

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)

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

\_\_/\_\_/\_\_

Ο Επιβλέπων

Ο Διευθυντής του Τομέα

Καζάκος Δημοσθένης  
Επίκουρος Καθηγητής

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  $\text{\LaTeX}$  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

**Ό**σο 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

1	LabVIEW	3
1.1	Εισαγωγή στο LabVIEW . . . . .	3
2	PID ελεγκτές	7
2.1	Εισαγωγή στους PID ελεγκτές . . . . .	7
2.1.1	Ιστορική αναδρομή . . . . .	8
2.1.2	Χρησιμότητα PID ελεγκτών . . . . .	11
2.2	Αρχές λειτουργίας . . . . .	12
2.2.1	Η Ανάδραση . . . . .	12
2.2.2	Αναλογικός Όρος . . . . .	13
2.2.3	Ολοκληρωτικός Όρος . . . . .	14
2.2.4	Παράγωγος Όρος . . . . .	16
2.2.5	Εξίσωση του PID ελεγκτή . . . . .	16
2.2.6	Τροποποιήσεις του PID αλγορίθμου . . . . .	18
3	Ρύθμιση του PID ελεγκτή	21
3.1	Εισαγωγή . . . . .	21
3.2	Τεχνικές ρύθμισης . . . . .	22
3.2.1	Χειροκίνητη ρύθμιση . . . . .	22
3.2.2	Μέθοδος Ziegler–Nichols . . . . .	23
	Βιβλιογραφία	25



# 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

**T**ο 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)

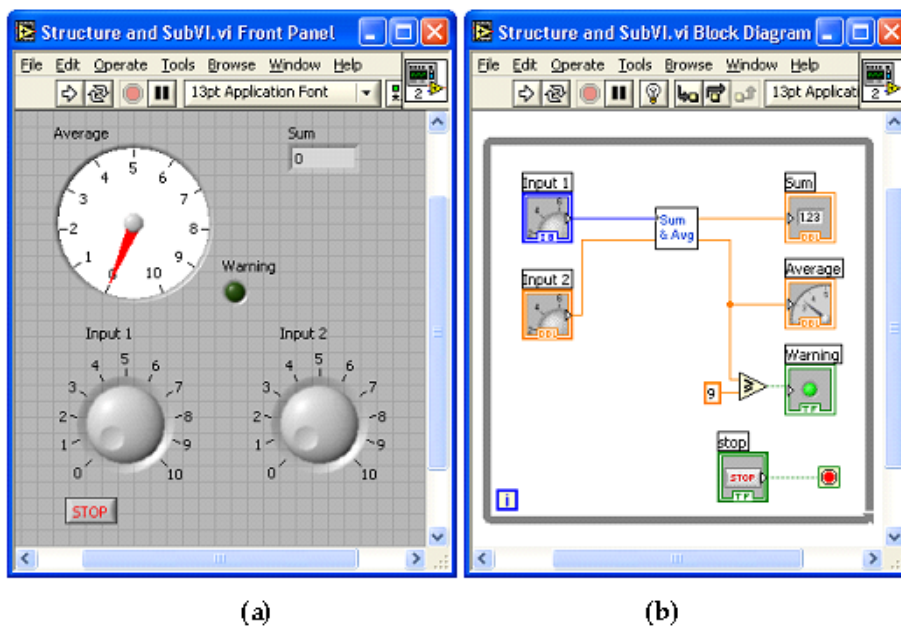
τότε αυτή η συνάρτηση ή το subVI θα εκτελεστεί. Η ροή της εκτέλεσης του προγράμματος καθορίζεται από τη δομή ενός γραφικού μπλοκ διαγράμματος (block diagram), που στην ουσία αποτελεί τον πηγαίο κώδικα του LabVIEW. Σε αυτό ο προγραμματιστής συνδέει διαφορετικές συναρτήσεις-κόμβους (function-nodes) τραβώντας 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 εμπειρογνομοσύνη 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



Σχήμα 1.1: LabVIEW: Front Panel και Block Diagram



## ΚΕΦΑΛΑΙΟ

# 2

## PID ΕΛΕΓΚΤΕΣ

### 2.1 Εισαγωγή στους PID ελεγκτές

**Ε**ΝΑΣ 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 υπό 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 διορθωμένα 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<sup>ο</sup> 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].

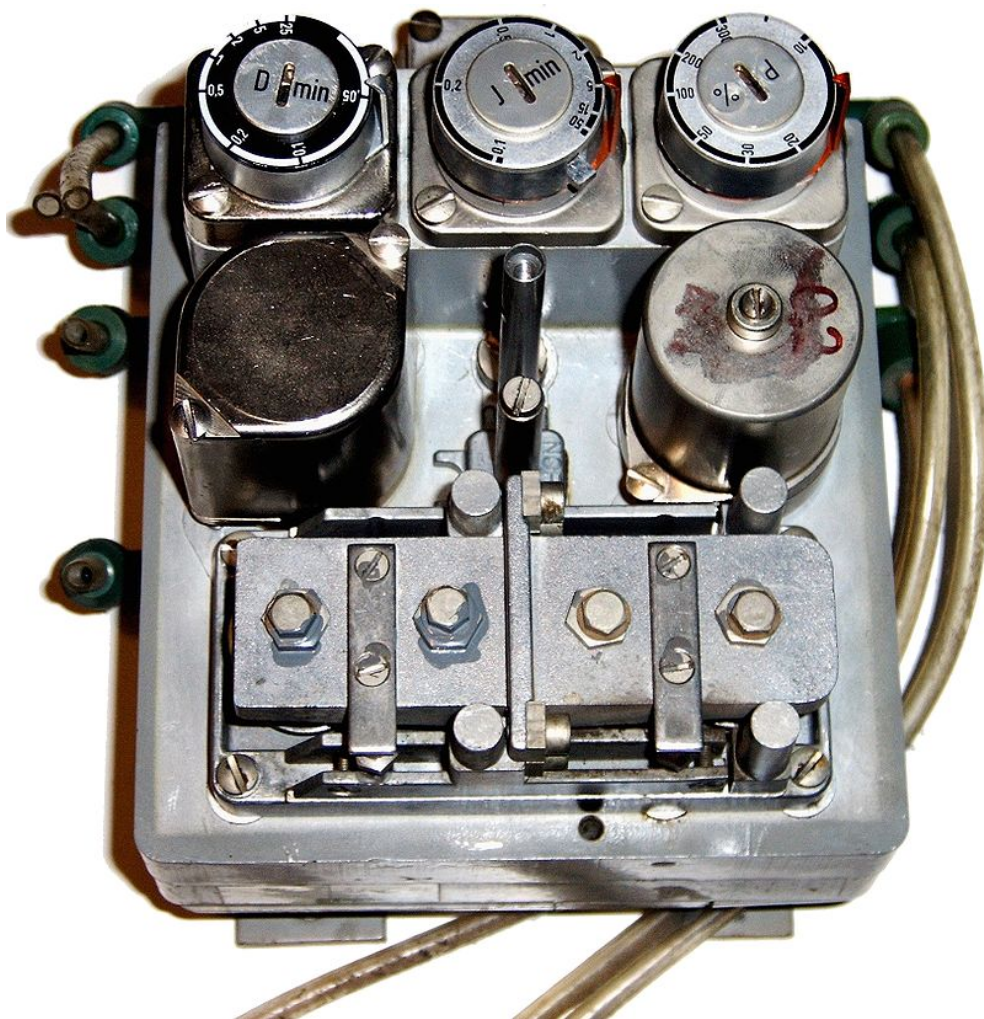
Το Πολεμικό Ναυτικό τελικά δεν υιοθέτησε το σύστημα, λόγω της αντίστασης του προσωπικού. Παρόμοια εργασία πραγματοποιήθηκε και δημοσιεύθηκε από αρκετούς άλλους στη δεκαετία του 1930.

## Βιομηχανικός έλεγχος

Η ευρεία χρήση των ελεγκτών ανάδρασης δεν κατέστη εφικτή μέχρις ότου αναπτύχθηκαν ενισχυτές υψηλού κέρδους και μεγάλης ζώνης (wideband high-gain amplifiers) για να χρησιμοποιηθεί η έννοια της αρνητικής ανάδρασης. Αυτοί είχαν αναπτυχθεί στην ηλεκτρονική τηλεφωνική μηχανική από τον Harold Black στα τέλη της δεκαετίας του 1920, αλλά δεν δημοσιεύθηκε μέχρι το 1934 [7]. Ανεξάρτητα, ο Clesson E Mason της εταιρείας Foxboro, το 1930, εφηύρε έναν ευρείας ζώνης πνευματικό ελεγκτή, συνδυάζοντας τον πνευματικό ενισχυτή με ακροφύσιο και πτερύγιο υψηλού κέρδους, που είχε εφευρεθεί το 1914, με αρνητική ανάδραση από την έξοδο του ελεγκτή. Αυτό αύξησε δραματικά το γραμμικό εύρος λειτουργίας του ακροφυσίου και του ενισχυτή πτερυγίου και ο ολοκληρωμένος έλεγχος μπορούσε επίσης να προστεθεί με τη χρήση μιας βαλβίδας εξαέρωσης ακριβείας και ενός φυσητήρα. Το αποτέλεσμα ήταν ο ελεγκτής "Stabilog" ο οποίος έδωσε αναλογικές και ολοκληρωμένες λειτουργίες χρησιμοποιώντας ανατροφοδοτούμενους φυσητήρες [7]. Αργότερα ο παράγωγος όρος

προστέθηκε από ένα άλλο φυσητήρα και ρυθμιζόμενο στόμιο.

Από το 1932 και μετά, η χρήση ευρυζωνικών ελεγκτών αυξήθηκε ραγδαία σε ποικίλες εφαρμογές ελέγχου. Ο πεπιεσμένος αέρας χρησιμοποιήθηκε τόσο για την παραγωγή της εξόδου του ελεγκτή όσο και για την τροφοδοσία της συσκευής διαμόρφωσης της διαδικασίας, όπως μια βαλβίδα ελέγχου που λειτουργεί με διαφράγματα. Ήταν απλές συσκευές χαμηλής συντήρησης που λειτουργούσαν καλά σε σκληρό βιομηχανικό περιβάλλον και δεν παρουσίαζαν κίνδυνο έκρηξης σε επικίνδυνες τοποθεσίες. Ήταν το βιομηχανικό πρότυπο για πολλές δεκαετίες μέχρι την εμφάνιση διακριτών ηλεκτρονικών ελεγκτών και καταναμιγμένων συστημάτων ελέγχου.



Σχήμα 2.1: Πνευματικός ελεγκτής PID. Οι συντελεστές των "τριών όρων" P, I και D ρυθμίζονται από τους διακόπτες στην κορυφή

Στη δεκαετία του 1950, όταν οι ηλεκτρονικοί ενισχυτές υψηλού κέρδους έγιναν φτηνοί και αξιόπιστοι, οι ηλεκτρονικοί ελεγκτές PID έγιναν δημοφιλείς και χρησιμοποιήθηκαν σήματα ρεύματος βρόχου 4 – 20mA

τα οποία εξομοιώνουν το πνευματικό πρότυπο. Ωστόσο, οι ενεργοποιητές πεδίου (field actuators) εξακολουθούν να χρησιμοποιούν ευρέως το πνευματικό πρότυπο λόγω των πλεονεκτημάτων της πνευματικής κινητήριας δύναμης για τις βαλβίδες ελέγχου στα περιβάλλοντα των μονάδων επεξεργασίας.

### Ηλεκτρονικοί αναλογικοί ελεγκτές

Οι ηλεκτρονικοί αναλογικοί βρόχοι ελέγχου PID βρίσκονταν συχνά μέσα σε πιο σύνθετα ηλεκτρονικά συστήματα, όπως για παράδειγμα η τοποθέτηση της κεφαλής μιας μονάδας σκληρού δίσκου, η ρύθμιση της ισχύος ενός τροφοδοτικού ή ακόμα και το κύκλωμα ανίχνευσης κίνησης ενός σύγχρονου σεισμομέτρου. Οι διακριτοί ηλεκτρονικοί αναλογικοί ελεγκτές έχουν αντικατασταθεί σε μεγάλο βαθμό από ψηφιακούς ελεγκτές που χρησιμοποιούν μικροελεγκτές ή FPGA, για την εφαρμογή αλγορίθμων PID. Ωστόσο, διακριτοί αναλογικοί ελεγκτές PID εξακολουθούν να χρησιμοποιούνται σε εξειδικευμένες εφαρμογές που απαιτούν απόδοση υψηλού εύρους ζώνης και χαμηλού θορύβου, όπως ελεγκτές με λέιζερ-δίοδο [15].

#### 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 καταναεμημένου 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 πραγματικότητα PI ελέγχου).

Η δημοτικότητα του PID οφείλεται σε μεγάλο βαθμό στην ευκολία εφαρμογής και την αποτελεσματικότητά του. Το κίνητρο για τη χρήση του PID προέρχεται από την αποδοτικότητα του κόστους υλοποίησής του: ο ελεγκτής PID σπάνια αποτελεί ένα βέλτιστο ελεγκτή αλλά είναι αρκετά καλός στις περισσότερες περιπτώσεις και έτσι το πρόσθετο κόστος και η πολυπλοκότητα ενός βέλτιστου ελεγκτή δεν αξίζουν την οριακή αύξηση στην απόδοση. Επιπλέον, ο έλεγχος PID δεν απαιτεί βαθιά κατανόηση των υποκείμενων λειτουργιών μιας διαδικασίας. Το μόνο που έχει σημασία είναι ότι μερικές μετρούμενες μεταβλητές της διαδικασίας να μπορούν να

επηρεαστούν έντονα από ορισμένες ελεγχόμενες μεταβλητές. Επίσης ένας PID ελεγκτής μπορεί εύκολα να μεταφερθεί από διεργασία σε διεργασία. Το μόνο που χρειάζεται είναι να προσαρμοστούν τα κέρδη των όρων του και τα όρια της εξόδου του έτσι ώστε να ταιριάζουν στην καινούρια διεργασία. 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.

Οι PID ελεγκτές έχουν επιβιώσει πολλές αλλαγές στην τεχνολογία πηγαίνοντας από τη χρήση πνευματικών συστημάτων στη χρήση μικροεπεξεργαστών μέσω ηλεκτρονικών σωλήνων, τρανζίστορ, ολοκληρωμένων κυκλωμάτων. Οι μικροεπεξεργαστές είχαν μεγάλη επίδραση στους PID ελεγκτές. Βασικά όλοι οι PID ελεγκτές σήμερα κατασκευάζονται με τη χρήση μικροεπεξεργαστών. Αυτό έδωσε τη δυνατότητα να συμπεριληφθούν επιπλέον χαρακτηριστικά στους PID ελεγκτές, όπως η αυτο-ρύθμισή τους που υλοποιείται σε αυτή την εργασία.

## 2.2 Αρχές λειτουργίας

**Ο** PID ελεγκτής είναι, κατά πολύ, ο πιο κοινός αλγόριθμος ελέγχου. Οι περισσότεροι βρόχοι ανάδρασης ελέγχονται από αυτόν τον αλγόριθμο ή από διαφοροποιήσεις του. Συνεπώς ο έλεγχος αυτός έχει διάφορους τρόπους με τους οποίους μπορεί να αντιμετωπιστεί. Μπορεί να θεωρείται ως συσκευή η οποία λειτουργεί με κάποιους εμπειρικούς κανόνες ή μπορεί και να προσεγγιστεί αναλυτικά. Σε αυτή την ενότητα θα παρουσιαστούν ο βασικός αλγόριθμος καθώς και οι μηχανισμοί που διέπουν τη λειτουργία του PID ελεγκτή. Καθώς αυτή η ενότητα αποτελεί μια σύντομη εισαγωγή στη θεωρία του PID ελεγκτή, η ανάλυση που θα γίνει είναι σύντομη και απαιτεί το ελάχιστο μαθηματικό υπόβαθρο. Όποιος ενδιαφέρεται για μια εις βάθος μαθηματική ανάλυση μπορεί να ανατρέξει στην πηγή [18].

### 2.2.1 Η Ανάδραση

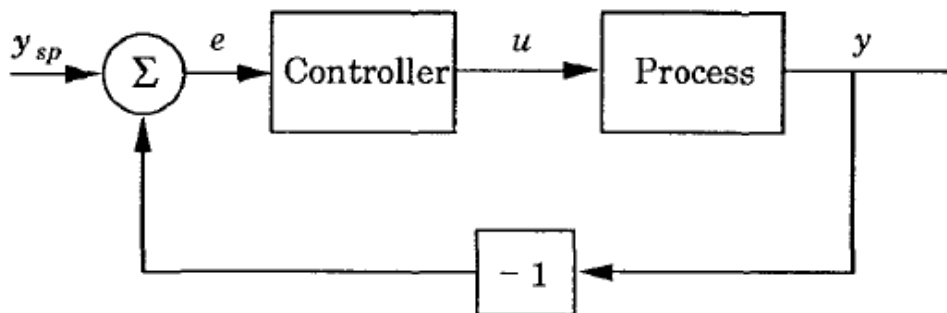
Όπως οι περισσότεροι ελεγκτές, έτσι και ο PID, βασίζεται στην έννοια της ανατροφοδότησης ή ανάδρασης. Συνεπώς αξίζει να γίνει μια σύντομη αναφορά στο τι είναι ανάδραση και πώς λειτουργεί. Η ανάδραση είχε μεγάλη επιρροή στην εξέλιξη της τεχνολογίας σε διάφορα πεδία, μεταξύ αυτών και ο αυτόματος έλεγχος. Χάριν απλότητας, ας υποθέσουμε ότι σε μία διαδικασία, αν αυξηθεί η έξοδος του ελεγκτή (manipulated variable) τότε θα αυξηθεί και η τιμή της μεταβλητής της διαδικασίας που μας ενδιαφέρει (process variable). Με αυτό το σκεπτικό, η ανάδραση μπορεί να



περιγράφει ως:

Αύξησε τη manipulated variable όταν η process variable είναι χαμηλότερη από την επιθυμητή τιμή και μείωσε τη manipulated variable όταν η process variable είναι μεγαλύτερη από την επιθυμητή τιμή.

Αυτού του είδους η ανάδραση ονομάζεται αρνητική (negative feedback) γιατί η manipulated variable κινείται αντίθετα από την process variable. Το Σχήμα 2.2 δείχνει ένα τυπικό παράδειγμα αρνητικής ανάδρασης. Ο λόγος που η αρνητική ανάδραση είναι τόσο σημαντική είναι επειδή κάνει την process variable να πλησιάζει την επιθυμητή τιμή, παρά την ύπαρξη διαταραχών και διακυμάνσεων στα χαρακτηριστικά της διεργασίας.



Σχήμα 2.2: Δομικό διάγραμμα μιας διεργασίας με αρνητική ανάδραση

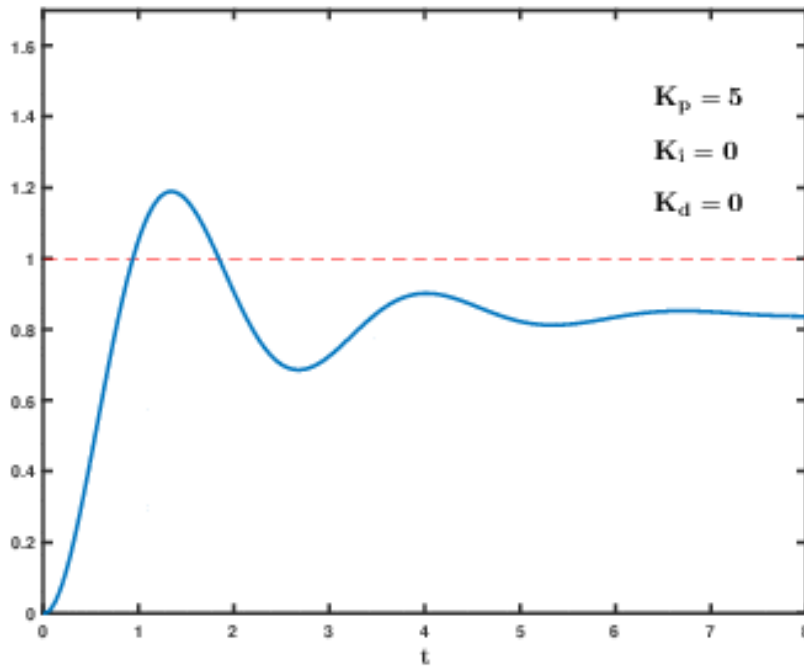
### 2.2.2 Αναλογικός Όρος

Ο αναλογικός όρος παράγει ένα σήμα εξόδου το οποίο είναι ανάλογο στην τρέχουσα τιμή του σφάλματος. Αυτό επιτυγχάνεται πολλαπλασιάζοντας το σφάλμα  $e(t) = y_{sp} - y$  με ένα συντελεστή  $K_p$  που ονομάζεται αναλογική σταθερά κέρδους (proportional gain constant). Ο αναλογικός όρος συνεπώς δίνεται από τη σχέση:

$$P_{out} = K_p e(t) \quad (2.1)$$

Ένα υψηλό αναλογικό κέρδος έχει ως αποτέλεσμα μια μεγάλη αλλαγή στην έξοδο για μια δεδομένη αλλαγή στο σφάλμα. Εάν το αναλογικό κέρδος είναι πολύ υψηλό, το σύστημα μπορεί να γίνει ασταθές (βλ. Το τμήμα σχετικά με τη ρύθμιση βρόγχου). Σε αντίθεση, ένα μικρό κέρδος έχει ως αποτέλεσμα μια μικρή απόκριση εξόδου σε ένα μεγάλο σφάλμα εισόδου και έναν λιγότερο ανταποκρίσιμο ή λιγότερο ευαίσθητο ελεγκτή. Αν το αναλογικό κέρδος είναι πολύ χαμηλό, η ενέργεια ελέγχου μπορεί να είναι πολύ μικρή όταν οφείλεται σε διαταραχές του συστήματος. Στις περισσότερες εφαρμογές ελέγχου, ο αναλογικός όρος είναι αυτός που συνεισφέρει το μεγαλύτερο μέρος στην έξοδο του ελεγκτή.

Επειδή ο αναλογικός όρος επιδρά στην τρέχουσα τιμή του σφάλματος, ένας αναλογικός ελεγκτής πάντα παρουσιάζει ένα σφάλμα μόνιμης κατάστασης (steady state error). Εξάιρεση αποτελεί η περίπτωση που η επιθυμητή τιμή είναι η τιμή στην οποία ο αναλογικός όρος ισούται με το μηδέν. Στο Σχήμα 2.3 φαίνεται η επίδραση μόνο του αναλογικού όρου σε μια διεργασία. Για την εξάλειψη του σφάλματος μόνιμης κατάστασης χρησιμοποιούμε τον όρο που αναφέρεται στη συνέχεια.



Σχήμα 2.3: Επίδραση του αναλογικού όρου σε μια διεργασία

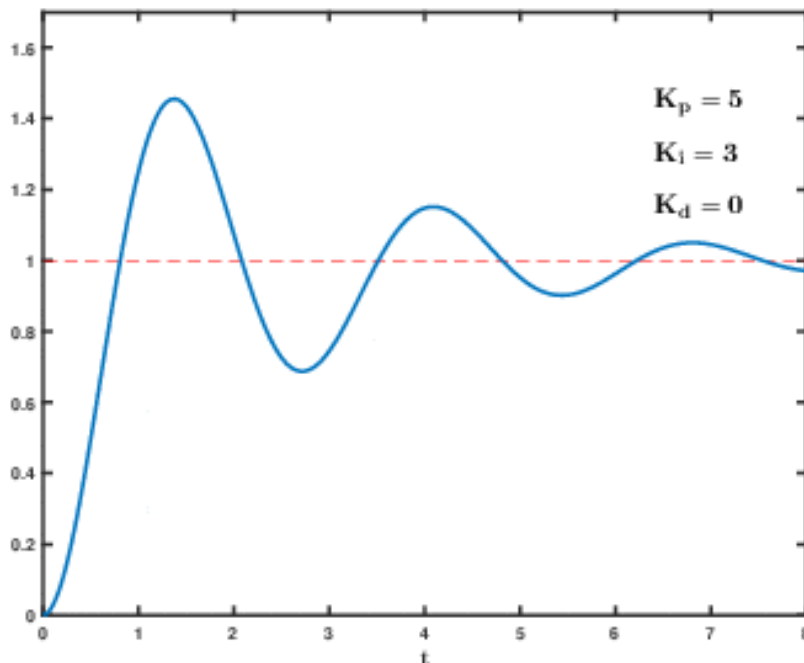
### 2.2.3 Ολοκληρωτικός Όρος

Η συμβολή του ολοκληρωτικού όρου είναι ανάλογη τόσο με το μέγεθος του σφάλματος όσο και με τη διάρκεια του. Το ολοκλήρωμα ενός ελεγκτή PID είναι το άθροισμα του στιγμιαίου σφάλματος με την πάροδο του χρόνου και δίνει τη συσσωρευμένη μετατόπιση που θα έπρεπε να είχε διορθωθεί προηγουμένως. Το συσσωρευμένο σφάλμα στη συνέχεια πολλαπλασιάζεται με το ολοκληρωτικό κέρδος ( $K_i$ ) και προστίθεται στην έξοδο του ελεγκτή. Ο ολοκληρωτικός όρος δίνεται από τη σχέση:

$$I_{out} = K_i \int_0^t e(\tau) d\tau \quad (2.2)$$

όπου  $\tau$  είναι η μεταβλητή ολοκλήρωσης και παίρνει τιμές από το 0 έως την τρέχουσα χρονική στιγμή  $t$ . Έτσι, αν η εφαρμοζόμενη δράση δεν είναι

αρκετή για να φέρει το σφάλμα στο μηδέν, αυτή η δράση θα αυξηθεί με το πέρασμα του χρόνου. Επίσης ο ολοκληρωτικός όρος επιταχύνει την κίνηση της process variable προς την επιθυμητή τιμή. Ένας καθαρός ολοκληρωτικός ελεγκτής “I” θα μπορούσε να φέρει το σφάλμα στο μηδέν, ωστόσο θα είχε πολύ αργή αντίδραση στην αρχή (επειδή η δράση θα ήταν μικρή, και θα χρειαζόταν χρόνο για να γίνει σημαντική), βίαιη (η δράση αυξάνεται όσο το σφάλμα είναι θετικό, ακόμη και αν το σφάλμα έχει αρχίσει να πλησιάζει το μηδέν) και αργή να τελειώσει (όταν το σφάλμα αλλάζει πρόσημο, αυτό για κάποιο χρονικό διάστημα μόνο θα μειώνει τη δύναμη της δράσης του ελεγκτή και δε θα το κάνει να αλλάζει και αυτή πρόσημο), προκαλώντας υπέρβαση και ταλαντώσεις. Επιπλέον, θα μπορούσε να προκαλέσει το σύστημα να αποκριθεί ακόμα και αν υπάρχει ήδη μηδενικό σφάλμα καθώς θυμάται ότι το σύστημα είχε σφάλμα και έτσι θα μπορούσε να προκαλέσει μια ενέργεια όταν αυτή δεν είναι απαραίτητη. Το Σχήμα 2.4 δείχνει πώς η προσθήκη ενός ολοκληρωτικού όρου εξαλείφει το σφάλμα μόνιμης κατάστασης που δεν κατάφερε να εξαλείψει ο αναλογικός όρος αλλά ταυτόχρονα εισάγει ταλαντώσεις στο σύστημα.



Σχήμα 2.4: Εφαρμογή του ολοκληρωτικού όρου στην προηγούμενη απόκριση κρατώντας το αναλογικό κέρδος σταθερό

### 2.2.4 Παράγωγος Όρος

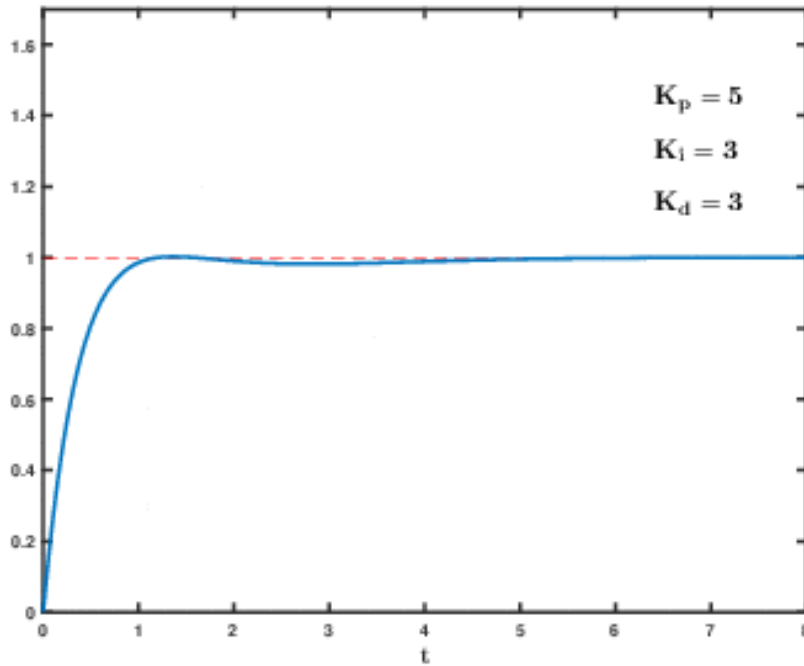
Η παράγωγος του σφάλματος της διαδικασίας υπολογίζεται καθορίζοντας την κλίση του σφάλματος με την πάροδο του χρόνου και πολλαπλασιάζοντας αυτόν τον ρυθμό μεταβολής με το κέρδος του παραγώγου  $K_d$ . Το μέγεθος της συμβολής του παραγώγου όρου στη συνολική δράση ελέγχου ονομάζεται παράγωγο κέρδος (derivative gain),  $K_d$ . Ο παράγωγος όρος δίνεται από τη σχέση:

$$D_{out} = K_d \frac{de(t)}{dt} \quad (2.3)$$

Ένας παράγωγος όρος δεν λαμβάνει υπόψιν του το σφάλμα (αυτό σημαίνει ότι δεν μπορεί να το φτάσει στο μηδέν: ένας καθαρός ελεγκτής “D” δεν μπορεί να φέρει το σύστημα στην επιθυμητή τιμή του), αλλά το ρυθμό αλλαγής του σφάλματος, προσπαθώντας να φέρει αυτόν τον ρυθμό στο μηδέν. Στόχος του είναι η τροχιά του σφάλματος να γίνει μια οριζόντια γραμμή, αποσβένοντας την εφαρμοζόμενη δύναμη και μειώνοντας έτσι την υπέρβαση (overshoot). Η εφαρμογή υπερβολικής ώθησης όταν το σφάλμα είναι μικρό και συνεχίζει να μειώνεται θα οδηγήσει σε υπέρβαση. Μετά την υπέρβαση, αν ο ελεγκτής εφαρμόσει μια μεγάλη διόρθωση στην αντίθετη κατεύθυνση και επανειλημμένα υπερβεί την επιθυμητή θέση, η έξοδος θα ταλαντώνεται γύρω από την επιθυμητή τιμή είτε σε ένα σταθερό, αναπτυσσόμενο ή σε αποσβεννόμενο ημιτονοειδές. Εάν το πλάτος των ταλαντώσεων αυξάνεται με το χρόνο, το σύστημα είναι ασταθές. Εάν μειώνεται, το σύστημα είναι σταθερό. Εάν οι ταλαντώσεις διατηρούν σταθερό πλάτος, το σύστημα είναι οριακά σταθερό. Ο παράγωγος όρος προβλέπει τη συμπεριφορά του συστήματος και επομένως βελτιώνει τον χρόνο που απαιτείται για να ηρεμήσει το σύστημα (settling time) καθώς και τη σταθερότητα του συστήματος [16], [17]. Ένα ιδανικός παράγωγος ελεγκτής δεν είναι αιτιατός, και έτσι οι εφαρμογές των PID ελεγκτών περιλαμβάνουν ένα πρόσθετο φίλτρο χαμηλής διέλευσης (low-pass filter) για τον παράγωγο όρο για να περιορίσουν το κέρδος και το θόρυβο υψηλής συχνότητας. Ο παράγωγος όρος χρησιμοποιείται πολύ πιο σπάνια στην πράξη από τους άλλους δύο όρους (P και I), λόγω της μεταβλητής επίδρασής του στη σταθερότητα του συστήματος σε πραγματικές εφαρμογές. Το Σχήμα 2.5 δείχνει πώς βελτιώθηκε η συνολική απόκριση του συστήματος και εξαλείφθηκαν οι ταλαντώσεις που είχε εισάγει ο ολοκληρωτικός όρος με την προσθήκη του όρου παραγώγου.

### 2.2.5 Εξίσωση του PID ελεγκτή

Οι τρεις όροι που αναλύθηκαν παραπάνω είναι αυτοί που δίνουν στον ελεγκτή το όνομά του. Το άθροισμα των τριών αυτών όρων αποτελεί τη μεταβλητή που μεταχειρίζεται ο ελεγκτής (manipulated variable) και ισούται με την έξοδό του. Ορίζοντας ως  $u(t)$  την έξοδο του ελεγκτή και



Σχήμα 2.5: Εφαρμογή του όρου παραγώγου στην προηγούμενη απόκριση κρατώντας τα άλλα κέρδη σταθερά

αθροίζοντας τις εξισώσεις 2.1, 2.2 και 2.3 η τελική μορφή του είναι:

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2.4)$$

Αυτή η μορφή του PID είναι γνωστή ως παράλληλη (parallel). Η μορφή του PID που χρησιμοποιείται περισσότερο στη βιομηχανία και η οποία χρησιμοποιήθηκε και σε αυτή την εργασία είναι η τυποποιημένη μορφή (standard form). Σε αυτή τη μορφή το κέρδος του αναλογικού όρου  $K_p$  εφαρμόζεται και στους άλλους δύο όρους και έτσι η εξίσωση που προκύπτει είναι η:

$$u(t) = MV(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.5)$$

όπου  $T_i$  είναι ο χρόνος ολοκλήρωσης και  $T_d$  είναι ο χρόνος παραγώγισης. Σε αυτή τη μορφή, οι παράμετροι έχουν σαφή φυσική σημασία. Συγκεκριμένα, η εσωτερική άθροιση παράγει μια νέα μοναδική τιμή σφάλματος η οποία αντισταθμίζεται για μελλοντικά και παρελθόντα σφάλματα. Η προσθήκη των αναλογικών και παραγώγων συνιστωσών προβλέπει την τιμή σφάλματος σε  $T_d$  δευτερόλεπτα (ή δείγματα) στο μέλλον, υποθέτοντας ότι ο έλεγχος βρόχου παραμένει αμετάβλητος. Το ολοκληρωτικό στοιχείο προσαρμόζει την τιμή σφάλματος για να αντισταθμίσει το άθροισμα όλων των

παρελθόντων σφαλμάτων, με σκοπό την πλήρη εξάλειψή τους  $T_i$  δευτερόλεπτα (ή δείγματα). Η προκύπτουσα αντισταθμισμένη τιμή μοναδικού σφάλματος κλιμακώνεται από το μοναδικό κέρδος  $K_p$ .

Οι συντελεστές της εξίσωσης (2.4) με αυτούς της εξίσωσης (2.5) συνδέονται με τις σχέσεις:  $K_i = \frac{K_p}{T_i}$  και  $K_d = K_p T_d$ . Η παράλληλη μορφή, όπου οι παράμετροι αντιμετωπίζονται ως απλά κέρδη, είναι η πιο γενική και ευέλικτη μορφή. Ωστόσο, είναι επίσης η μορφή όπου οι παράμετροι αυτοί έχουν τη λιγότερη φυσική ερμηνεία και γενικά προορίζονται για θεωρητική ανάλυση του PID ελεγκτή. Η τυποποιημένη μορφή, παρά το γεγονός ότι είναι λίγο πιο περίπλοκη μαθηματικά, είναι πιο συνηθισμένη στη βιομηχανία.

**Σύνοψη** Ο PID ελεγκτής έχει τρεις όρους. Ο αναλογικός όρος (“P”) αφορά τον αναλογικό έλεγχο και επιδρά στο τρέχον σφάλμα. Ο ολοκληρωτικός όρος (“I”) παρέχει μια δράση ελέγχου που είναι ανάλογη στο χρονικό ολοκλήρωμα του σφάλματος. Αυτό εξασφαλίζει ότι το σφάλμα μόνιμης κατάστασης γίνεται μηδενικό. Ο παράγωγος όρος (“D”) είναι ανάλογος της χρονικής παραγώγου του σφάλματος. Αυτός ο όρος προβλέπει το μελλοντικό σφάλμα. Εκτός από αυτές τις δύο μορφές του PID υπάρχουν και άλλες, η καθεμία με διαφορετικές ιδιότητες αλλά οι δύο που αναφέρθηκαν είναι οι πιο γνωστές. Κάποιος που θέλει να μελετήσει και τις άλλες μπορεί να απευθυνθεί στην πηγή [18].

## 2.2.6 Τροποποιήσεις του PID αλγορίθμου

### Integral Windup

Όταν σχεδιάζεται ένας πρακτικός PID ελεγκτής που θα ελέγχει πραγματικούς ενεργοποιητές όπως βαλβίδες, διακόπτες κτλ, θα πρέπει να λαμβάνονται υπόψη και οι περιορισμοί του εξοπλισμού αυτού. Για παράδειγμα ένας κινητήρας έχει περιορισμένη ταχύτητα, μια βαλβίδα έχει όρια στο πόσο ανοιχτή ή πόσο κλειστή μπορεί να είναι και ούτω καθεξής. Συνεπώς, αν ένα σύστημα δεν μπορεί να φτάσει την επιθυμητή τιμή χωρίς να ξεπεράσει αυτά τα όρια τότε θα υπάρχει πάντα ένα σφάλμα μόνιμης κατάστασης, ακόμα και αν έχει προστεθεί ο ολοκληρωτικός όρος στον PID ελεγκτή. Αυτό έχει σαν αποτέλεσμα ο ολοκληρωτικός όρος να συνεχίζει να συσσωρεύει σφάλμα και έτσι αυτός μπορεί να γίνει πολύ μεγάλος και να οδηγήσει το σύστημα σε ανεπιθύμητη συμπεριφορά. Αυτό αναφέρεται στη διεθνή βιβλιογραφία ως integral windup. Αυτό το πρόβλημα μπορεί να αντιμετωπιστεί με κάποια από τις ακόλουθες τροποποιήσεις στο βασικό αλγόριθμο:

- Απενεργοποίηση της ολοκλήρωσης έως ότου η μεταβλητή της διεργασίας εισέλθει στην ελεγχόμενη περιοχή
- Αποτροπή της συσσώρευσης του ολοκληρωτικού όρου πάνω ή κάτω

από τα προκαθορισμένα όρια

- Υπολογισμός εκ των υστέρων του ολοκληρωτικού όρου για τον περιορισμό της εξόδου του ελεγκτή εντός εφικτών ορίων

### Bumpless λειτουργία

Διάφορες αλλαγές μπορεί να επέλθουν σε έναν PID που είναι ενεργός. Μπορεί να αλλάξουν οι παράμετροί του ή να αλλάξει η λειτουργία του από χειροκίνητη σε αυτόματη. Οι ελεγκτές PID υλοποιούνται συχνά με ένα χαρακτηριστικό “εξομάλυνσης” έτσι ώστε η έξοδός τους να μεταβάλλεται με ομαλό τρόπο κατά τη διάρκεια αυτών των αλλαγών [19].

### Περισσότερες τροποποιήσεις

Φυσικά μετά από τόσα χρόνια εφαρμογής του PID αλγορίθμου στη βιομηχανία, πολλές εναλλακτικές μορφές του έχουν προταθεί, πέρα από αυτές τις δύο προαναφερθέντες. Η καθεμία από αυτές τις τροποποιήσεις βελτιώνει πολλά από τα προβλήματα του βασικού αλγορίθμου. Κάποιος που θέλει να ενημερωθεί για όλες τις τροποποιήσεις του αλγορίθμου αναλυτικά, καλείται να ανατρέξει στην πηγή [18].





## ΚΕΦΑΛΑΙΟ

### 3

## ΡΥΘΜΙΣΗ ΤΟΥ PID ΕΛΕΓΚΤΗ

### 3.1 Εισαγωγή

**Η** ρύθμιση του PID ελεγκτή έχει να κάνει με την απόδοση τιμών στο συντελεστή κάθε όρου, έτσι ώστε καθένας από αυτούς να επηρεάσει θετικά την απόκριση του συστήματος, ενώ ταυτόχρονα να μετριαστούν όσο γίνεται περισσότερο τα αρνητικά του κάθε όρου. Ο τύπος της διεργασίας επηρεάζει το τι είναι επιθυμητό από τον έλεγχό της. Για παράδειγμα, κάποιες διεργασίες μπορεί να μην ανέχονται υπέρβαση (overshoot) της απόκρισής τους και αυτό να θέτει και περιορισμούς στο χρόνο ανύψωσης (rise time). Αντιθέτως, άλλες διεργασίες μπορεί να παρουσιάζουν ανοχή σε ένα ποσοστό υπέρβασης και έτσι να επιτρέπουν να αυξηθεί ο χρόνος ανύψωσης.

Παρόλο που η υλοποίηση του PID ελεγκτή είναι σχετικά ευθύς και απλή διαδικασία, η σωστή ρύθμισή του είναι πιο περίπλοκο ζήτημα. Αυτό συμβαίνει επειδή απαιτεί κατανόηση του τρόπου με τον οποίο κάθε όρος του PID επηρεάζει τη συνολική απόκριση. Ένας κακώς ρυθμισμένος PID ελεγκτής θα εμφανίσει αρκετά προβλήματα απόδοσης όπως: ταλαντώσεις, μη επαρκή απόσβεση, υπέρβαση, αργούς χρόνους ανόδου ή καθόδου και άλλα. Σε αυτό το κεφάλαιο θα γίνει μια περιγραφή κάποιων διαδεδομένων τεχνικών ρύθμισης που συναντάει κανείς στη βιβλιογραφία (υπάρχουν πολλές πιο εξελιγμένες μέθοδοι ρύθμισης αλλά υπόκεινται σε εμπορικές πατέντες) καθώς και των προβλημάτων που οφείλονται σε κακή ρύθμιση του PID ελεγκτή.

Η ανάλυση των τεχνικών ρύθμισης θα γίνει με αναφορά στα κέρδη  $K_p$ ,  $K_i$  και  $K_d$ , δηλαδή θα αφορά την παράλληλη μορφή του PID ελεγκτή

της Εξίσωσης (2.4) και όχι την τυποποιημένη μορφή της Εξίσωσης (2.5), παρόλο που η δεύτερη χρησιμοποιήθηκε στην εργασία αυτή. Αυτό συμβαίνει γιατί ο αναγνώστης είναι πιο εύκολο να καταλάβει πώς επηρεάζουν τα κέρδη την απόκριση του συστήματος χρησιμοποιώντας την πρώτη εξίσωση. Φυσικά, οι ίδιοι κανόνες ισχύουν και για τις παραμέτρους  $T_i$  και  $T_d$  της τυποποιημένης μορφής, έχοντας πάντα υπόψιν τις εξισώσεις που συνδέουν τις παραμέτρους των δύο αυτών σχέσεων.

## 3.2 Τεχνικές ρύθμισης

### 3.2.1 Χειροκίνητη ρύθμιση

Η πρώτη και πιο φυσική μέθοδος είναι η χειροκίνητη ρύθμιση του ελεγκτή από έναν χειριστή. Σε αυτή τη μέθοδο αρχικά τα κέρδη  $K_i$  και  $K_d$  ισούνται με το μηδέν. Στη συνέχεια, αυξάνουμε το αναλογικό κέρδος  $K_p$  μέχρι το σύστημα να αρχίσει να ταλαντώνεται ελαφρώς και ο χρόνος ανύψωσης να είναι ικανοποιητικός. Έπειτα, αυξάνουμε το ολοκληρωτικό κέρδος  $K_i$  έως ότου να εξαλειφθεί το σφάλμα μόνιμης κατάστασης, σε λογικό χρόνο για τη συγκεκριμένη διεργασία. Σε αυτό το σημείο πρέπει να έχουμε στο μυαλό μας ότι μεγαλύτερο ολοκληρωτικό κέρδος σημαίνει και μεγαλύτερη αστάθεια του συστήματος. Τέλος, αν χρειάζεται, αυξάνουμε το κέρδος παραγώγου  $K_d$  για να βελτιώσουμε τη σταθερότητα του συστήματος καθώς και την απόκρισή του σε αλλαγή φορτίου ή σε κάποια διαταραχή. Εδώ πρέπει να έχουμε υπόψη μας ότι μεγάλο κέρδος παραγώγου θα προκαλέσει υπερβολική ανταπόκριση σε μεταβολές και υπέρβαση.

Ένας βρόχος PID που έχει ρυθμιστεί για να επιτύχει μια γρήγορη απόκριση συνήθως θα υπερβεί ελαφρώς την επιθυμητή τιμή ως συνέπεια της γρήγορης “ρύθμισης” του. Εντούτοις, μερικά συστήματα δεν μπορούν να ανεχθούν οποιαδήποτε υπέρβαση, οπότε στην περίπτωση αυτή απαιτείται σύστημα κλειστού βρόχου, το οποίο θα πρέπει να έχει μικρότερο αναλογικό κέρδος  $K_p$  από αυτό ενός συστήματος που μπορεί να ανεχτεί κάποιο ποσοστό υπέρβασης. Στον Πίνακα 3.1 φαίνεται πώς ο κάθε όρος του ελεγκτή επηρεάζει την απόκριση του συστήματος υπό έλεγχο.

Παράμετρος	Χρόνος ανύψωσης	Υπέρβαση	Μόνιμο σφάλμα	Χρόνος ηρεμίας	Ευστάθεια
$K_p$	Μείωση	Αύξηση	Μείωση	Μικρή αλλαγή	Χειροτέρευση
$K_i$	Μείωση	Αύξηση	Εξάλειψη	Αύξηση	Χειροτέρευση
$K_d$	Μικρή αλλαγή	Μείωση	Καμία αλλαγή	Μείωση	Βελτίωση

Πίνακας 3.1: Επίδραση στο σύστημα, της αύξησης κάθε παραμέτρου ανεξάρτητα από τις άλλες

### 3.2.2 Μέθοδος Ziegler–Nichols

Ίσως η πιο γνωστή μέθοδος ρύθμισης ενός PID ελεγκτή είναι η Ziegler–Nichols. Η μέθοδος αυτή έχει πάρει το όνομά της από τους John G. Ziegler και Nathaniel B. Nichols που την παρουσίασαν το 1942 και χρησιμοποιείται ακόμα και σήμερα. Όπως και στην προηγούμενη μέθοδο, στην αρχή τα κέρδη  $K_i$  και  $K_d$  τίθενται ίσα με το μηδέν. Στη συνέχεια αυξάνουμε το κέρδος  $K_p$  μέχρι το σύστημα αυξάνεται μέχρι την τιμή που θα οδηγήσει το σύστημα να εκτελεί ταλαντώσεις σταθερού πλάτους και σταθερής περιόδου. Η τιμή αυτή του κέρδους ονομάζεται απόλυτο κέρδος,  $K_u$ , και η περίοδος των ταλαντώσεων ονομάζεται απόλυτη περίοδος,  $T_u$ . Αυτές τις δύο παραμέτρους τις χρησιμοποιούμε για να ορίσουμε τα κέρδη όπως φαίνεται στον Πίνακα 3.2. Στον πίνακα αυτόν έχουν υπολογιστεί οι παράμετροι  $T_i$  και  $T_d$  της τυποποιημένης μορφής του PID αλγορίθμου [20]. Αυτό έγινε επειδή αυτοί οι παράμετροι θα χρησιμοποιηθούν αργότερα για την αυτόματη ρύθμιση του ελεγκτή της εργασίας.

Τύπος ελέγχου	$K_p$	$T_i$	$T_d$
P	$0.50K_u$	-	-
PI	$0.45K_u$	$T_u/1.2$	-
PD	$0.80K_u$	-	$T_u/8$
PID	$0.60K_u$	$T_u/2$	$T_u/8$

Πίνακας 3.2: Παράμετροι Ziegler–Nichols ανάλογα τον τύπο του ελεγκτή

Οι κανόνες ρύθμισης Ziegler–Nichols, συνήθως οδηγούν σε συστήματα με ιδιαίτερα υψηλή υπέρβαση και “επιθετική” (“aggressive”) απόκριση. Δεν παρέχουν δηλαδή μια έτοιμη λύση ρύθμισης αλλά τις περισσότερες φορές προσφέρουν ένα αρκετά ικανοποιητικό σημείο εκκίνησης από το οποίο ο χειριστής μπορεί να ξεκινήσει να τροποποιεί τα κέρδη έτσι ώστε να έχει την επιθυμητή απόκριση.



# 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.
- [15] Neuhaus, Rudolf. “Diode Laser Locking and Linewidth Narrowing”. Retrieved June 8, 2015.
- [16] “Introduction: PID Controller Design”. University of Michigan.
- [17] Tim Wescott (October 2000). “PID without a PhD”. *EE Times-India*.
- [18] Karl-Johan Åström and Tore Hägglund. “PID Controllers: Theory, Design and Tuning”. Instrument Society of America, 2 edition, 1995.
- [19] Cooper, Douglas. “PI Control of the Heat Exchanger”. *Practical Process Control by Control Guru*. Retrieved 2014-02-27.
- [20] “Ziegler-Nichols Tuning Rules for PID”, Microstar Laboratories

