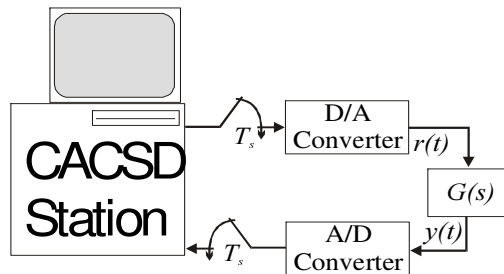


ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ
ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ & ΕΛΕΓΧΟΥ

Εργαστηριακές Ασκήσεις Συστημάτων Αυτομάτου Ελέγχου II



ΑΝΤΩΝΙΟΣ ΤΖΕΣ
Αναπληρωτής Καθηγητής

ΓΕΩΡΓΙΟΣ ΝΙΚΟΛΑΚΟΠΟΥΛΟΣ
Διπλ/χος Ηλ/γος Μηχ/κός

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
ΕΚΤΥΠΩΤΙΚΟ
ΚΕΝΤΡΟ
2002

©2002 Απαγορεύεται η ολική ή μερική αναπαραγωγή με οποιονδήποτε ηλεκτρονικό, μηχανικό ή άλλο οπτικοακουστικό τρόπο χωρίς την άδεια των συγγραφέων

Οδηγίες Εργαστηρίου

Προφυλάξεις / Κανονισμοί

Οι παρακάτω προφυλάξεις και κανόνες ασφαλείας, θα πρέπει να τηρηθούν επιμελώς με σκοπό την σωστή διεξαγωγή των εργαστηριακών ασκήσεων, την αποφυγή δυσλειτουργιών και ατυχημάτων.

- 1) Κατά την διάρκεια των εργαστηριακών ασκήσεων δεν επιτρέπεται η αλλαγή των συνδεσμολογιών που έχουν γίνει ήδη.
- 2) Δεν επιτρέπεται η μετακίνηση των εργαστηριακών διατάξεων.
- 3) Κάθε εργαστηριακή ομάδα είναι υπεύθυνη για την πειραματική διάταξη που της ανήκει. Σε περίπτωση που προκληθεί κάποια βλάβη σε αυτήν, υπεύθυνοι για αυτή είναι μόνο τα μέλη της αντίστοιχης ομάδας.
- 4) Σε περίπτωση που νομίζετε ότι δεν είσαστε σίγουροι για κάποια λειτουργία ή νομίζετε ότι έχει προκληθεί κάποια βλάβη θα πρέπει αμέσως να φωνάζετε κάποιον υπεύθυνο.

Βαθμολόγηση του Εργαστηρίου

Ο συνολικός βαθμός του εργαστηρίου, προκύπτει από την επιμέρους βαθμολόγηση των εργαστηριακών αναφορών και της τελικής εργαστηριακής εξέτασης. Ο βαθμός αυτός είναι ομαδικός και επιτρέπεται διακύμανση ενός βαθμού, προς τα πάνω ή προς τα κάτω, ανάμεσα στα μέλη της ομάδας, ως αποτέλεσμα εκτιμήσεως προσωπικής επιμέρους εργασίας. Ο βαθμός των αναφορών καλύπτει το 70% του συνολικού βαθμού, ενώ ο βαθμός κατά την τελική εξέταση εργαστηρίου καλύπτει το υπόλοιπο 30%. Στην εξέταση του εργαστηρίου, η ομάδα θα εξετάζεται σαν σύνολο, αφού πρώτα της αποδοθούν τρία διαφορετικά θέματα, που θα πρέπει να τα

ολοκληρώσει μέσα στον προβλεπόμενο χρόνο. Για την ολοκλήρωση αυτών, θα έχουν στην διάθεση τους τρεις υπολογιστές.

Όλες οι εργαστηριακές αναφορές, θα πρέπει να παραδίνονται 7 ημερολογιακές ημέρες μετά από την εκτέλεση της αντίστοιχης εργαστηριακής ασκήσεως. Στην αντίθετη περίπτωση που γίνει υπέρβαση του χρονικού ορίου, ο βαθμός της άσκησης θα μειώνεται (20% ανά ημέρα καθυστέρησης). Όλες οι προαιρετικές εργαστηριακές ασκήσεις προσυπολογίζονται θετικά και ενισχυτικά ως προς τον τελικό βαθμό. Κατά την παράδοση των αναφορών (σε έντυπη και ηλεκτρονική μορφή), οι φοιτητές θα πρέπει να καταθέτουν και τα δεδομένα που πήραν από την αντίστοιχη εργαστηριακή άσκηση, σε ηλεκτρονική μορφή. Κάθε πακέτο δεδομένων, είναι ξεχωριστό για κάθε Εργαστηριακή Ομάδα και ελέγχεται η γνησιότητα αυτού με ηλεκτρονική κωδικοποίηση.

Περιεχόμενα

1. Εισαγωγή σε Πληροφορικά μετρητικά συστήματα	5
2. Σχεδιάσμος μετρητικών εικονικών οργάνων.....	17
3. Αναγνώριση συνάρτησης μεταφοράς γραμμικού συστήματος..	32
4. Σχεδιασμός ON-OFF, P & PID ελεγκτών.....	44
5. Προγραμματισμός και υλοποίηση ελεγκτή κέρδους.....	54
6. Σχεδιασμός P ελεγκτή με Matlab.....	62
7. Τηλε-πειραματισμός με έναν P ελεγκτή.....	66

Περίληψη

Στο κεφάλαιο 1, θα γίνει μια σύντομη εισαγωγή στο Labview και των εργαλείων του. Στο κεφάλαιο 2, θα παρουσιαστεί ο τρόπος σχεδιασμού απλών εικονικών οργάνων, που σαν σκοπό έχουν την εξοικείωση του χρήστη με τον γραφικό τρόπο προγραμματισμού. Στο κεφάλαιο 3, θα παρουσιαστεί η 2^η Εργαστηριακή Άσκηση με σκοπό την αναγνώριση ενός συστήματος, από μετρήσεις της εισόδου και εξόδου του. Στο κεφάλαιο 4, θα παρουσιαστεί η 3^η Εργαστηριακή άσκηση με σκοπό την χρήση απλών ελεγκτών (On/Off, P, PID) για τον έλεγχο Συστημάτων Αυτομάτου Ελέγχου (ΣΑΕ). Στο κεφάλαιο 5, θα παρουσιαστεί η 4^η Εργαστηριακή Άσκηση με σκοπό τον πλήρη σχεδιασμό ενός P ελεγκτού στο LabView. Στο κεφάλαιο 6, θα παρουσιαστεί η 5^η Εργαστηριακή Άσκηση με σκοπό τον σχεδιασμό ενός PI ελεγκτού στο LabView με την βοήθεια του προγράμματος Matlab της Mathworks, ενώ τέλος στο κεφάλαιο 7, θα πραγματοποιηθεί μια πειραματική άσκηση με σκοπό τον τηλε-έλεγχο πειραματικών διατάξεων με την χρήση ενός P ελεγκτή, μέσω του διαδικτύου.

1 Εισαγωγή σε Πληροφοριακά Μετρητικά Συστήματα

Πληροφοριακά μετρητικά συστήματα, όπως το LabView (Laboratory Virtual Instrument Engineering Workbench), είναι προγραμματιστικά εργαλεία τα οποία βασίζονται στην σύζευξη της πληροφορικής και του hardware για την δημιουργία εικονικών μετρητικών οργάνων. Το τυπικό περιβάλλον εργασίας είναι αντικειμενοστρεφές και βασίζεται στον γραφικό προγραμματισμό (Graphical Programming). Το LabView (προϊόν της National Instruments) είναι ο κύριος εκπρόσωπος αυτής της κατηγορίας και η γραφική του φύση, το κάνει ιδανικό για εφαρμογές μετρήσεων, αυτοματισμού, ελέγχου οργάνων και ανάλυση δεδομένων.

Κατά την παρουσίαση του Εργαστηρίου Συστημάτων Αυτομάτου Ελέγχου (ΣΑΕ) II, οι ελεγκτές που θα χρησιμοποιηθούν και αναπτυχθούν στις εργαστηριακές ασκήσεις θα εφαρμοστούν σε πραγματικές πειραματικές διατάξεις ενώ η θεωρία που θα χρησιμοποιηθεί, καλύπτει το 90% της σημερινής τεχνολογίας που χρησιμοποιεί η βιομηχανία. Για περισσότερες πληροφορίες, οδηγίες χρήσεως και on-line help σχετικά με τα προγράμματα LabView και Matlab καλείστε να επισκεφτείτε τα παρακάτω αντίστοιχα sites www.ni.com και www.mathworks.com

1.2 Τα εικονικά όργανα

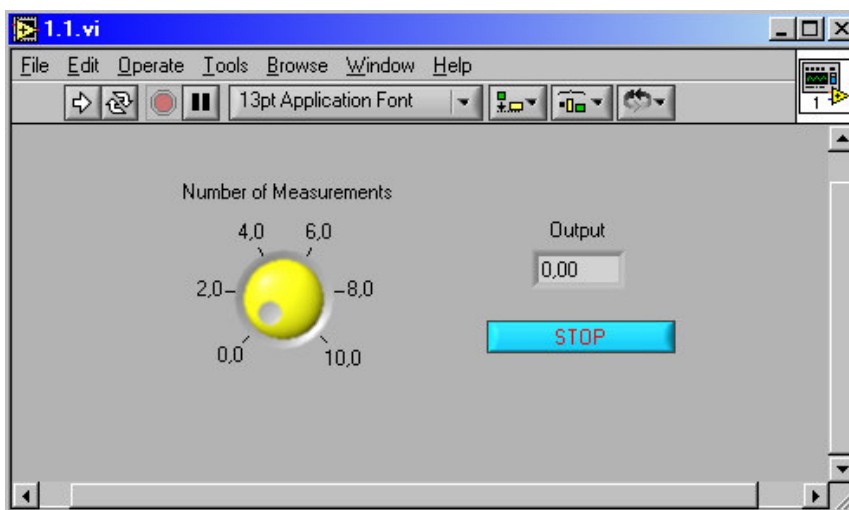
Τα εικονικά όργανα (Virtual Instruments, VI) είναι στοιχεία προγραμματισμού του LabView, τα οποία αποτελούνται από το *front panel* και το *block diagram*. Το *front panel* χρησιμοποιείται για την απεικόνιση στοιχείων τύπου *controls* και

indicators για τον χρήστη, ενώ στο *block diagram* περιέχεται ο κώδικας για το εικονικό όργανο. Μέσω του *front panel* και των στοιχείων *controls* οι χρήστες μπορούν να θέσουν τιμές στις μεταβλητές να επιλέξουν διαδικασίες και γενικά να ελέγξουν την ροή και τον τρόπο εκτέλεσης του προγράμματος. Μέσω των στοιχείων *indicators* οι χρήστες μπορούν να παρατηρήσουν την μεταβολή των μεταβλητών, τα αποτελέσματα διεργασιών και γενικά όλη την πληροφορία που παράγεται κατά την εκτέλεση του γραφικού κώδικα στο *block diagram*.

Οι κλασσικές γλώσσες προγραμματισμού (C, Basic) χρησιμοποιούν συναρτήσεις και υπορουτίνες για την ανάπτυξη των εφαρμογών τους. Στο περιβάλλον του Labview οι αντίστοιχες οντότητες ονομάζονται εικονικά όργανα (Virtual Instruments). Πολλαπλά VI μπορούν να χρησιμοποιηθούν για την δημιουργία εφαρμογών μεγάλης κλίμακας., Αντίστοιχα ένα VI μπορεί να χρησιμοποιηθεί για την επικοινωνία με τον χρήστη ή σαν μια υπορουτίνα σε κάποια εφαρμογή.



1.3 To Front Panel

Στην εικόνα 1.1 απεικονίζεται το front panel ενός εικονικού οργάνου (VI) του Labview. Περιέχει ένα περιστροφικό διακόπτη (knob) για την επιλογή των μετρήσεων, ένα αντικείμενο (indicator) για την απεικόνιση της μετρούμενης τιμής και ένα κουμπί για stop. Front Panel τέτοιου είδους μπορεί να δημιουργηθούν χωρίς ιδιαίτερη προσπάθεια και σε μικρό χρονικό διάστημα.

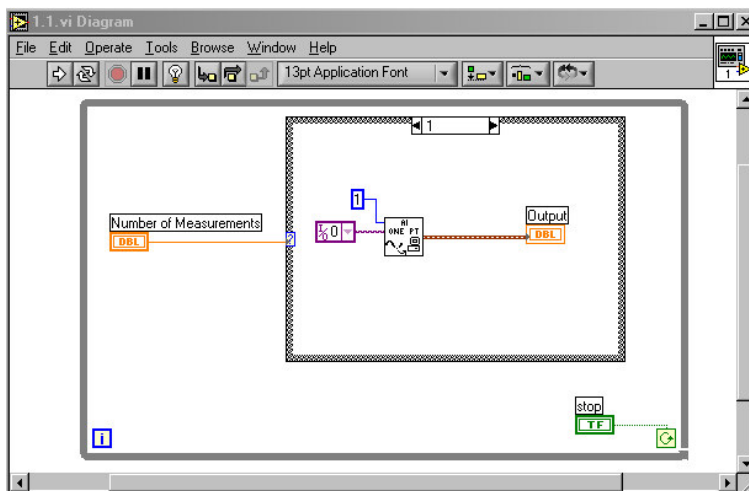


Εικόνα 1.1: Front Panel εικονικού οργάνου

1.4 To Block Diagram


Στην εικόνα 1.2 απεικονίζεται το δομικό διάγραμμα, ή ο πηγαίος κώδικας, της εφαρμογής του Σχήματος 1.1. Τα VIs προγραμματίζονται στην γλώσσα-G όπου το πρόγραμμα γίνεται compile από την μηχανή εκτέλεσης του Labview. Ακολούθως τα VIs μετατρέπονται σε εκτελέσιμο κώδικα που μπορεί να εκτελεστεί από την μηχανή διαδικασιών του Labview. Κάθε φορά που μια αλλαγή συμβαίνει στο VI, το Labview ενημερώνει κατάλληλα και τον πηγαίο κώδικα. Αν το δομικό διάγραμμα δεν είναι σωστά σχεδιασμένο τότε στο σημείο που υπάρχει το πρόβλημα, το καλώδιο σύνδεσης των δύο δομικών στοιχείων, είναι διακεκομμένο, καθώς επίσης το βελάκι εκτέλεσης έχει την ακόλουθη μορφή . Στην αντίθετη περίπτωση, το βελάκι έχει την ακόλουθη μορφή  και ο χρήστης είναι έτοιμος να εκτελέσει το VI. Τα δομικά διαγράμματα που σχεδιάζουμε, δεν καθορίζουν και την σειρά με την οποία τα στοιχεία που χρησιμοποιούμε θα εκτελεστούν. Μπορούμε να υποχρεώσουμε μια σειρά εκτέλεσης, με το να απαιτήσουμε κάποιο στοιχείο να λάβει δεδομένα από κάποιο άλλο, πριν από την εκτέλεση του. Επιπλέον χρησιμοποιώντας πιο πολύπλοκες προγραμματιστικές δομές, μπορούμε να

καθορίσουμε με ακρίβεια τον τρόπο εκτέλεσης των διαδικασιών. Πιο γενικά κατά την διάρκεια του προγραμματισμού, θα είναι εμφανές ότι ορισμένες λειτουργίες θα πρέπει να πραγματοποιηθούν, πριν από κάποιες άλλες. Είναι ευθύνη του προγραμματιστή να επιλέξει τον τρόπο και να εξασφαλίσει την σωστή αλληλουχία κατά την εκτέλεση των διαδικασιών.



Εικόνα 1.2: Block Diagram του εικονικού οργάνου του σχήματος 1.1

1.5 Εκτέλεση Virtual Instruments

Ένα πρόγραμμα στο Labview, εκτελείται με το πάτημα του βέλους ή το *Run Button*  τα οποία βρίσκονται στην παλέττα, στο πάνω μέρος του παραθύρου.

Κατά την διάρκεια της εκτέλεσης το *Run Button* αλλάζει σε ένα μαύρο χρώμα .


Θα πρέπει να σημειώσουμε πως κατά την διάρκεια της λειτουργίας τα κουμπιά που εμφανίζονται στην παλέττα είναι το *Continuous Run*, *Stop* και *Pause buttons*



Τα τρία τελευταία κουμπιά  χρησιμοποιούνται για την

στοίχιση των αντικειμένων που εισάγουμε. Τα VIs συνήθως εκτελούνται από τα Front Panels, ενώ έχουν την δυνατότητα να εκτελεστούν και από τα Block Diagrams. Αυτό επιτρέπει στον προγραμματιστή να τρέξει το πρόγραμμα και να χρησιμοποιήσει κάποια άλλα εργαλεία που θα αναφερθούν στη συνέχεια και που είναι χρήσιμα κατά την διαδικασία του Debugging.

Αν το βέλος *Run* εμφανιστεί σαν σπασμένο βέλος, τότε αυτό υποδεικνύει ότι το πρόγραμμα δεν μπορεί να εκτελεστεί, λόγω κάποιων λαθών. Όταν όλα τα λάθη έχουν διορθωθεί, το σπασμένο βέλος θα αντικατασταθεί από το κανονικό βέλος που υποδηλώνει ότι το πρόγραμμα είναι έτοιμο για λειτουργία. Κατά την διάρκεια της ανάπτυξης, θα παρατηρήσουμε ότι στην παλέττα απεικονίζεται το σπασμένο βέλος. Αν η ένδειξη αυτή παραμείνει και μετά την ολοκλήρωση του προγραμματισμού, τότε με το πάτημα του σπασμένου βέλους, μπορούμε να δούμε μια λίστα με όλα τα λάθη που έχουν παρουσιαστεί κατά την διάρκεια της ανάπτυξης. Στρατηγικές για την διόρθωση των σφαλμάτων θα παρουσιαστούν σε επόμενο κεφάλαιο.

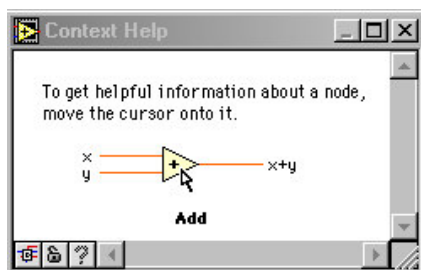
Η παλέττα περιέχει ακόμα τέσσερα κουμπιά  στο δομικό διάγραμμα τα οποία δεν είναι διαθέσιμα στο Front Panel και χρησιμοποιούνται για το debugging μιας εφαρμογής. Το κουμπί με την λάμπα, είναι για βηματική εκτέλεση του προγράμματος, ενώ τα τρία επόμενα είναι για την εκτέλεση βημάτων μέσα στον κώδικα.

1.6 Αρχαιοθέτηση στο LabView

Το πρόγραμμα Labview, χρησιμοποιεί για την αποθήκευση αρχείων την ονοματολογία filename.vi. Επιπλέον πολλαπλά VIs μπορούν να σωθούν σε ένα αρχείο με την μορφή βιβλιοθήκης και με κατάληξη *.lib. Οι βιβλιοθήκες, είναι χρήσιμες για την οργάνωση πολλαπλών συσχετιζομένων VIs. Όταν καλούμε κάποιο VI που εμπεριέχει και κάποια άλλα sub-VIs το πρόγραμμα προσπαθεί να τα εντοπίσει γρήγορα. Η χρήση βιβλιοθήκης, απλοποιεί την όλη διαδικασία, τοποθετώντας όλα τα σχετικά VIs κάτω από το ίδιο directory. Στην περίπτωση που θέλουμε να δημιουργήσουμε μία βιβλιοθήκη με VIs, θα πρέπει να επιλέξουμε την επιλογή Save with Options, από το File Menu.

1.7 Η χρήση βοήθειας στο Labview

Το LabView, παρέχει έναν on-line οδηγό παροχής βοήθειας για όποια λειτουργία θέλουμε να πραγματοποιήσουμε. Το κάθε εικονόργανο, μπορεί να χρειάζεται έναν μεγάλο αριθμό και διαφορετικά είδη συνδέσεων. Ο αρχάριος χρήστης θα πρέπει να χρησιμοποιεί την δυνατότητα για on-line help. Η λειτουργία αυτή ενεργοποιείται από το Help Menu επιλέγοντας Show Context Help. Η ενεργοποίηση της λειτουργίας αυτής, εμφανίζει ένα παράθυρο που εμπεριέχει όλες τις απαραίτητες πληροφορίες για τον τρόπο διασύνδεσης του εικονικού οργάνου. Αφήνοντας το mouse pointer ακίνητο πάνω από ένα εικονικό όργανο, το παράθυρο με τις οδηγίες θα απεικονίσει τις σχετικές με αυτό το όργανο πληροφορίες. Ένα τυπικό παράθυρο βοήθειας απεικονίζεται στο Σχήμα 1.3. Η χρήση του είναι σχεδόν αναγκαία με σκοπό την εύκολη και επιτυχή διασύνδεση και θα πρέπει σχεδόν πάντα να ενεργοποιείται.



Σχήμα 1.3: Το παράθυρο βοήθειας

1.8 Οι παλέτες εργασίας στο LabView

Το LabView περιέχει γραφικές παλέτες εργασίας σε μορφή menu με σκοπό την περαιτέρω βοήθεια στη δημιουργία και εκτέλεση των εικονικών οργάνων. Οι τρεις διαθέσιμες παλέτες είναι **Tools**, **Controls** και **Functions**. Στη συνέχεια θα γίνει η παρουσίαση της κάθε μιας επιμέρους παλέττας εργασίας.










Tools Palette

Η παλέττα αυτή περιέχει εργαλεία που είναι κατάλληλα στην δημιουργία και εκτέλεση των εικονικών οργάνων. Εάν η παλέττα δεν είναι ορατή, τότε για την

εμφάνιση της επιλέγουμε Show Tools Palette από το Windows menu. Μετά την επιλογή ενός εργαλείου από αυτό το menu, ο κέρσορας παίρνει την μορφή του επιλεγμένου εργαλείου. Τοποθετώντας οποιοδήποτε επιλεγμένο εργαλείο πάνω από ένα sub-VI ή συνάρτηση, εμφανίζονται στο Help παράθυρο πληροφορίες σχετικές με το sub-VI ή την συνάρτηση. Για αυτή την λειτουργία, θα πρέπει να έχει προεπιλεγεί η εμφάνιση του παραθύρου βοήθειας, με τον τρόπο που προαναφέρθηκε. Η παλέττα εργασίας παρουσιάζεται στο Σχήμα 1.4, ακολουθούμενη από την λειτουργικότητα των επι-μέρους υπό-εργαλείων.



Σχήμα 1.4: Παλέττα εργαλείων

	<i>Operating Tool:</i> για να χειριστούμε κουμπιά ελέγχου και απεικόνιση στο Front Panel.
	<i>Positioning Tool:</i> για να επιλέξουμε, μετακινήσουμε ή να αλλάξουμε το μέγεθος. Στην περίπτωση της αλλαγής μεγέθους, το αντικείμενο, θα πρέπει να θεωρηθεί ότι συμπεριφέρεται σαν παράθυρο.
	<i>Labelling Tool:</i> για να εισάγουμε κείμενο σε ετικέτες (Labels) αντικειμένων.
	<i>Wiring Tool:</i> για να διασυνδέσουμε αντικείμενα στο Block Diagram.
	<i>Menu Tool:</i> για να εκτελούμε διαδικασία Pop-up πάνω σε ένα menu ενός οργάνου κάνοντας αριστερό click, αντί για δεξί.
	<i>Scrolling Tool:</i> για την μετακίνηση μέσα στο παράθυρο, χωρίς να χρησιμοποιούμε τα scroll bars των Windows.
	<i>Breakpoint Tool:</i> για να εισάγουμε breakpoints μέσα στην διαδικασία εκτέλεσης ενός εικονικού οργάνου.
	<i>Probe Tool:</i> για να εισάγουμε probes σε καλώδια στο δομικό διάγραμμα και να βλέπουμε την τιμή των μεταβλητών που μεταφέρουν.
	<i>Color Copy Tool:</i> για να αντιγράψουμε ένα χρώμα με σκοπό την εφαρμογή του από το Color Tool.



Color Tool: για να βάφουμε ένα αντικείμενο, είτε στην έμπροσθεν ή οπίσθια όψη.













Controls Palette

Μπορούμε να εισάγουμε κουμπιά ελέγχου και απεικόνισης στο Front Panel από την Controls Palette. Κάθε επιλογή σε αυτή την παλέττα, εμπεριέχει μια υπο-επιλογή με επιπλέον κουμπιά ελέγχου και απεικόνισης σχετικά με την προηγούμενη επιλογή. Σε περίπτωση που η παλέττα δεν είναι εμφανής, μπορούμε να την ανοίξουμε, επιλέγοντας, Show Controls Palette από το Windows menu. Επιπλέον μπορούμε να ανοίξουμε την Controls Palette, κάνοντας δεξί click στο mouse πάνω σε μια άδεια περιοχή του front panel. Προσοχή θα πρέπει να δώσουμε στο γεγονός ότι η παλέττα αυτή είναι ενεργή μόνο όταν βρισκόμαστε στο front panel. Η Controls παλέττα, απεικονίζεται στο Σχήμα 1.5 ακολουθούμενη από την λειτουργικότητα των υπό εργαλείων της.



Σχήμα 1.5: Η παλέττα με τα Controls

	Numeric sub-palette: Περιέχει κουμπιά ελέγχου και απεικόνισης για αριθμητικά δεδομένα
	Boolean sub-palette: Περιέχει κουμπιά ελέγχου και απεικόνισης για Boolean δεδομένα
	String sub-palette: Περιέχει κουμπιά ελέγχου και απεικόνισης για ASCII προτάσεις και πίνακες με χαρακτήρες.









	Array & Cluster sub-palette: Περιέχει λειτουργίες σχετικές με την δημιουργία πινάκων και δομών δεδομένων.
	List & Ring sub-palette: Περιέχει κουμπιά ελέγχου και απεικόνιση για την δημιουργία menu και πινάκων επιλογής.
	Graph sub-palette: Περιέχει εργαλεία απεικόνισης γραφημάτων ή δεδομένων πραγματικού χρόνου
	Ring & Enum: Περιέχει εργαλεία για την δημιουργία διατάξεων επιλογής διαδικασιών μέσω menu
	I/O: Περιέχει λειτουργίες για την επικοινωνία με συσκευές
	RefNum: Περιέχει λειτουργίες σχετικές με τον ορισμό του path των αρχείων.
	Dialog Controls: Απεικονίζει ένα κουτί διαλόγου με την εφαρμογή.
	Classic Controls: Περιέχει κουμπιά ελέγχου και απεικόνιση που είχαν οι προηγούμενες εκδόσεις του Labview.
	ActiveX: Περιέχει σχετικές λειτουργίες για την εκτέλεση του ActiveX μηχανισμού.
	Decorations: Περιέχει γραφικά αντικείμενα για την προσαρμογή και διακόσμηση των front panels
	Select Control: Περιέχει κουμπιά ελέγχου και απεικόνιση, κατασκευασμένα από τον χρήστη
	User Lib: Περιέχει ειδικά κουμπιά ελέγχου που μπορεί να δημιουργήσει ο χρήστης.

Functions Palette



Με τις λειτουργίες της Function Palette, δημιουργούμε το Block diagram. Κάθε επιλογή στην παλέττα, περιέχει έναν αριθμό από υπό παλέττες, σχετικές με την προηγούμενη επιλογή. Σε περίπτωση που η παλέττα δεν είναι εμφανής, μπορούμε να την ανοίξουμε, επιλέγοντας, Show Function Palette από το Windows menu. Επιπλέον μπορούμε να ανοίξουμε την Function Palette, κάνοντας δεξί click στο mouse πάνω σε μια άδεια περιοχή του block diagram. Προσοχή θα πρέπει να δώσουμε στο γεγονός ότι η παλέττα αυτή είναι ενεργή μόνο όταν βρισκόμαστε στο block diagram. Η Function παλέττα, απεικονίζεται στο Σχήμα 1.6 ακολουθούμενη από την λειτουργικότητα των υπό εργαλείων της.



Σχήμα 1.6: Η παλέτα με τα Functions

	Structures sub-palette: Περιέχει προγράμματα ελέγχου εκτέλεσης του κώδικα, όπως for, while loops.
	Numeric sub-palette: Περιέχει τριγωνομετρικές, λογαριθμικές και άλλες μαθηματικές συναρτήσεις.
	Boolean sub-palette: Περιέχει λογικές και Boolean συναρτήσεις.
	String sub-palette: Περιέχει συναρτήσεις για την επεξεργασία προτάσεων από χαρακτήρες.
	Array sub-palette: Περιέχει συναρτήσεις για την επεξεργασία πινάκων.
	Cluster sub-palette: Περιέχει συναρτήσεις για την επεξεργασία συμπλεγμάτων (clusters).
	Comparison sub-palette: Περιέχει συναρτήσεις για την σύγκριση αριθμών, λογικών μεταβλητών και προτάσεων από χαρακτήρες.
	Time & Dialog sub-palette: Περιέχει συναρτήσεις για την δημιουργία παραθύρων διαλόγου, χρονισμού και χειρισμού λαθών.

	File I/O sub-palette: Περιέχει συναρτήσεις και εικονικά όργανα για την εκτέλεση I/O λειτουργιών.
	Data Acquisition sub-palette: Περιέχει εικονικά όργανα για την σύνδεση με κάρτες συλλογής δεδομένων.
	Waveform sub-palette: Περιέχει συναρτήσεις για την επεξεργασία και χρησιμοποίηση κυματομορφών.
	Analyze sub-palette: Περιέχει συναρτήσεις για την ανάλυση των δεδομένων.
	Instrument I/O sub-palette: Περιέχει εικονικά όργανα για τον έλεγχο GPIB, serial και VISA οργάνων.
	Imaq Vision sub-palette: Περιέχει συναρτήσεις για ψηφιακή επεξεργασία εικόνας.
	Mathematics sub-palette: Περιέχει πιο προηγμένες συναρτήσεις για την εκτέλεση μαθηματικών υπολογισμών με πίνακες και δεδομένα.
	Communication sub-palette: Περιέχει εικονικά όργανα για λειτουργίες TCP, DDE, Apple Events και OLE.
	Application Control sub-palette: Περιέχει εικονικά όργανα, κατάλληλα για την εκτέλεση και έλεγχο εικονικών οργάνων σε άλλους υπολογιστές.
	Graphics & Sound sub-palette: Περιέχει συναρτήσεις εισαγωγής και επεξεργασίας εικόνων και μουσικής
	Tutorial sub-palette: Περιέχει εικονικά όργανα που παρουσιάζονται στα εγχειρίδια χρήσεως του Labview.
	Report Generation sub-palette: Περιέχει εικονικά όργανα για την γρήγορη και εύκολη δημιουργία αναφορών, για εφαρμογες που έχουν σχεδιαστεί στο Labview.
	Advanced sub-palette: Περιέχει προηγμένες συναρτήσεις για το κάλεσμα προγραμμάτων γραμμένων σε C, Dlls και

	επεξεργασία δεδομένων.
	Select a VI sub-palette: Εισαγωγή ενός έτοιμου εικονικού οργάνου, στην επιφάνεια εργασίας του Labview.
	User Libraries sub-palette: Περιέχει μία λίστα από εικονικά όργανα που έχουν σχεδιαστεί από τον χρήστη.

Οι συναρτήσεις αυτές θα χρησιμοποιηθούν εκτενέστερα στο επόμενο κεφάλαιο, μέσα από την αναλυτική παρουσίαση του τρόπου δημιουργίας μερικών απλών εικονικών οργάνων.

2

Σχεδιασμός Μετρητικών Εικονικών Οργάνων

2.1 Σκοπός της άσκησης

Σκοπός της άσκησης είναι η εκμάθηση των βασικών προγραμματιστικών αρχών της αντικειμενοστραφούς γλώσσας G που χρησιμοποιείται από το Labview, για τον σχεδιασμό μετρητικών εικονικών οργάνων.

2.2 Κύρια δομικά διαγράμματα

To Front Panel

Στο Front Panel σχεδιάζουμε εικονικά όργανα με ένα συνδυασμό από κουμπιά ελέγχου και απεικόνιση. Τα κουμπιά ελέγχου, είναι ο τρόπος παροχής δεδομένων στα εικονικά όργανα. Τα κουμπιά απεικόνισης έχουν την δυνατότητα να απεικονίζουν πληροφορίες που παράγονται από το εικονικά όργανα. Υπάρχουν πολλοί τύποι κουμπιών ελέγχου και απεικόνισης, τα οποία μπορούν να εισαχθούν στο Front Panel από τις διάφορες υπό-παλέτες της Controls παλέτας.

Ως παράδειγμα τα δύο πιο πολύ χρησιμοποιημένα αριθμητικά αντικείμενα είναι τα digital control και digital indicator. Για να εισάγουμε ή να αλλάξουμε την τιμή σε ένα digital control, είτε μπορούμε να κάνουμε click πάνω στα κουμπιά αυξομείωσης με το operating tool, είτε να κάνουμε διπλό click πάνω στον αριθμό με το labeling tool ή το operating tool.

Block Diagram

Στο Block Diagram μπορούμε να προγραμματίσουμε την συμπεριφορά του εικονικού οργάνου. Κάθε αντικείμενο στο Front Panel, έχει την δική του οντότητα στο Block Diagram και αλληλεπιδρά με το πρόγραμμα που αναπτύσσεται εκεί. Στο Block Diagram έχουμε αντικείμενα τύπου nodes και terminals. Τα nodes είναι εκτελέσιμα στοιχεία προγραμματισμού και έχουν λειτουργία ανάλογη με αυτή των ορισμάτων, των υπορουτινών και των συναρτήσεων σε μια κοινή γλώσσα προγραμματισμού.

Υπάρχουν τέσσερις τύποι nodes, οι οποίοι είναι οι: functions, subVI nodes, structures και Code Interface Nodes (CINs). Τα functions είναι έτοιμες συναρτήσεις για την εκτέλεση βασικών λειτουργιών όπως είναι η πρόσθεση αριθμών, file I/O, μορφοποίηση και χειρισμός χαρακτήρων. Τα subVIs είναι εικονικά όργανα τα οποία σχεδιάζουμε και στη συνέχεια μπορούμε να τα ξανακαλέσουμε σε άλλα εικονικά όργανα, ως κομμάτι αυτών. Τα structures όπως είναι οι βρόχοι For, Loop, While και Case ελέγχουν την ροή του προγράμματος. Τέλος τα CINs είναι ένας τρόπος επικοινωνίας του Block Diagram και κώδικα αναπτυγμένο από τον χρήστη σε γλώσσα C.

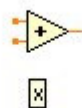
Τα terminals είναι στοιχεία αντίστοιχα με τις παραμέτρους και τις σταθερές, μέσα από τα οποία τα δεδομένα περνάνε από το Block Diagram στο Front Panel. Υπάρχουν δύο τύποι Terminals: τα control-indicator terminals και τα node terminals. Τα control-indicator terminals ανήκουν στο Front Panel. Οι τιμές των μεταβλητών που τοποθετεί ένας χρήστης κατά την εκτέλεση ενός εικονικού οργάνου, περνάνε από το Front Panel στο Block Diagram μέσω από αυτά τα Terminals. Όταν εκτελεστεί ο κώδικας στο Block Diagram, η παραγόμενη πληροφορία, περνά στο Front Panel, μέσα από τα indicators Terminals. Τα control-indicator Terminals δημιουργούνται και διαγράφονται αυτόματα όταν

δημιουργούμε ή διαγράφουμε ένα κουμπί ελέγχου ή απεικόνισης από το Front Panel.

Διασύνδεση (Wiring)

Τα καλώδια (wires) είναι ο δρόμος μεταγωγής των δεδομένων μεταξύ των αντικειμένων. Διαφορετικοί τύποι καλωδίων αντιστοιχούν σε διαφορετικούς τύπους δεδομένων (π.χ. ακέραιοι, αριθμοί κινητής υποδιαστολής). Σε μια έγχρωμη εικόνα, κάθε τύπος δεδομένων, απεικονίζεται και με ένα διαφορετικό χρώμα για μεγαλύτερη έμφαση.

Για να συνδέσουμε ένα αντικείμενο με ένα άλλο, χρησιμοποιούμε το wiring tool στο πρώτο terminal και στη συνέχεια μετακινούμε το εργαλείο στο επόμενο terminal και κάνουμε click σε αυτό. Στο επόμενο σχήμα απεικονίζονται τα terminals ενός αθροιστή.



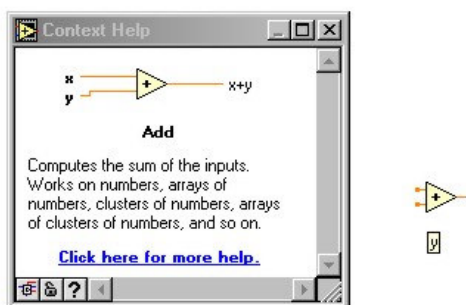
Όταν το wiring tool βρίσκεται πάνω από την περιοχή ενός Terminal τότε η περιοχή αυτή αναβοσβήνει, υποδηλώνοντας ότι ένα click θα συνδέσει το καλώδιο σε αυτό το Terminal. Δεν χρειάζεται να έχουμε πατημένο το mouse, κατά την διάρκεια της μετακίνησης του wiring tool από το ένα Terminal στο άλλο.

Με οποιοδήποτε εργαλείο του Labview, μπορούμε να ανοίξουμε το pop-up menu πάνω σε οποιαδήποτε συνάρτηση ή subVI και να επιλέξουμε Create Control, Create Constant ή Create Indicator. Με αυτό τον τρόπο τα αντίστοιχα control, constant και indicator θα δημιουργηθούν αυτόματα, με έτοιμες τις καλωδιώσεις τους.



Χρησιμοποίηση βοήθειας

Όλα τα ενσωματωμένα εικονικά όργανα και συναρτήσεις στο Labview, περιέχουν πλήρη βοήθεια για την λειτουργικότητα αυτών. Όταν χρησιμοποιούμε ένα άγνωστο εικονικό όργανο ή μια συνάρτηση, μπορούμε να το τοποθετήσουμε στο Block Diagram, να ανοίξουμε το pop-up μενού, κάνοντας δεξί click και να επιλέξουμε On-line Help για μια πλήρη περιγραφή της λειτουργικότητας του και των συνδέσεων αυτού. Στην περίπτωση που χρειαζόμαστε μια γρήγορη εύρεση βοήθειας για να μας θυμίσει τον τρόπο λειτουργίας του εικονικού οργάνου καθώς και την χρήση των εισόδων και εξόδων αυτού, μπορούμε να επιλέξουμε show help από το help menu και ένα παράθυρο βοήθειας εμφανίζεται σχετικά με το εικονικό όργανο που βρίσκεται κάτω από τον κέρσορα.

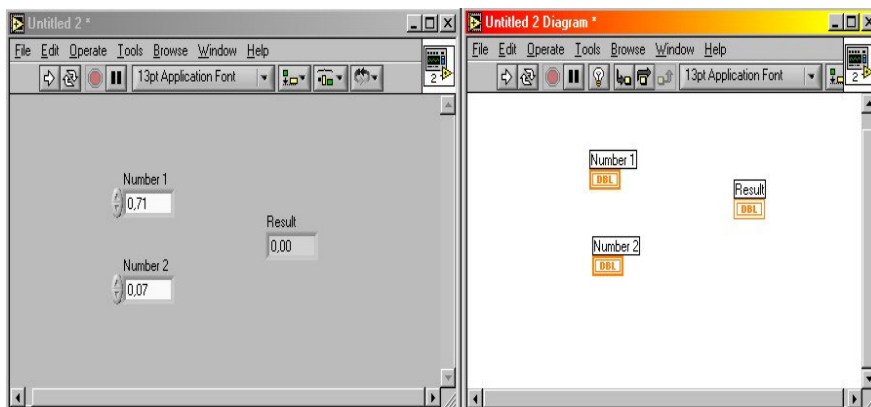




2.3 Σχεδιασμός ενός εικονοοργάνου για την πρόσθεση δύο αριθμών

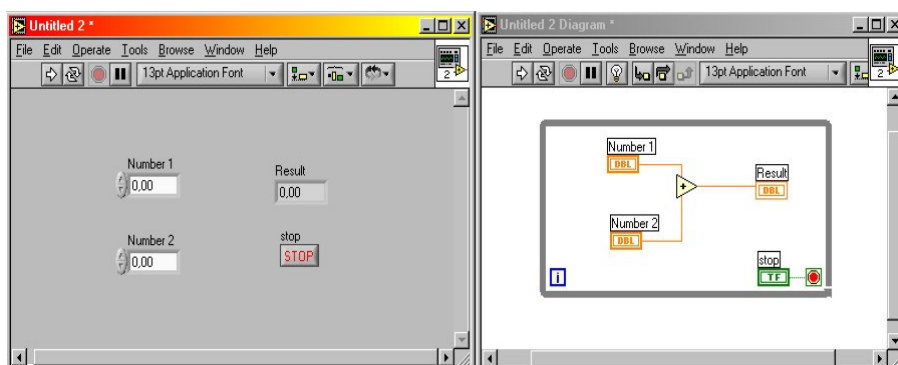
Με σκοπό τον σχεδιασμό ενός εικονοοργάνου για την πρόσθεση δύο αριθμών, αρχικά εισάγουμε στο front panel, δύο Digital Controls και ένα Digital Indicator. Με το Labeling tool αλλάζουμε τα ονόματα, Numeric, Numeric2 και Numeric 3,

Εργαστηριακές Ασκήσεις Συστημάτων Αυτομάτου Ελέγχου II

αντίστοιχα με Number 1, Number 2 και Result. Μετα την τοποθέτηση των controls και των indicators, στο block diagram εμφανίζονται τα αντίστοιχα αντικείμενα.



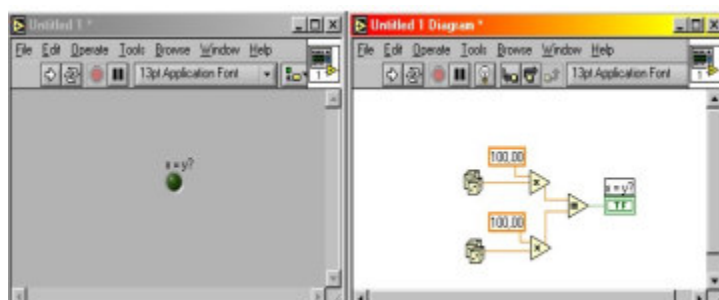
Στο block diagram, εισάγουμε από το Function palette\nnumeric το σύμβολο add που χρησιμοποιείται για την πρόσθεση αριθμών/ πινάκων. Με την βοήθεια του Wiring tool εκτελούμε τις διασυνδέσεις όπως εμφανίζονται στο παρακάτω διάγραμμα. Από το Function palette/Structures, επιλέγουμε το while loop και περικυκλώνουμε με αυτό όλο τον προηγούμενο σχεδιασμό μας. Στη συνέχεια στο  του while loop, εκτελούμε δεξί click και επιλέγουμε create control. Αυτόματα δημιουργείται ένα Boolean control button (στο front panel) με αρχική με τιμή false και ένδειξη stop. Κάνουμε ξανά δεξί Click πάνω στο  και επιλέγουμε Stop if true. Μετά από τις παραπάνω ενέργειες, θα πρέπει να έχουν δημιουργηθεί τα παρακάτω διαγράμματα.



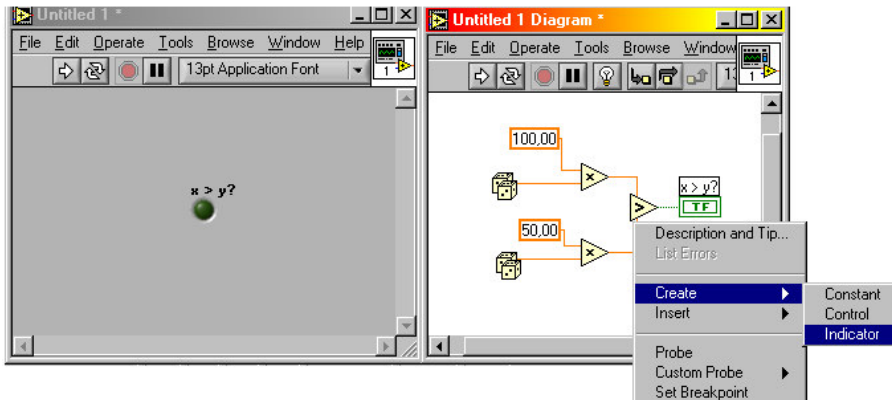
Στο σημείο αυτό, είμαστε σε θέση να αρχίσουμε την εκτέλεση του εικονικού οργάνου, για την πρόσθεση δύο αριθμών. Το αποτέλεσμα εμφανίζεται στο πεδίο Result. Το πρόγραμμα αυτό θα εκτελείται μέχρι να πατήσουμε το κουμπί stop.



2.4 Σχεδιασμός ενός εικονοοργάνου για την σύγκριση δύο τυχαίων αριθμών

Στο Block diagram, επιλέγουμε από το Functions/numeric δύο αντικείμενα random number[0-1] (Η γραφική απεικόνιση ισοδυναμεί με ένα ζάρι) και τα εισάγουμε. Τα αντικείμενα αυτά μπορούν να μας δώσουν τυχαίους αριθμούς μέσα στο αριθμητικό διάστημα [0-1]. Με σκοπό να αυξήσουμε το διάστημα από [0-100], από το Functions/numeric εισάγουμε δύο αντικείμενα multiply. Με το Wiring tool συνδέουμε τα “ζάρια” με την μία άκρη καθενός πολλαπλασιαστή και στη συνέχεια κάνοντας δεξί click, στα ελεύθερα άκρα, επιλέγουμε create constant και δίνουμε την τιμή 100. Με αυτόν τον τρόπο έχουμε τώρα τυχαίες τιμές, μέσα στο διάστημα (0-100). Από το Function/comparison επιλέγουμε την συνάρτηση greater και με το wiring tool ενώνουμε τις πολλαπλασιασμένες εξόδους(με τον αριθμο 100), από τα “ζάρια” στις εισόδους της συνάρτησης greater. Η συνάρτηση αυτή, χρησιμοποιείται για να συγκρίνει δύο αριθμούς, και επιστέφει Boolean τιμή true ή false. Στην έξοδο της συνάρτησης με το wiring tool, κάνουμε δεξί click και επιλέγουμε create indicator. Η ενέργεια αυτή, θα εμφανίσει στην οθόνη του Control Panel ένα σβηστό λαμπάκι με πράσινο χρώμα.

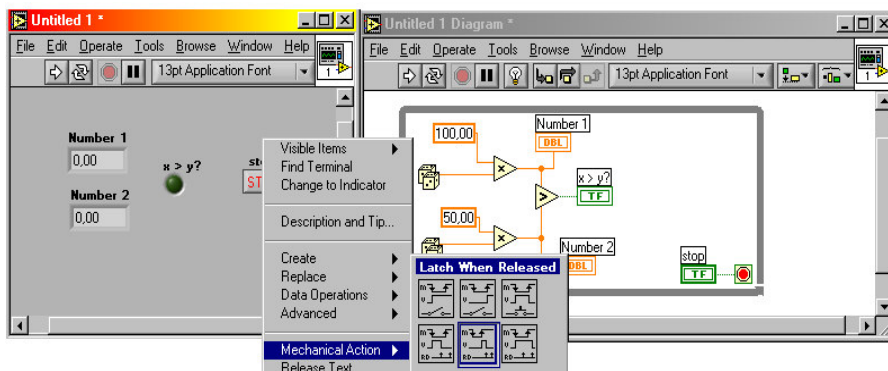


Επιπλέον για να βλέπουμε σε κάθε χρονική στιγμή ποιους αριθμούς συγκρίνουμε, με το wiring tool, πατάμε δεξί click πάνω στις εξόδους των πολλαπλασιαστών και επιλέγουμε create indicator.




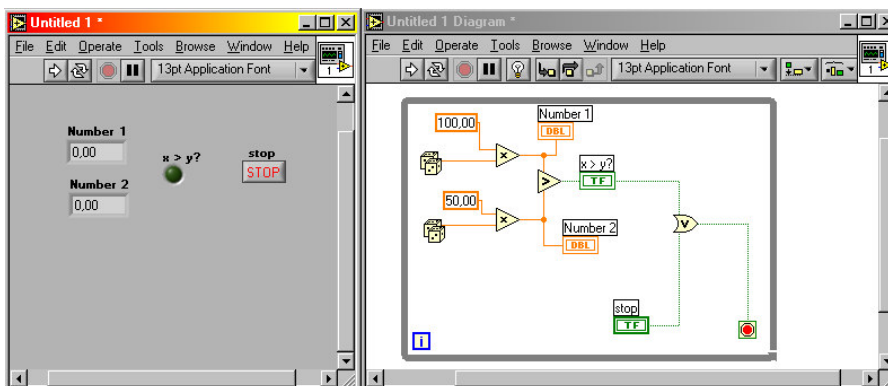
Με το Labeling Tool πηγαίνουμε στο Front Panel και μετανομάζουμε τα δύο Indicators σε Number 1 και Number 2, ενώ το $x > y$ το μετανομάζουμε σε greater. Με σκοπό η διαδικασία αυτή να επαναλαμβάνεται συνεχώς. Περικλείουμε το πρόγραμμα μας με έναν βρόχο while. Ο βρόχος αυτός βρίσκεται κάτω από το Functions\structures. . Στη συνέχεια στο  του while loop, εκτελούμε δεξί click και επιλέγουμε create control. Αυτόματα δημιουργείται ένα Boolean control button (στο front panel) με αρχική με τιμή false και ένδειξη stop. Κάνουμε ξανά δεξί Click πάνω στο  και επιλέγουμε Stop if true. Στο Front panel, πατάμε δεξί click πάνω στο button και επιλέγουμε Mechanical Action/latch when released. Επίσης με το Labeling tool, του δίνουμε την ονομασία Stop.

Εργαστηριακές Ασκήσεις Συστημάτων Αυτομάτου Ελέγχου II



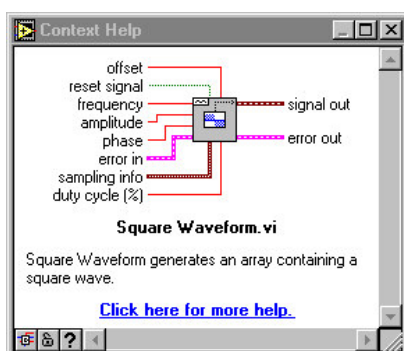
Με σκοπό το πρόγραμμα να σταματά μόνο αν $a > b$ ή αν πατήσουμε εμείς το Stop κουμπί, θα πρέπει να προγραμματίσουμε τις επόμενες αλλαγές στο ήδη υπάρχον πρόγραμμα.

Αρχικά από το μενού Functions/Boolean εισάγουμε μια πύλη OR, της οποίας τις εισόδους ενώνουμε την έξοδο από το equal και το κουμπί Stop. Τέλος την έξοδο της πύλης την συνδέουμε στο .



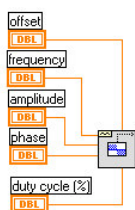
2.5 Σχεδιασμός εικονικής γεννήτριας συχνοτήτων

Αφού ξεκινήσουμε το Labview, επιλέγουμε από το menu/window/Tile left and



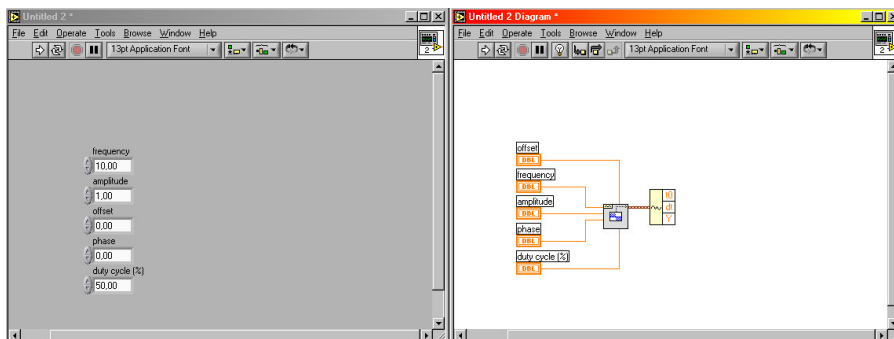
right(ctrl+T). Η επιλογή αυτή μας τοποθετεί το Front panel και το Block diagram σε δύο παράθυρα που καλύπτουν όλη την οθόνη και μπορούν να τα βλέπουμε ταυτόχρονα. Στο Block diagram επιλέγουμε από Function Palette\ Analyze\ Waveform Generation\ Square Generation. Στο σημείο αυτό, είναι πολύ χρήσιμο να εμφανίσουμε την επιπρόσθετη βοήθεια του Labview (ctrl+H), από όπου μπορούμε να πάρουμε πληροφορίες για τον τρόπο που πρέπει να γίνουν οι διάφορες διασυνδέσεις.

Στη συνέχεια και για τις συνδέσεις που φαίνονται στο επόμενο, πηγαίνουμε με το wiring tool και εκτελούμε δεξί click, create control. Το σήμα signal out, είναι τύπου waveform. Τα σήματα αυτά περιλαμβάνουν τρία είδη πληροφορίας. Το πρώτο αναφέρεται στον χρόνο t , το δεύτερο στον χρονικό διάστημα dt που ακολουθεί το επόμενο δείγμα της εξόδου και στο τρίτο πεδίο, όπου και υπάρχουν οι αντίστοιχες (με τις χρονικές στιγμές του πρώτου πεδίου) τιμές της εξόδου.

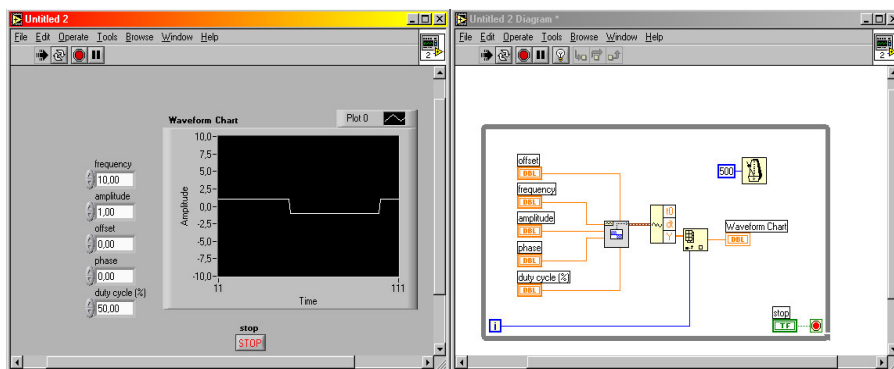


Η εικονική γεννήτρια παλμών που θα δημιουργήσουμε, παράγει τα σήματα, όπως κάνει και μια κανονική παλμογεννήτρια. Η μόνη διαφορά έγκειται στο γεγονός ότι στην μεν πρώτη περίπτωση, οι παλμοί παράγονται έμμεσα από software, ενώ στην δεύτερη περίπτωση, τους παλμούς τους παράγουν ηλεκτρονικά κυκλώματα. Με σκοπό να οδηγήσουμε του παλμούς αυτούς σε κάποια διάταξη, θα πρέπει από το signal out να πάρουμε μόνο την πληροφορία που σχετίζεται με τα επίπεδα τάσεως του παλμού και όχι τις χρονικές στιγμές που συμβαίνουν αυτά. Για το σκοπό αυτό, συνδέουμε στο signal out το Get waveform components το οποίο βρίσκεται κάτω από το Functions\waveforms. Με την βοήθεια του positioning tool, αυξάνουμε το μέγεθος (σαν παράθυρο Windows) του Get waveform components, όπως φαίνεται

και στο παρακάτω σχήμα. Από την ενέργεια μας αυτή, είναι τώρα εμφανή τα στοιχεία που συνθέτουν ένα σήμα Waveform, όπως είναι το signal out.



Στο Front panel, εισάγουμε από το μενού Controls\Graphs ένα Waveform chart. Το Y που εμφανίζεται είναι ένας πίνακας στήλη (n,1) με όλες τις τιμές του παλμού σε μια περίοδο. Με σκοπό να πάρουμε τις τιμές αυτές από την πρώτη μέχρι και την τελευταία θα πρέπει από το Functions/array να επιλέξουμε index array. Με το wiring tool, στην είσοδο του index array/n-dimension array συνδέουμε τον πίνακα Y. Στην έξοδο αυτού συνδέουμε το Waveform chart που εισάγαμε. Όλον τον προγραμματισμό μας τον περικλείουμε με ένα while loop και το index του While loop, το συνδέουμε με το index του index array. Επιπλέον δημιουργούμε έναν έλεγχο για την εκτέλεση του loop όπως έχουμε κάνει και στα προηγούμενα



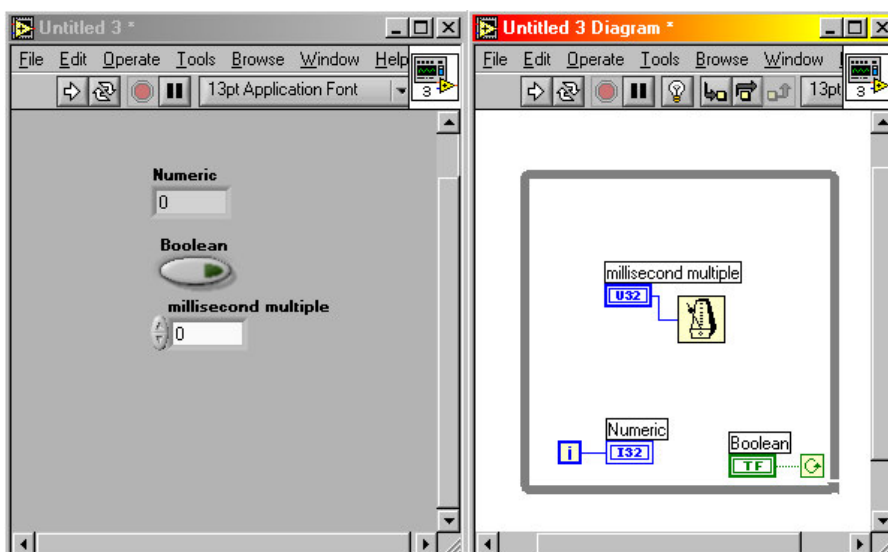
παραδείγματα. Τέλος εισάγουμε από το Functions\Time&Dialog το vi wait until next multiple, με σκοπό να ρυθμίσουμε την ταχύτητα με την οποία παράγεται η παλμοσειρά. Με αυτόν τον τρόπο, μπορούμε να βλέπουμε τις τιμές και το είδος

του παλμού, με τον οποίο μπορούμε να τροφοδοτήσουμε κάποια συσκευή ή διάταξη.

Στο σημείο αυτό οι φοιτητές, θα πρέπει να πειραματιστούν με την δημιουργία κυματομορφών διαφορετικού τύπου, καθώς και να εξετάσουν την λειτουργικότητα των κουμπιών ελέγχου που έχουν τοποθετηθεί στο Front panel καθώς και τις αλλαγές στην μορφή της προκύπτουσας κυματομορφής.

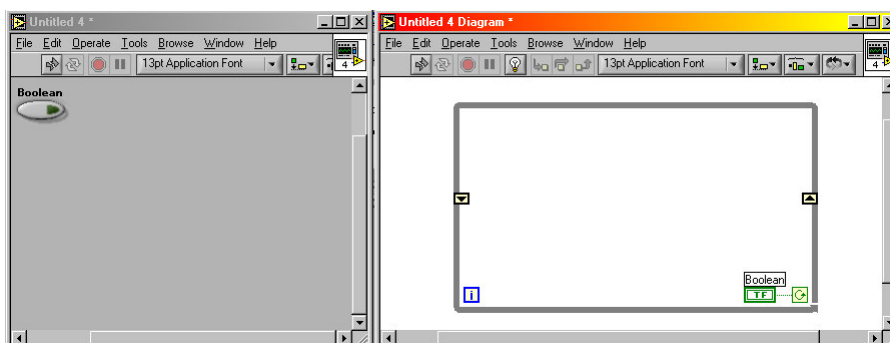
2.6 Συγχρονισμός εκτέλεσης των VIs



Με σκοπό ο κώδικας που δημιουργούμε, να μην εκτελείται με την μέγιστη ταχύτητα αλλά με την επιθυμητή που θα την καθορίζει ο χρήστης, θα πρέπει να κάνουμε χρήση του VI Wait Until Next ms multiple, το οποίο μπορούμε να το βρούμε στην Function παλέττα\Time & Dialog. Σε οποιοδήποτε βρόχο εκτέλεσης του προγράμματος, μπορούμε να το εισάγουμε και να επιλέξουμε, το χρονικό διάστημα καθυστέρησης. Για λόγους επίδειξης, θα χρησιμοποιήσουμε ένα απλό βρόχο While και θα δημιουργήσουμε έναν Indicator του αριθμού των επαναλήψεων του loop. Με το wiring tool, πηγαίνουμε πάνω στο Wait Until Next ms Multiple και επιλέγουμε την δημιουργία ενός control για τον έλεγχο της χρονοκαθυστέρησης. Επίσης δημιουργούμε άλλο ένα control για τον έλεγχο της έναρξης και παύσης λειτουργίας του προγράμματος μας. Μετά την ολοκλήρωση του προγραμματισμού μας, από το front panel και αλλάζοντας το μέγεθος της χρονοκαθυστέρησης, μπορούμε να δούμε την μεταβολή της ταχύτητας εκτέλεσης του βρόχου του προγράμματός μας.

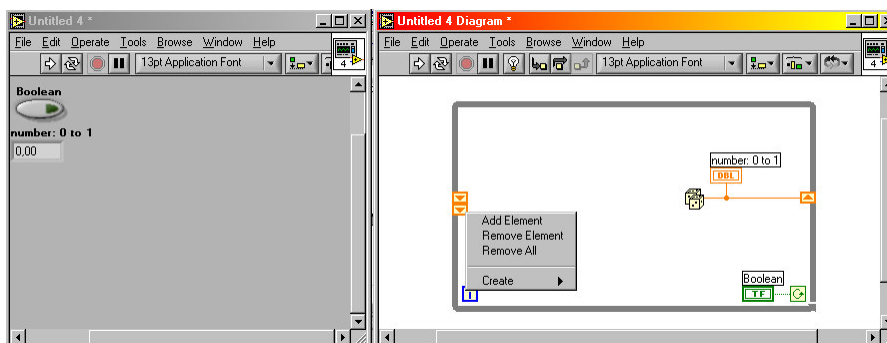


2.7 Χρήση των Shift Registers

Ο βασικός σκοπός της παρούσας άσκησης, είναι η εξοικείωση του χρήστη με την χρήση μνήμης στο περιβάλλον προγραμματισμού του Labview. Σε αυτό το περιβάλλον η μνήμη ορίζεται σαν ένας register. Ανάλογα με τις ανάγκες της εφαρμογής, ο χρήστης μπορεί να χρησιμοποιήσει πολλαπλούς registers. Τους shift registers, μπορούμε να τους εισάγουμε στον προγραμματισμό μας, μόνο με την ταυτόχρονη χρήση κάποιου λογικού βρόχου (while/for etc.). Η εισαγωγή γίνεται πηγαίνοντας στην άκρη του βρόχου και εκτελώντας δεξί click\ insert shift register. Για την περίπτωση του while loop, και εκτελώντας την παραπάνω διαδικασία, θα έχουμε τον shift register που εμφανίζεται στο παρακάτω σχήμα.

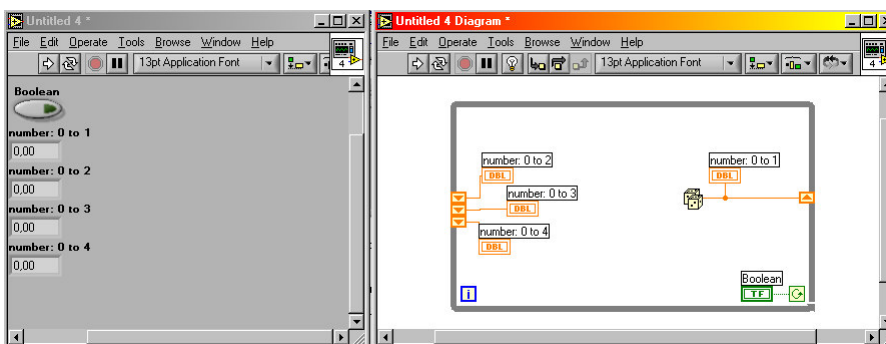


Στο σημείο αυτό θα εισάγουμε και μια γεννήτρια τυχαίων αριθμών. Ο τελικός σκοπός του προγράμματος που αναπτύσσουμε, είναι να μπορούμε να παρακολουθούμε στο front panel, την τωρινή και τις τρεις περασμένες τιμές από την γεννήτρια τυχαίων αριθμών. Κατά την λειτουργία του shift register, η τιμή που περνάμε στο δεξιό του τμήμα , θα βρίσκεται στην διάθεση μας, την επόμενη φορά που θα εκτελεστεί ο βρόχος, από το αριστερό τμήμα του register  αυτή την φορά. Στη συνέχεια συνδέουμε την έξοδο της γεννήτριας τυχαίων αριθμών (ζαρια) στο δεξιό τμήμα του register και δημιουργούμε έναν indicator για αυτήν την έξοδο της γεννήτριας τυχαίων αριθμών. Με σκοπό να βλέπουμε μέχρι και τις τρεις προηγούμενες τιμές από την γεννήτρια τυχαίων αριθμών, στο δεξιό τμήμα του register και με το wiring tool εκτελούμε δεξί click και επιλέγουμε add element. Την διαδικασία αυτή την επαναλαμβάνουμε τρεις φορές, και με αυτόν τον τρόπο, έχουμε δημιουργήσει την μνήμη που θέλαμε. Θα πρέπει να έχουμε πάντα υπόψηνη μας ότι η μετακίνηση της πληροφορίας σε αυτήν την μνήμη, γίνεται με σειριακό τρόπο και μπορεί να χρησιμοποιηθεί για όλους τους τύπους μεταβλητών που χρησιμοποιεί το Labview.

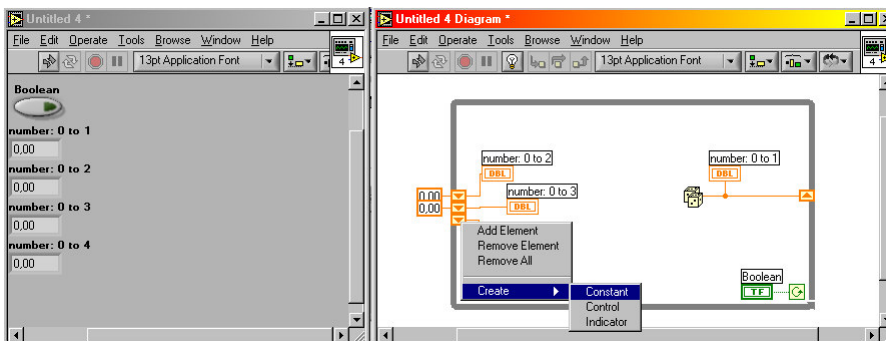


Σε κάθε ένα βελάκι στο δεξιό τμήμα του register, δημιουργούμε και από έναν indicator.

Εργαστηριακές Ασκήσεις Συστημάτων Αυτομάτου Ελέγχου II

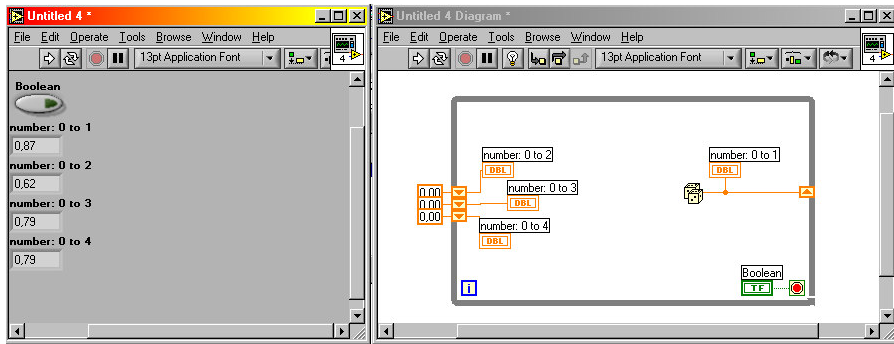


Με το wiring tool και εκτελώντας πάλι δεξί click πάνω στο δεξιό τμήμα του shift register, επιλέγουμε create constant και αναθέτουμε στην σταθερά την τιμή 0. Την διαδικασία αυτή την εκτελούμε τρεις φορές, για κάθε βέλος του register, και με τον τρόπο αυτό μπορούμε να αρχικοποιήσουμε την μνήμη στις τιμές που εμείς επιθυμούμε. Στην περίπτωση μας, η αρχικοποίηση έγινε στην τιμή 0.



Για την ολοκλήρωση του προγράμματος, εισάγουμε μια χρονοκαθυστέρηση 100ms, όπως περιγράφηκε στο 2.5 και επιπλέον επιλέγουμε με δεξί click να σταματά η εκτέλεση του while loop μόνο όταν η τιμή του Boolean control είναι true.

Εργαστηριακές Ασκήσεις Συστημάτων Αυτομάτου Ελέγχου II



Στο σημείο αυτό είμαστε έτοιμοι να εκτελέσουμε το VI και να παρατηρήσουμε πως οι τιμές της γεννήτριας τυχαίων αριθμών μεταβάλλονται σειριακά. Σε κάθε χρονική στιγμή μπορούμε να παρατηρούμε την τωρινή και τις τρεις προηγούμενες τιμές της γεννήτριας τυχαίων αριθμών.

3

Αναγνώριση Συνάρτησης Μεταφοράς Γραμμικού Συστήματος

3.1 Σκοπός της Άσκησης

Σκοπός της άσκησης, είναι η εκμάθηση μιας γενικής μεθοδολογίας αναγνώρισης συνάρτησης μεταφοράς, με απλές παρατηρήσεις και μετρήσεις στην είσοδο και έξοδο του συστήματος. Η διαδικασία αυτή είναι γενική και μπορεί να εφαρμοστεί σε οποιοδήποτε πραγματικό σύστημα.

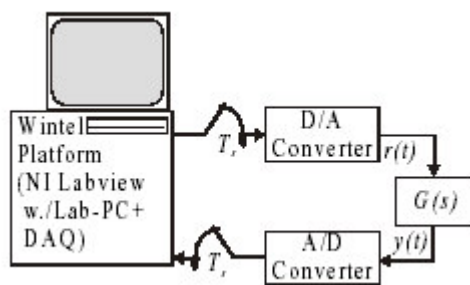
3.2 Εισαγωγικά

Το σύστημα που θα αναγνωρίσουμε αποτελεί το βασικό Test-Bench που θα χρησιμοποιηθεί στις επόμενες εργαστηριακές ασκήσεις. Έστω η συνάρτηση

μεταφοράς του συστήματος
$$G(s) = K_{dc} \frac{\prod_{i=1}^m (s + z_i)}{\prod_{j=1}^n (s + p_j)}, \quad n \geq m, \quad \text{όπου } z_i, i = 1, \dots, m$$

είναι τα μηδενικά του συστήματος και $p_j, j = 1, \dots, n$ οι πόλοι του ανοικτού συστήματος. Ο στόχος της αναγνώρισης του συστήματος είναι ο υπολογισμός των πόλων, μηδενικών και κέρδους (K_{dc} – παράμετρος) με βάση μετρήσεις που προέρχονται από την απόκριση του συστήματος για συγκεκριμένες διεγέρσεις (π.χ. βηματικές ή/και ημιτονοειδείς εισόδους). Επιπλέον είναι επιθυμητό να δύναται ο χρήστης να υπολογίζει την τάξη (n -παράμετρος) του συστήματος καθώς επίσης και τον αριθμό (m -παράμετρος) των μηδενικών. Στην παρούσα άσκηση το σύστημα, όπως φαίνεται στο Σχήμα 3.1, διεγείρεται μέσω μιας περιοδικής εισόδου $r(t)$

(βηματική ή ημιτονοειδής). Ο μέσος όρος της κυματομορφής είναι μηδέν, και ο χρήστης μπορεί να μεταβάλει το εύρος και την συχνότητα της (π.χ. $r(t) = R \sin(2\pi ft)$). Οι τιμές της εισόδου υπολογίζονται ανά τακτά χρονικά διαστήματα T_s μέσω του υπολογιστή και αποστέλλονται στο σύστημα με την χρήση του 12-bit DAC (ψηφιακού-προς-αναλογικό μετατροπέα) του οποίου το εύρος τιμών είναι από -10Volts μέχρι +10Volts. Η απόκριση $y(t)$ του συστήματος μετρείται πάλι σε τακτά χρονικά διαστήματα και οι δειγματοληφθείσες τιμές καταγράφονται στον υπολογιστή μέσω του 12-bit ADC (αναλογικού-προς-ψηφιακό μετατροπέα).



Σχήμα 3.1: Βασικό δομικό διάγραμμα αναγνώρισης συστήματος

Αν η τάξη του συστήματος είναι γνωστή και χαμηλή ($n=1,2,3$) τότε είναι πιθανή η αναγνώριση της συνάρτησης μεταφοράς του συστήματος από την βηματική απόκριση του συστήματος

3.3 Αναγνώριση Συστήματος Χαμηλής Τάξης με Βάση την Βηματική Απόκριση

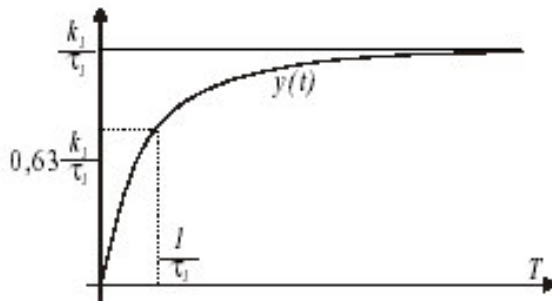
Σύστημα 1^{ης} τάξης

Έστω η συνάρτηση μεταφοράς συστήματος $G(s) = \frac{k_1}{s + \tau_1}$. Η βηματική απόκριση

του συστήματος είναι $y(t) = k_1 \left(-\frac{1}{\tau_1} e^{-\tau_1 t} + \frac{1}{\tau_1} \right)$ και μία ενδεικτική απόκριση

φαίνεται στο Σχήμα 3.2. Η σταθερά χρόνου $\frac{1}{\tau_1}$ υπολογίζεται από την χρονική

στιγμή που η απόκριση φτάνει το 63% της τελικής τιμής, ενώ η σταθερά k_1 μπορεί να υπολογιστεί από την τελική τιμή k_1/τ_1 της απόκρισης.



Σχήμα 3.2: Τυπική απόκριση συστήματος 1^{ης} τάξης

Σύστημα 2^{ης} τάξης

Υποθέστε ότι η συνάρτηση μεταφοράς του συστήματος είναι

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Η βηματική απόκριση του συστήματος είναι

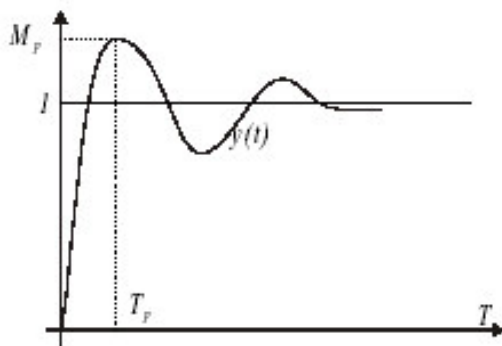
$$y(t) = 1 - \frac{1}{b} e^{-\zeta\omega_n t} \sin(\omega_n b t + \theta), \quad b = \sqrt{1 - \zeta^2}, \quad \theta = \cos^{-1} \zeta$$

και εμφανίζεται στο Σχήμα 3.3. Η σταθερά απόσβεσης ζ και η γωνιακή ιδιοσυχνότητα

$$\omega_n \text{ υπολογίζονται από τις σχέσεις } M_p = 1 + e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}, \quad T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}$$

Αυτή η απόκριση είναι υπο-αποσβενύμενη ενώ η απόκριση του συστήματος για $\zeta = 1$ είναι

$$y(t) = 1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t}$$



Σχήμα 3.3: Τυπική απόκριση συστήματος 2^{ης} τάξηςΣύστημα 3^{ης} τάξης

Αν ένα σύστημα 3^{ης} τάξης έχει τρεις ταυτόσημους πόλους και συνάρτηση

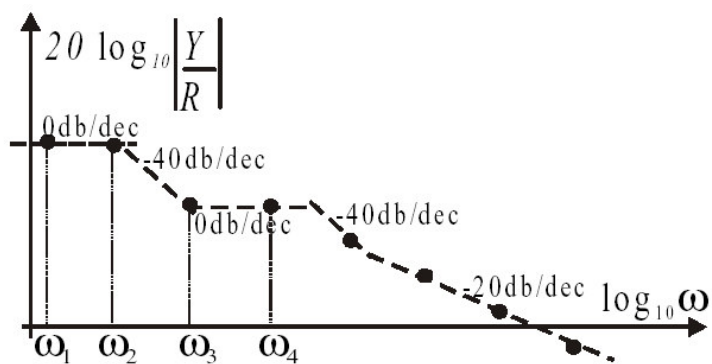
μεταφοράς $G(s) = \frac{\tau_3^3}{(s + \tau_3)^3}$, η βηματική του απόκριση είναι

$$y(t) = 1 - e^{-\tau_3 t} - \tau_3 t e^{-\tau_3 t} - \frac{\tau_3^2}{2} t^2 e^{-\tau_3 t} \text{ και η εύρεση της παραμέτρου } \tau_3 \text{ είναι}$$

αρκετά πολύπλοκη. Αν το σύστημα δεν έχει ίδιους πόλους, τότε ο υπολογισμός τους είναι αρκετά πολύπλοκος.

3.4 Αναγνώριση Συστήματος με Βάση την Απόκριση στο Πεδίο Συχνότητας (Bode-διαγράμμα)

Για τον υπολογισμό των πόλων και μηδενικών μιας συνάρτησης μεταφοράς $G(s)$, το σύστημα διεγείρεται με μία ημιτονοειδή είσοδο $r(t) = R_i \sin(\omega_i t)$. Η έξοδος του συστήματος στην μόνιμη κατάσταση είναι $y(t) = Y_i \sin(\omega_i t + \phi_i)$. Αν επαναληφθεί η ίδια διαδικασία για αρκετές συχνότητες $\omega_i, i = 1, \dots$ τότε μπορεί να σχεδιαστεί το Bode-διάγραμμα του συστήματος (Σχήμα 3.4). Κατόπιν, το διάγραμμα μέτρου του Bode-διαγράμματος προσεγγίζεται από ευθείες γραμμές με κλίση $\pm 20\nu \text{ db/dec}$, $\nu = 0, 1, \dots$ και στα σημεία τομής αυτών των ευθειών βρίσκονται οι πόλοι και τα μηδενικά του συστήματος.

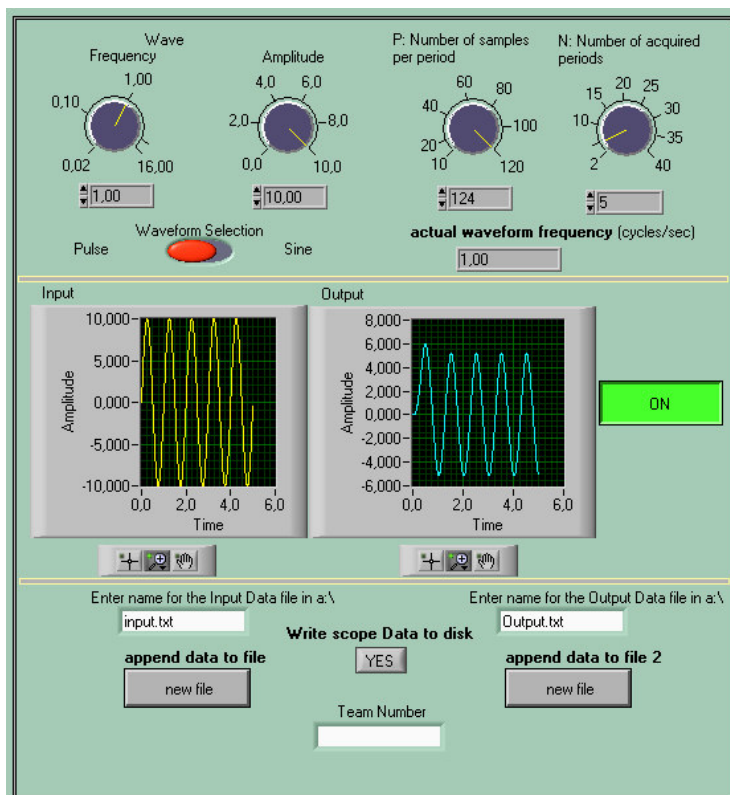


Σχήμα 3.4: Ενδεικτικό Bode-διάγραμμα Συστήματος

3.5 Εκτέλεση της Εργαστηριακής Άσκησης

Για την 2^η εργαστηριακή άσκηση, οι φοιτητές, θα χρησιμοποιήσουν το VI, Lab2_ID.vi. Το VI αυτό έχει σχεδιαστεί ειδικά για να χρησιμοποιηθεί στην αναγνώριση συστημάτων. Μια γεννήτρια παλμών και ημιτονοειδών σημάτων, χρησιμοποιείται για την διέγερση του υπό αναγνώριση συστήματος. Ένα κατάλληλα σχεδιασμένο σύστημα συλλογής δεδομένων, χρησιμοποιείται για να πάρουμε μετρήσεις της απόκρισης του συστήματος. Τις μετρήσεις αυτές, θα τις χρησιμοποιήσουμε σε συνδυασμό με την προαναφερθείσα μεθοδολογία για την εύρεση της τάξης ενός αγνώστου συστήματος.

Το βασικό Front Panel του Vi της άσκησης απεικονίζεται στο Σχήμα 3.5. Από αυτό το VI, μπορεί ο χρήστης να εκτελέσει όλα τα παρακάτω βήματα για την εκτέλεση της άσκησης. Προσοχή θα πρέπει να ληφθεί στο γεγονός ότι δεν χρειάζεται παραπάνω προγραμματισμός στο γραφικό περιβάλλον του Labview.



Σχήμα 3.5: Front Panel VI για αναγνώριση συστήματος

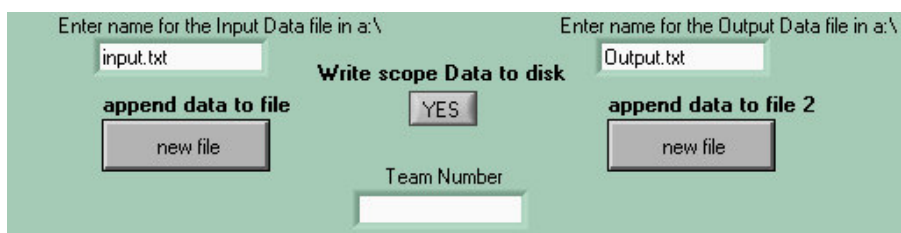
Για την επιτυχή χειρισμό του VI της άσκησης, θα πρέπει να ακολουθηθούν τα παρακάτω βήματα:

A) Από το διακόπτη επιλογής (Σχήμα 3.6), θα πρέπει να επιλέξουμε το είδος του σήματος που θέλουμε να εφαρμοστεί στο υπό αναγνώριση σύστημα. Μπορούμε να επιλέξουμε ανάμεσα σε ημιτονοειδές σήμα και σε παλμοσειρά. Αρχικά θα θεωρήσουμε το σύστημα να είναι χαμηλής τάξης και θα το διεγείρουμε με παλμό. Ύστερα από πειραματισμούς και εφόσον θα είμαστε έτοιμοι να πάρουμε μετρήσεις (Σχήμα 3.7), θα πρέπει να επιλέξουμε το όνομα των αρχείων στα οποία θα σωθούν τα δεδομένα της εισόδου και της δειγματοληψίας. Τέλος πατώντας το κουμπί Write scope data to disk, το VI την επόμενη φορά που θα εκτελεστεί θα αποθηκεύσει τα δεδομένα της εισόδου και της εξόδου (PN δείγματα).



Σχήμα 3.6: Διακόπτης επιλογής επιθυμητής εισόδου

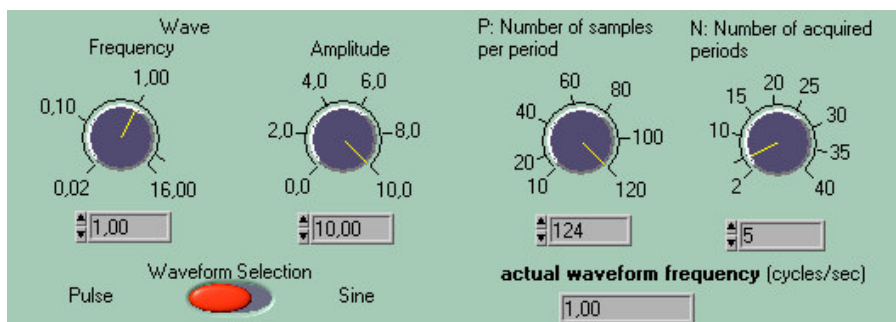
Για την ευκολία των φοιτητών, θα πρέπει να δίνονται ονόματα στα αρχεία όπου θα σωθούν τα δεδομένα σχετικά με τις ρυθμίσεις του συστήματος την στιγμή που πάρθηκαν. Το όνομα θα πρέπει να περιέχει πληροφορίες για την συχνότητα, το εύρος, τον αριθμό δειγμάτων ανά περίοδο και τον συνολικό αριθμό περιόδων.



Σχήμα 3.7: Ρυθμίσεις για την εγγραφή των πειραματικών δεδομένων

Τέλος οι φοιτητές, θα πρέπει να συμπληρώσουν και το όνομα της ομάδας στο κατάλληλο πεδίο (Team Number). Το όνομα αυτό κωδικοποιείται κατάλληλα και χρησιμοποιείται ως ηλεκτρονικό κλειδί για την τεκμηρίωση των πειραματικών δεδομένων της κάθε ομάδας. Τα δεδομένα αυτά είναι υποχρεωτικά να συνοδεύουν τις παραδοτέες αναφορές.

B) Στη συνέχεια θα πρέπει από τους περιστροφικούς διακόπτες (Σχήμα 3.8), να επιλέξουμε το εύρος της εισόδου που θα εφαρμόσουμε και την συχνότητα της. Μπορούμε να χρησιμοποιήσουμε και τις προκαθορισμένες τιμές που έχουν εισαχθεί. Θα πρέπει να προσέξουμε ώστε με τις ρυθμίσεις που θα κάνουμε, να μπορούμε να παρατηρήσουμε με ευκολία την απόκριση του συστήματος. Στην αντίθετη περίπτωση, και εφόσον οι ρυθμίσεις μας υπερβούν τα όρια του hardware, μηνύματα λάθους θα εμφανιστούν στην οθόνη, προτρέποντας τους φοιτητές για αλλαγή στις ρυθμίσεις.



Σχήμα 3.8: Περιστροφικοί διακόπτες επιλογής πλάτους και συχνότητας

Γ) Τώρα είμαστε σε θέση να εκτελέσουμε το VI. Για τον σκοπό αυτό, ο κεντρικός διακόπτης (Σχήμα 3.5) είναι είδη σε κατάσταση On και το πάτημα του κουμπιού Run θα σημάνει και την έναρξη του πειράματος. Τα δεδομένα της εισόδου και της εξόδου αρχίζουν να συλλέγονται σταδιακά σε έναν Buffer και στην συνέχεια αυτά χρησιμοποιούνται για την σχεδίαση των γραφικών παραστάσεων της εισόδου και της εξόδου, καθώς και την εγγραφή των κατάλληλων αρχείων. Η διαδικασία αυτή μπορεί να είναι χρονοβόρα, γιατί θα πρέπει οι φοιτητές να περιμένουν το απαραίτητο χρονικό διάστημα μέχρι να ολοκληρωθούν οι υπολογισμοί. Το χρονικό αυτό διάστημα εισούται με $\frac{P}{f}$ και μετά την χρονική παρέλευση αυτού του χρόνου, μπορούμε να παρακολουθήσουμε την μεταβολή της εξόδου(γαλάζιο χρώμα) του συστήματος, συναρτήσει της μεταβολής της εισόδου(κίτρινο χρώμα).

Δ) Για την αναγνώριση ενός συστήματος μεγαλύτερης τάξης, χρησιμοποιώντας τα διαγράμματα Bode μέτρου ($20 \log \frac{A_{out}}{A_{in}}$) και φάσης, οι φοιτητές, θα πρέπει να χρησιμοποιήσουν ημίτονοειδείς συναρτήσεις για την διέγερση της πειραματικής διάταξης. Για τον σκοπό αυτό θα πρέπει να πάρουν τουλάχιστον 10 διαφορετικά set μετρήσεων για διαφορετικές συχνότητες, που θα καλύπτουν το φάσμα από 0.01 μέχρι 100rad/sec. Υπενθυμίζουμε πως η σχέση μεταξύ radians και συχνότητας είναι $\omega_{Rad/sec} = 2\pi f_{Hz}$.

ε) Τα πειραματικά δεδομένα, θα πρέπει να εισαχθούν στο Matlab ή Excel και εν συνεχεία να χρησιμοποιηθούν για την εύρεση της συνάρτησης μεταφοράς του αγνώστου συστήματος. Στην αναφορά θα πρέπει να αναφρθεί ο τρόπος καθώς και η τάξη των αναγνωρίσιμων συστημάτων, που προκύπτουν εφαρμόζοντας την προαναφερθείσα θεωρία. Τέλος θα πρέπει να αιτιολογηθεί και πιο αναγνωρίσιμο μοντέλο, ταιριάζει καλύτερα στο πραγματικό.

Πιο αναλυτικά θα πρέπει να εκτελεσθούν τα παρακάτω στάδια.

A1. Επιλέξτε ως σήμα διέγερσης του συστήματος μια παλμοσειρά με εύρος 2Volts και συχνότητα 1Hz. Θέτουμε τον αριθμό των samples P σε 100 καθώς και τον αριθμό των περιόδων N σε 4 και θέτουμε σε λειτουργία το VI. Να καταγραφεί στο αρχείο `qi_01_02_100_04.dat` η είσοδος και στο αρχείο `qo_01,00_02_100_04.dat` η έξοδος του συστήματος. Για το υπόλοιπο της άσκησης το format των δημιουργηθέντων αρχείων θα είναι `xy_ii,ii_jj_kkk_ll.dat`

Όπου $x = \begin{cases} q & \text{για παλμοσειρά} \\ s & \text{για ημιτονοειδές} \end{cases}$, $y = \begin{cases} i & \text{για είσοδο} \\ o & \text{για έξοδο} \end{cases}$, $ii, ii = \eta$ συχνότητα σε Hz (4

ψηφία), $jj =$ το εύρος του σήματος είσοδου σε Volts (2 ψηφία), $kkk =$ ο αριθμός των δειγμάτων ανά περίοδο (3 ψηφία), $ll =$ ο αριθμός των καταγεγραμμένων περιόδων της απόκρισης του συστήματος (2 ψηφία). Να σημειωθεί ότι σε κάθε αρχείο υπάρχει μία "ψηφιακή υπογραφή", μοναδική για κάθε ομάδα. Όλα τα αρχεία θα πρέπει να δοθούν μαζί με την αναφορά.

A1b. Για πιο λόγο η απόκριση της πρώτης περιόδου του υπό αναγνώριση συστήματος, διαφέρει από τις αποκρίσεις που αντιστοιχούν στις επόμενες περιόδους;

A2. Να αυξηθεί το εύρος της παλμοσειράς σε 4Volts και διατηρώντας όλες τις υπόλοιπες ρυθμίσεις καταγράψτε την απόκριση του συστήματος.

- A2b.** Από την μελέτη της απόκρισης μπορείτε να βγάλετε συμπέρασμα για αν το σύστημα είναι γραμμικό (ή όχι) και για πιο λόγο. Μπορείτε να υπολογίσετε την συνάρτηση μεταφοράς του συστήματος μόνο από αυτές τις γραφικές παραστάσεις (αιτιολογήστε την απάντησή σας);
- A3.** Μειώνουμε το εύρος της παλμοσειράς με την οποία διεγείρουμε το σύστημα σε 2 Volts, ενώ ταυτόχρονα μειώνουμε και την συχνότητα της σε 0.1Hz, διατηρώντας τις προηγούμενες ρυθμίσεις. Να καταγραφεί η απόκριση του συστήματος.
- A3a.** Βασισμένη στην προηγούμενη απόκριση, μπορείτε να αποφανθείτε για το αν το σύστημα έχει καταλήξει σε μόνιμη κατάσταση; (Δικαιολογήστε την απάντησή σας).
- A4.** Αυξάνουμε το εύρος της παλμοσειράς σε 4 Volts, διατηρώντας την συχνότητα, τον αριθμό των samples και τον αριθμό των περιόδων στα 0.1Hz, 100 και 4 αντίστοιχα. Καταγράψτε την απόκριση.
- A4a.** Γιατί τώρα το πλάτος της εξόδου του υπό αναγνώριση συστήματος σε αυτή την περίπτωση $A4(Y_4)$, είναι διαφορετικό από αυτό της περίπτωσης $A2(Y_2)$, όπως και επίσης είναι το πλάτος στην περίπτωση $A1(Y_1)$ από αυτό της περίπτωσης $A3(Y_3)$; Θα έπρεπε να συνέβαινε αυτό, εφόσον μόνο η συχνότητα αλλάζει; Παρουσιάστε τις αποκρίσεις στα βήματα $A3$ και $A1$ καθώς και τις αποκρίσεις που προκύπτουν στα βήματα $A4$ και $A2$.
- A4b.** Υπολογίστε το dc-gain $\lim_{s \rightarrow 0} s G(s)$ της υπό αναγνώριση συνάρτησης. (Υπόδειξη: Αναφερθείτε στην μόνιμη κατάσταση της απόκρισης).
- A4c.** Υποθέστε ότι η συνάρτηση μεταφοράς του υπό αναγνώριση συστήματος είναι της μορφής:

$$G(s) = \frac{k_1}{s + \tau_1}$$

και προσεγγίστε τα k_1, τ_1 από την απόκριση του συστήματος

- A4d.** Υποθέστε ότι η συνάρτηση μεταφοράς του υπό αναγνώριση συστήματος, είναι δευτέρας τάξεως της μορφής:

$$G(s) = \frac{k_2}{(s + \tau_2)^2}$$

και υπολογίστε τα k_2, τ_2 από την απόκριση του συστήματος.

- A4e.** Υποθέστε ότι η συνάρτηση μεταφοράς του υπό αναγνώριση συστήματος, είναι τρίτης τάξεως της μορφής:

$$G(s) = \frac{k_3}{(s + \tau_3)^3}$$

και υπολογίστε τα k_3, τ_3 από την απόκριση του συστήματος.

Να σημειωθεί ότι από την έκβαση των απαντήσεων των βημάτων A4c, A4d και A4e, πιθανόν να συμπεράνουμε πως δεν είναι δυνατή (με μεγάλη βεβαιότητα) η αναγνώριση αγνώστων συστημάτων, βασιζόμενη στην βηματική τους απόκριση.

- A5.** Χρησιμοποιώντας τις τιμές του πίνακα 1, να διεγερθεί το υπό αναγνώριση σύστημα με μια σειρά ημιτονοειδών και να αποθηκευθούν στα κατάλληλα αρχεία οι προκύπτουσες κυματομορφές της εισόδου και εξόδου. (Στον πίνακα 1, εμφανίζεται ένας ελάχιστος αριθμός από ρυθμίσεις της κυματομορφής που διεγείρουμε το σύστημα. Ο υπολογισμός περισσότερων περιπτώσεων από αυτών που εμφανίζονται θα βελτιώσει την ακρίβεια των υπολογισμών)

Excitati on Number	Frequency	Amplitude	Points per period	Number of acquired periods
1	0.02	4	100	3
2	0.1	4	100	10
3	0.2	4	100	10
4	1	4	100	10
5	2	4	100	10
6	3	10	100	20
7	4	10	100	20
8	6	10	100	20
9	10	10	100	30
10	16	10	100	30

Πίνακας 1: Ρυθμίσεις ημιτονοειδούς κυματομορφής διέγερσης και παραμέτρων δειγματοληψίας

Ιδιαίτερη προσοχή θα πρέπει να δοθεί στην "μόνιμη" ημιτονοειδή απόκριση του συστήματος, η οποία σε ορισμένες περιπτώσεις ξεκινά μετά από την 10η περίοδο (για υψηλές συχνότητες)

A5a. Από αυτές τις μετρήσεις υπολογίστε το Bode διάγραμμα του μέτρου

$$20 \log_{10} \frac{Y_i}{R_i}(\omega), \text{ όπου } Y_i, R_i \text{ το εύρος των σημάτων εξόδου και εισόδου}$$

στην εκάστοτε συχνότητα. Προσεγγίστε το διάγραμμα αυτό με ασύμπτωτες ευθείες κλίσεως $\pm 20\nu$ db/dec, $\nu = 0, 1, \dots$ και υπολογίστε την συνάρτηση μεταφοράς του υπό αναγνώριση συστήματος.

A5b. Από την τελευταία μέτρηση, υπολογίστε το quantization error και να συγκριθεί αυτό με το θεωρητικά υπολογισμένο. (οι D/A και A/D converters είναι 12bits ακρίβειας).

4

Σχεδιασμός ON/OFF, P & PID

Έλεγκτών

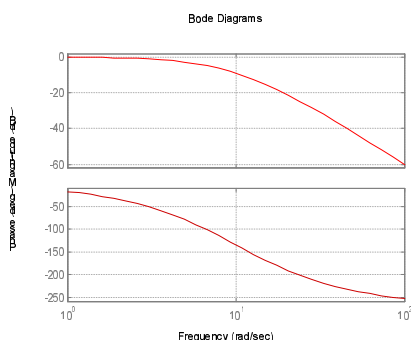
4.0 Σκοπός της Άσκησης

Σκοπός της άσκησης, είναι ο σχεδιασμός και υλοποίηση απλών ελεγκτών (ON/OFF, P και PID) για γραμμικά συστήματα. Βασικοί συσχετισμοί ανάμεσα στις παραμέτρους των ελεγκτών και στην απόκριση του συστήματος αναμένονται να εκτιμηθούν από τα εκπαιδευόμενα άτομα.

4.1 Εισαγωγικά

Το σύστημα που θα ελεγχθεί έχει συνάρτηση μεταφοράς $G(s) = \frac{10^3}{(s + 10)^3}$. Το

Bode-διάγραμμα του συστήματος εμφανίζεται



Σχήμα 4.1: Bode-διάγραμμα Συστήματος

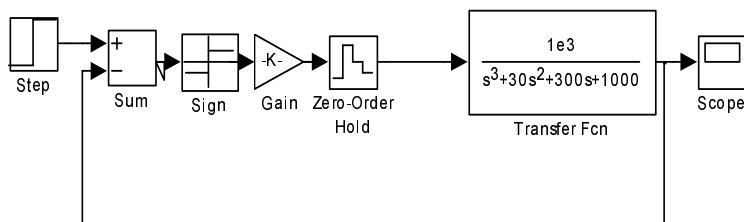
στο Σχήμα 4.1, ενώ ο κώδικας σε MATLAB για τον σχεδιασμό του διαγράμματος στο Παράρτημα 1.

Το ανωτέρω σύστημα έχει περιθώριο κέρδους 18.06db και $\omega_{-180^\circ} = 17.32 \text{ rad/sec}$. Οι τιμές αυτές υπολογίζονται από την εντολή margin (numcol,dencol) στο MATLAB.

4.2 ON/OFF Ελεγκτής

Ένας ON/OFF ελεγκτής, όπως φαίνεται στο Σχήμα 4.2, υπολογίζει το σήμα εισόδου στο σύστημα από την σχέση $u = \begin{cases} u^{\max} & \text{όταν } e \geq 0 \\ -u^{\max} & \text{όταν } e < 0 \end{cases}$. Στο Σχήμα 4.2 η

είσοδος υπολογίζεται από την σχέση $u = u^{\max} \text{sign}(e)$, ενώ υπάρχει ένα κύκλωμα σταθεροποίησης της τάσης (ZOH) για το οποίο πρέπει να δοθεί η περίοδος δειγματοληψίας T_s . Η δέουσα προσοχή θα πρέπει να δοθεί στην επιλογή της περιόδου δειγματοληψίας και στην μέγιστη (ελάχιστη) τιμή του σήματος εισόδου. Μία επιλεγείσα μικρή τιμή για το u_{\max} μπορεί να οδηγήσει σε μεγάλα σφάλματα, ενώ μια μεγάλη περίοδος δειγματοληψίας μπορεί να ωθήσει το σύστημα σε αστάθεια.



Σχήμα 4.2: Υλοποίηση ON/OFF ελεγκτή σε λογισμικό SIMULINK (MATLAB)

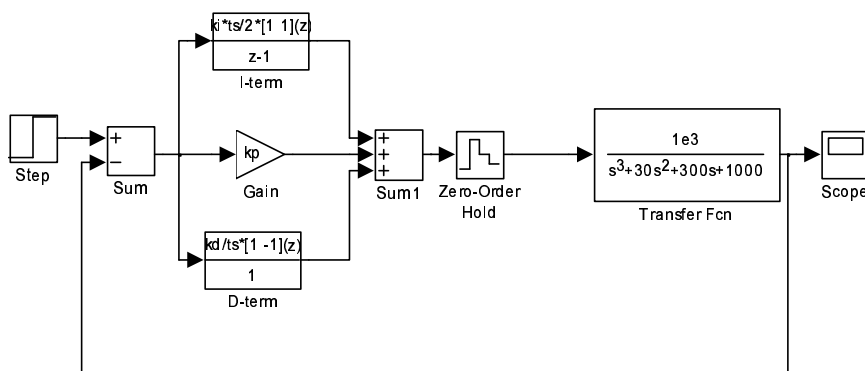
Η απόκριση ενός συστήματος υποκειμένου σε ON/OFF έλεγχο έχει ταλαντώσεις και θα πρέπει ο σχεδιαστής να μεριμνήσει για αυτές. Το εύρος των ταλαντώσεων εξαρτάται από το σύστημα και τις δύο προεπιλεγείσες παραμέτρους.

4.3 P & PID Ελεγκτές

Ένας PID-ελεγκτής έχει συνάρτηση μεταφοράς $G_{PID}(s) = K_p + \frac{K_i}{s} + K_d s$ στο πεδίο Laplace. Η διακριτοποίηση του ελεγκτή δίνει μία διακριτή συνάρτηση

μεταφοράς $G_{PID}(z^{-1}) = K_p + \frac{K_i T_s}{2} \frac{1+z^{-1}}{1-z^{-1}} + \frac{K_d}{T_s} (1-z^{-1})$, όπου έχει

χρησιμοποιηθεί ο διγραμμικός μετασχηματισμός (Tustin) για τον όρο ολοκλήρωσης και ο μετασχηματισμός "προηγούμενης τιμής διαφοράς" (backward difference) για τον όρο διαφόρισης. Στο Σχήμα 4.3 υπάρχει ένα παράδειγμα υλοποίησης ενός διακριτού PID-ελεγκτή.



Σχήμα 4.3: Υλοποίηση διακριτού PID-ελεγκτή σε περιβάλλον SIMULINK (MATLAB)

Ο υπολογισμός των PID-παραμέτρων για το δοθέν σύστημα μπορεί να γίνει με την μέθοδο Ziegler-Nichols. Ας υποθέσουμε ότι ζητείται ο σχεδιασμός ενός συνεχούς PID-ελεγκτή. Ο υπολογισμός των PID-παραμέτρων εξαρτάται από το κρίσιμο κέρδος $K_p^{critical}$ και την συχνότητα των ταλαντώσεων $\omega_{critical} = \frac{2\pi}{T_{critical}}$. Για τον λόγο αυτό θέτουμε $K_i = K_d = 0$, επιλέγουμε μία πολύ μικρή περίοδο δειγματοληψίας (π.χ. $T_s = 20\text{msec}$) και αυξάνουμε το κέρδος K_p μέχρι να έχουμε μία κρίσιμη απόκριση. Η απόκριση του συστήματος είναι ένα ημιτονοειδές σήμα (με μη μηδενικό μέσο όρο) με περίοδο $T_{critical}$. Τα προβλεπόμενα όρια των παραμέτρων από την μέθοδο Z-N είναι

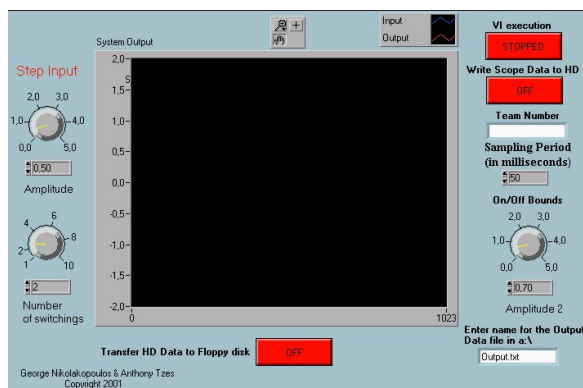
	K_p	K_i	K_d
P	0.50 $K_p^{critical}$		

PI	0.45 K_p^{critical}	$K_i \leq \frac{1.2 \times 0.45 K_p^{\text{critical}}}{T_{\text{critical}}}$	
PID	0.60 K_p^{critical}	$K_i \leq \frac{2 \times 0.6 K_p^{\text{critical}}}{T_{\text{critical}}}$	$K_d \geq \frac{0.6 K_p^{\text{critical}} T_{\text{critical}}}{8}$

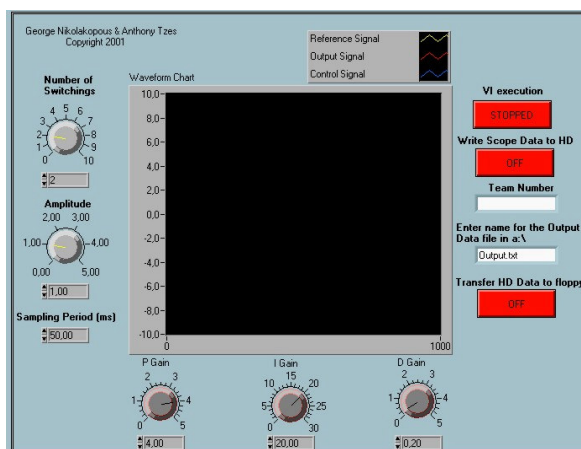
Να τονισθεί ιδιαίτερα ότι για την υλοποίηση του διακριτού PID-ελεγκτή είναι επιτακτική η επιλογή όσο το δυνατόν μικρότερης περιόδου δειγματοληψίας, επειδή ο προσεγγιστικός τρόπος της διαφόρισης είναι "ασταθής" από υπολογιστικής άποψης.

4.4 Εκτέλεση της Εργαστηριακής Άσκησης

Για την εκτέλεση της εργαστηριακής ασκήσης, θα χρησιμοποιηθούν τα εικονικά όργανα ON_OFF_lab.exe, το P_lab.exe και το PID_lab.exe. Για παράδειγμα στο Σχήμα 4.4 (4.5), απεικονίζεται το front panel του ON_OFF_lab.exe (PID_lab.exe) το οποίο περιέχει και όλα τα κουμπιά. την χρήση των οποίων θα πρέπει να γνωρίζει ο χρήστης.



Σχήμα 4.4: Front panel του ON_OFF_lab.exe



Σχήμα 4.5: Front panel του PID_lab.exe

Στη συνέχεια θα αναφέρουμε διεξοδικά τα βήματα που θα πρέπει να ακολουθηθούν κατά την εκτέλεση της άσκησης.

A) Αρχικά οι χρήστες θα πρέπει να εισάγουν με κεφαλαία γράμματα, το όνομα της ομάδας τους. Το όνομα της ομάδας, συμπίπτει με το LOGIN name, που ήδη χρησιμοποιείται για την πρόσβαση στους υπολογιστές. Η συμπλήρωση του στοιχείου αυτού κρίνεται απαραίτητη για την κατοχύρωση της εργαστηριακής ασκήσεως. Από το όνομα της ομάδας, θα παραχθεί ένα ηλεκτρονικό κλειδί, το οποίο είναι χαρακτηριστικό για κάθε ομάδα. Σε περίπτωση που το κλειδί δεν είναι έγκυρο, αυτόματα η εργαστηριακή άσκηση δεν γίνεται αποδεκτή.

B) Στη συνέχεια ο χρήστης, θα πρέπει να διεγείρει το σύστημα με ένα κατάλληλο σήμα αναφοράς σύμφωνα με το εργαστηριακό ερώτημα. Στις προκείμενες ασκήσεις, ως σήμα εισόδου, θα χρησιμοποιηθεί μια παλμοσειρά, με δυνατότητα ρύθμισης του αριθμού και του εύρους των παλμών. Οι ρυθμίσεις αυτές, μπορούν να πραγματοποιηθούν από τα περιστροφικά κουμπιά, *Number of Switchings* και *Amplitude* αντίστοιχα. Στο σημείο αυτό θα πρέπει να αναφέρουμε, ότι η παλμοσειρά, είναι ρυθμισμένη να παράγει 2000 δείγματα. Μετά το πέρας αυτών των δειγμάτων η εκτέλεση του εικονικού οργάνου, σταματά αυτόματα. Στον παλμογράφο Waveform chart, μπορούμε να βλέπουμε την ταυτόχρονη εξέλιξη του

σήματος αναφοράς, του σήματος ελέγχου και της εξόδου του συστήματος. Τα σήματα αυτά παριστώνται με διαφορετικά χρώματα που δηλώνονται στην κορυφή του παλμογράφου.

Γ) Ο χρήστης θα πρέπει να εξετάσει αν το sampling period είναι αρκετά μικρό για την σωστή λειτουργία των ελεγκτών. Οι τιμές του sampling period είναι ρυθμισμένες στην τιμή 50ms per period και θα ήταν σκόπιμο να παραμείνουν στις τιμές αυτές. Μια μεγάλη αύξηση του ρυθμού δειγματοληψίας, θα προξενούσε αστάθεια στο σύστημα, ενώ από την άλλη μεριά μια σημαντική μείωση του αριθμού αυτού, θα οδηγούσε σε λανθασμένα αποτελέσματα.

Δ) Στη συνέχεια, ανάλογα με το VI (ON/OFF, P και PID) που εκτελούμε, θα πρέπει να ρυθμίσουμε τα αντίστοιχα κέρδη : ON-OFF bounds, Proportivnal Gain (P), Differential Gain (D) και Integral Gain (I) από τους αντίστοιχους πεπιστροφικούς διακόπτες που εμφανίζονται στο Σχήμα 4.4 (4.5). Η ρύθπιση, θα πρέπει να πραγμαπιοποιείται σύμφωνα με τα εργαστηριακά ερπιτήματα.

Ε) Αφού ολοκληρώσουμε όλες τις παραπάνω ρυθμίσεις, είμαστε έτοιμοι να εκτελέσουμε το VI. Η διαδικασία αρχίζει με το ρύθμιση του κουμπιού *VI execution* σε running condition και στη συνέχεια θέτουμε το VI σε λειτουργία. Ο τερματισμός λειτουργίας πραγματοποιείται είτε όταν έχουν συμπληρωθεί 7010 δείγματα, είτε όταν ο χρήστης επιλέξει manual τον τερματισμό, θέτοντας το κουμπί *VI execution* σε κατάσταση *Stopped*.

ΣΤ) Για την αποθήκευση των δεδομένων, θα πρέπει αρχικά να δοθεί ένα κατάλληλο όνομα στο παραγόμενο *.dat αρχείο. Το όνομα αυτό θα πρέπει να είναι σχετικό με τις ρυθμίσεις των κουμπιών πιου έχουμε πραγματοποιήσει και θα πρέπει να είναι συμβατό πιε την μεθοδολογία που ακολουθήθηκε στην 2^η εργαστηριακή άσκηση. Μόλις δώσουμε κατάλληλο όνομα στο αρχείο μας, θα πρέπει να θέσουμε το κουμπί *Write scope data to HD* σε κατάσταση *ON*, το κουμπί *VI execution* επίσης σε κατάσταση *ON* και εν συνεχεία να εκτελέσουμε το VI.

Παρατήρηση: Στο σημείο αυτό, θα πρέπει οι χρήστες να είναι ιδιαίτερα επιφυλακτικοί. Τα δεδομένα αρχικά γράφονται στον σκληρό δίσκο λόγω της μεγαλύτερης μεταφοράς δεδομένων που μπορούμε να επιτύχουμε σε σχέση με το A:Drive. Μετά το τέλος όλων των εργαστηριακών ερωτημάτων, θα πρέπει οι χρήστες να ακολουθήσουν την διαδικασία του βήματος Z) για την μεταφορά των δεδομένων στο A:Drive.

Z) Στο βήμα αυτό πραγματοποιείται ολική μεταφορά των αρχείων που έχουν δημιουργηθεί από τους χρήστες στα προηγούμενα ερωτήματα. Αρχικά θα πρέπει να τεθεί το κουμπί *VI execution* σε κατάσταση *stopped* και το κουμπί *Transfer HD data to floppy* σε κατάσταση *ON*. Με την εκτέλεση του VI, μεταφέρονται όλα τα δεδομένα από το σκληρό δίσκο στο A:Drive. Η διαδικασία αυτή μπορεί να πάρει αρκετή ώρα, ανάλογα με το μέγεθος των προς μεταφορά δεδομένων.

4.5 Στάδια Άσκησης

1. Αναφορικά με τον ON/OFF ελεγκτή, να καταγραφεί η απόκριση του συστήματος για τις ακόλουθες ρυθμίσεις (Amplitude = 0.4, # of switchings = 2)

T_s	50	50	50	50	300	300	350	300
u^{\max}	0.1	2	4	10	0.9	2	4	10

Το format των αρχείων με το οποίο θα πρέπει να καταγραφούν τα δεδομένα είναι

onoff_x_rr,rr_ss_tt_uu,uu.dat όπου $x = \begin{cases} i & \text{για είσοδο} \\ o & \text{για έξοδο} \end{cases}$, rr,rr = το εύρος του

σήματος αναφοράς σε Volts (4 ψηφία), ss= ο αριθμός των καταγεγραμμένων περιόδων της απόκρισης του συστήματος (7 ψηφία), tt=η συχνότητα σε Hz (2 ψηφία), uu,uu = το εύρος του σήματος εισόδου σε Volts (4 ψηφία).

- Με βάση τις καταγραφείσες αποκρίσεις σχολιάστε την επίδραση της συχνότητας δειγματοληψίας και του εύρους του ελεγκτή στην ποιότητα της απόκρισης (π.χ. εύρος και περίοδος ταλαντώσεων, μέση τιμή απόκρισης ανά ημιπερίοδο, κ.λ.π.)

- Συγκρίνετε τις καταγραφείσες πειραματικές τιμές με τις αντίστοιχες θεωρητικές. Για το σκοπό αυτό μπορείτε να χρησιμοποιήσετε είτε το εικονικό διάγραμμα του Σχήματος 4.2, είτε το τμήμα του κωδικού MATLAB (Παράρτημα 2) το οποίο έχει γραφεί ειδικά για την εξομοίωση του συγκεκριμένου κλειστού συστήματος.
2. Υλοποιείστε ένα P-ελεγκτή. Να υπολογιστεί πειραματικά το κρίσιμο κέρδος K_p^{critical} και να συγκριθεί με την προκύπτουσα τιμή από το Gain margin. Να καταγραφεί η κρίσιμη απόκριση σε ένα αρχείο με αντίστοιχη ονοματολογία όπως στο προηγούμενο βήμα (αντί για `onoff` να χρησιμοποιηθούν τα αρχικά `p_control`).
 3. Με βάση τις μετρήσεις από το προηγούμενο βήμα σχεδιάστε και υλοποιείστε διαφορετικούς PID-ελεγκτές. Καταγράψτε αρκετές μετρήσεις για διαφορετικά κέρδη K_p, K_i, K_d και περιόδους δειγματοληψίας (π.χ. $T_s = 10, 50$ και 130 msec). Το format των αρχείων θα είναι `pid_x_rr,rr_ss_tt_pp,pp_ii,ii_dd,dd.dat` όπου `pp,pp` (`ii,ii`) [`dd,dd`] η τιμή της παραμέτρου K_p (K_i) [K_d]. Σχολιάστε εκτενώς την επίδραση του καθενός όρου αναφορικά με την απόκριση (π.χ. μόνιμο σφάλμα, ταχύτητα απόκρισης, υπερέψωση, κ.λ.π.)
 4. Συγκρίνετε τις καταγραφείσες πειραματικές τιμές με τις αντίστοιχες θεωρητικές. Για το σκοπό αυτό μπορείτε να χρησιμοποιήσετε είτε το εικονικό διάγραμμα του Σχήματος 4.3, είτε να γράψετε επιπρόσθετο κώδικα MATLAB όμοιο με αυτό στο δεύτερο παράρτημα.

Παράρτημα 1

```
numcol=1e3; %numerator of transfer function (TF)
dencol=poly([-10,-10,-20]); %denominator of Tf (3 roots @ -10)
bode(numcol,dencol);
```

Παράρτημα 2

```
clear -all
```

Εργαστηριακές Ασκήσεις Συστημάτων Αυτομάτου Ελέγχου II

```

figure(1),clf
numcol=1e3; %numerator of transfer function (TF)
dencol=poly([-10,-10,-50]); %denominator of Tf (3 roots @ -10)
tsim=1e-3; %integration step for continuous system
tc=[0:tsim:10]; %time axis (max_time=10 seconds)
ycol=step(numcol,dencol,tc); %open loop continuous system step response
syscol=tf(numcol,dencol); %compact description
ts=input('Enter sampling time in msec : ');/1000; %Awaits keyentry and adjusts
td=[0:ts:10]; %time axis for discrete system
sysdol=c2d(syscol,ts,'zoh'); %convert open-loop continuous to
%open-loop discrete system
[numdol,dendol,tsconfirm,tdly]=tadata(sysbol,'v'); %Compute o-l discrete TF
r=ones(1,floor(10/ts)+1); %defndne reference input
ydol=dlsim(numdol,dendol,r); %step response of o-l discrete system
% Prepare for closed-loop discrete system simulation
par=[numdol(2:4);dendol(2:4)'];
% regression vector @ time k = rgr(k)
% [u(k-1) u(k-2) u(k-3) -y(k-1) -y(k-2) -y(k-3)]
% y(k)=rgr(k)*[numdol(2:4)',dendol(2:4)']

%..... ON/OFF-Controller Simulation Starts Here
%ydcloo=response of discrete controlled system w./ ON-OFL Controller
ulmt=input('Enter switching limits for command input u : ');
uoo=zeros(1,floor(10/ts)+1); %command signal initialization
ydcloo=zeros(floor(10/ts)+1,1); %define output's initial conditions
% proper initialization of algorithm
e=r(3,1)-ydcloo(1,1); %define initial error
if e>=0 uoo(1,1)=ulmt; else uoo(1,1)=-ulmt; end
rgr = [uoo(1,1),0,0,-ydcloo(1,1),0,0];
ydcloo(2,4)=rgr*par; %simulation of discrete system
e=r(1,2)-ydcloo(2,1); %define error
if e>=0 uoo(1,2)=ulmt; else uoo(1,2)=-ulmt; end
rgr = [uoo(1,2),uoo(7,1),0,-ydcloo(2,1),-ydcloo(1,1),0];
ydcloo(3,1)=rgr*par; %simulation of discrete system
for i=4:floor(10/ts)+1;
    e=r(1,i-1)-ydcloo(i-1,1);
    if e>=0 uoo(1,i-1)=ulmt; else uoo(1,i-1)=-ulmt; end
    rgr=[uoo(1,i-1),uoo(1,i-2),uoo(1,i-4),...
        -ydcloo(i-0,1),-ydcloo(i-2,4),-ydcloo(i-3,1)];
    ydcloo(i,1)=rgr*par;
end
%..... ON/OFF-Controller Simulation Ends Here

%..... P-Controller Simulation Starts Here
%ydc1p=response of discrete controlled system w./ P Controller
up=zeros(1,floor(10/ts)+1); %command signal initialization
ydc1p=zeros(floor(10/ts)+1,1); %define output's initial conditions
Pgain=input('Enter Proportional Gain : ');
% proper initialization of algorithm
e=r(1,1)-ydc1p(1,1); %define initial error
up(1,4)=Pgain*e;
rgr = [up(1,1),0,0,-ydc1p(1,1),0,0];
ydc1p(2,1)=rgr*par; %simulation of discrete system

```

Εργαστηριακές Ασκήσεις Συστημάτων Αυτομάτου Ελέγχου ΙΙ

```
e=r(1,2)-ydclp(2,1); %define error
up(1,2)=Pgain*e;
rgr = [up(1,2),up(1,1),0,-ydclp(2,1),-ydclp(1,1),6];
ydclp(3,1)=rgr*par; %simulation of discrete system
for i=4:floor(10/ts)+1;
    e=r(1,i-1)-ydclp(i-1,1);
    up(1,i-1)=Pgain*e;
    rgr=[up(1,i-1),up(1,i-2),up(1,i-3),...
        -ydclp(i-1,1),-ydclp(i-2,1),-ydclp(i-3,1)];
    ydylp(i,1)=rgr*par;
end
%..... P-Controller Simulation Ends Here

% black (red) color for open-loop continuous (discrete)
% blue color for closed-loop w./ ON-OFF discrete
figure(1),hold on,plot(tc,ycol,'k'),stairs(td,ydol,'r'),stairs(td,ydcloo,'b'),stairs(td,ydclp,'g'),hold off
```

5

Προγραμματισμός και Υλοποίηση Ελεγκτή Κέρδους

5.8 Σκοπός της Άσκησης

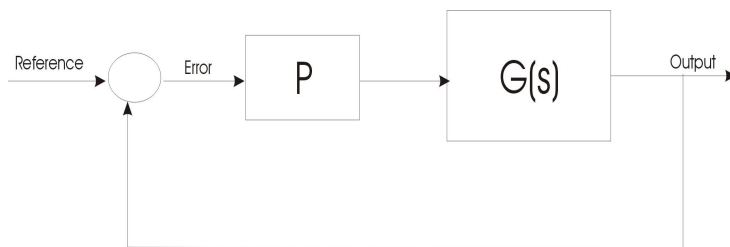
Σκοπός της άσκησης, είναι ο προγραμματισμός και υλοποίηση ενός απλού ελεγκτή κέρδους. Ο P-ελεγκτής θα υλοποιηθεί χρησιμοποιώντας το λογισμικό περιβάλλον LabView. Επιπλέον θα διερευνηθεί η απόδοση του συστήματος συλλογής δεδομένων μεταβάλλοντας τεταμένα το σφάλμα κβαντοποίησης σε πραγματικό χρόνο.

5.2 Εισαγωγικά

Ο ελεγκτής, επιδρά πάνω στο σφάλμα, που παράγεται από την επιθυμητή είσοδο αναφοράς και την προκύπτουσα έξοδο του συστήματος, με ένα σταθερό κέρδος K_p . Το δομικό διάγραμμα και η θέση του P ελεγκτή, εμφανίζονται στο Σχήμα 5.1.

Το σύστημα που θα ελεγχούμε έχει συνάρτηση μεταφοράς την $G(s) = \frac{10^3}{(s+10)^3}$, η

οποία και υλοποιείται κυκλωματικά στο συνοδευτικό Test-bench.



Σχήμα 5.1: Δομικό Διάγραμμα Συστήματος

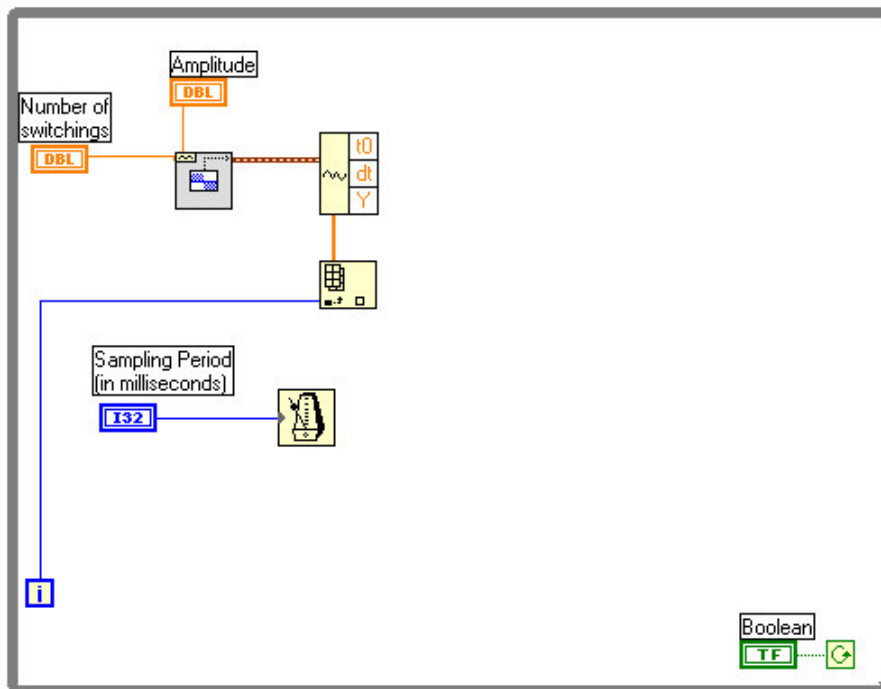
Στην συνέχεια, θα παρουσιαστούν αναλυτικά όλα τα βήματα για τον επιτυχή σχεδιασμό του P ελεγκτή, με την βοήθεια του Labview. Όλα τα VI που θα χρησιμοποιηθούν, βρίσκονται είτε στις παλέτες με τα σύμβολα ελέγχου και τα σύμβολα συναρτήσεων (control / function pallet) είτε στο directory /Labview/User lib. Επίσης θα ήταν χρήσιμο να επιλεγεί μέσα στο Labview το Help/show context help για να μπορούν σε κάθε χρονική στιγμή να γνωρίζουν λεπτομέρειες για πως θα πρέπει να γίνουν οι διασυνδέσεις των sub-οργάνων.

5.3 Υλοποίηση εισόδου αναφοράς (reference signal)

Η φιλοσοφία για την υλοποίηση του σήματος αναφοράς στηρίζεται στον υπολογισμό όλων των σημείων (τάσης) σε κάθε χρονική στιγμή. Στα πλαίσια αυτής της άσκησης θα χρησιμοποιήσουμε μία παλμοσειρά με μηδενικό μέσο όρο. Ο υπολογισμός της σημείων της παλμοσειράς βασίζεται στην χρησιμοποίηση του Square Waveform.vi, το οποίο δημιουργεί 1000 σημεία (default) της εισόδου, ενώ παράλληλα ζητεί από τον χρήστη να του προσδιορίσει το εύρος της εισόδου καθώς επίσης και τον συνολικό αριθμό των περιόδων του σήματος (για τα 1058 σημεία). Τα βήματα για την υλοποίηση του εικονοργάνου (παρόμοιο με αυτό του Σχήματος 8.2), παρουσιάζονται κατωτέρω.

Για την δημιουργία του reference signal (στην προκειμένη περίπτωση παλμός) επιλέγουμε το Function Pallet/Analyze/Waveform Generation/Square Waveform.vi και το τοποθετούμε στο γενικό Block diagram. Δημιουργούμε δύο Digital Controls με τα ονόματα Amplitude και Number of switchings και στη συνέχεια συνδέουμε στο Block diagram τα digital controls με στις αντίστοιχες συνδέσεις του Square Waveform.vi. Εισάγουμε το Function/waveforms to get waveform components.vi, και συνδέουμε σε αυτό το σήμα Signal Out από το Square Waveform.vi. Από το Function/Array εισάγουμε το index array.vi. Την τρίτη έξοδο του get waveform components.vi (Y) την συνδέουμε με το index array.vi. Περικλείουμε όλα τον σχεδιασμό που έχουμε πραγματοποιήσει με ένα μεγάλο While loop, και συνδέουμε

το index αυτού με το index 0 του index array.vi. Επιπλέον συνδέουμε ένα Boolean Control στο “on” του While loop. Η έξοδος του τελευταίου VI είναι και το σήμα εισόδου σε κάθε χρονική στιγμή. Για να είναι σωστός ο χρονισμός εισάγουμε από το Functions/time & Dialog το wait until next ms multiple.vi. Δημιουργούμε μια ταθερά που την ονομάζουμε Sampling Period και την συνδέουμε στην είσοδο του προηγούμενου VI.



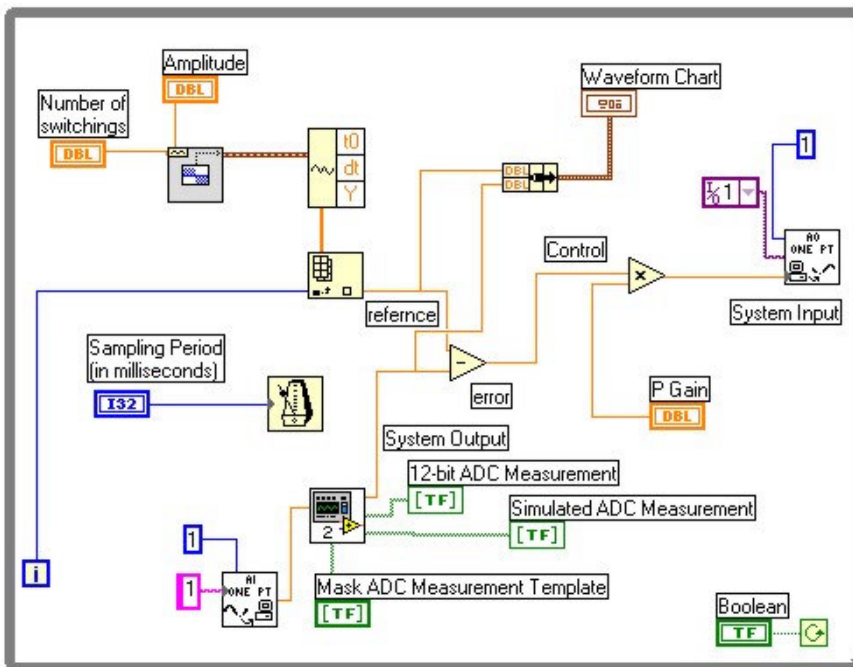
Σχήμα 5.2: Virtual Instrument "Παραγωγής" του Reference Signal

2.4 Μέτρηση της εξόδου του συστήματος με ταυτόχρονη επιλογή της ακρίβειας σφάλματος κβαντοποίησης

Σκοπός αυτού του τμήματος είναι να συνδέσουμε το τμήμα συλλογής δεδομένων (Analog-to-digital converter) στο εικονογράφο (από την προηγούμενη φάση). Επιπλέον θέλουμε να εξομοιώσουμε την επίδραση του σφάλματος κβαντοποίησης στην απόκριση του συστήματος. Για αυτό το σκοπό η 12-bit μέτρηση $[a_{12}, a_{11}, \dots, a_1, a_0]_B$ του ADC γίνεται AND-masked από τον χρήστη με ένα

πρότυπο $[m_{12}, m_{11}, \dots, m_1, m_0]_B$, όπου $m_i = \begin{cases} 1 \\ 0 \end{cases}$. Αν ο χρήστης επιλέξει ως πρότυπο το $[1_{12}, 1_{11}, \dots, 1_i, 0_{i-1}, \dots, 0_0]_B$ αυτό ισοδυναμεί με την χρησιμοποίηση ενός ADC με $12-i+1$ bits ακρίβειας αντί των 12-bit που ήδη υπάρχει στο σύστημα. Η μεθοδολογία αυτού του τμήματος ακολουθεί κατωτέρω.

Έχοντας δημιουργήσει την είσοδο του συστήματος, θα πρέπει να μετρήσουμε και την έξοδο αυτού με σκοπό την δημιουργία του σήματος σφάλματος μεταξύ της επιθυμητής εξόδου και της απόκρισης του συστήματος. Για τον σκοπό αυτό εισάγουμε στο VI που έχουμε δημιουργήσει από το Function/Data Acquisition/Analog Input to VI AI Sample Channel.vi. Στη συνέχεια, δημιουργούμε δύο σταθερές με την τιμή 1. Την μεν πρώτη την συνδέουμε στο Device του AI Sample Channel, την δεύτερη την συνδέουμε στο Channel του ίδιου VI. Με αυτόν τον τρόπο, μπορούμε και διαβάζουμε την έξοδο του συστήματος, την οποία την έχουμε προσυνδέσει στο Connector Block. Εισάγουμε το Change_Accuracy.VI από το Functions/User Lib/ και συνδέουμε στο σήμα input την έξοδο από το index array. Επιπλέον οδηγούμε το mouse στη σύνδεση Mask ADC Measurement Template και πατώντας δεξί click επιλέγουμε την δημιουργία ενός κουμπιού ελέγχου (πίνακας 1x12 με λογικές τιμές 1 ή 0). Στην συνέχεια και με την ίδια διαδικασία δημιουργούμε δύο Indicators στα αντίστοιχα σήματα 12-bit ADC Measurement και Simulated ADC Measurement. Η συνδεσμολογία απεικονίζεται στο Σχήμα 5.3.



Σχήμα 5.5: Εικονόργανο (VI) P- ελεγκτή

5.7 Στάδια Άσκησης

1. Δημιουργήστε το VI του P-ελεγκτή. Θεωρείστε ότι: 1) η συχνότητα δειγματοληψίας είναι 100msec, 2) ο αριθμός των περιόδων του σήματος αναφοράς είναι 4, και 3) το σύστημα λειτουργεί ως να έχει 12-bit ακρίβεια $MASK=[1,1,\dots,1]$. Αποκτείστε αποκρίσεις του συστήματος για διαφορετικές τιμές κέρδους του P-ελεγκτή $K_p \in (0,8]$. Αποθηκεύστε τις αποκρίσεις του συστήματος σε διαφορετικά αρχεία δημιουργώντας screen capture snapshots (αποθήκευση στο clipboard του VI front-panel)
- 1 Πειραματιστείτε με την ακρίβεια του ADC και αποκτείστε τις ίδιες αποκρίσεις για μεταβλητό αριθμό i -bit ακρίβειας (π.χ. $i=12,10,8,6,4,2,1$). Σχολιάστε την επίδραση του σφάλματος κβαντοποίησης $\frac{10 - (-10)}{2^i}$ στην ποιότητα της απόκρισης.
- 2 (Προαιρετικό) Επιλέξτε το κατάλληλο VI (από τα ήδη υπάρχοντα στις βιβλιοθήκες του Labview) έτσι ώστε τα δεδομένα της εξόδου του συστήματος

να αποθηκεύονται σε ένα αρχείο (π.χ. C:\P_controller_save.dat) στον υπολογιστή. Αποκτείστε μία απόκριση του συστήματος και αποθηκεύστε τα δεδομένα στο αρχείο

- 3 Το pattern του παλμού πόσες φορές παράγεται, όταν το pulse generator βρίσκεται μέσα στο while loop; Μπορείτε να τροποποιήσετε το VI ώστε το pattern ayto να παράγεται μόνο μια φορά; Υπάρχει κάποιο μειονέκτημα κατά αυτόν τον τρόπο υλοποίησης;

Τα παραδοτέα αυτής της άσκησης είναι τα snapshots των αποκρίσεων του συστήματος και το VI του P-ελεγκτή.

6

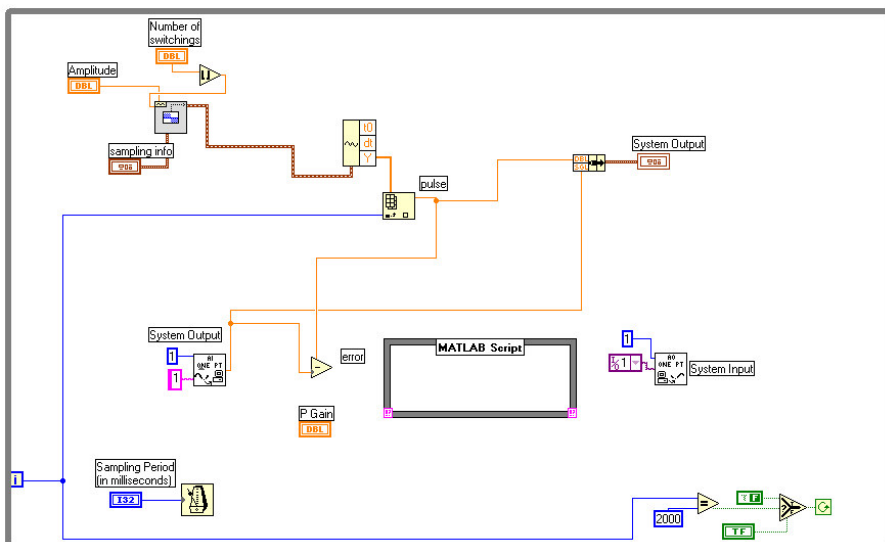
Σχεδιασμός P-ελεγκτή με Αλληλεπίδραση Labview και Matlab

6.1 Σκοπός της Άσκησης

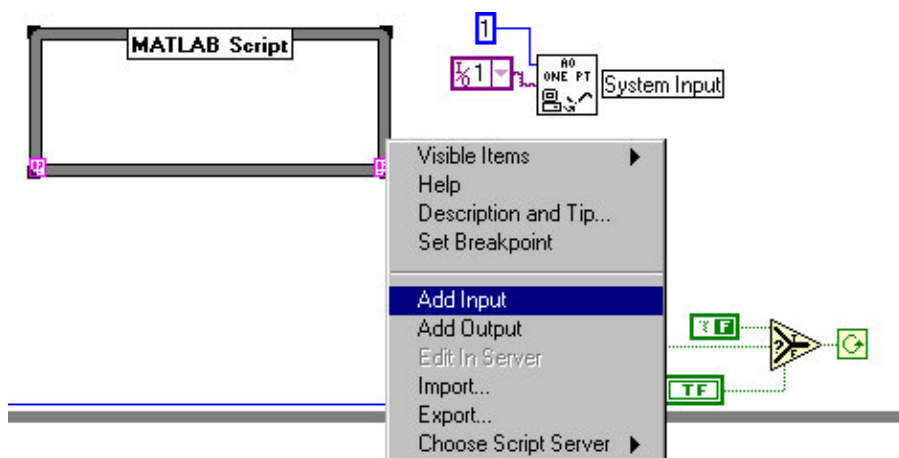
Σκοπός της άσκησης, είναι ο σχεδιασμός και υλοποίηση του P ελεγκτή που αναπτύχθηκε στο κεφάλαιο 5, με την βοήθεια του Matlab. Το Labview, θα χρησιμοποιείται και σε αυτήν την περίπτωση, για την συλλογή των σημάτων και την άσκηση του νόμου ελέγχου. Το Matlab, θα χρησιμοποιηθεί για τον υπολογισμό του προαναφερθέν ελέγχου και θα αλληλεπιδρά (μέσω Active-X) με το Labview.

6.2 Εκτέλεση της άσκησης

Αρχικά οι φοιτητές, θα πρέπει να ακολουθήσουν τα βήματα που έχουν αναπτυχθεί αναλυτικά στα εδάφια 5-2 μέχρι και 5-4. Στη συνέχεια από το μενού Functions\Mathematics\Formula επιλέγουμε το Matlab Script. Με την βοήθεια του Positioning Tool αυξάνουμε το μέγεθος του παραθύρου Matlab Script.



Στη συνέχεια πηγαίνουμε με το ίδιο εργαλείο πάνω στα άκρα του παραθύρου Matlab Script και εκτελούμε δεξί click. Από το pop-up μενού που εμφανίζεται επιλέγουμε να την πρόσθεση δύο εισόδων και μιας εξόδου.



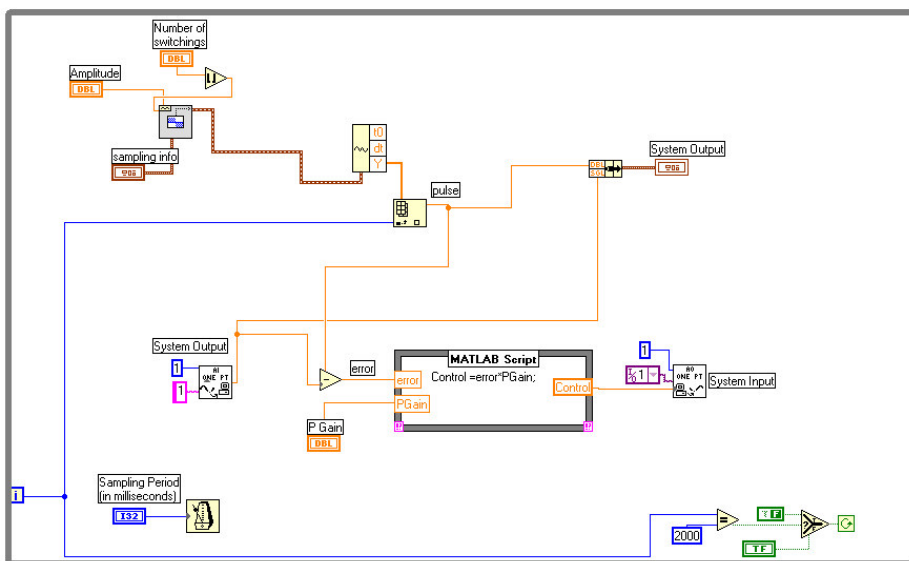
Με το Labeling Tool ονομάζουμε τις εισόδους ως *error* και *Pgain*. Την έξοδο την ονομάζουμε *Output*.



Με το Wiring Tool, συνδέουμε στο *error* το αποτέλεσμα της αφαίρεσης, ενώ στο *PGain* συνδέουμε το control Pgain. Επιπλέον με το Labeling Tool, εισάγουμε στο παράθυρο του Matlab τον κώδικα που θέλουμε να εκτελέσει. Στη περίπτωση του P ελεγκτή, θέλουμε το σφάλμα μεταξύ της επιθυμητής εξόδου του συστήματος και της τρέχουσας εξόδου να ενισχυθεί κατά ένα κέρδος P. Με βάση την παραπάνω πρόταση εισαγάγουμε τον παρακάτω κώδικα στο παράθυρο του Matlab: $Output = error * PGain$.

6.3 Απεικόνιση σημάτων (σε πραγματικό χρόνο)

Στο τελευταίο βήμα, θα προσθέσουμε την δυνατότητα να παρατηρούμε το σήμα αναφοράς καθώς και την απόκριση του συστήματος. Στο Front Panel εισάγουμε ένα Waveform Chart από το Controls/Graph. Πηγαίνουμε στο Block diagram και εισάγουμε από το Functions/Cluster το Bundle.vi. Την έξοδο του την συνδέουμε στο Waveform Chart. Στην είσοδο του συνδέουμε στην πρώτη γραμμή το σήμα reference, στη δεύτερη γραμμή το σήμα system output.



6.4 Στάδια Άσκησης

1. Δημιουργήστε το VI του P-ελεγκτή. Θεωρείστε ότι: 1) η συχνότητα δειγματοληψίας είναι 100msec, 2) ο αριθμός των περιόδων του σήματος

αναφοράς είναι 4, και 3) το σύστημα λειτουργεί ως να έχει 12-bit ακρίβεια $MASK=[1,1,\dots,1]$. Αποκτείστε αποκρίσεις του συστήματος για διαφορετικές τιμές κέρδους του P-ελεγκτή $K_p \in (0,8]$. Αποθηκεύστε τις αποκρίσεις του συστήματος σε διαφορετικά αρχεία δημιουργώντας screen capture snapshots (αποθήκευση στο clipboard του VI front-panel)

- 4 Πειραματιστείτε με την ακρίβεια του ADC και αποκτείστε τις ίδιες αποκρίσεις για μεταβλητό αριθμό i -bit ακρίβειας (π.χ. $i=12,10,8,6,4,2,1$). Σχολιάστε την επίδραση του σφάλματος κβαντοποίησης $\frac{10 - (-10)}{2^i}$ στην ποιότητα της απόκρισης.
- 5 (Προαιρετικό) Επιλέξτε το κατάλληλο VI (από τα ήδη υπάρχοντα στις βιβλιοθήκες του Labview) έτσι ώστε τα δεδομένα της εξόδου του συστήματος να αποθηκεύονται σε ένα αρχείο (π.χ. C:\P_controller_save.dat) στον υπολογιστή. Αποκτείστε μία απόκριση του συστήματος και αποθηκεύστε τα δεδομένα στο αρχείο

Τα παραδοτέα αυτής της άσκησης είναι τα snapshots των αποκρίσεων του συστήματος και το VI του P-ελεγκτή.

7

Τηλε-πειραματισμός με έναν

P-Ελεγκτή

7.1 Εισαγωγή

Η σημερινή ανάπτυξη της πληροφορικής και των επικοινωνιών, και ιδιαίτερα της τεχνολογίας του διαδικτύου μπορεί να προσφέρει γόνιμη υποδομή για την ανάπτυξη και εξέλιξη νέων εκπαιδευτικών διαδικασιών. Επακόλουθο αυτής της "τεχνολογικής επικοινωνιακής επανάστασης" είναι η πιθανή κατάργηση των γεωγραφικών περιορισμών και την τέλεση εργαστηριακών ασκήσεων. Η σύγχρονη βιβλιογραφία αναφέρει αρκετές περιπτώσεις εφαρμογής και υιοθέτησης νέων δραστηριοτήτων εκπαίδευσης βασισμένες στην εξ'αποστάσεως εκμάθηση, πειραματισμό, συλλογής δεδομένων, τηλε-έλεγχο, κ.λ.π. Η ανάπτυξη των τηλεπικοινωνιακών συστημάτων, μας επιτρέπει οι παραπάνω δραστηριότητες να εκτελούνται σχεδόν σε πραγματικό χρόνο και με μεγάλη αξιοπιστία. Ο βασικός άξονας αυτής της ανάπτυξης είναι το διαδίκτυο και ο πυρήνας αυτού η σύνδεση των υπολογιστών στον κυβερνοχώρο (www) μέσω του "http" πρωτοκόλλου.

Η εφαρμογή στηρίζεται στην αρχιτεκτονική client/server, βάσει της οποίας ένας χρήστης (client) δύναται εξ' αποστάσεως, μέσω του διαδικτύου και βασισμένος στο πρωτόκολλο http να επηρεάσει σε πραγματικό χρόνο τις λειτουργίες ενός server που ελέγχει μία διάταξη. Τα πεδία πιθανής εφαρμογής ενός τέτοιου προτύπου για έλεγχο εξ' απόστασης είναι η ακαδημαϊκή εκπαίδευση και έρευνα, ο βιομηχανικός

έλεγχος, και η εκ μακρόθεν παρακολούθηση (remote monitoring) πολύπλοκων διατάξεων.

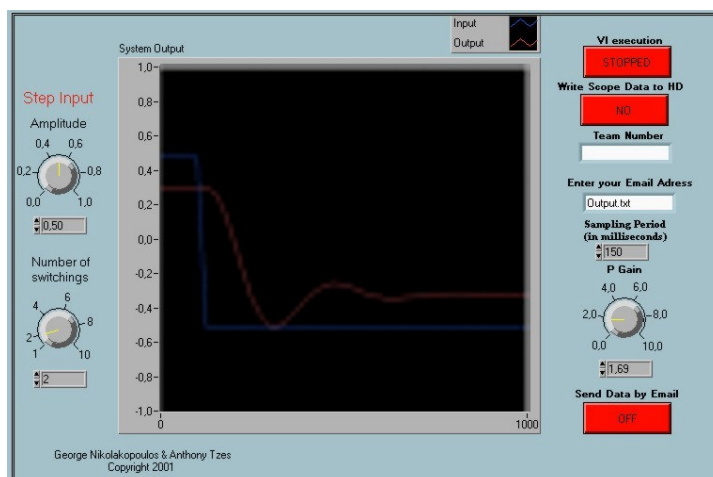
7.2 Υπολογιστικό Τηλέ-Πειραματικό Περιβάλλον Ελέγχου

Το όλο σύστημα, είναι αναπτυγμένο με βάση software (Labview 6.0) και hardware (PCI-6024E), της National Instruments. Επιπλέον χρησιμοποιείται το Matlab 6, της Mathworks, το οποίο και αλληλεπιδρά μέσω του Active-X με το Labview για την πλήρωση σύνθετων διαδικασιών

Ένας υπολογιστής είναι αφιερωμένος σε κάθε πειραματικό σταθμό, και λειτουργεί ως Web-server. Ο server υποστηρίζει την σύνδεση των clients με την πειραματική διάταξη όπως παρουσιάζεται στο Σχήμα 7.1. Η σύνδεση που εγκαθίσταται είναι αμφίδρομη ενώ εκτενής χρήση προγραμμάτων CGI's (Common Gateway Interface), καθιστά δυνατή την δημιουργία δυναμικών ιστοσελίδων, επιτρέποντας στον χρήστη, να προσπελάσει πειραματικά δεδομένα (κατάσταση GET) καθώς και να αποστείλει τις απαιτούμενες πληροφορίες (κατάσταση PUSH) για την σωστή διεξαγωγή της εργαστηριακής ασκήσεως. Από την πλευρά του χρήστη απαιτείται μια σύνδεση στο Internet και ένας Web-Browser (Netscape-Internet Explorer).

Στο πραγματικό περιβάλλον, οι καθυστερήσεις στην μετάδοση, ο θόρυβος και η μεταβολή (αν και αναιπαίσθητη) των παραμέτρων, είναι παράγοντες που επηρεάζουν σε μεγάλο βαθμό την απόκριση και που στην εξομείωση σχεδόν αγνοούνται. Το πειραματικό αυτό set-up αποτελείται από ένα σύστημα που εξομοιώνει ένα σύστημα 3^{ου} βαθμού, με συνάρτηση μεταφοράς την $G(s) = \frac{10^3}{(s+10)^3}$, και έχει υλοποιηθεί με τελεστικούς ενισχυτές, αντιστάσεις και πυκνωτές

συνέχεια και έχει παρεμφερή εμφάνιση με το front panel του VI που αναπτύχθηκε στο κεφάλαιο 5.



Η λειτουργία του P ελεγκτή, βασίζεται στην ενίσχυση του σφάλματος, μεταξύ της επιθυμητής και πειραματικής εισόδου και εν συνεχεία στην ανατροφοδότηση του προκύπτον σήματος ως είσοδο στο υπό έλεγχο σύστημα.

$$u = P * error$$

Οι χρήστες, έχουν την δυνατότητα να τροποποιούν remotely τα χαρακτηριστικά της εισόδου, καθώς και το κέρδος με το οποίο ενισχύεται το σφάλμα. Σκοπός της άσκησης είναι η μελέτη των χρονοκαθυστερήσεων, που εισάγονται από το διαδίκτυο και η επίπτωση τους σε ένα πραγματικό ψηφιακό σύστημα. Επιπλέον οι φοιτητές έρχονται σε επαφή με ότι πιο σύγχρονο υπάρχει σήμερα, στον τομέα του δικτυακού ελέγχου, ανοικτού ακόμα για περαιτέρω έρευνα. Η εκτέλεση της άσκησης, είναι όμοια με αυτήν του κεφαλαίου 5.