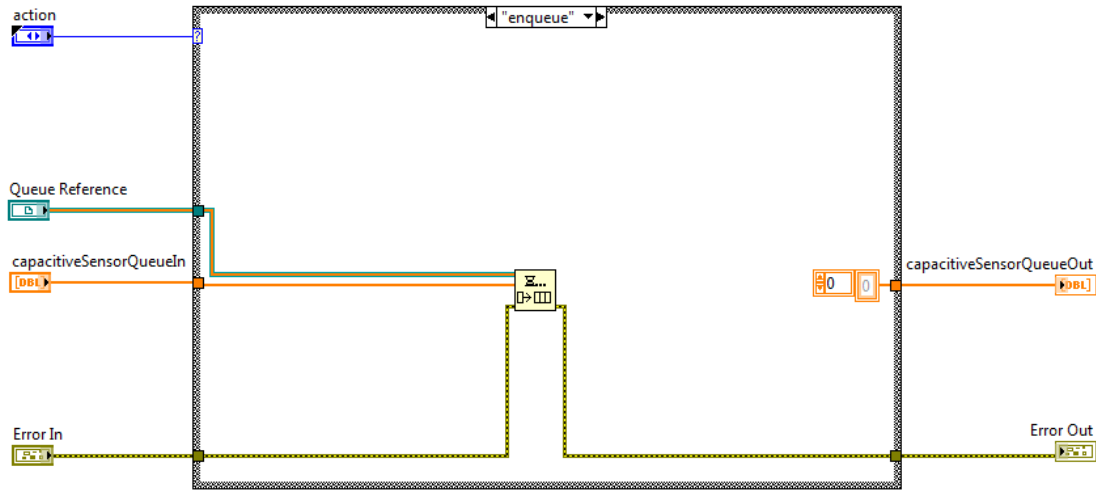


Figure 3.31: “Enqueue” state of data FGVs in client application

The idea behind data management in the client application is that new elements are enqueued from the network stream at regular intervals and at the same time a parallel VI undertakes the task to write the data to the tsv files. This last step involves performing a “dequeue” on the FGVs and writing the extracted data to the tsv files.

The “dequeue” state is similar for the capacitive sensor, signal generator feedback, load cell, current, strain gage and PT100 FGVs and is illustrated in Fig. 3.32.

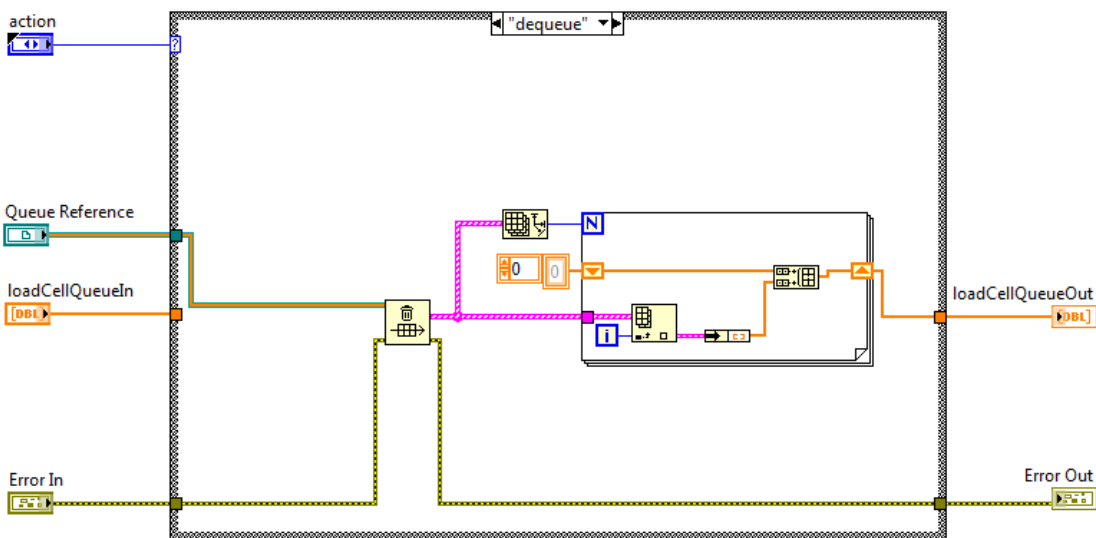


Figure 3.32: “Dequeue” state for most of the data FGVs in client application



The code is similar to the “dequeue” state found in many FGVs in the server application described above and the result is a *1-dimensional array of double* containing all the available queue elements of 1-dimensional array of double placed one after the other.

The situation is different, however, when it comes to the “dequeue” state of the accelerometer and FFT FGVs, shown in Fig. 3.33.

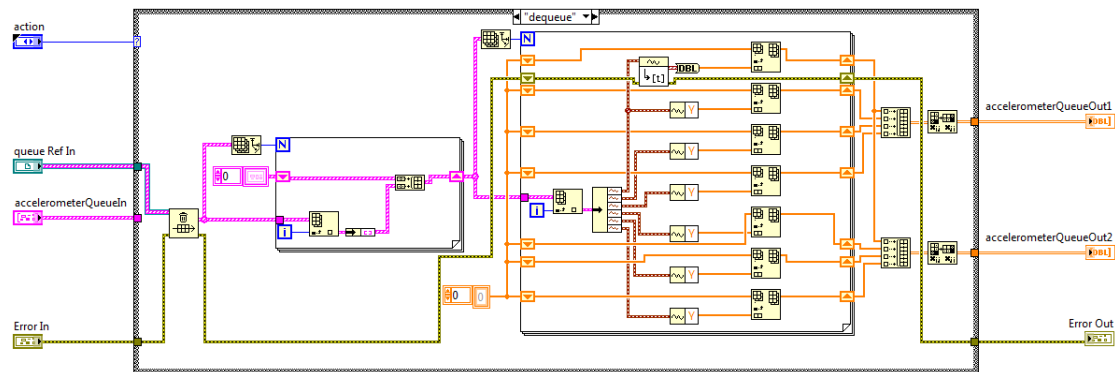


Figure 3.33: Accelerometer and FFT “Dequeue” FGV state in client application

What the code above does is extract all available queue elements of type *1-dimensional array of cluster of 6 waveforms of double* and place them one after the other in a new 1-dimensional array (of clusters of 6 waveforms of double). A for loop then, which runs as many times as the array size, splits each cluster into the six waveforms (X,Y,Z axis for accelerometer 1 and 2 respectively) and creates a new array for each of them containing the respective data from all the clusters put one after the other. In the end, the three axis of each accelerometer along with the time (for the accelerometer FGV) or the frequency (for the FFT FGV) are bundled together to create two separate *2-dimensional arrays of double* containing four columns (time or frequency, X, Y, Z). In other words, the end result is two separate 2-d arrays (one for each accelerometer sensor) each containing the X, Y, Z and time or frequency components of all available data in the queue.

After the data are dequeued in the “*dataLoggingProcessClient*” subvi (Fig. 3.34), they are saved in the respective tsv files using the “*Write To Spreadsheet File*” function. The tsv files have a pre-defined directory and the data logging process is done in parallel and not in serial with the data reception because in principal writing to the

hard disk is a process with varying time demands. With the specific implementation, data reception is performed without interruptions at regular intervals and data logging is performed when the necessary computer resources are available, writing all available data at that time to the tsv files.

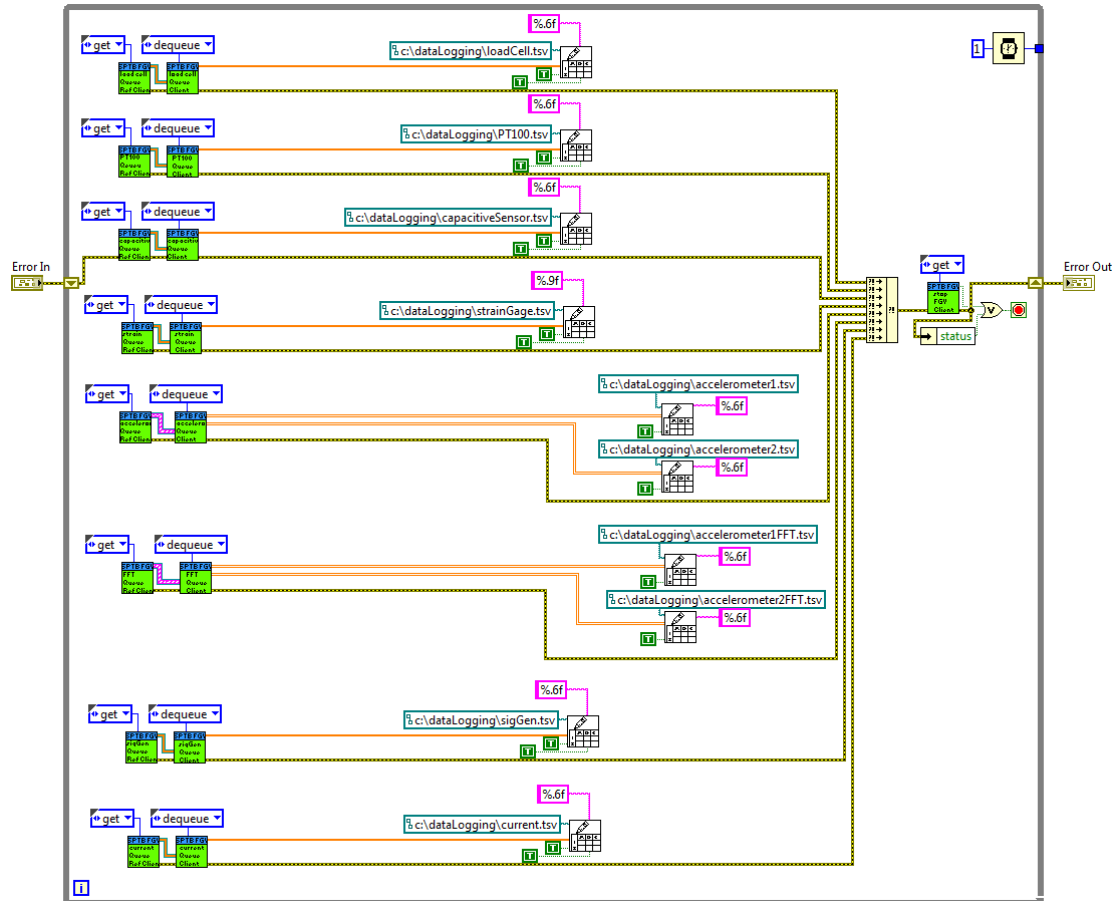


Figure 3. 34: Data logging in client application

Once the data reception and logging is complete, the tsv files can be copied to a new location, used to draw the graphs or used to generate the reports with the help of the Matlab scripts. The last two cases will be covered in detail in the next sections.

### 3.5 DAQ Synchronization

The synchronization of the DAQ devices on the PXI proved to be one of the biggest challenges. It was crucial to ensure that all readings were in sync with each other, as

otherwise false conclusions could be drawn and the characterization process would not be reliable.

The devices that are required to be in absolute sync with each other are slots 5 (PXI-4461), 7 (PXIe-4330) and 9 (PXI-4461). Slot 5 acquires readings from the load cell and the current sensing board, slot 7 acquires readings from the strain gage while slot 9 generates the output and acquires readings from the capacitive sensor and the signal generator feedback. All of these modules utilize delta-sigma analog-to-digital converters, which imposes some limitations to the synchronization process. Modules in slot 2 (PXIe-4357), acquiring readings from the PT100, and slot 3 (PXIe-4496), acquiring readings from the two accelerometers, do not require the same level of accuracy on synchronization and this is why their tasks are created as usual in “*PT100DAQInit*” and “*accelerometerDAQInit*” subvis.

The synchronization of slots 5, 7 and 9 is achieved in “*initSyncStartStatic*” for the Creep and static-execution Hysteresis tests and “*initSyncStartDynamic*” for the dynamic-execution Hysteresis test respectively. The specific implementation allows synchronization of multiple DSA cards at different sampling rates.

In both aforementioned subvis, PXIe-4330 of slot 7 is set as the master synchronization device. Slots 5 and 9 are set as slave. The devices are synchronized with the reference clock synchronization method. That means that both master and slave devices lock their analog-to-digital converters and digital-to-analog converters over the sample clock to a shared reference clock on the PXIe chassis backplane. To achieve synchronization, the slave slots are set to accept a trigger produced by the master device as their start trigger. This trigger is exported from the master device when its task is started. To be picked up from the slave devices though, it is necessary that their tasks are started before that of the master device. This is the reason for the Flat Sequence structure found in both subvis.

In addition, because of the nature of PXI-4461 cards (present in slots 5 and 9), it is necessary to remove the delay that is introduced by the digital filter. Those cards use a digital filter to remove frequency components above the Nyquist frequency. This digital filter imparts delay because a minimum number of samples are needed before the digital filter can work effectively. To do that, we perform a “dump” reading after the tasks of the devices have started, where we read as many elements as the

multiplication of delay (*AI.FilterDelay* DAQmx Channel property node) and sample clock rate of the specific channel (*SampClk.Rate* DAQmx Timing property node).

Another noteworthy point is that before starting the slot 9 output task, we need to perform a dump write. This happens because by default an output task cannot start with nothing being output. The solution to this as shown is to write a small array of zero elements before starting the task. The minimum amount of required points to output depends on the sample rate – and in our case of 2 kS/s sample rate it is 34 samples – but because Analog Output Regeneration<sup>12</sup> is enabled by default, we can get away with an array as small as 2 elements of zero.

Up to this point, the two aforementioned subvis are the same. The difference lays in the output task parameters. In the case of dynamic-execution (dynamic-execution Hysteresis test), as previously mentioned, the termination condition is the completion of the desired load change. That means that the signal generator output has no predefined duration or iterations and continues to produce the selected output until the load completes its change. This makes things straightforward for the output task parameters, as a Continuous task with a large buffer is sufficient. The “*initSyncStartDynamic*” subvi is shown in Fig. 3.35.

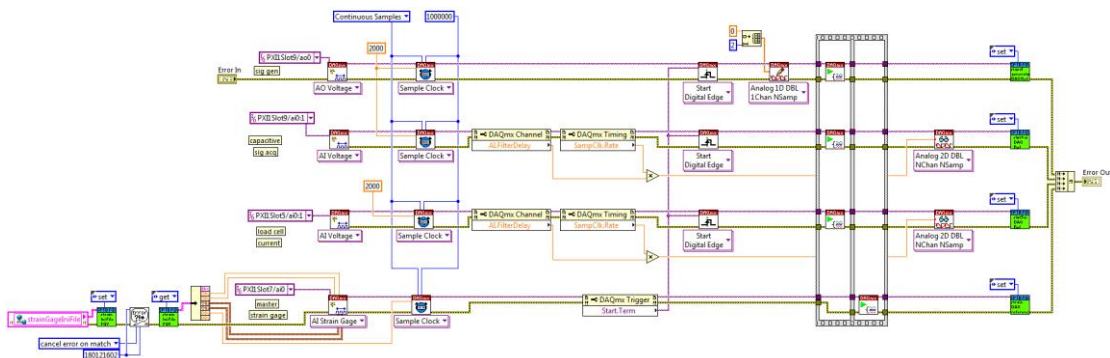


Figure 3. 35: “*initSyncStartDynamic*” subvi

<sup>12</sup> Regeneration refers to the DAQ device's ability to 'regenerate' samples that it previously generated. So, when regeneration is enabled, data written to either the user buffer or the FIFO is reused by the DAQ device. In this way, a continuous output can be achieved without LabVIEW having to continuously write new data to the buffer.

Things are more complicated when it comes to static-execution (Creep or static-execution Hysteresis test). The termination condition of the test in this case is the production of the desired output. That means that the output task has to be set to “Finite Samples” mode and the number of samples to be produced be exactly defined. The calculation of the exact points to be produced is different depending on the type of test (Creep or static-execution Hysteresis test). In case of static-execution Hysteresis test, the total number of points that will be generated in the output is the multiplication of the “Iterations” field from the respective Front Panel (Fig. 3.18, lower-left corner) with the sample rate. That case can be seen in the “*initSyncStartStatic*” subvi (illustrated in Fig. 3.36) for the “Hysteresis” state of the case structure.

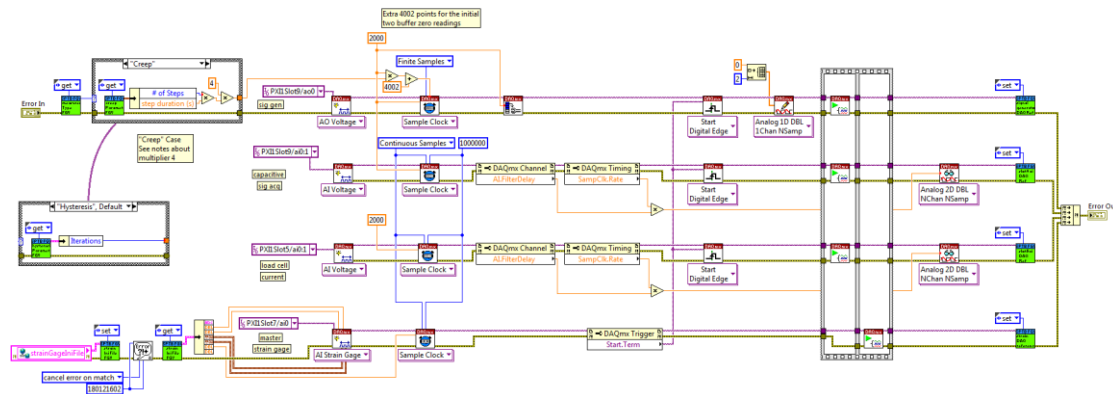


Figure 3.36: “*initSyncStartStatic*” subvi

In the case of creep test (Fig. 3.36, “Creep” state of the case structure), the number of points to be generated is calculated by multiplying the “# of steps” with the “step duration” fields from the respective Creep test Front Panel seen in Fig. 3.16. That result is then multiplied by 4 because of the creep excitation signal pattern (the result is doubled because of the alternation between the fixed level and the progressing level of the voltage and is doubled again because of the ascending and descending nature of the pattern). That new result is finally multiplied by the sample rate to give us the number of points to be generated.

In both cases, the output buffer size is set to the same value as the sample rate (2000) and the number of points to be generated is increased by a size of a little more than

two buffers (4002 points in our case) for reasons that will become apparent in the next paragraph.

One important property of the delta-sigma PXI-4461 card that is used for the output generation is that because of the oversampling that occurs because of the delta-sigma converters, the output cannot change instantly from its initial zero value that was defined before the start of the task. It takes two sampling periods, which translates to 4000 points, in case of Finite Samples (Creep and static-execution Hysteresis test) and one sampling period, which translates to 2000 points, in case of Continuous samples (dynamic-execution Hysteresis test) for the output to start producing the new values. So, for static execution, it will take 2 seconds from when the signal generation starts until the output starts generating the defined values. Until then, the output will produce the value 0. The same applies to dynamic execution with the difference that the delay is 1 second.

This behaviour is not a problem by itself, but since the signal generation and the acquisition from the sensors run in parallel and start at the same time, the data of the first seconds was useless. The solution to this is to not start the data acquisition (“SyncedAcquisitionProcess” subvi) and the data transmission over network stream (“dataSend” subvi, Fig. 3.26) until the total samples generated in the output (*TotalSampPerChanGenerated* DAQmx Write property node) are 4000 for static execution and 2000 for dynamic execution. This task is undertaken by “removeJunkSamples” subvi seen in Fig. 3.8. Its code is presented in Fig. 3.37. The “JunkSamples” argument is 4000 in case of static execution and 2000 in case of dynamic execution.

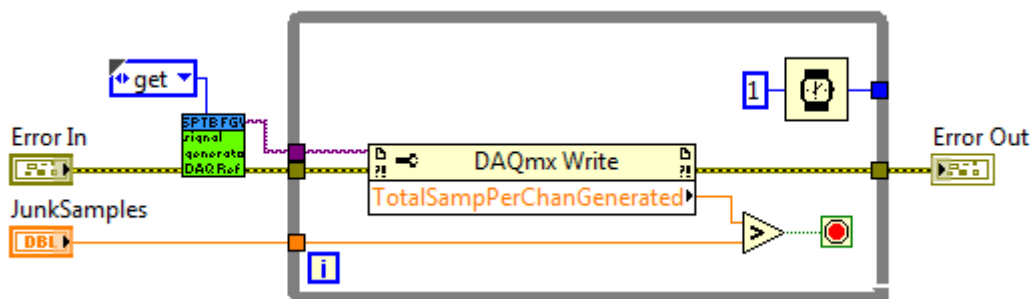


Figure 3. 37: “removeJunkSamples” subvi

Even still, there is one last necessary step to achieve complete discard of the junk samples. From the moment the input tasks start, their input buffers are filling with data. The data from the input buffers are extracted whenever a Read function is performed. With that in mind, the first “Read” functions performed in the “*SyncedAcquisitionProcess*” subvi would enqueue and subsequently transmit junk data. To avoid this, we dump the very first Read data and in their place void arrays are passed. This situation is presented in Fig. 3.38. The next Read occurs so fast that no valuable data are lost.

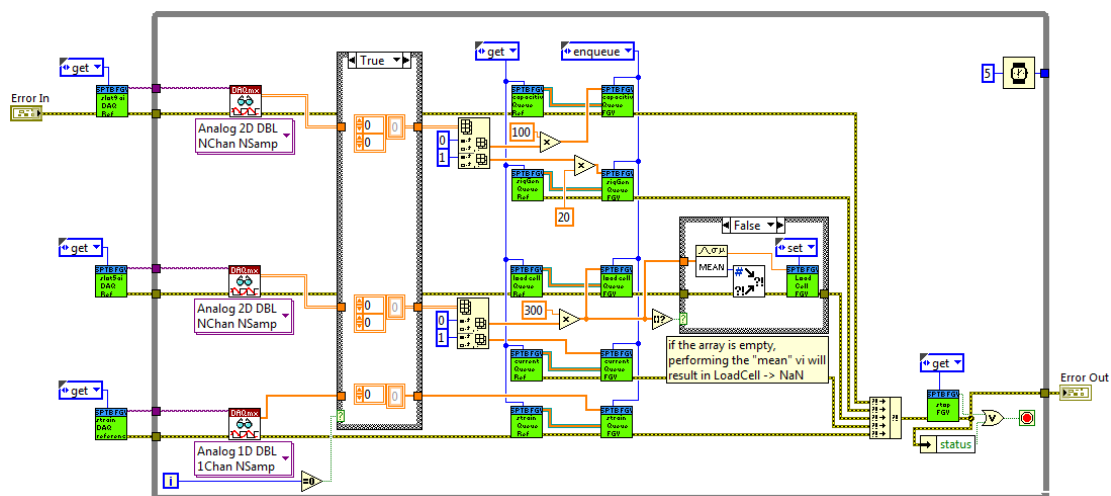


Figure 3.38: “*SyncedAcquisitionProcess*” subvi 1<sup>st</sup> iteration

### 3.6 Graph creation

The data saved in the tsv files described in subchapter 3.4 “Data Management” can be used to draw the graphs. The task is undertaken by subvi “*graphCreationClient*” in the “Post-processing options and Front Panel” state of the client application, the front panel of which can be seen in Fig. 3.39.

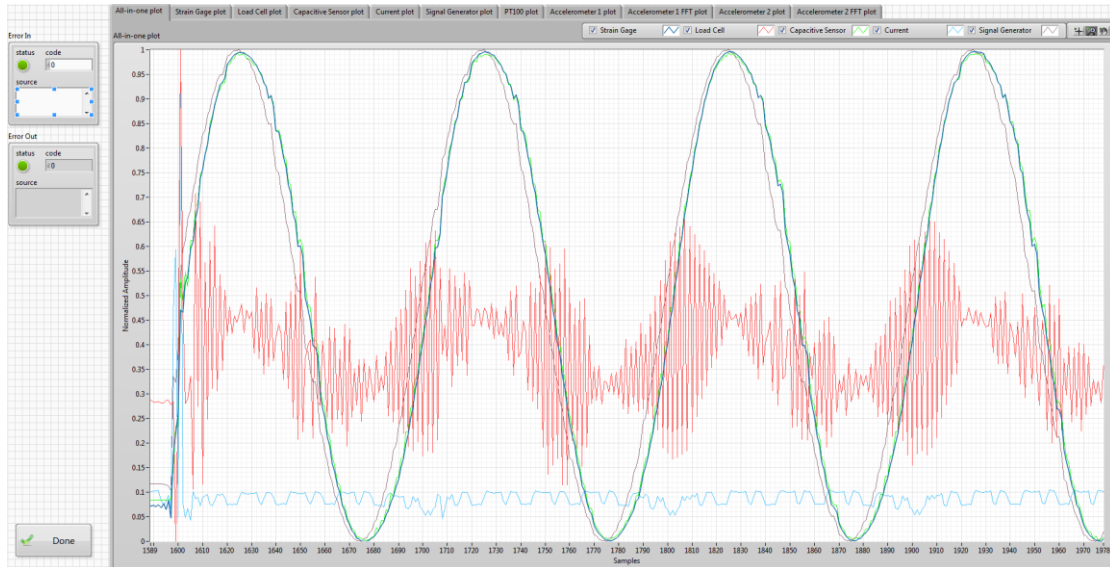


Figure 3.39: “graphCreationClient” subvi

On the upper part of the front panel there are tabs to select which plot to show. The options are All-in-one plot (a collective plot that draws the strain gage, load cell, capacitive sensor, current and signal generator feedback plots in the same plot and in sync with each other), strain gage plot, load cell plot, capacitive sensor plot, current plot, signal generator feedback plot, PT100 plot, accelerometer 1 plot, accelerometer 1 FFT plot, accelerometer 2 plot and accelerometer 2 FFT plot. On the lower left side of the front panel, the button labelled “Done” closes the subvi.

The code of the subvi can be seen in Fig. 3.40. The strain gage, load cell, capacitive sensor, current, signal generator feedback and PT100 plots are drawn simply by reading the data from the respective tsv files. The accelerometer 1 and 2 rely on the same logic and the end result for each accelerometer is a plot with three components – X, Y and Z (that can be turned on and off for better observation).



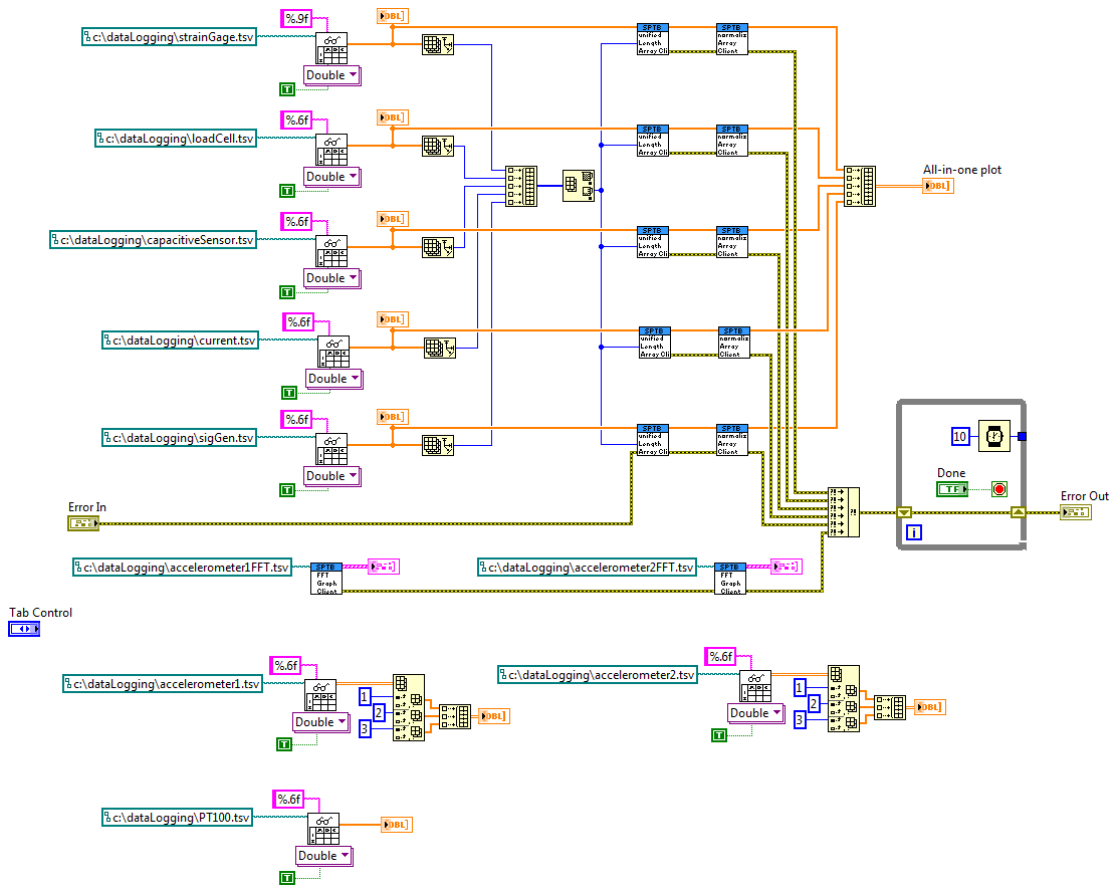



Figure 3. 40: “graphCreationClient” subvi code

Drawing the FFT plots posed a challenge as the data should first be sorted by ascending frequency. The saved data in the FFT tsv file have the structure that the “dequeue” state passed – that is, *1-dimensional array of cluster of 6 waveforms of double* placed one after the other in a new 1-dimensional array. That means that practically the format is that of the left side of Fig. 3.41. In that figure only 4 frequencies have been isolated but the logic is the same. To make the data useful, it needs to be sorted in an ascending frequency (column denoted “F” in Fig. 3.41) order. The desired result can be seen on the right side of the figure. This task is undertaken by “*FFTGraphClient*” subvi (Fig. 3.42), which sorts the data in an ascending frequency order and then does a mean calculation for each frequency for each axis (X, Y and Z). The results are then drawn on the FFT plot which has three components – X, Y and Z – that can be turned on and off for better observation.

f	X	Y	Z
0	0.514107	1.020329	1.135178
9.765625	0.510089	1.010979	1.119166
19.53125	0.498094	0.983271	1.071929
29.29688	0.478305	0.938213	0.995816
0	0.258195	0.49617	0.717426
9.765625	0.259848	0.493711	0.712039
19.53125	0.264513	0.486558	0.695951
29.29688	0.271385	0.47535	0.669416
0	0.301238	0.538313	0.61198
9.765625	0.304762	0.537017	0.611854
19.53125	0.314646	0.533105	0.61114
29.29688	0.329094	0.526503	0.608925



f	X	Y	Z
0	0.514107	1.020329	1.135178
0	0.258195	0.49617	0.717426
0	0.301238	0.538313	0.61198
9.765625	0.510089	1.010979	1.119166
9.765625	0.259848	0.493711	0.712039
9.765625	0.304762	0.537017	0.611854
19.53125	0.498094	0.983271	1.071929
19.53125	0.264513	0.486558	0.695951
19.53125	0.314646	0.533105	0.61114
29.29688	0.478305	0.938213	0.995816
29.29688	0.271385	0.47535	0.669416
29.29688	0.329094	0.526503	0.608925

Figure 3. 41: initial FFT data structure (left) and desired structure (right)

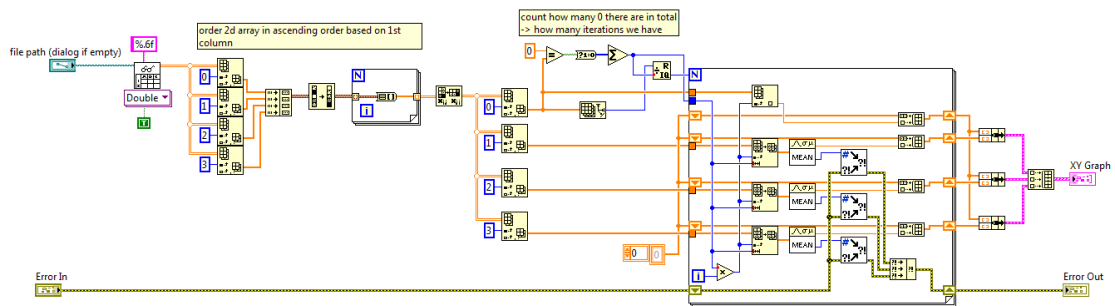


Figure 3. 42: “FFTGraphClient” subvi

Regarding the All-in-one plot, subvis “*unifiedLengthArrayClient*” and “*normalizeArrayClient*” ensure the correct display.

Subvi “*unifiedLengthArrayClient*” (Fig. 3.43) ensures that all data extracted from the different tsv files have the same length (by downsampling by means of calculating the mean value of the omitted elements), while subvi “*normalizeArrayClient*” (Fig. 3.44) normalizes the values, so that the possible value range is between 0 and 1 and the plots are drawn one on top of the other.

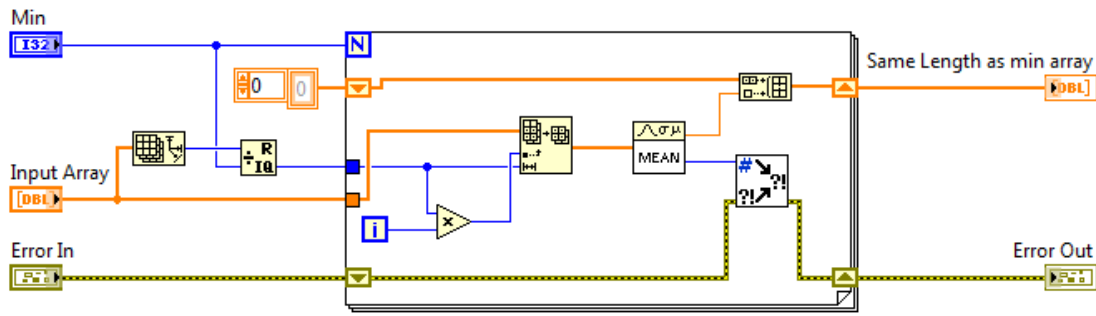


Figure 3. 43: “unifiedLengthArrayClient” subvi

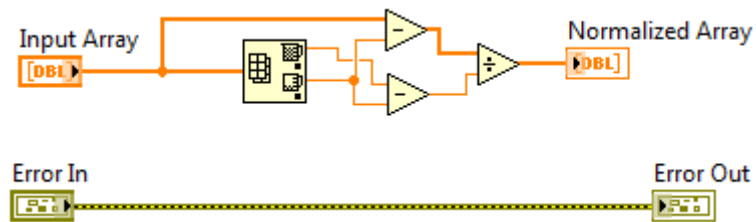


Figure 3. 44: “normalizeArrayClient” subvi

### 3.7 Matlab Post-Processing

National Instruments offers the ability to natively add text-based signal processing, analysis and math into the graphical development environment of LabVIEW with the add-on module “MathScript RT”. This module provides the ability to run commands directly written in the MathScript node or run existing .m files. The flexibility that MathScript RT introduces is enormous as it allows users to perform complex mathematical computations using existing .m files and perform analysis while they are acquiring data. An example of the MathScript node can be seen in Fig. 3.45, where the declared inputs of the node are found in the left side of the frame, the code in the middle and the declared output in the right side of the frame. [63]

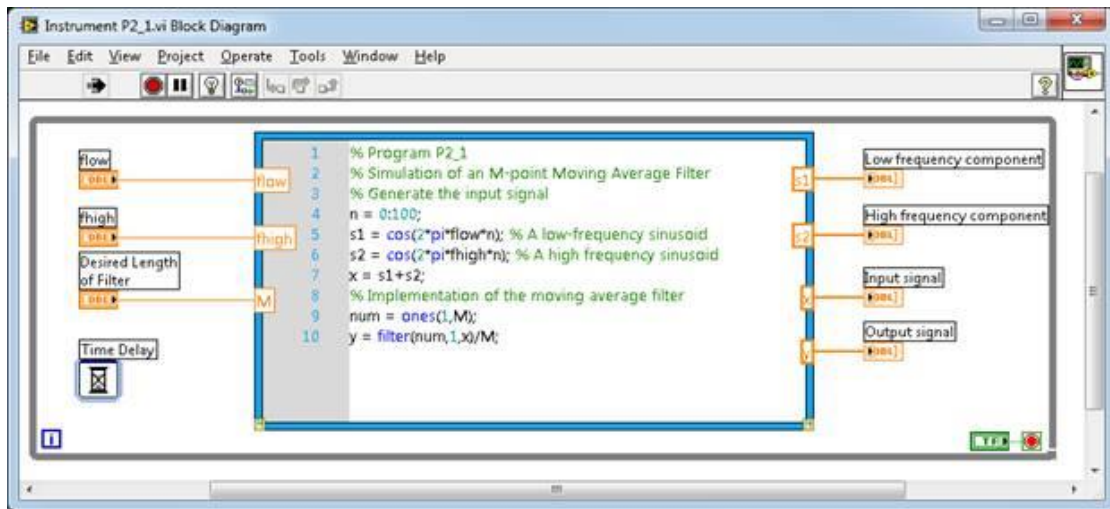


Figure 3. 45: MathScript node

That ability to use Matlab files directly is indispensable in our case, as it allows us to automatically calculate the creep and hysteresis attributes of interest by running “creep.m” and “hysteresis.m” Matlab scripts respectively (both of which are presented in Appendix A).

Specifically, we are able to **plot the hysteresis loop** and calculate the **peak-to-peak displacement, hysteresis percentage and hysteresis loop area**.

Regarding the creep effect, as explained in subchapter 1.5.2, the equation that describes the creep effect is:

$$L(t) = L_o \left[ 1 + \gamma \log_{10} \left( \frac{t}{0.1} \right) \right],$$

where  $L(t)$  is the actuator’s displacement for any fixed input voltage,  $L_o$  is a nominal constant displacement value which is the displacement of 0.1 s after applying the input voltage and  $\gamma$  is the creep factor that determines the rate of the logarithm.

The equation can be rewritten as:

$$L(t) = a + b * \log_{10} \left( \frac{t}{0.1} \right),$$

where coefficient  $b$  is called **creep coefficient**. Through “creep.m” Matlab script, we are able to calculate the creep coefficient for all different voltage levels of the creep test and for both negative and positive saturation (ascending voltage levels and descending respectively) and **plot the creep effect** in each case.

### 3.8 Report Generation

As previously mentioned, one of the options offered after the data logging has finished is the automatic report generation. This task is handled by “reportOptionsClient” subvi, which presents the user with the Front Panel seen in Fig. 3.46. All the possible panels are presented in the figure.

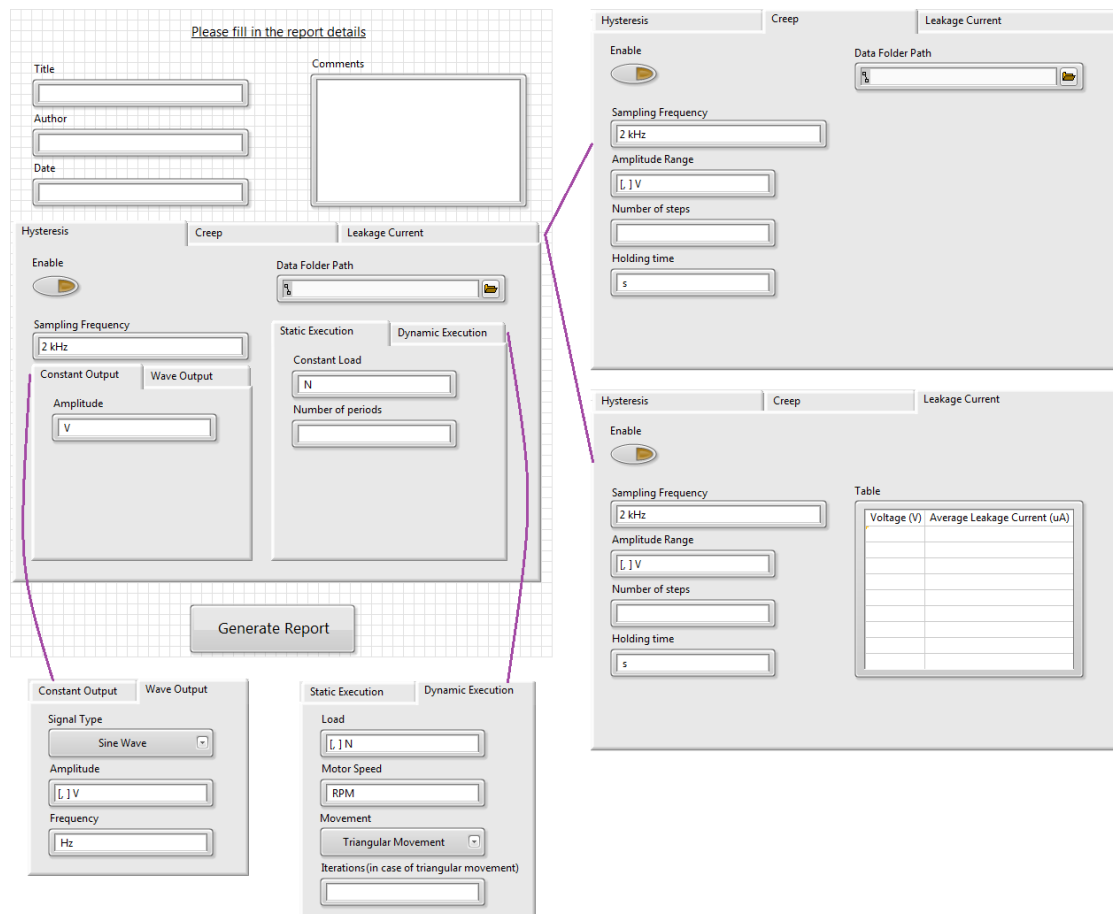


Figure 3. 46: Front Panel of “reportOptionsClient” subvi with all options shown

Currently, the subvi offers the ability to generate a Hysteresis, Creep or Leakage Current report although the list can be expanded in the future. The user can generate

any combination of the aforementioned types of report by pushing the respective “Enable” button found in each tab.

The user can fill in the “Title”, “Author” and “Date” fields of the report but in case they are left empty, the default values of “REPORT on Piezo Characterization”, “Bench User” and the system date and time at the instance the report was generated respectively are loaded. The comments box is optional. The user can then select the desired report types and fill in the parameter fields.

Once the “Generate Report” button is pressed, subvi “*reportGenerationClient*” is triggered with the defined report parameters. The subvi is shown in Fig. 3.47.

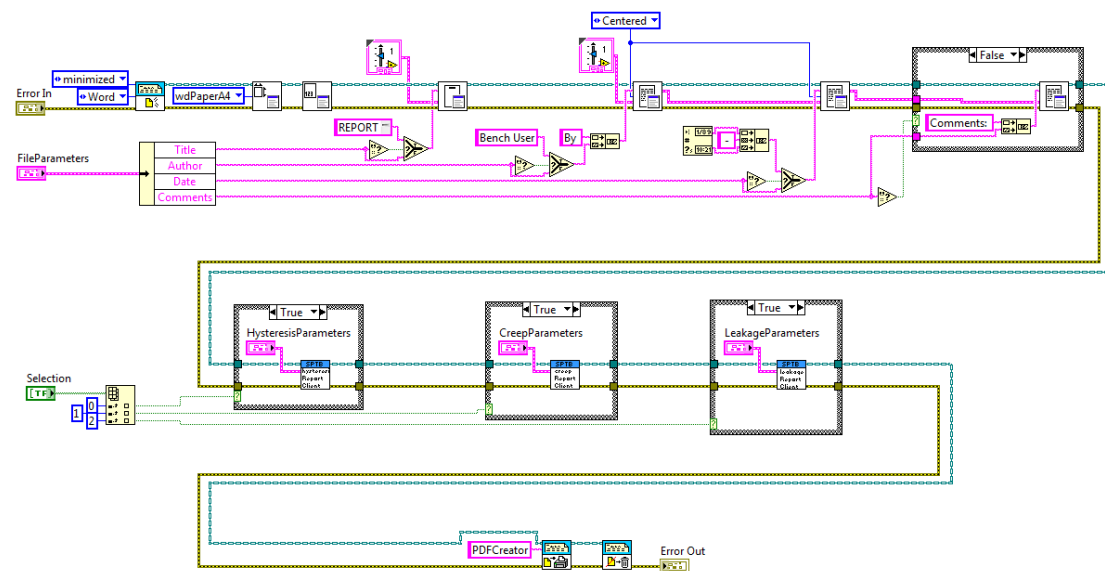


Figure 3.47: “*reportGenerationClient*” subvi

The subvi starts by creating a Word report, where the Title, Author and Date fields are placed on top. As previously mentioned, there are case selectors that load default values for these fields if they are left empty in the Front Panel. Another case selector recognizes if the comments box was left empty and in case it wasn’t, it adds the comments below the date. The following code segment recognizes which report types have been enabled and runs the respective subvis.

The subvi that handles the Hysteresis report is called “*hysteresisReportClient*” and can be seen in Fig. 3.48.

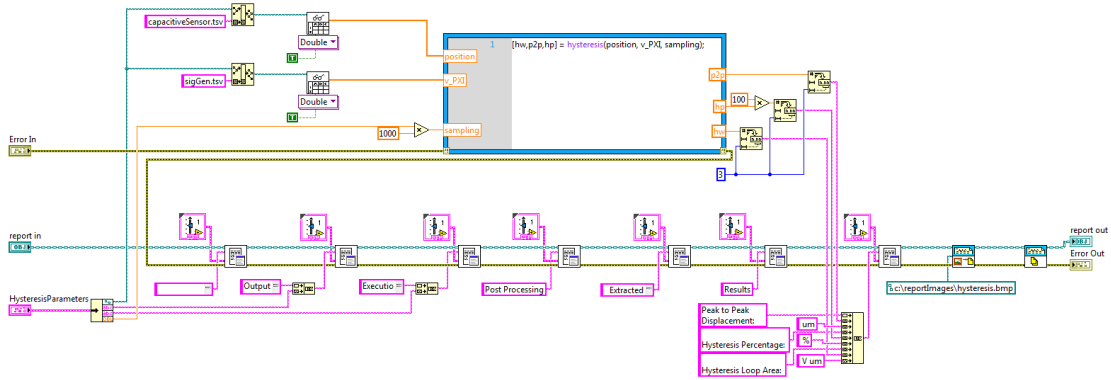


Figure 3. 48: “hysteresisReportClient” subvi

The subvi includes a MathScript node that runs “hysteresis.m” Matlab file on the data files that are located in the folder indicated in the “Data Folder Path” field of the report generation Front Panel (Fig. 3.46). The post-processing results and the corresponding plots are then appended to the report.

Two noteworthy points are that both “hysteresis.m” and “creep.m” Matlab files reside in the default MathScript directory, which is “\My Documents\LabVIEW Data”. The default MathScript directory can be changed from the LabVIEW Tools menu but in any case in order for the any Matlab file to be visible to MathScript, it has to be placed in the specified default directory. The other point is that plots generated by Matlab cannot be passed directly to LabVIEW. This is the reason why in both “hysteresis.m” and “creep.m” Matlab files the plots are stored in a specified disk location and are then loaded from that location and appended to the report.

The same applies to the Creep report, in which case the subvi that handles the Creep report is called “creepReportClient” and can be seen in Fig. 3.49.

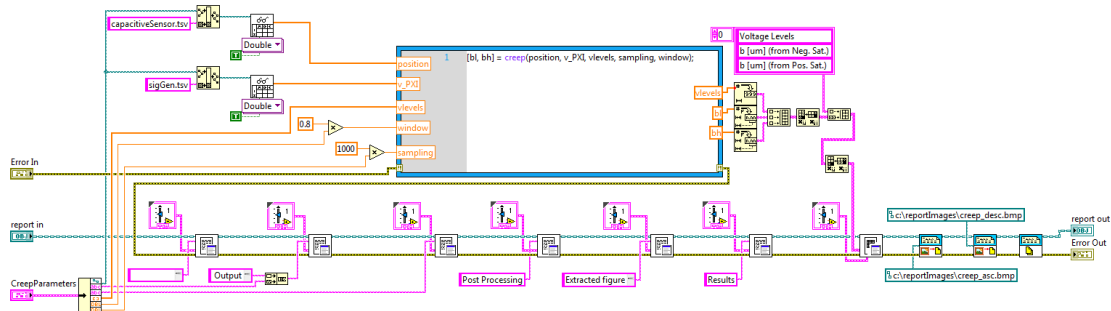


Figure 3. 49: “creepReportClient” subvi

In this subvi, the MathScript node runs “creep.m” Matlab file on the data files that are located in the folder indicated in the “Data Folder Path” field of the report generation Front Panel seen in Fig. 3.46. The post-processing results and the corresponding plots are then appended to the report.

Lastly, the Leakage Current report is handled by “*leakageCurrentReportClient*” subvi, which is illustrated in Fig. 3.50. The Leakage Current test provides a way to check whether the piezo actuator is in good working order or broken (even if it cannot be observed by the naked eye). By measuring the leakage current we are able to check non-destructively if the electrical properties of the piezo correspond to the manufacturer specifications or not. Usually the manufacturers provide a leakage current threshold for various voltage levels under which the actuator is considered to function properly.

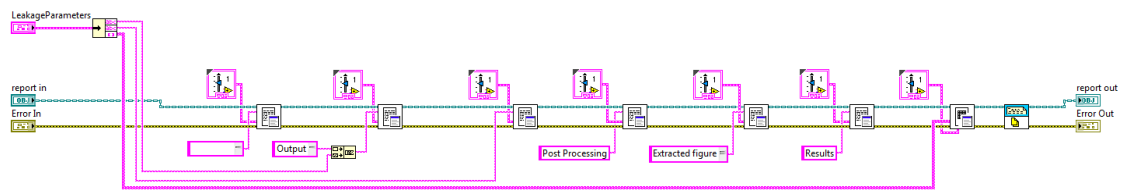


Figure 3. 50: “*leakageCurrentReportClient*” subvi

Compared to the Hysteresis and Creep report subvis, this subvi does not run any Matlab code. The leakage current is for the time being measured with external devices and the results of the test can be passed to the respective Table found in the “Leakage Current” tab of the report generation Front Panel (Fig. 3.46). So, in this case, the leakage current table is appended to the report.

Finally, the last stage of the “*reportGenerationClient*” subvi (Fig. 3.47) is the saving of the report to the hard disk. LabVIEW does not offer a way to save a report as a pdf file. As mentioned previously, the report in our case is a Word file and thus a way to convert it to pdf automatically within LabVIEW had to be found.

The solution that was eventually implemented was the utilization of a virtual printer. As can be seen in the code of the “*reportGenerationClient*” subvi in Fig. 3.47, after all selected elements have been appended to the report, the report is being sent to the printer named “PDFCreator”. After the report is printed as a pdf file, it is disposed as there is no need to save the Word file.



There are many virtual printers available online but in this case the requirement that defined the selection was the ability to automatically save the report without any pop-up window from the printer. Most virtual printers show up a pop-up window where the user can select the directory and other parameters of the document. To make the report generation process as automatic and uninterrupted as possible, a virtual printer that could automatically save the report to a predefined location was deemed necessary.

The virtual printer that fulfils all the requirements in this case is the “PDFCreator”. This is a virtual pdf printer that is available online for free and is set to automatically store the produced pdf files in a predefined location of the hard disk (“*My Documents*” in our case). The system date and time are passed as the name of the produced pdf file.



# 4

## *Results*

The purpose of this chapter is to present some indicative results of a typical piezo actuator characterization process. The tests performed are not exhaustive and do not cover all cases but they can be used to showcase the report and results generated by the software.

In this case, three different tests were performed on the same piezoelectric actuator: a hysteresis, a creep and a leakage current test. Regarding the first, it is a static-execution hysteresis test where the load is constant at 500N and the signal generator produces 10 periods of a 1Hz sine wave ranging from -20V to 150V. Regarding the creep test, the voltage range is [-20, 150] V with 5 steps (in-between voltage levels are 14V, 48V, 82V and 116V). The holding time (step duration) was set to 5s. Lastly, the leakage current test was performed on a Keithley 237 High-Voltage Source-Measure Unit. The leakage current was measured for two voltage levels – 5V and 150V. The sampling frequency of the specific device is 1 kHz according to the manufacturer.

The report and the results generated are attached in the following pages.

# **REPORT on Piezo Characterization**

By Bench User

13/05/2014 - 17:48

Comments: Keithley 237 High-Voltage Source-Measure Unit used for leakage current test.

Current threshold acceptance limit as proposed by the actuator manufacturer: <100uA at 150V, <3.3uA at 5V.

## **Measurement Description (Hysteresis Loop)**

Output Parameters: Sine Wave, Amplitude [-20, 150] V, Frequency 1 Hz

Execution Parameters: Static Execution, Constant Load 500 N, 10 periods, Sampling Frequency 2 kHz

## **Post Processing**

Extracted figures of merit: Peak to Peak Displacement, Hysteresis Percentage, Hysteresis Loop Area.

- o Peak to Peak Displacement: average the cycles and compute the difference between the maximum and minimum displacement values.

- o Hysteresis Percentage: on the averaged loop, compute the maximum difference of the displacement between the ascending and the descending branches for a given input voltage. Normalize it to the peak to peak displacement.

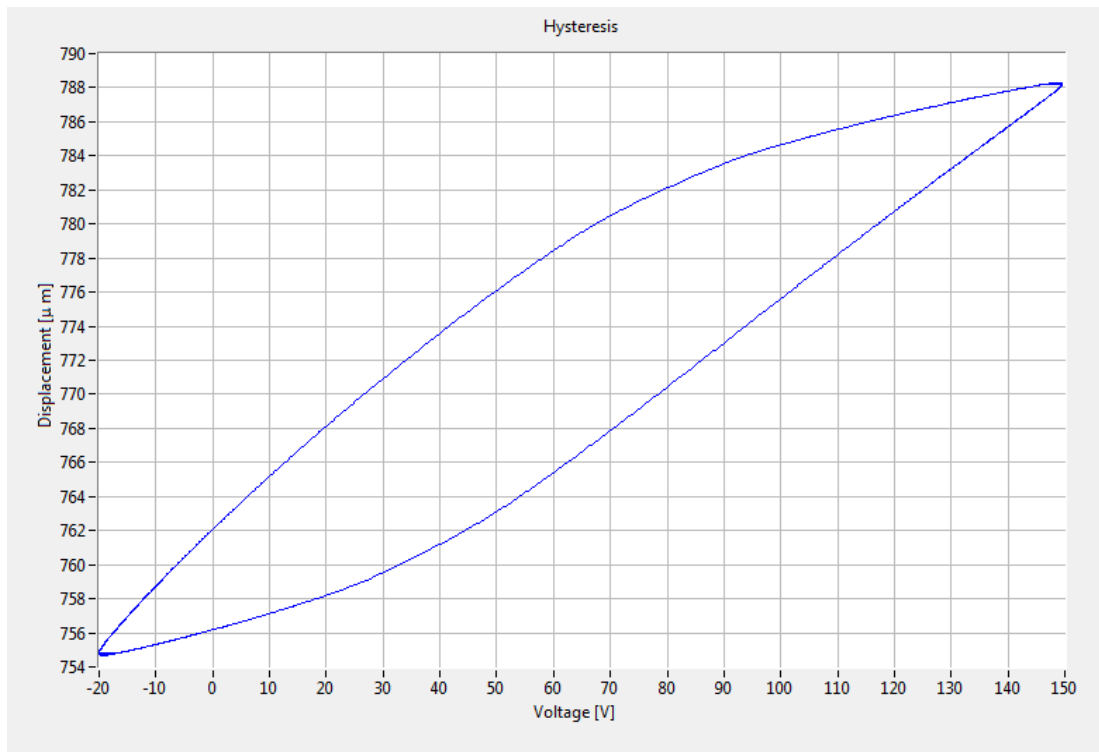
- o Hysteresis Loop Area: on the averaged loop, compute the area within the ascending and descending branches.

## **Results**

Peak to Peak Displacement: 33.648 um

Hysteresis Percentage: 38.879 %

Hysteresis Loop Area: 13.082 V um



## Measurement Description (Creep)

Output Parameters: Step-like waveform, Two sequences, Amplitude Range [-20, 150] V, 5 steps, Holding time 5 s

Sampling Frequency 2 kHz

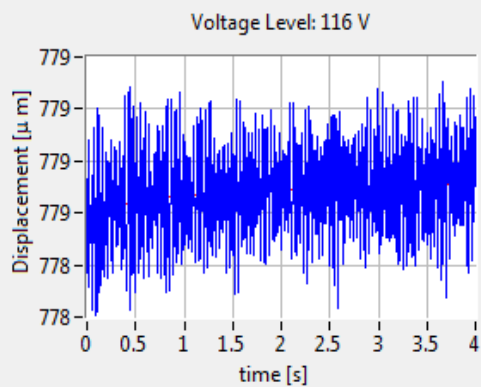
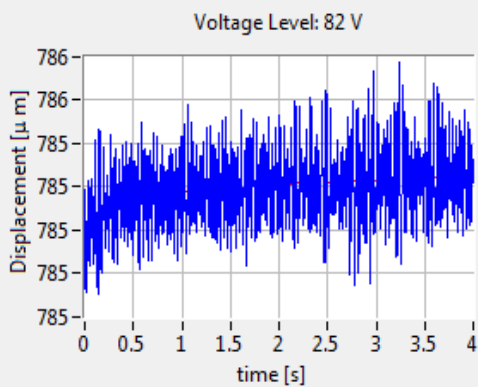
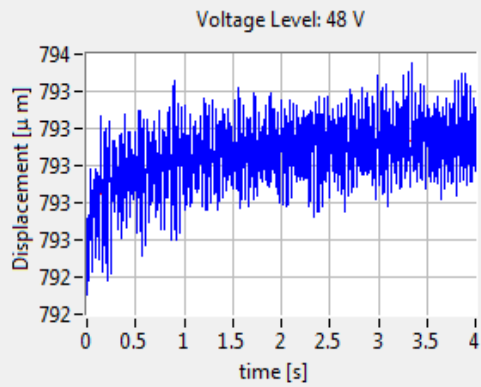
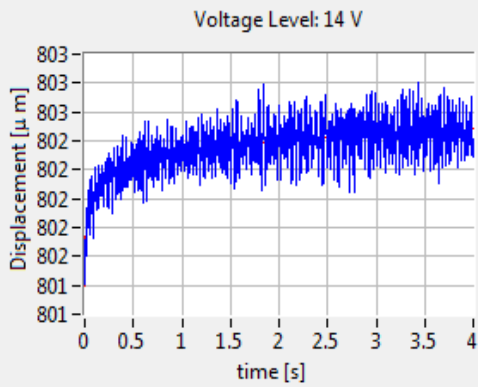
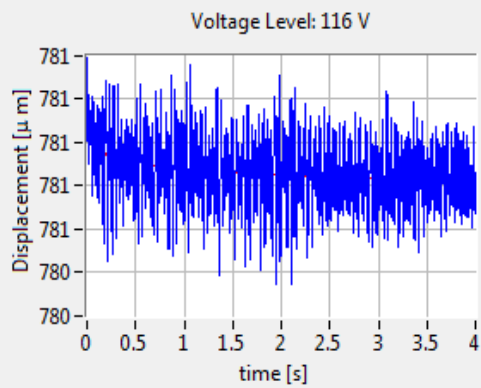
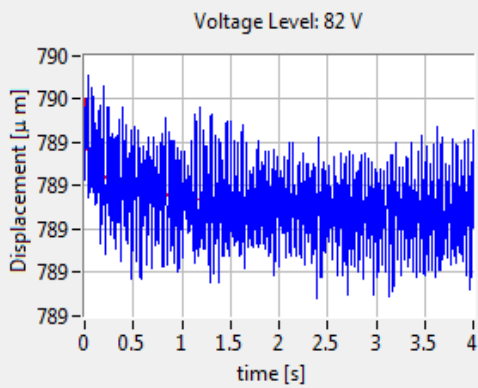
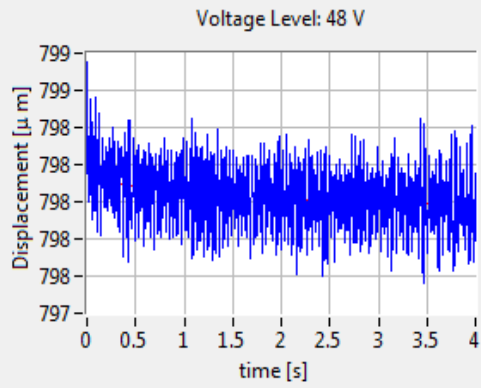
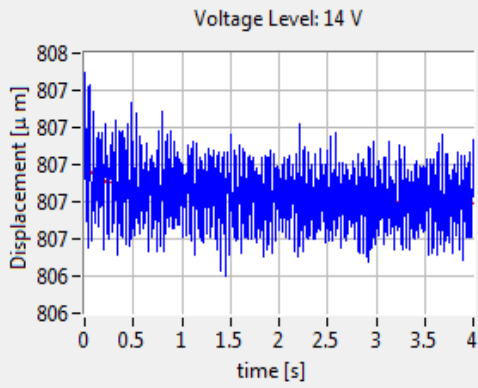
## Post Processing

Extracted figure of merit: Creep Coefficient – b (V).

It is obtained by a linear fit on the displacement acquired waveform over a logarithmic time scale.

## Results

Voltage Levels	14	48	82	116
b [um] (from Neg. Sat.)	-0.039625	-0.048713	-0.059391	-0.041306
b [um] (from Pos. Sat.)	0.151352	0.090981	0.048826	0.033962



## Measurement Description (Leakage Current)

Output Parameters: Amplitude Range 5V, 150V, 2 steps, Holding time 60 s

Sampling Frequency 1 kHz

## Post Processing

Extracted table of merit: Leakage current value.

It is obtained computing the average value of the steady state current during the holding time of the step-like input voltage waveform.

## Results

Voltage	Average Leakage Current
5V	40nA
150V	0.6uA



# 5

## *Conclusion and Future Work*

### *5.1 Conclusion*

The developed structure and accompanying LabVIEW code have successfully provided us with a test bench, which can perform reliable piezoelectric actuators' characterization and provide instant results and reports. The reliability has been verified against industrial piezoelectric actuators with known and recorded characteristics.

The test bench constitutes an indispensable tool which can be used to perform in-house characterization processes at any time. It can also be used in order to cross-check the characteristics specification sheet supplied by the piezo actuators' manufacturers or re-check previously measured actuators – for example after they have been irradiated, to check how their characteristics change with increasing levels of radiation.

### *5.2 Future Work*

The fact that the software is written in LabVIEW means that it can be easily modified and can be adapted to fit any kind of forthcoming need.

Two updates that would further expand the functionality of the test bench are the following. Firstly, the test bench could be modified to fully accommodate a Leakage Current test. That means that instead of performing the Leakage Current test on an external device and then manually passing the results on the Leakage Current Report Generation tab, the whole Leakage Current test could be undertaken by the test bench. If that proves too complex, a possibility would be to establish a connection to the external measuring device, so that the results are automatically passed.

The same applies for the Impedance test. The whole impedance test could be added as a function of the test bench or if that is too complicated, a connection to the measuring device making the transfer of the results automated, would suffice.

# 6

## *Bibliography*

- [1] “About CERN.” [Online]. Available: <http://home.web.cern.ch/about>. [Accessed: 04-Apr-2014].
- [2] “CERN,” *Wikipedia, the free encyclopedia*. 03-Apr-2014.
- [3] “Introduction of CERN, LHC and the ATLAS Experiment.” [Online]. Available: <http://zimmer.csufresno.edu/~yogao/ATLAS/CERN-LHC-ATLAS.html>. [Accessed: 04-Apr-2014].
- [4] “Large Electron–Positron Collider,” *Wikipedia, the free encyclopedia*. 02-Apr-2014.
- [5] “CERN LHC guide.” .
- [6] “The Large Hadron Collider.” [Online]. Available: <http://home.web.cern.ch/topics/large-hadron-collider>. [Accessed: 04-Apr-2014].
- [7] E. Koukovini-Platia, “Study of the  $\chi_{b0}$  meson via the decay  $\chi_{b0} \rightarrow J/\psi J/\psi \rightarrow \mu^+\mu^-\mu^+\mu^-$  in the ATLAS experiment,” National Technical University of Athens, School of Applied Mathematical and Physical Sciences, 2010.
- [8] “Large Hadron Collider,” *Wikipedia, the free encyclopedia*. 05-Apr-2014.
- [9] “Higgs boson,” *Wikipedia, the free encyclopedia*. 08-Apr-2014.
- [10] “The LHC: a look inside | Science in School.” [Online]. Available: <http://www.scienceinschool.org/2008/issue10/lhchow>. [Accessed: 08-Apr-2014].
- [11] “Pulling together: Superconducting electromagnets.” [Online]. Available: <http://home.web.cern.ch/about/engineering/pulling-together-superconducting-electromagnets>. [Accessed: 08-Apr-2014].
- [12] “Magnets.” [Online]. Available: <http://lhc-machine-outreach.web.cern.ch/lhc-machine-outreach/components/magnets.htm>. [Accessed: 08-Apr-2014].
- [13] “Radiofrequency cavities.” [Online]. Available: <http://home.web.cern.ch/about/engineering/radiofrequency-cavities>. [Accessed: 08-Apr-2014].
- [14] “LHC beam dumps.” [Online]. Available: <http://lhc-machine-outreach.web.cern.ch/lhc-machine-outreach/components/beam-dump.htm>. [Accessed: 08-Apr-2014].
- [15] “New beam dump for a veteran accelerator.” [Online]. Available: <http://home.web.cern.ch/about/updates/2013/10/new-beam-dump-veteran-accelerator>. [Accessed: 08-Apr-2014].

- [16]“LHC collimation.” [Online]. Available: [http://lhc-collimation-project.web.cern.ch/lhc%2Dcollimation%2Dproject/lhc\\_collimation\\_\\_introduction.htm](http://lhc-collimation-project.web.cern.ch/lhc%2Dcollimation%2Dproject/lhc_collimation__introduction.htm). [Accessed: 08-Apr-2014].
- [17]“Taking a closer look at LHC - LHC.” [Online]. Available: <http://www.lhc-closer.es/1/3/4/0>. [Accessed: 08-Apr-2014].
- [18]“The collimation system: defence against beam loss - CERN Courier.” [Online]. Available: <http://cerncourier.com/cws/article/cern/54384>. [Accessed: 08-Apr-2014].
- [19]T. Weiler, O. Aberle, R. Assmann, R. Chamizo, Y. Kadi, J. Lettry, R. Losito, and S. Redaelli, “LHC collimation system hardware commissioning,” in *IEEE Particle Accelerator Conference, 2007. PAC, 2007*, pp. 1625–1627.
- [20]“UA9.” [Online]. Available: <http://home.web.cern.ch/about/experiments/ua9>. [Accessed: 09-Apr-2014].
- [21]W. Scandale, “UA9 Report for 2013,” Oct. 2013.
- [22]S. Montesano, “Status of crystals for LHC,” presented at the LHC Collimation Working Group, CERN, 17-Feb-2014.
- [23]S. Montesano, “UA9 - Crystal collimation at CERN-SPS,” presented at the ICT, Trento, Italy, 11-Feb-2013.
- [24]“Piezoelectricity,” *Wikipedia, the free encyclopedia*. 07-Apr-2014.
- [25]N. Richardt, “An Introduction to the Piezo World,” Noliac, Denmark, Jun. 2007.
- [26]“Material characteristics | Hysteresis | Creep | Linearity | Thermal properties | Power dissipation.” [Online]. Available: [http://www.noliac.com/Material\\_characteristics\\_-143.aspx](http://www.noliac.com/Material_characteristics_-143.aspx). [Accessed: 10-Apr-2014].
- [27]A. Ledoux, “Theory of Piezoelectric Materials and Their Application in Civil Engineering,” MIT, Massachusetts Institute of Technology, Massachusetts, 2011.
- [28]M. S. Vijaya, *Piezoelectric materials and devices applications in engineering and medical sciences*. Boca Raton, FL: CRC Press, 2013.
- [29]H. Jung and D.-G. Gweon, “Creep characteristics of piezoelectric actuators,” *Rev. Sci. Instrum.*, vol. 71, no. 4, pp. 1896–1900, Apr. 2000.
- [30]“Fundamentals of Piezotechnology.” [Online]. Available: [http://www.piezo.ws/piezoelectric\\_actuator\\_tutorial/Piezo\\_Design\\_part2.php](http://www.piezo.ws/piezoelectric_actuator_tutorial/Piezo_Design_part2.php). [Accessed: 04-Apr-2014].
- [31]L. S. Zhang, Y. B. Liu, C. L. Pan, and Z. H. Feng, “Leakage current characterization and compensation for piezoelectric actuator with charge drive,” *Sens. Actuators Phys.*, vol. 199, pp. 116–122, Sep. 2013.
- [32]“Tutorial: Piezo Actuator Electrical Fundamentals, High Precision Piezo Motion, Nano Positioning Mechanics:” [Online]. Available: <http://www.physikinstrumente.com/en/products/prdetail.php?sortnr=400600.75>. [Accessed: 10-Apr-2014].
- [33]“Electrical impedance,” *Wikipedia, the free encyclopedia*. 06-Apr-2014.
- [34]“LabVIEW,” *Wikipedia, the free encyclopedia*. 05-Apr-2014.
- [35]“Visual programming language,” *Wikipedia, the free encyclopedia*. 10-Apr-2014.
- [36]“What Is LabVIEW? - National Instruments.” [Online]. Available: <http://www.ni.com/newsletter/51141/en/>. [Accessed: 11-Apr-2014].
- [37]“Data flow - LabVIEW Wiki.” [Online]. Available: [http://labviewwiki.org/Data\\_flow](http://labviewwiki.org/Data_flow). [Accessed: 11-Apr-2014].
- [38]“VI - LabVIEW Wiki.” [Online]. Available: <http://labviewwiki.org/VI>. [Accessed: 11-Apr-2014].
- [39]D. Audigier, C. Richard, C. Descamps, M. Troccaz, and L. Eyraud, “PZT uniaxial stress dependence: experimental results,” *Ferroelectrics*, vol. 154, no. 1, pp. 219–224, 1994.
- [40]M. W. Hooker, L. R. Center, and S. A. Wise, *Characterization of multilayer piezoelectric actuators for use in active isolation mounts*. Hampton, Va: National Aeronautics and Space Administration, Langley Research Center, 1997.
- [41]S. Mekid and B. Lim, “Characteristics comparison of piezoelectric actuators at low electric field: analysis of strain and blocking force,” *Smart Mater. Struct.*, vol. 13, no. 5, p. N93, Oct. 2004.

- [42] M. Badescu, S. Sherrit, X. Bao, J. Aldrich, Y. Bar-Cohen, and C. Jones, "Extended life PZT stack test fixture," 2008, pp. 692903–692903–7.
- [43] G. Yang, H. Gou, Y. Ji, Z. Yue, and L. Li, "Influence of uniaxial compressive stress on the ferroelectric behaviors of piezoelectric stack actuators," Mar. 2008.
- [44] M. Mitrovic, G. P. Carman, and F. K. Straub, "Response of piezoelectric stack actuators under combined electro-mechanical loading," *Int. J. Solids Struct.*, vol. 38, no. 24–25, pp. 4357–4374, Jun. 2001.
- [45] R. Yimnirun, S. Ananta, A. Ngamjarujana, and S. Wongsanmai, "Uniaxial stress dependence of ferroelectric properties of xPMN-(1-x)PZT ceramic systems," *Appl. Phys. A*, vol. 81, no. 6, pp. 1227–1231, Nov. 2005.
- [46] D. Zhou, M. Kamlah, and D. Munz, "Effects of uniaxial prestress on the ferroelectric hysteretic response of soft PZT," *J. Eur. Ceram. Soc.*, vol. 25, no. 4, pp. 425–432, Apr. 2005.
- [47] C. S. Lynch, "The effect of uniaxial stress on the electro-mechanical response of 8/65/35 PLZT," *Acta Mater.*, vol. 44, no. 10, pp. 4137–4148, Oct. 1996.
- [48] "Stainless steel," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [49] "Anodizing," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [50] "Series and parallel springs," *Wikipedia, the free encyclopedia*. 23-Mar-2014.
- [51] "Backlash (engineering)," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [52] "Torque limiter," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [53] "Resistance thermometer," *Wikipedia, the free encyclopedia*. 17-Apr-2014.
- [54] "PCI eXtensions for Instrumentation," *Wikipedia, the free encyclopedia*. 15-Apr-2014.
- [55] "What Is PXI? - National Instruments." [Online]. Available: <http://www.ni.com/pxi/whatis/>. [Accessed: 17-Apr-2014].
- [56] "What is a Real-Time Operating System (RTOS)? - National Instruments." [Online]. Available: <http://www.ni.com/white-paper/3938/en/>. [Accessed: 17-Apr-2014].
- [57] "NI Real-Time Hypervisor for Windows - National Instruments." [Online]. Available: <http://sine.ni.com/nips/cds/view/p/lang/en/nid/207302>. [Accessed: 17-Apr-2014].
- [58] *LabView Basics I Introduction Course Manual Course Software Version 8.0*. National Instruments, 2006.
- [59] *LabView Basics II Development Course Manual Course Software Version 8.0*. National Instruments, 2006.
- [60] "INI file," *Wikipedia, the free encyclopedia*. 03-May-2014.
- [61] "Tab-separated values," *Wikipedia, the free encyclopedia*. 03-May-2014.
- [62] "Fast Fourier transform," *Wikipedia, the free encyclopedia*. 05-May-2014.
- [63] "What is the NI LabVIEW MathScript RT Module? - National Instruments." [Online]. Available: <http://www.ni.com/white-paper/6206/en/>. [Accessed: 08-May-2014].



# Appendix A

## *Matlab Code*

### *A1. “Creep.m”*

```
function [bl,bh] = creep(position,v_PXI,vlevels,sampling>window)

v = v_PXI;
x = position;
Voltage_levels = vlevels;
t = 1:max(length(x),length(v));
t = t/sampling;
tolerance = 2;
dt = 0.1;

v_low_sat = v(1:floor(numel(v)/2));
x_low_sat = x(1:floor(numel(v)/2));
time_low_sat = t(1:floor(numel(v)/2));
v_high_sat = v(floor(numel(v)/2)+1:end);
x_high_sat = x(floor(numel(v)/2)+1:end);
time_high_sat = t(floor(numel(v)/2)+1:end);

figure('Position', [0 0 800 800]);
hold on;

for i = 1:numel(Voltage_levels)
    ndx = find(abs(v_low_sat-Voltage_levels(i)) < tolerance);
    ndx = ndx(1);
    onset_time(i) = time_low_sat(ndx);
    ndx = find(time_low_sat > onset_time(i) & time_low_sat <
onset_time(i) + window);

    t_red = time_low_sat(ndx)-onset_time(i);
    x_red = x_low_sat(ndx);
    p_low_sat(i,:) = polyfit(log(t_red/dt),x_red,1);

    subplot(ceil(numel(Voltage_levels)/2),2,i);
```

```

    hold on;
    plot(t_red,x_red);
    plot(t_red,p_low_sat(i,2)+p_low_sat(i,1)*log(t_red/dt),'r--');
    ylabel('Displacement [\mu m]');
    xlabel('time [s]');
    grid on;
    title(strcat('Voltage Level: ',num2str(Voltage_levels(i)), ' V'));
end

print('c:\reportImages\creep_asc.bmp');
close;

figure('Position', [0 0 800 800]);
hold on;

for i = 1:numel(Voltage_levels)
    ndx = find(abs(v_high_sat-Voltage_levels(i)) < tolerance);
    ndx = ndx(1);
    onset_time(i) = time_high_sat(ndx);
    ndx = find(time_high_sat > onset_time(i) & time_high_sat <
onset_time(i) + window);

    t_red = time_high_sat(ndx)-onset_time(i);
    x_red = x_high_sat(ndx);
    p_high_sat(i,:) = polyfit(log(t_red/dt),x_red,1);

    subplot(ceil(numel(Voltage_levels)/2),2,i);
    hold on;
    plot(t_red,x_red);
    plot(t_red,p_high_sat(i,2)+p_high_sat(i,1)*log(t_red/dt),'r--');
    ylabel('Displacement [\mu m]');
    xlabel('time [s]');
    grid on;
    title(strcat('Voltage Level: ',num2str(Voltage_levels(i)), ' V'));
end

print('c:\reportImages\creep_desc.bmp');
close;

bl = p_low_sat(:,1);
bh = p_high_sat(:,1);

```



## A2. “Hysteresis.m”

```
function [hw, p2p, hp] = hysteresis(position,v_PXI,sampling)

num_samples = min(length(position),length(v_PXI));
N= floor(num_samples/sampling);
ndx=1:1*sampling;
pos_avg = zeros(numel(ndx),1)';
v_PXI_avg = zeros(numel(ndx),1)';

for i = 2:(N)
    ndx = ((i-1)*sampling)+1:((i)*sampling);
    pos_avg = pos_avg + position(ndx);
    v_PXI_avg = v_PXI_avg + v_PXI(ndx);
end

pos_avg = pos_avg/(N-1);
v_PXI_avg = v_PXI_avg/(N-1);
[vmin,ndx] = min(v_PXI_avg);

v_PXI_avg = [v_PXI_avg(ndx+1:end) v_PXI_avg(1:ndx)];
pos_avg = [pos_avg(ndx+1:end) pos_avg(1:ndx)];

[vmin,ndx] = max(v_PXI_avg);
v_PXI_avg_ascending = v_PXI_avg(1:ndx-1);
v_PXI_avg_descending = v_PXI_avg(ndx:end);
pos_avg_ascending = pos_avg(1:ndx-1);
pos_avg_descending = pos_avg(ndx:end);

minv = min(v_PXI_avg_ascending);
maxv = max(v_PXI_avg_descending);
v_resampled = linspace(minv,maxv,1000);

[v_PXI_avg_ascending,I,J] = unique(v_PXI_avg_ascending);
pos_avg_ascending = pos_avg_ascending(I);
[pos_avg_ascending,I,J] = unique(pos_avg_ascending);
v_PXI_avg_ascending = v_PXI_avg_ascending(I);
[v_PXI_avg_descending,I,J] = unique(v_PXI_avg_descending);
pos_avg_descending = pos_avg_descending(I);
[pos_avg_descending,I,J] = unique(pos_avg_descending);
v_PXI_avg_descending = v_PXI_avg_descending(I);

pos_asc_res = interpolateld(v_PXI_avg_ascending, pos_avg_ascending,
v_resampled, 'linear');
pos_desc_res =
interpolateld(v_PXI_avg_descending,pos_avg_descending,v_resampled,
'linear');

[value,ndx]=max(pos_desc_res-pos_asc_res);

hysteresis_width = abs(pos_desc_res(ndx)-pos_asc_res(ndx));
peak2peak = max(pos_avg)-min(pos_avg);
hysteresis_percentage_PXI = hysteresis_width/peak2peak;

disp('Hysteresis Percentage, PXI voltage:');
disp(hysteresis_percentage_PXI);

disp('Hysteresis peak2peak:');
disp(peak2peak);
```

```

areatot=polyarea([v_resampled rot90(v_resampled,2)]*20,-[pos_asc_res
rot90(pos_desc_res,2)]); %V*mu_m
disp('Hysteresis Area:');
disp(areatot);

figure('Position', [0 0 800 600]);
plot([v_PXI_avg_ascending v_PXI_avg_descending(end:-
1:1)], [pos_avg_ascending pos_avg_descending(end:-1:1)]);
xlabel('Voltage [V]');
ylabel('Displacement [\mu m]');
grid on;
title('Hysteresis');
print('c:\reportImages\hysteresis.bmp');
close

hw = hysteresis_width;
p2p = peak2peak;
hp = hysteresis_percentage_PXI;

```

# Appendix B

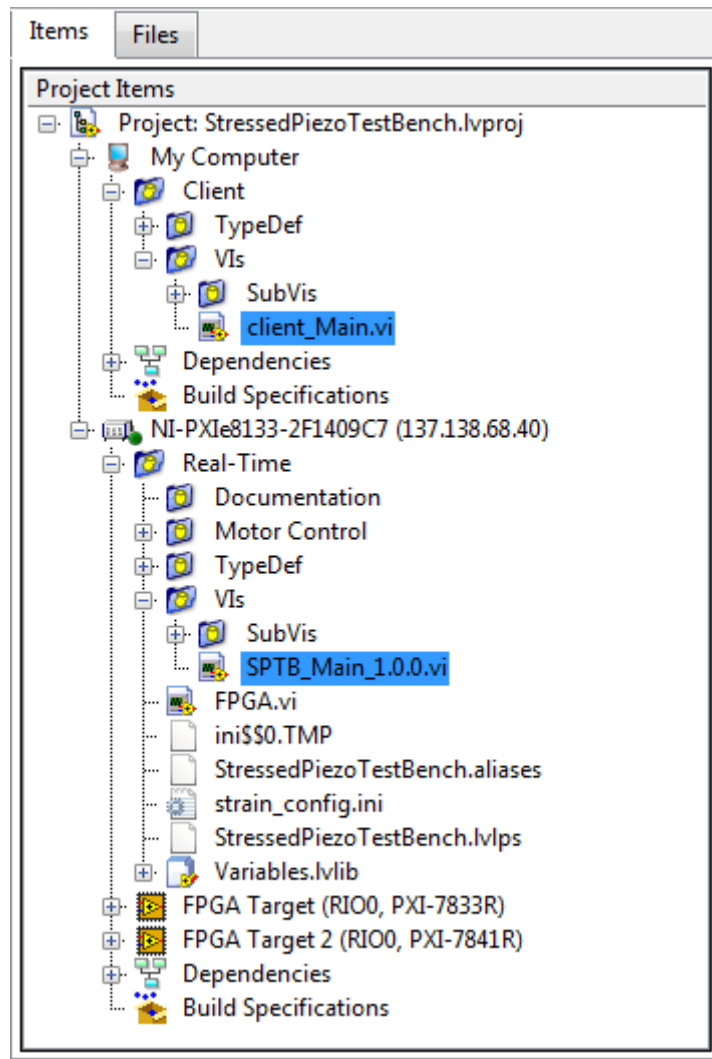
## *LabVIEW Project Instructions*

The work presented was carried out using National Instrument's LabVIEW 2011 Service Pack 1 edition software.

The LabVIEW Project file, which opens the project, is located in the "Real-Time" folder and is called "StressedPiezoTestbench.lvproj".

Inside the project, the server application, named "SPTB\_Main\_1.0.0.vi", is located under "NI-PXIE8133-F1409C7\Real-Time\VIs\". The client application, named "client\_Main.vi", is located under "My Computer\Client\VIs\". Both of them are highlighted in the following figure.

The Matlab files ("creep.m" and "hysteresis.m"), as mentioned, must be placed in the default MathScript directory in order for them to work properly. They are located in "Support Files\Matlab scripts\".



Lastly, there have been reports of incompatibilities of the National Instruments 'Report Generation Toolkit For Microsoft Office' version 11.0.0 with Microsoft Office editions newer than 2003. Specifically, the "Append Table to Report.vi" does not function properly and the problem is that column labels are printing where the data should be. This happens because recent Word editions treat cell separators while pasting table contents in a different way than previously. The report generation toolkit uses line feed (LF, \n) to separate both columns and rows, filling one cell at a time, from left to right, and then continuing with the next row. However, newer Word editions expect a tab (\t) character to separate each column and then a line feed to continue on the next row.

A workaround to this problem is to replace the original "Word\_Update\_Table.vi" and "Word\_Insert\_Table.vi" files with the modified versions located in "Support Files\Word Array Fix\". The original files can be accessed in "\National

Instruments\LabVIEW 2011\vi.lib\addons\\_office\\_wordsub.llb” inside the installation directory (usually “C:\Program Files”).