ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ - ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΤΟΜΕΑΣ: Τομεας Συστηματων και Αυτοματου Ελεγχου
ΕΡΓΑΣΤΗΡΙΟ: Εργαστηριο Αυτοματου Ελεγχου

Διπλωματικη Εργασια

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών

Μιχαηλ-Άγγελου Τριανταφυλλη του Άγγελου
(σε γενικη πτωση)

ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: 228214

Θέμα

Υλοποίηση αυτο-ρυθμιζόμενων PID ελεγκτών με χρήση Labview

Επιβλέπων

Επίκουρος Καθηγητής Καζάκος Δημοσθένης

Πάτρα, Ιανουάριος 2018

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η διπλωματική εργασία με θέμα

Υλοποίηση αυτο-ρυθμιζόμενων PID ελεγκτών με χρήση Labview

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Μιχαήλ-Άγγελου Τριανταφύλλη του Άγγελου
(σε γενική πτώση)

(Α.Μ.: 228214)

παρουσιάτηκε δημόσια και εξετάστηκε στο τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών στις

__/__/___

Ο Επιβλέπων

Καζάκος Δημοσθένης
Επίκουρος Καθηγητής

Ο Διευθυντής του Τομέα

Κούσουλας Νικόλαος
Καθηγητής

Στοιχεία διπλωματικής εργασίας

Θέμα: Υλοποίηση αυτο-ρυθμιζόμενων PID ελεγκτών με χρήση Labview

Φοιτητής: Μιχαήλ-Άγγελος Τριανταφύλλης του Άγγελου
(σε ονομαστική πτώση)

Ομάδα επίβλεψης
Επίκουρος Καθηγητής Καζάκος Δημοσθένης
Βαθμίδα και Ονοματεπώνυμο Συνεπιβλέποντα
Ονοματεπώνυμο Διδακτορικού Φοιτητή

Εργαστήρια
Εργαστήριο Αυτομάτου Ελέγχου

Περίοδος εκπόνησης της εργασίας:
Μήνας Έτος - Μήνας Έτος

Η εργασία αυτή γράφτηκε στο XeLaTeX και χρησιμοποιήθηκε η γραμματοσειρά GFS Didot του Greek Font Society.

Περίληψη

Η εργασία αυτή ασχολείται με την αυτόματη ρύθμιση (self-regulation) PID ελεγκτών. Παρόλο που οι PID ελεγκτές αποτελούν ένα πολύ διαδεδομένο είδος ελεγκτών με ευρεία χρήση σε βιομηχανικές, και όχι μόνο, εφαρμογές η σωστή ρύθμιση τους απαιτεί εμπειρία από το χειριστή και συνήθως αποτελεί χρονοβόρα διαδικασία. Μέσω προσομοίωσης στο περιβάλλον Labview γίνεται προσπάθεια να αυτοματοποιηθεί η διαδικασία αυτή και να φανεί ποιοι είναι οι περιορισμοί ενός αυτο-ρυθμιζόμενου PID ελεγκτή.

Ευχαριστίες

Όσο κι αν φαίνεται σαν ατομική δουλειά η παρούσα εργασία, στην πραγματικότητα βοήθησαν αρκετοί άνθρωποι (ο καθένας με το δικό του τρόπο) για να ολοκληρωθεί.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ

Η εργασία αυτή έχει γίνει προσπάθεια να γραφεί σε ανεξάρτητα κεφάλαια, τα οποία θα δώσουν στον αναγνώστη τις απαιτούμενες γνώσεις ώστε να καταλάβει σε βάθος τις τεχνικές που χρησιμοποιούνται. Σε κάθε κεφάλαιο γίνεται αναλυτική παρουσίαση των τεχνικών καθώς και του υπόβαθρου που πρέπει να έχει κάποιος ώστε τις κατανοήσει, ωστόσο θεωρείται πως ο αναγνώστης έχει ήδη κάποιες γνώσεις στο χώρο του αυτομάτου ελέγχου και στην ανάλυση συστημάτων. Έτσι, βασικές έννοιες και μηχανισμοί της ανωτέρω περιοχής θα θεωρούνται δεδομένοι και δε θα γίνει κάποια ανάλυσή τους στο κείμενο αυτό, εκτός αν κρίνεται απαραίτητο.

ΚΕΦΑΛΑΙΟ

1

LABVIEW

1.1 Εισαγωγή στο LabVIEW

Το λογισμικό που χρησιμοποιήθηκε για την υλοποίηση αυτής της εργασίας είναι το LabVIEW από την εταιρία National Instruments (NI). Συνεπώς κρίνεται χρήσιμη μια σύντομη αναφορά σε αυτό και στον τρόπο που λειτουργεί. Καθώς το LabVIEW είναι ένα πολύ διαδεδομένο λογισμικό, με ευρεία χρήση στους κλάδους των μηχανικών, κάποιος που θέλει περισσότερες πληροφορίες μπορεί να τις βρει εύκολα στο διαδίκτυο. Κάποιες πηγές που χρησιμοποιήθηκαν για τη συγγραφή αυτής της εργασίας είναι [1], [2] και [3]. Το LabVIEW (Laboratory Virtual Instrument Engineering Workbench) είναι ένα περιβάλλον ανάπτυξης για μία οπτική γλώσσα προγραμματισμού. Σε αντίθεση με τα κοινά προγραμματιστικά περιβάλλοντα, στο LabVIEW δε χρησιμοποιείται κώδικας για να γραφτούν οι εντολές που θα εκτελεστούν αλλά γραφικά όπως κουτιά και σύμβολα. Για παράδειγμα, υπάρχουν πολλές οπτικές γλώσσες, που είναι γνωστές σαν γλώσσες ροής δεδομένων (dataflow), που βασίζονται στην ιδέα ”τετράγωνα και βέλη” (”boxes and arrows”), όπου τα τετράγωνα (ή άλλου τύπου αντικείμενα) της οθόνης θεωρούνται οντότητες που συνδέονται από βέλη, γραμμές ή ακμές, που αναπαριστούν σχέσεις μεταξύ τους.

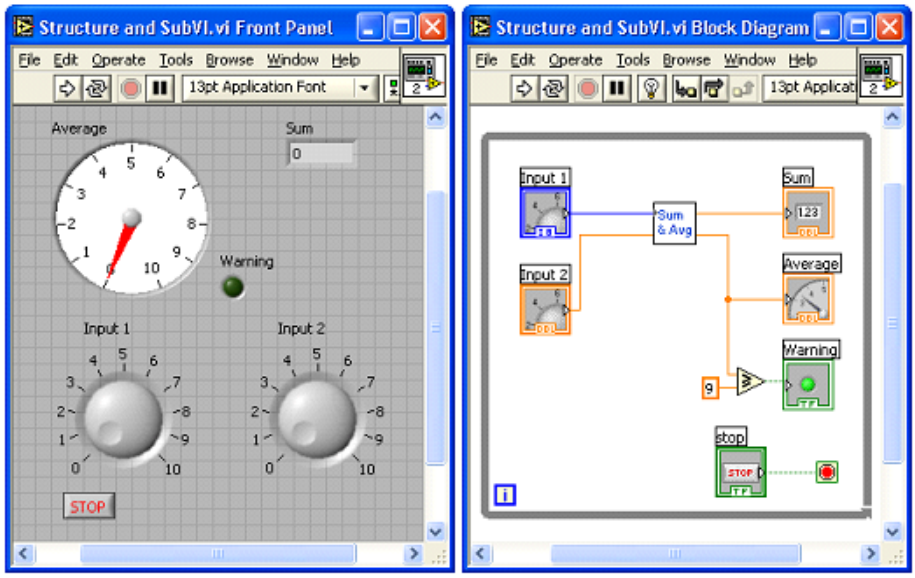
Dataflow Programming Η οπτική γλώσσα προγραμματισμού του LabVIEW ονομάζεται “G” και βασίζεται στη λογική του dataflow προγραμματισμού που αναφέρθηκε προηγουμένως. Αυτό σημαίνει ότι αν υπάρχουν αρκετά δεδομένα διαθέσιμα σε μία συνάρτηση ή ένα subVI (σύνολο συναρτήσεων)

τότε αυτή η συνάρτηση ή το subVI θα εκτελεστεί. Η ροή της εκτέλεσης του προγράμματος καθορίζεται από τη δομή ενός γραφικού μπλοκ διαγράμματος (block diagram), που στην ουσία αποτελεί τον πηγαίο κώδικα του LabVIEW. Σε αυτό ο προγραμματιστής συνδέει διαφορετικές συναρτήσεις-κόμβους (function-nodes) τραβώντας καλώδια. Αυτά τα καλώδια διαδίδουν τις μεταβλητές και κάθε κόμβος μπορεί να εκτελεστεί μόλις όλα τα δεδομένα στην είσοδό του είναι διαθέσιμα. Δεδομένου ότι αυτό μπορεί να συμβαίνει για πολλαπλούς κόμβους ταυτόχρονα, το LabVIEW μπορεί να εκτελεστεί εγγενώς παράλληλα. Περισσότερα για το dataflow programming μπορείτε να βρείτε στο [4].

Graphical Programming Το LabVIEW ενσωματώνει τη δημιουργία διεπαφών χρήστη, που ονομάζονται εμπρόσθιοι πίνακες (front panels) στον κύκλο ανάπτυξης. Τα προγράμματα-υπορουτίνες LabVIEW ονομάζονται εικονικά όργανα (VIs). Κάθε VI διαθέτει τρία στοιχεία: ένα block diagram, ένα front panel και ένα πάνελ σύνδεσης (connection panel). Το τελευταίο χρησιμοποιείται για να αντιπροσωπεύει το VI στα block diagrams άλλων, καλώντας τα VI. Το front panel κατασκευάζεται με χειριστήρια (controls) και δείκτες (indicators). Τα controls είναι είσοδοι: επιτρέπουν σε ένα χρήστη να παρέχει πληροφορίες στο VI. Τα indicators είναι έξοδοι: υποδηλώνουν ή εμφανίζουν τα αποτελέσματα με βάση τις εισόδους που δίδονται στο VI. Το πίσω πλαίσιο, το οποίο είναι ένα block diagram, περιέχει τον γραφικό πηγαίο κώδικα. Όλα τα αντικείμενα που τοποθετούνται στο front panel εμφανίζονται στην πίσω πλευρά ως τερματικά (terminals). Ο πίσω πίνακας περιέχει επίσης δομές και λειτουργίες οι οποίες εκτελούν εργασίες στα controls και παρέχουν δεδομένα στα indicators. Οι δομές και οι λειτουργίες βρίσκονται στην παλέτα λειτουργιών και μπορούν να τοποθετηθούν στον πίσω πίνακα. Οι συλλογικοί έλεγχοι, οι δείκτες, οι δομές και οι λειτουργίες θα αναφέρονται ως κόμβοι. Οι κόμβοι συνδέονται μεταξύ τους με τη χρήση καλωδίων, π.χ. δύο controls και μια ενδεικτική λυχνία μπορούν να συνδεθούν με τη λειτουργία προσθήκης (addition function) έτσι ώστε η ένδειξη να εμφανίζει το άθροισμα των δύο controls. Έτσι, ένα VI μπορεί να λειτουργήσει είτε ως πρόγραμμα, με τον μπροστινό πίνακα να λειτουργεί ως διεπαφή χρήστη, είτε, όταν πέσει ως κόμβος στο block diagram, το front panel ορίζει τις εισόδους και εξόδους του κόμβου μέσω του παραθύρου σύνδεσης. Αυτό σημαίνει ότι κάθε VI μπορεί εύκολα να δοκιμαστεί πριν να ενσωματωθεί ως υπορουτίνα σε ένα μεγαλύτερο πρόγραμμα.

Η γραφική προσέγγιση επιτρέπει επίσης στους μη προγραμματιστές να χτίσουν προγράμματα με μεταφορά και απόθεση εικονικών αναπαραστάσεων του εργαστηριακού εξοπλισμού με τον οποίο είναι ήδη εξοικειωμένοι. Το περιβάλλον προγραμματισμού LabVIEW, με τα παραδείγματα και την τεκμηρίωση που περιλαμβάνονται, καθιστά απλή τη δημιουργία μικρών εφαρμογών. Αυτό είναι ένα πλεονέκτημα από τη μια πλευρά, αλλά υπάρχει επίσης ο κίνδυνος να υποτιμηθεί η εμπειρογνωμοσύνη που

απαιτείται για τον προγραμματισμό G υψηλής ποιότητας. Για σύνθετους αλγορίθμους ή κώδικα μεγάλης κλίμακας, είναι σημαντικό ο προγραμματιστής να έχει εκτεταμένη γνώση της σύνταξης του LabVIEW και της τοπολογίας της διαχείρισης μνήμης της. Τα πιο εξελιγμένα συστήματα ανάπτυξης LabVIEW προσφέρουν τη δυνατότητα δημιουργίας αυτόνομων εφαρμογών. Επιπλέον, είναι δυνατή η δημιουργία κατανεμημένων εφαρμογών, οι οποίες επικοινωνούν με ένα μοντέλο πελάτη-εξυπηρετητή και έτσι είναι ευκολότερο να εφαρμοστούν λόγω της εγγενώς παράλληλης φύσης της προγραμματιστικής γλώσσας G. Ένα τυπικό περιβάλλον προγραμματισμού στο LabVIEW φαίνεται στο σχήμα 1.1



(a) (b)

Σχήμα 1.1: LabVIEW: Front Panel και Block Diagram

ΚΕΦΑΛΑΙΟ

2

PID ΕΛΕΓΚΤΕΣ

2.1 Εισαγωγή στους PID ελεγκτές

Ένας αναλογικός-ολοκληρωτικός-παραγωγικός ελεγκτής (proportional-integral-derivative controller) ή όπως είναι πιο γνωστός PID controller, είναι ένας μηχανισμός ανάδρασης (feedback) βρόχου ελέγχου (control loop) που χρησιμοποιείται ευρέως σε βιομηχανικά συστήματα ελέγχου καθώς και σε μια ποικιλία άλλων εφαρμογών που απαιτούν συνεχή διαμορφωμένο έλεγχο. Η διαδικασία λειτουργίας είναι κοινή για όλους τους ελεγκτές αυτού του είδους. Ένας PID ελεγκτής υπολογίζει συνεχώς μια τιμή σφάλματος $e(t)$ ως διαφορά μεταξύ μιας επιθυμητής τιμής ρύθμισης (setpoint ή SP) και μεταξύ μιας μεταβλητής της διαδικασίας ύπο έλεγχο (process value ή PV) και εφαρμόζει μια διόρθωση βασισμένη στον αναλογικό, ολοκληρωτικό και παραγωγικό όρο του (P, I, D αντίστοιχα) οι οποίοι δίνουν και στον ελεγκτή το όνομά του.

Στην πράξη, εφαρμόζει αυτόματα διορθωμένη και ακριβή διόρθωση σε μια λειτουργία ελέγχου. Ένα καθημερινό παράδειγμα είναι ο έλεγχος ταχύτητας σε οδικό όχημα. όπου εξωτερικές επιδράσεις, όπως κλίσεις, θα προκαλούσαν αλλαγές στην ταχύτητα του οχήματος. Ο αλγόριθμος PID επαναφέρει την ταχύτητα του αυτοκινήτου στην επιθυμητή από τον οδηγό τιμή της με τον βέλτιστο τρόπο, χωρίς καθυστέρηση ή υπέρβαση, ελέγχοντας την ισχύ εξόδου του κινητήρα του οχήματος.

2.1.1 Ιστορική αναδρομή

Προέλευση Ο συνεχής έλεγχος, προτού καταστούν πλήρως κατανοητοί και εφαρμοσμένοι οι ελεγκτές PID, έχει μία από τις πηγές του στον φυγοκεντρικό ρυθμιστή ο οποίος χρησιμοποιεί περιστρεφόμενα βάρη για να ελέγξει μια διαδικασία. Αυτό είχε εφευρεθεί από τον Christian Huygens τον 17ο αιώνα για να ρυθμίσει το χάσμα μεταξύ των μυλόπετρων στους ανεμόμυλους ανάλογα με την ταχύτητα περιστροφής και έτσι να αντισταθμίσει την μεταβλητή ταχύτητα της τροφοδότησης των σιτηρών [5], [6].

Με την εφεύρεση της σταθερής ατμομηχανής υψηλής πίεσης, υπήρχε ανάγκη για αυτόματο έλεγχο ταχύτητας και ο αυτοδιαμορφωμένος ρυθμιστής "κωνικού εκκρεμούς" του James Watt, ένα σύνολο περιστρεφόμενων χαλύβδινων σφαιρών προσαρτημένων σε κάθετο άξονα με βραχίονες σύνδεσης, έγινε πρότυπο της βιομηχανίας [7].

Ωστόσο, ο περιστρεφόμενος έλεγχος ταχύτητας του ρυθμιστή εξακολουθούσε να είναι μεταβλητός υπό συνθήκες μεταβαλλόμενου φορτίου, και έτσι το μειονέκτημα του ελέγχου που πλέον είναι γνωστός ως αναλογικός έγινε προφανές. Το σφάλμα μεταξύ της επιθυμητής ταχύτητας και της πραγματικής ταχύτητας αυξανόταν με την αύξηση του φορτίου. Τον 19^o αιώνα, η θεωρητική βάση για τη λειτουργία των ρυθμιστών περιγράφηκε για πρώτη φορά από τον James Clerk Maxwell το 1868. Εξερεύνησε τη μαθηματική βάση για τη σταθερότητα του ελέγχου και προχώρησε σε έναν καλό δρόμο προς μια λύση, αλλά έκανε μια έκκληση σε μαθηματικούς να εξετάσουν το πρόβλημα [7], [8]. Το πρόβλημα εξετάστηκε περαιτέρω από τον Edward Routh το 1874, τον Charles Sturm και το 1895 από τον Adolf Hurwitz, που όλοι συνέβαλαν στην καθιέρωση κριτηρίων σταθερότητας ελέγχου [7]. Στην πράξη, οι ρυθμιστές ταχύτητας βελτιώθηκαν περαιτέρω, κυρίως από τον αμερικανικό επιστήμονα Willard Gibbs, ο οποίος το 1872 ανέλυσε θεωρητικά τον κωνικό κυβερνήτη εκκρεμούς του Watt.

Περίπου εκείνη την εποχή, η εφεύρεση της τορπίλης Whitehead έθεσε ένα πρόβλημα ελέγχου το οποίο απαιτούσε ακριβή έλεγχο του βάθους λειτουργίας. Η χρήση μόνο ενός αισθητήρα πίεσης βάθους αποδείχθηκε ανεπαρκής και έτσι ένα εκκρεμές που μετρούσε το εμπρόσθιο και οπίσθιο βήμα της τορπίλης συνδυάστηκε με τη μέτρηση βάθους για να γίνει ο έλεγχος εκκρεμούς και υδροστάτη (pendulum-and-hydrostat control). Ο έλεγχος πίεσης παρείχε μόνο ένα αναλογικό έλεγχο, το οποίο αν το κέρδος ελέγχου ήταν πολύ υψηλό, θα ήταν ασταθές και θα έπεφτε σε υπέρβαση, με σημαντική αστάθεια στη διατήρηση βάθους. Το εκκρεμές προσέθεσε αυτό που είναι τώρα γνωστό ως παράγωγο έλεγχο (derivative control), το οποίο εξασθένισε τις ταλαντώσεις ανιχνεύοντας τη γωνία κατάδυσης / ανόδου της τορπίλης και επομένως τον ρυθμό μεταβολής του βάθους [9]. Αυτή η εξέλιξη (που ονομάστηκε από το Whitehead ως "Το Μυστικό" για να μην δώσει καμιά ένδειξη για τη δράση της) ήταν περίπου το 1868 [10].

Ένα άλλο πρώιμο παράδειγμα ελεγκτή τύπου PID αναπτύχθηκε

από τον Elmer Sperry το 1911 για την πλοήγηση, αν και το έργο του ήταν διαισθητικό και όχι μαθηματικό [11].

Η πρώτη θεωρητική ανάλυση και πρακτική εφαρμογή αφορούσε το αυτόματο σύστημα διεύθυνσης πλοίων, το οποίο αναπτύχθηκε από τις αρχές της δεκαετίας του 1920 και μετά από τον μηχανικό Nicolas Minorsky [12]. Ο Minorsky ερευνούσε και σχεδίαζε την αυτόματη καθοδήγηση πλοίων για το Πολεμικό Ναυτικό των ΗΠΑ και βάσισε την ανάλυσή του στις παρατηρήσεις ενός πηδαλιούχου. Σημείωσε ότι ο πηδαλιούχος κατεύθυνε το πλοίο με βάση όχι μόνο το τρέχον σφάλμα πορείας, αλλά και το λάθος του παρελθόντος, καθώς και τον τρέχοντα ρυθμό αλλαγής [13]. Αυτό μοντελοποιήθηκε μαθηματικά από τον Minorsky [7]. Ο στόχος του ήταν η σταθερότητα, όχι ο γενικός έλεγχος, ο οποίος απλοποίησε σημαντικά το πρόβλημα. Ενώ ο αναλογικός έλεγχος παρείχε σταθερότητα έναντι μικρών διαταραχών, ήταν ανεπαρκής για να αντιμετωπίσει μια σταθερή διαταραχή (λόγω σφάλματος σταθερής κατάστασης), η οποία απαιτούσε την προσθήκη του ολοκληρωτικού όρου. Τέλος, ο παράγωγος όρος προστέθηκε για να βελτιώσει τη σταθερότητα και τον έλεγχο.

Διεξήχθησαν δοκιμές στο USS New Mexico, με τον ελεγκτή να ελέγχει τη γωνιακή ταχύτητα (όχι τη γωνία) του πηδαλίου. Ο έλεγχος PI απέδωσε σταθερή στροφή (γωνιακό σφάλμα) $\pm 2^\circ$. Η προσθήκη του στοιχείου D οδήγησε σε σφάλμα εκτροπής $\pm 1/6^\circ$, καλύτερα από ότι θα μπορούσαν να επιτύχουν οι περισσότεροι πηδαλιούχοι [14].

Το Πολεμικό Ναυτικό τελικά δεν υιοθέτησε το σύστημα, λόγω της αντίστασης του προσωπικού. Παρόμοια εργασία πραγματοποιήθηκε και δημοσιεύθηκε από αρκετούς άλλους στη δεκαετία του 1930.

Βιομηχανικός έλεγχος

2.1.2 Χρησιμότητα PID ελεγκτών

Ο PID ελεγκτής έχει διάφορα σημαντικά χαρακτηριστικά: παρέχει ανατροφοδότηση ελέγχου, έχει την ικανότητα να εξαλείφει το σφάλμα μόνιμης κατάστασης (steady-state error) μέσω του ολοκληρωτικού του όρου, μπορεί να προβλέπει το μελλοντικό σφάλμα μέσω του παραγωγικού του όρου. Οι PID ελεγκτές παρέχουν ικανοποιητικό έλεγχο σε πολλά προβλήματα ελέγχου, ειδικά όταν οι δυναμικές που διέπουν τη διεργασία είναι ήπιες και οι απαιτήσεις ελέγχου μέτριες. Οι ελεγκτές αυτού του είδους έρχονται σε αρκετές διαφορετικές μορφές. Υπάρχουν αυτόνομα συστήματα μέσα σε κουτιά για έναν ή περισσότερους βρόχους και παράγονται εκατοντάδες χιλιάδες μονάδες κάθε χρόνο. Ο PID έλεγχος είναι σημαντικό στοιχείο ενός κατανεμημένου συστήματος ελέγχου. Οι ελεγκτές είναι επίσης ενσωματωμένοι σε πολλά, ειδικού σκοπού, συστήματα ελέγχου. Στον έλεγχο διεργασιών (process control), περισσότερο από το 95% των βρόχων ελέγχου είναι PID τύπου, οι περισσότεροι βρόχοι είναι στην πραγματικό-

τητα PI ελέγχου. Ο PID έλεγχος χρησιμοποιείται στο χαμηλότερο επίπεδο: ο πολλαπλών μεταβλητών ελεγκτής (multivariable controller) δίνει το σημείο λειτουργίας στους ελεγκτές στο χαμηλότερο επίπεδο. Ο PID ελεγκτής μπορεί λοιπόν να ειπωθεί ότι αποτελεί το "ψωμί και βούτυρο" της μηχανικής ελέγχου. Είναι, συνεπώς, ένα σημαντικό στοιχείο στην εργαλειοθήκη κάθε μηχανικού ελέγχου.

ΚΕΦΑΛΑΙΟ

3

SUPER RESOLUTION ΜΕΣΩ IMAGE REGISTRATION

3.1 Μοντελοποίηση προβλήματος

Πριν δούμε στην πράξη πώς λειτουργούν οι τεχνικές super resolution θα πρέπει να περιγράψουμε το πρόβλημα με το οποίο θα δουλέψουμε με μαθηματικούς όρους. Η μοντελοποίηση αυτή θα μας επιτρέψει να χρησιμοποιήσουμε μαθηματικά εργαλεία και τεχνικές και να ορίσουμε σε μια αυστηρή "γλώσσα" (αυτή των μαθηματικών) τις λειτουργίες που επιτελούνται από κάθε μέθοδο, ούτως ώστε να πετύχουμε το τελικό αποτέλεσμα.

Ξεκινώντας, θα πρέπει να περιγράψουμε τον τρόπο με τον οποίο λαμβάνουμε εικόνες χαμηλής ανάλυσης από μια φυσική σκηνή μέσω μιας κάμερας. Η κάμερα σαν όργανο καταγραφής εισάγει κάποιες ατέλειες, όπως είδαμε στο προηγούμενο κεφάλαιο. Τέτοιες ατέλειες μπορεί να είναι σφάλματα καταγραφής από τον αισθητήρα της κάμερας, θόλωμα λόγω αστοχιών των οπτικών στοιχείων κλπ. Επιπλέον, αν λάβουμε διαδοχικές εικόνες μιας φυσικής σκηνής, οι εικόνες αυτές περιμένουμε να έχουν κάποιες μετατοπίσεις ως προς αυτό που απεικονίζουν. Οι μετατοπίσεις αυτές μπορεί να οφείλονται είτε στον άνθρώπινο παράγοντα (που χειρίζεται την κάμερα) είτε στο αντικείμενο της φυσικής σκηνής που μπορεί να μην είναι σταθερό. Στην εικόνα που λαμβάνεται τελικά, λαμβάνονται δείγματα σε χαμηλή χωρική συχνότητα και λόγω ατελειών του οργάνου μπορεί να έχουμε και παρουσία θορύβου. Για μια εικόνα υψηλής ανάλυσης X (την

οποία θα προσπαθήσουμε να ανακατασκευάσουμε), μπορούμε να αναπτύξουμε το παραπάνω μοντέλο ως εξής:

$$Y_i = S_i T_i H_i X + n_i \tag{3.1}$$

όπου ορίζουμε για την i-οστή εικόνα χαμηλής ανάλυσης Y_i τους τελεστές που ενεργούν στην X :

- S_i για υποδειγματοληψία,
- T_i για μετατόπιση,
- H_i για θόλωμα (blurring),
- n_i για προσθετικό θόρυβο.

Οι υποθέσεις που κάνουμε για το παραπάνω πρόβλημα είναι ότι το blurring είναι ίδιο σε όλο το χώρο και είναι γνωστό στον αλγόριθμο super resolution, ο θόρυβος είναι λευκός Gaussian με την ίδια διασπορά σε όλες τις εικόνες χαμηλής ανάλυσης και ότι ο γεωμετρικός μετασχηματισμός αφορά μόνο την καθολική μετατόπιση.

Στην περίπτωση που θεωρήσουμε αμελητέο θόρυβο και θόλωμα, τα παραπάνω βήματα αρκούν για να υπολογίσουμε την εικόνα υψηλής ανάλυσης X, την οποία λαμβάνουμε σαν αποτέλεσμα του αλγορίθμου.

Data: dx, dy, Y_i, N, W, H, S
Output: High resolution reconstructed X
$X \leftarrow 0$;
for $n \leftarrow 1$ to N do
  $\tilde{Y} \leftarrow Y_i$;
  $i \leftarrow 1 : W/S$;
  $j \leftarrow 1 : H/S$;
  $px = i * S + dx_n$;
  $py = j * S + dy_n$;
  $X_{px,py} = \tilde{Y}_{i,j}$;
end

Αλγόριθμος 1: Ανακατασκευή shift-add fusion

Συγκρίνοντας τη μέθοδο αυτή με την ανακατασκευή μέσω του μετασχηματισμού Fourier, παρατηρούμε την απλότητα και την αποδοτικότητά της καθώς βρίσκει την εικόνα υψηλής ανάλυσης μόνο με μετακινήσεις pixel.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] https://en.wikipedia.org/wiki/LabVIEW

[2] http://www.ni.com/en-us/shop/labview.html

[3] Jeffrey Travis, Jim Kring, "LabVIEW for Everyone: Graphical Programming Made Easy and Fun (3rd ed.)", Prentice Hall Professional, July 27, 2006, ISBN-13: 978-0-13-185672-1.

[4] Johnston, W.M., Hanna, J.R.P. and Millar, R.J. "Advances in dataflow programming languages". ACM Computing Surveys 36 (1): 1–34. 2004. doi: 10.1145/1013208.1013209

[5] Hills, Richard L (1996), "Power From the Wind", Cambridge University Press.

[6] Richard E. Bellman (December 8, 2015). "Adaptive Control Processes: A Guided Tour". Princeton University Press.

[7] Bennett Stuart (1996) "A brief history of automatic control", IEEE Control Systems Magazine, IEEE. 16 (3): 17–25. doi:10.1109/37.506394.

[8] Maxwell, J. C. (1868). "On Governors". Proceedings of the Royal Society. 100.

[9] Newpower, Anthony (2006). "Iron Men and Tin Fish: The Race to Build a Better Torpedo during World War II". Praeger Security International. ISBN 0-275-99032-X. p. citing Gray, Edwyn (1991), "The Devil's Device: Robert Whitehead and the History of the Torpedo", Annapolis, MD: U.S. Naval Institute, p. 33.

[10] Sleeman, C. W. (1880), "Torpedoes and Torpedo Warfare", Portsmouth: Griffin Co., pp. 137–138.

[11] "A Brief Building Automation History". Retrieved 2011-04-04.

[12] Minorsky, Nicolas (1922). "Directional stability of automatically steered bodies". J. Amer. Soc. Naval Eng. 34 (2): 280–309. doi:10.1111/j.1559-3584.1922.tb04958.x.

[13] Bennett 1993, p. 67

[14] Bennett, Stuart (June 1986). A history of control engineering, 1800-1930. IET. pp. 142–148. ISBN 978-0-86341-047-5.

Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών
Μιχαήλ-Άγγελος Τριανταφύλλης του Άγγελου
(σε ονομαστική πτώση)