

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

<b>1</b>	<b>LabVIEW</b>	<b>3</b>
1.1	Εισαγωγή στο LabVIEW . . . . .	3
<b>2</b>	<b>PID ελεγκτές</b>	<b>7</b>
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 . . . . .	15
2.2.5	Εξίσωση του PID ελεγκτή . . . . .	16
2.2.6	Τροποποιήσεις του PID αλγορίθμου . . . . .	18
<b>3</b>	<b>Ρύθμιση του PID ελεγκτή</b>	<b>21</b>
3.1	Εισαγωγή . . . . .	21
3.2	Τεχνικές ρύθμισης . . . . .	22
3.2.1	Χειροκίνητη ρύθμιση . . . . .	22
3.2.2	Μέθοδος Ziegler-Nichols . . . . .	23
3.2.3	Μέθοδος Tyreus-Luyben . . . . .	23
3.2.4	Μέθοδος Cohen-Coon . . . . .	24
3.2.5	Άλλες μέθοδοι και περαιτέρω πληροφορίες . . . . .	24
3.3	Περιορισμοί και προβλήματα κακής ρύθμισης . . . . .	24
3.3.1	Γραμμικότητα . . . . .	25
3.3.2	Θόρυβος στον Παράγωγο Όρο . . . . .	25

3.3.3	Ταλαντώσεις . . . . .	25
3.3.4	Χρόνος Ανόδου και Υπέρβαση . . . . .	27
3.3.5	Ανοχή σε Διαταραχές . . . . .	28
3.3.6	Θόρυβος . . . . .	29
3.3.7	Συχνότητα . . . . .	29
3.3.8	Συμπεράσματα . . . . .	30
<b>4</b>	<b>Ο Αυτο-Ρυθμιζόμενος PID Ελεγκτής</b>	<b>33</b>
4.1	Εισαγωγή . . . . .	33
4.2	Θεωρία . . . . .	33
4.2.1	Relay Method . . . . .	34
4.3	Το Πρόγραμμα . . . . .	37
4.3.1	Το Front Panel και το Block Diagram . . . . .	37
4.3.2	Tab Control . . . . .	39
4.3.3	Simulation Loop . . . . .	40
4.3.4	Controller or Relay subVI . . . . .	41
4.3.5	Record Oscillations Magnitude subVI . . . . .	42
4.3.6	Record Oscillations Time subVI . . . . .	43
4.3.7	Extract Amplitude and Period subVI . . . . .	43
4.3.8	Calculate PID Gains subVI . . . . .	44
4.3.9	Update PID Gains subVI . . . . .	47
4.3.10	Automatic Mode subVI . . . . .	47
4.3.11	Oscillations Counting subVI . . . . .	49
4.4	Σύνοψη . . . . .	50
<b>5</b>	<b>Πειραματική Διαδικασία και Πειραματικά Αποτελέσματα</b>	<b>51</b>
5.1	Εισαγωγή . . . . .	51
	<b>Βιβλιογραφία</b>	<b>53</b>

# 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)

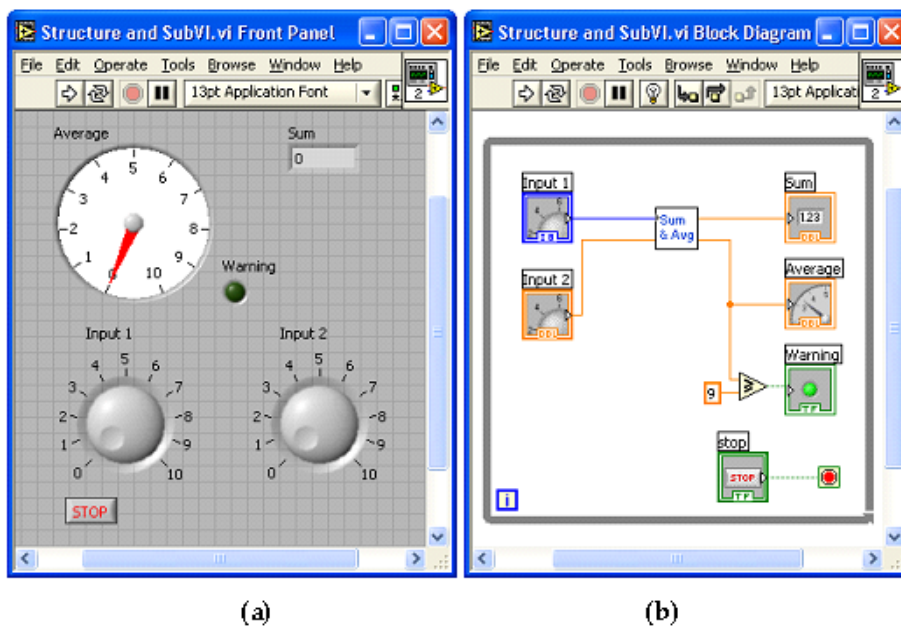
τότε αυτή η συνάρτηση ή το subVI θα εκτελεστεί. Η ροή της εκτέλεσης του προγράμματος καθορίζεται από τη δομή ενός γραφικού μπλοκ διαγράμματος (*block diagram*), που στην ουσία αποτελεί τον πηγαίο κώδικα του LabVIEW. Σε αυτό ο προγραμματιστής συνδέει διαφορετικές συναρτήσεις-κόμβους (*function-nodes*)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

**Έ**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 κατευθύνει 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.

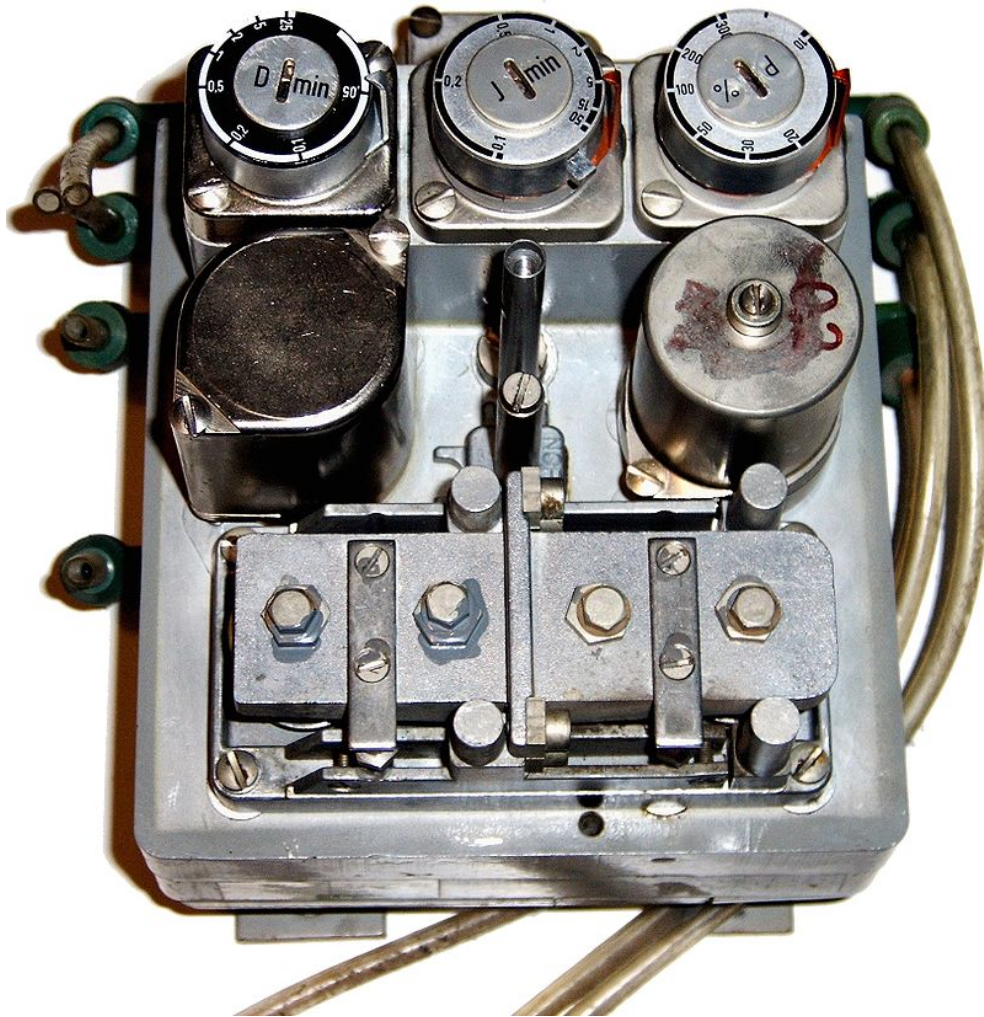
Διεξήχθησαν δοκιμές στο USS New Mexico, με τον ελεγκτή να ελέγχει τη γωνιακή ταχύτητα (όχι τη γωνία) του πηδαλίου. Ο έλεγχος PI απέδωσε σταθερή στροφή (γωνιακό σφάλμα)  $\pm 2^\circ$ . Η προσθήκη του στοιχείου D οδήγησε σε σφάλμα εκτροπής  $\pm 1/6^\circ$ , καλύτερα από ότι θα μπορούσαν να επιτύχουν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

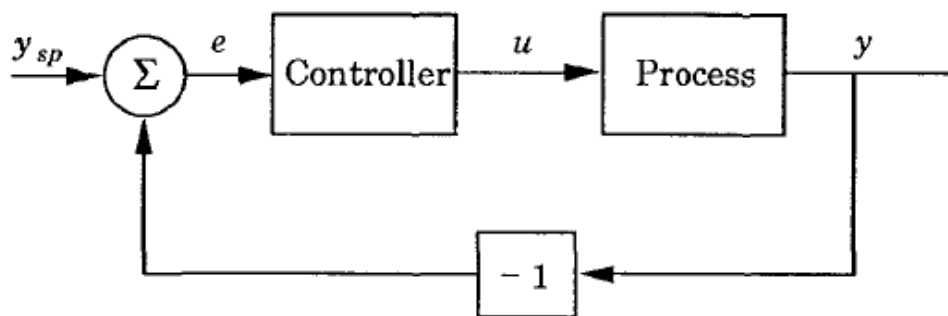
### 2.2.1 Η Ανάδραση

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 περιγραφεί 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

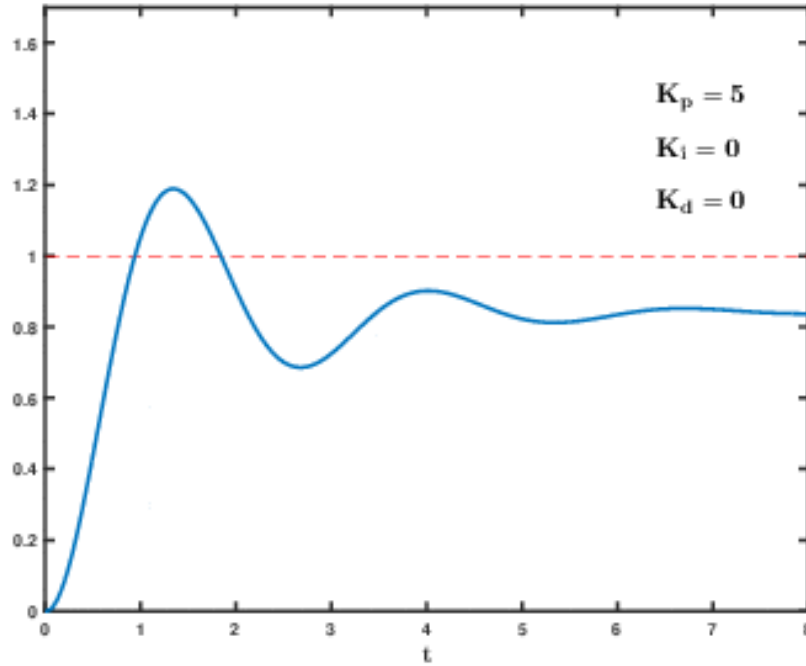
Ο αναλογικός όρος παράγει ένα σήμα εξόδου το οποίο είναι ανάλογο στην τρέχουσα τιμή του σφάλματος. Αυτό επιτυγχάνεται πολλαπλασιάζοντας το σφάλμα  $e(t) = y_{sp} - y$  με ένα συντελεστή  $K_p$  που ονομάζεται αναλογική σταθερά κέρδους (*proportional gain constant*). Ο αναλογικός όρος συνεπώς δίνεται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

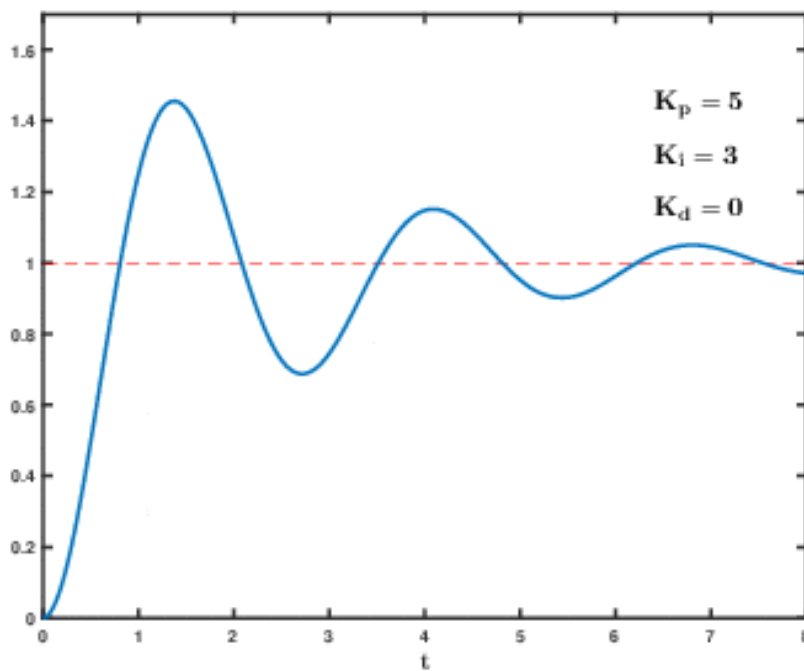
### 2.2.3 Ολοκληρωτικός Όρος

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 ακόμη 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 πολ-

λαπλασιάζοντας αυτόν τον ρυθμό μεταβολής με το κέρδος του παραγώγου  $K_d$ . Το μέγεθος της συμβολής του παραγώγου όρου στη συνολική δράση ελέγχου ονομάζεται παράγωγος κέρδος (*derivative gain*),  $K_d$ . Ο παράγωγος όρος δίνεται από τη σχέση:

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

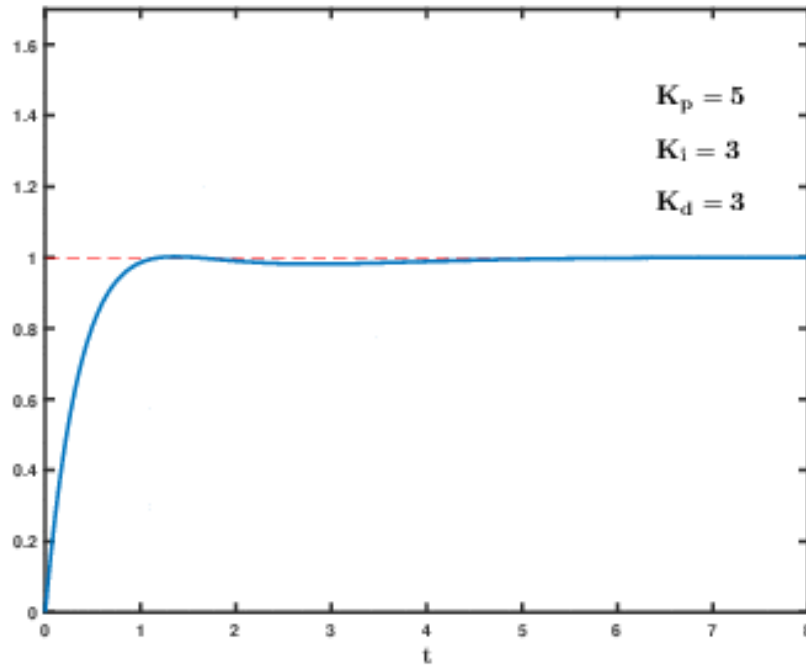
Ένας παράγωγος όρος δεν λαμβάνει υπόψιν του το σφάλμα (αυτό σημαίνει ότι δεν μπορεί να το φτάσει στο μηδέν: ένας καθαρός ελεγκτής “D” δεν μπορεί να φέρει το σύστημα στην επιθυμητή τιμή του), αλλά το ρυθμό αλλαγής του σφάλματος, προσπαθώντας να φέρει αυτόν τον ρυθμό στο μηδέν. Στόχος του είναι η τροχιά 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 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 χρησι-



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

μποιήθηκε 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 υπόψιν 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].





## 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 έλεγχο της. Για παράδειγμα, κάποιες διεργασίες μπορεί να μην ανέχονται υπέρβαση (*overshoot*) της απόκρισης τους και αυτό να θέτει και περιορισμούς στο χρόνο ανόδου (*rise time*). Αντιθέτως, άλλες διεργασίες μπορεί να παρουσιάζουν ανοχή σε ένα ποσοστό υπέρβασης και έτσι να επιτρέπουν να αυξηθεί ο χρόνος ανόδου.

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 ένας από τους όρους 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 της πρώτης μεθόδου 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 τα κέρδη επηρεάζουν 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 τα κέρδη  $K_i$  και  $K_d$  αρχικά 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 αυτή η μορφή των παραμέτρων θα χρησιμοποιηθεί 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.

### 3.2.3 Μέθοδος Tyreus-Luyben

Μια μέθοδος πολύ παρόμοια με τη μέθοδο Ziegler-Nichols που περιγράφηκε πριν είναι η μέθοδος Tyreus-Luyben. Και αυτή βασίζεται στην απόλυτη συχνότητα ( $T_u$ ) και στο απόλυτο κέρδος ( $K_u$ ) του συστήματος. Το μόνο που αλλάζει σε σχέση με την προηγούμενη μέθοδο είναι οι τιμές των κερδών ανάλογα με τον τύπο ελέγχου. Οι τιμές αυτές φαίνονται στον Πίνακα 3.3.

Τύπος ελέγχου	$K_p$	$T_i$	$T_d$
PI	$K_u/3.2$	$2.2T_u$	–
PID	$K_u/2.2$	$2.2T_u$	$T_u/6.3$

Πίνακας 3.3: Παράμετροι Tyreus-Luyben ανάλογα τον τύπο του ελεγκτή

### 3.2.4 Μέθοδος Cohen-Coon

Αυτή η μέθοδος αναπτύχθηκε το 1953 και βασίζεται σε ένα μοντέλο συστήματος πρώτης τάξης με μία χρονοκαθυστέρηση. Παρόμοια με τη μέθοδο Ziegler-Nichols, η μέθοδος αυτή προτείνει ένα σύνολο παραμέτρων συντονισμού για να δώσει μια απόκριση κλειστού βρόχου με λόγο απόσβεσης  $1/4$ . Σε αντίθεση όμως με τις δύο προηγούμενες μεθόδους που χρησιμοποιούν στοιχεία της απόκρισης συχνότητας του συστήματος για να βγάλουν υπολογίσουν τα κέρδη του ελεγκτή, αυτή η χρησιμοποιεί στοιχεία από την βηματική απόκριση (*Step response*) του συστήματος.

### 3.2.5 Άλλες μέθοδοι και περαιτέρω πληροφορίες

Στην ενότητα αυτή περιγράφηκαν οι πιο γνωστές και διαδεδομένες τεχνικές ρύθμισης ενός PID ελεγκτή. Βέβαια τα πολλά χρόνια εφαρμογής των PID ελεγκτών στη βιομηχανία, καθώς και η ανάγκη για όλο και πιο λεπτομερή και στιβαρό έλεγχο έχουν οδηγήσει στην ανάπτυξη πολλών ακόμα μεθόδων και τεχνικών. Βέβαια πολλές από αυτές προστατεύονται από νόμους περί πνευματικής ιδιοκτησίας. Εδώ αναλύθηκαν μόνο οι μέθοδοι που χρησιμοποιήθηκαν στο πρακτικό κομμάτι της εργασίας. Κάποιος που ενδιαφέρεται για μια αναλυτική προσέγγιση σε πολλές μεθόδους ρύθμισης καθώς και να μάθει για τα πλεονεκτήματα και τους περιορισμούς της κάθε μίας μπορεί να ανατρέξει στις πηγές [18], [21], [22].

## 3.3 Περιορισμοί και προβλήματα κακής ρύθμισης

**Π**ΑΡΟΛΟ που οι ελεγκτές PID είναι εφαρμόσιμοι σε πολλά προβλήματα ελέγχου και συχνά παρέχουν ικανοποιητικό έλεγχο χωρίς βελτιώσεις ακόμα και όταν έχουν ρυθμιστεί “στο περίπου”, μπορούν να έχουν χαμηλή απόδοση σε ορισμένες εφαρμογές και σχεδόν ποτέ δεν παρέχουν βέλτιστο έλεγχο. Η βασική δυσκολία με τον έλεγχο PID είναι ότι είναι ένα σύστημα ελέγχου ανατροφοδότησης, με σταθερές παραμέτρους και χωρίς άμεση γνώση της διαδικασίας και έτσι η συνολική απόδοση είναι αντιδραστική και συμβιβαστική.

Οι ελεγκτές PID, όταν χρησιμοποιούνται μόνοι τους, μπορεί να έχουν χαμηλή απόδοση αν τα κέρδη τους πρέπει να μειωθούν, έτσι ώστε το σύστημα ελέγχου να μην παρουσιάζει υπέρβαση, να μην ταλαντώνεται ή να μην κυνηγάει την τιμή ρύθμισης του ελέγχου. Παρουσιάζουν επίσης προβλήματα όταν εμφανίζονται μη γραμμικότητες, μπορεί να ανταλλάσουν την πιο ομαλή απόκριση έναντι του χρόνου ανόδου, να μην αντιδρούν στην αλλαγή της συμπεριφοράς της διαδικασίας (ας πούμε, κάποια χαρακτηριστικά της διαδικασίας έχουν αλλάξει λόγω μεταβολής της θερμοκρασίας) και να έχουν καθυστέρηση στην αντιμετώπιση μεγάλων διαταραχών.

### 3.3.1 Γραμμικότητα

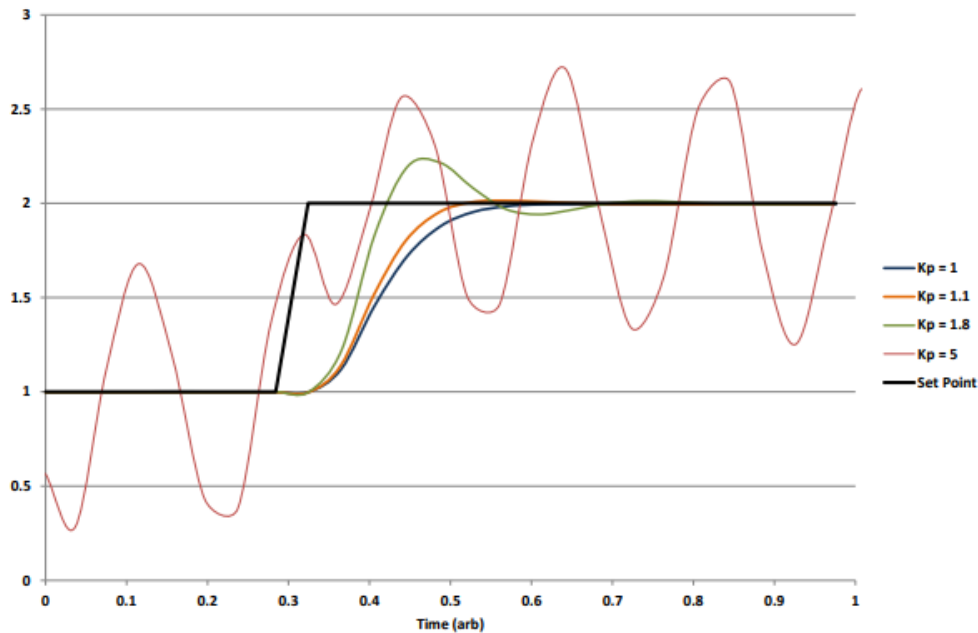
Ο PID ελεγκτής είναι ένα γραμμικό και ιδιαίτερα συμμετρικό σύστημα. Συνεπώς, η απόκρισή του όταν εφαρμόζεται σε ένα μη γραμμικό σύστημα είναι μεταβλητή. Για παράδειγμα, στον έλεγχο της θερμοκρασίας, μια κοινή περίπτωση χρήσης είναι η ενεργή θέρμανση (μέσω θερμοαντικού στοιχείου), αλλά η παθητική ψύξη (θέρμανση χωρίς ψύξη), οπότε η υπέρβαση που μπορεί να προκαλεί ο ελεγκτής διορθώνεται αργά - δεν μπορεί να εξαναγκαστεί προς τα κάτω. Σε αυτή την περίπτωση ο PID θα πρέπει να ρυθμιστεί ώστε να παρέχει υπεραποσβενομένη απόκριση, για να αποτρέψει ή να μειώσει την υπέρβαση, αν και αυτό μειώνει την απόδοση του συστήματος (αυξάνει το χρόνο ηρεμίας).

### 3.3.2 Θόρυβος στον Παράγωγο Όρο

Ένα πρόβλημα με τον παράγωγο όρο είναι ότι ενισχύει τη μέτρηση υψηλής συχνότητας ή το θόρυβο της διαδικασίας, με αποτέλεσμα να προκαλούνται μεγάλες αλλαγές στην έξοδο. Συχνά είναι χρήσιμο να φιλτράρονται οι μετρήσεις με φίλτρο χαμηλής διέλευσης (*low-pass filter*) για να αφαιρείται ο θόρυβος υψηλής συχνότητας. Καθώς το φιλτράρισμα χαμηλής διέλευσης και ο παράγωγος έλεγχος μπορούν να ακυρώνουν ο ένας στον άλλο, η ποσότητα φιλτραρίσματος είναι περιορισμένη. Μπορεί επίσης να χρησιμοποιηθεί ένα μη γραμμικό διάμεσο φίλτρο, το οποίο βελτιώνει την αποδοτικότητα φιλτραρίσματος και την πρακτική απόδοση [23]. Σε ορισμένες περιπτώσεις, ο παράγωγος όρος μπορεί να απενεργοποιηθεί με λίγες διαφοροποιήσεις στη συνολική απόδοση.

### 3.3.3 Ταλαντώσεις

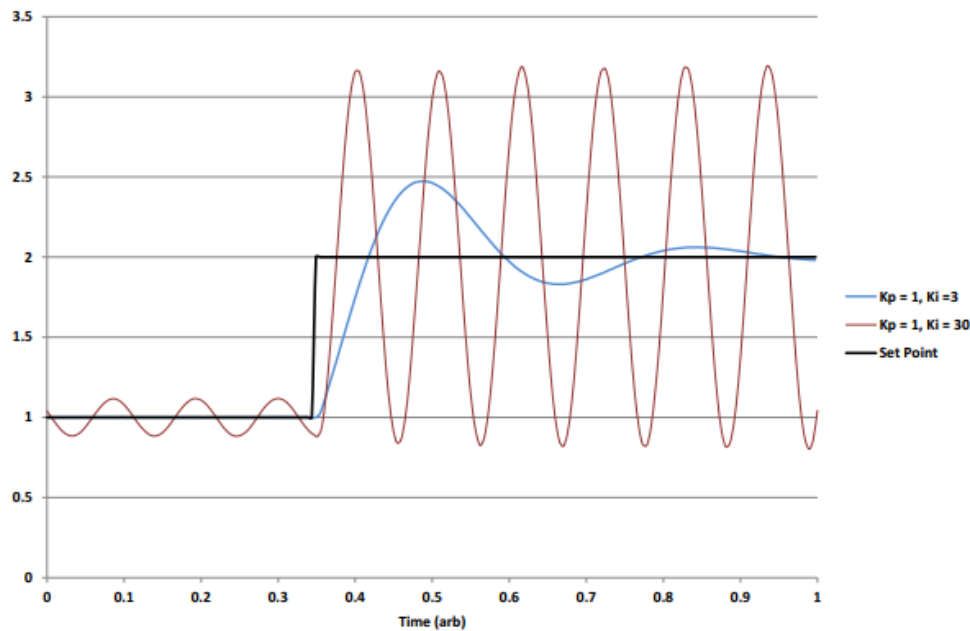
Η ταλάντωση είναι ένα από τα πιο περίπλοκα ζητήματα επίδοσης στους PID ελεγκτές και μπορεί να είναι αποτέλεσμα πολλών παραγόντων. Η πιο συνηθισμένη αιτία είναι το πολύ μεγάλο αναλογικό κέρδος (Σχήμα 3.1). Μια δεύτερη αιτία είναι ένας ελεγκτής που βασίζεται κυρίως στον ολοκληρωτικό του όρο (Σχήμα 3.2) επειδή το σφάλμα που έχει συσσωρευτεί ενώ η μετρούμενη μεταβλητή είναι κάτω από το καθορισμένο σημείο χρειάζεται χρόνο για να διορθωθεί ενώ η μετρούμενη μεταβλητή είναι πάνω από το καθορισμένο σημείο. Ο ολοκληρωτικός όρος στη συνέχεια αρχίζει να συσσωρεύει σφάλμα στην αντίθετη κατεύθυνση, το οποίο δεν θα διορθωθεί μέχρις ότου η μετρούμενη μεταβλητή να διασχίσει ξανά το καθορισμένο σημείο. Σε αυτή την περίπτωση ο αναλογικός όρος ενεργεί ως αποσβεστήρας. Τρίτον, κάτι που εύκολα παραβλέπεται πολλές φορές, είναι ότι το φυσικό σύστημα που ελέγχεται θα μπορούσε να έχει από μόνο του ταλαντευόμενο χαρακτήρα και συνεπώς η ύπαρξη ταλαντώσεων στο σύστημα ίσως είναι αποδεκτή. Αυτό δεν σημαίνει απαραίτητα κακή ρύθμιση του PID ελεγκτή.



Σχήμα 3.1: Απόκριση της μετρούμενης μεταβλητής στην αύξηση του αναλογικού κέρδους

Η συχνότητα του ελεγκτή είναι ο ρυθμός με τον οποίο ο βρόχος ελέγχου λειτουργεί και η έξοδος τροφοδοσίας ενημερώνεται μία φορά ανά κύκλο. Αυτή η συμπεριφορά απλής ενημέρωσης μπορεί να είναι προβληματική αν το αναλογικό κέρδος είναι υπερβολικά υψηλό. Ένα μικρό σφάλμα παράγει μια μεγάλη έξοδο, η οποία προκαλεί μεγάλη μεταβολή στη μετρούμενη μεταβλητή, ακόμη και σε έναν κύκλο. Εάν η αλλαγή αυτή μεταβάλλει τη μετρούμενη μεταβλητή πέρα από το καθορισμένο σημείο, τότε η έξοδος αντιστρέφει την πολικότητα για τον επόμενο κύκλο. Αν η μετρούμενη μεταβλητή συνεχίζει να πηδάει από το κύκλο σε κύκλο, τότε ο ελεγκτής ταλαντώνεται. Αν το κέρδος είναι αρκετά υψηλό και προκαλεί την μεταπήδηση της μετρούμενης μεταβλητής με αυξανόμενο μέγεθος σφάλματος, τότε ο ελεγκτής θεωρείται ασταθής. Η πιο γρήγορη λύση είναι η μείωση του αναλογικού κέρδους. Ωστόσο, η αύξηση της συχνότητας του ελεγκτή, αν αυτό είναι δυνατόν, θα βοηθήσει επίσης.

Αυτός είναι εγγενής περιορισμός ενός ψηφιακού ελεγκτή. Θα υπάρχει πάντα ταλάντωση σε κάποιο βαθμό. Εκτός από το ζήτημα συχνότητας βρόχου που έχει ήδη περιγραφεί, η έξοδος της τροφοδοσίας μπορεί να μειωθεί μόνο σε κάποια πεπερασμένη τιμή και τίποτα μικρότερο. Το πρόβλημα τότε γίνεται ζήτημα πόσο μπορεί να μειωθεί η ταλάντωση. Ένας απόλυτα συνεχής ελεγκτής δεν έχει αυτό το πρόβλημα επειδή η έξοδος συνεχώς ενημερώνεται μαζί με το μεταβαλλόμενο σφάλμα. Ωστόσο, η ταλάντωση από τους περιορισμούς συχνότητας εξακολουθεί να συμβαίνει στον συνεχή έλεγχο PID. Αυτοί οι ελεγκτές είναι αναλογικά κυκλώματα και εξακολου-



Σχήμα 3.2: Ταλαντώσεις λόγω υψηλού ολοκληρωτικού κέρδους

θούν να έχουν μια αποτελεσματική συχνότητα λειτουργίας που προκύπτει από αντιδραστικά στοιχεία που προκαλούν χρονική υστέρηση μεταξύ της εξόδου και της εισόδου.

Η ταλάντωση λόγω ενός υψηλού ολοκληρωτικού κέρδους μπορεί να μειωθεί είτε μειώνοντας το ολοκληρωτικό κέρδος (προφανής λύση) είτε αυξάνοντας το αναλογικό κέρδος (όχι άμεσα προφανές). Η αναλογική δράση θα επιβραδύνει τις ταλαντώσεις που προκαλούνται από την ολοκληρωτική δράση. Αυτό το παράδειγμα παρουσιάζει τη δυσκολία συντονισμού του PID ελεγκτή. Ο χρήστης πρέπει να αποφασίσει ποια χαρακτηριστικά των αποκρίσεων του ελεγκτή είναι απαραίτητα, επιθυμητά και ποια είναι μη αποδεκτά. Κάθε όρος έχει υπέρ και κατά και πρέπει να καθοριστεί η προτεραιότητα των χαρακτηριστικών του ελεγκτή πριν από την προσπάθεια ρύθμισης του συστήματος.

### 3.3.4 Χρόνος Ανόδου και Υπέρβαση

Ο χρόνος ανόδου της μετρούμενης μεταβλητής είναι ο χρόνος που απαιτείται για την μετρούμενη μεταβλητή να φτάσει στο καθορισμένο σημείο μετά από μια αλλαγή. Ένας γρήγορος χρόνος ανόδου είναι επιθυμητός για προφανείς λόγους, αλλά ένας πολύ μικρός χρόνος ανόδου θα προκαλέσει υπέρβαση της μετρούμενης μεταβλητής από το καθορισμένο σημείο. Η υπέρβαση μπορεί στη συνέχεια να μετατραπεί σε αποσβενούμενες ταλαντώσεις για όσο χρονικό διάστημα η μετρούμενη μεταβλητή προσπαθεί να ισορροπήσει στο καθορισμένο σημείο. Για κάποιες εφαρμογές, μια μι-

κρή υπέρβαση είναι αποδεκτή παραχώρηση έτσι ώστε να επιτευχθεί ένας μικρός χρόνος ανόδου. Για άλλες διεργασίες, καμία υπέρβαση δεν είναι ανεκτή και η προσεκτική επιλογή των κερδών του PID ελεγκτή πρέπει να έχει ως αποτέλεσμα τον ταχύτερο χρόνο ανόδου χωρίς υπέρβαση.

Ο χρόνος ανόδου επηρεάζεται κυρίως από το αναλογικό και το παράγωγο κέρδος. Η παράγωγος ενέργεια αντιδρά έντονα σε μία απότομη αλλαγή αλλά επιβραδύνει την προσέγγιση στο καθορισμένο σημείο. Η ολοκληρωτική συνιστώσα δεν επηρεάζει αισθητά τον χρόνο ανόδου λόγω της φύσης της που αφορά τη συσσώρευση σφάλματος, αλλά μπορεί να προκαλέσει υπέρβαση. Επιπλέον, όπως περιγράφηκε προηγουμένως, θα υπάρξει ένα μεγάλο ενιαίο κέρδος προκαλούν ταλαντώσεις.

Ο χρόνος άνοδος τυπικά ορίζεται ως ο χρόνος για την αύξηση της μεταβλητής ενδιαφέροντος από 10% της τελικής τιμής στο 90% της τελικής τιμής. Αυτός είναι ο ορισμός που χρησιμοποιείται εδώ.

### 3.3.5 Ανοχή σε Διαταραχές

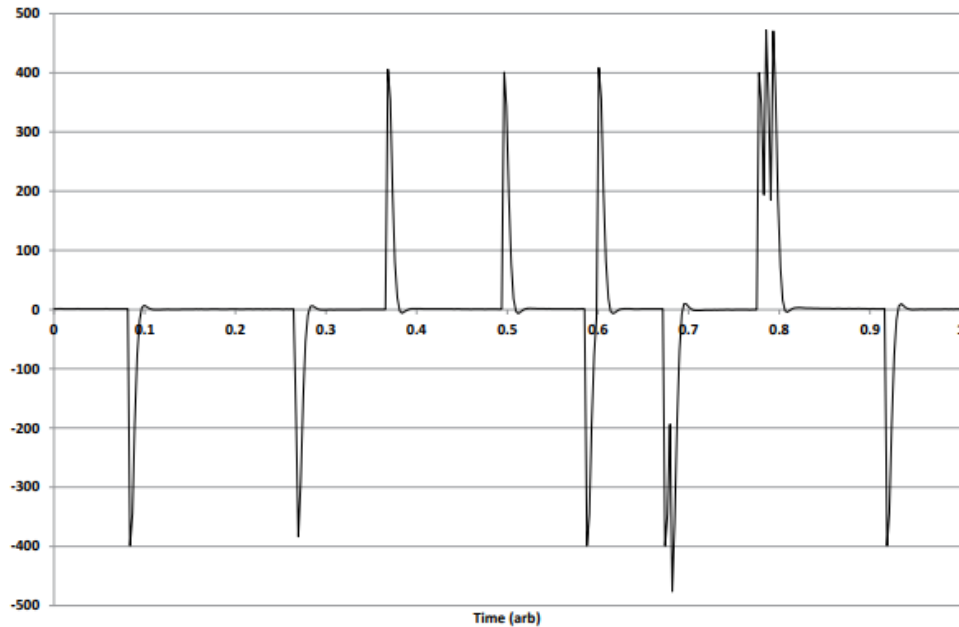
Μια διαταραχή εμφανίζεται όταν οι ανωμαλίες σε οποιαδήποτε από τις μεταβλητές της διεργασίας προκαλούν μια απότομη ή ασυνήθιστη αλλαγή στη μετρούμενη μεταβλητή. Τέτοια διαταραχή μπορεί για παράδειγμα να είναι μια έκρηξη σωλήνα σε μια πρώτη ύλη η οποία προκαλεί μια δραματική πτώση της πίεσης του δοχείου (η μετρούμενη μεταβλητή). Μια δυσλειτουργία στη διαδικασία προθέρμανσης ενός συστήματος, η οποία προκαλεί τη θερμοκρασία του συστήματος να πέσει σε χαμηλότερη τιμή από την κανονική λειτουργία, είναι ένα άλλο χαρακτηριστικό παράδειγμα. Η ανοχή σε διαταραχές (*disturbance robustness*) είναι τότε μια ιδιότητα ενός PID ελεγκτή για να χειριστεί αποτελεσματικά αυτές τις αλλαγές.

Οι διαταραχές μπορούν ή όχι να ληφθούν υπόψη στη σχεδίαση ενός συστήματος ελέγχου. Αυτό σημαίνει ότι ένας ελεγκτής που προσπαθεί να ελέγξει μια διαδικασία χωρίς βρόχο ανατροφοδότησης, ασφαλώς δεν θα αντιληφθεί μια διαταραχή επειδή είναι δύσκολο να εξεταστεί κάθε πιθανή πηγή μιας διαταραχής. Ένα από τα μεγαλύτερα πλεονεκτήματα ενός ελεγκτή PID είναι ότι μπορεί να συντονιστεί για να απορρίψει τις διαταραχές, ακόμα και αν προέρχονται από κάτι που δεν φαίνεται να σχετίζεται με τη διαδικασία.

Μια απότομη μεταβολή της μετρούμενης μεταβλητής μπορεί να θεωρηθεί ως λειτουργικά ισοδύναμη με μια αλλαγή του καθορισμένου σημείου (Σχήμα 3.3). Επομένως, η ρύθμιση του ελεγκτή που εκτελείται για να εξασφαλιστεί η καλύτερη απόκριση σε μια βαθμιαία μεταβολή του καθορισμένου σημείου θα κάνει το ίδιο και για την απόρριψη της διαταραχής. Όμως, αττό δε συμβαίνει αν η διαταραχή επηρεάζει την ικανότητα του ελεγκτή PID να ελέγχει τη μετρούμενη μεταβλητή, είτε επειδή ο ενεργοποιητής τίθεται σε κίνδυνο είτε επειδή η διαταραχή φέρνει τη μετρούμενη μεταβλητή κοντά ή εκτός των ορίων δράσης του ενεργοποιητή. Η έκρηξη στο



σωλήνα που αναφέρθηκε προηγουμένως θα ήταν ένα παράδειγμα εκτεθειμένου ενεργοποιητή διότι έλεγχος της πίεσης μέσα σε ένα παραμορφωμένο σωλήνα σίγουρα θα επιβαρύνει υπερβολικά την αντλία.



Σχήμα 3.3: Κάθε κορυφή είναι αποτέλεσμα είτε διαταραχής είτε αλλαγής του καθορισμένου σημείου. Είναι αδύνατον να υπάρξει διαχωρισμός μεταξύ των δύο.

### 3.3.6 Θόρυβος

Ένα πρόβλημα των PID ελεγκτών, που σχετίζεται κυρίως με τον παράγωγο όρο, είναι ο θόρυβος. Μικρές, γρήγορες αλλαγές στη μετρούμενη μεταβλητή μπορεί να μεγεθύνονται σε μεγάλες αποκρίσεις του ελεγκτή μόνο από την παράγωγο δράση. Το πρόβλημα είναι ότι το παράγωγο κέρδος πρέπει να είναι σχετικά μεγάλο για να έχει επιρροή στην απόδοση του συστήματος, αλλά όσο μεγαλύτερο είναι τόσο περισσότερο ενισχύει τον θόρυβο. Η πιο εύκολη λύση σε αυτό είναι να συμπεριληφθεί στη σχεδίαση του ελεγκτή μια νεκρή ζώνη (*dead-band*) για την αλλαγή του σφάλματος. Όταν η μεταβολή του σφάλματος θα είναι μέσα στα όρια της νεκρής ζώνης η έξοδος του παράγωγου όρου θα είναι μηδενική.

### 3.3.7 Συχνότητα

Η συχνότητα του βρόχου ελέγχου θεωρείται ότι είναι ο ρυθμός με τον οποίο αυτός λειτουργεί. Το αντίστροφο είναι η περίοδος ( $T$ ) και αντιστοιχεί στον χρόνο μεταξύ δύο διαδοχικών επαναλήψεων του βρόχου. Η

συχνότητα είναι μια σημαντική παράμετρος, επειδή καθορίζει το είδος των συστημάτων που μπορεί να ελέγξει επαρκώς ο PID ελεγκτής. Θεωρούμε ότι κάθε φυσικό σύστημα έχει μια χρονική σταθερά που επηρεάζει τη μετρούμενη μεταβλητή. Με άλλα λόγια, πόσο γρήγορα αλλάζει η μετρούμενη μεταβλητή με μια αντίστοιχη αλλαγή στον ενεργοποιητή;

Αυτό είναι σημαντικό επειδή, αν η περίοδος του βρόχου PID είναι πολύ μεγαλύτερη από τη χρονική σταθερά της μετρούμενης μεταβλητής, τότε η μετρούμενη μεταβλητή δεν θα μπορέσει να φθάσει στο καθορισμένο σημείο σε μια τάξη χρόνου γρηγορότερα από  $T$  (ίσως να μην φτάσει και καθόλου). Για παράδειγμα, εάν η έξοδος καθορίζεται για να αυξήσει τη μετρούμενη μεταβλητή για τα επόμενα  $T$  δευτερόλεπτα, τότε η μετρούμενη μεταβλητή θα αυξηθεί με κάποιο ρυθμό που σχετίζεται με τη χρονική σταθερά του συστήματος. Η μετρούμενη μεταβλητή μπορεί να υπερβεί το καθορισμένο σημείο σε αυτό τον χρόνο, ενδεχομένως κατά πολύ. Η έξοδος δεν θα ενημερωθεί μέχρι την επόμενη επανάληψη, η οποία στη συνέχεια θα ανταποκριθεί στη μείωση της μετρούμενης μεταβλητής για τα επόμενα  $T$  δευτερόλεπτα. Το τελικό αποτέλεσμα είναι μια ταλάντωση και ο μόνος τρόπος να μειωθεί, αλλά όχι να εξαιρεθεί, αυτή η ταλάντωση είναι ο ελεγκτής να αποκρίνεται ασθενώς (χαμηλοί συντελεστές κέρδους) στο σφάλμα. Στην πραγματικότητα, αυτό αυξάνει τη χρονική σταθερά του φυσικού συστήματος στην περίοδο του βρόχου ελέγχου και χάνεται η πιθανότητα για γρήγορο έλεγχο.

Εάν η περίοδος του ελεγκτή είναι πολύ μικρότερη από τη χρονική σταθερά του φυσικού συστήματος, τότε ο ελεγκτής έχει σχεδιαστεί υπερβολικά για την εφαρμογή. Αυτό δεν έχει ένα μειονέκτημα στον έλεγχο του συστήματος, αλλά ο ίδιος ο ελεγκτής μπορεί να είναι ακριβότερος ή πιο δύσκολος να κατασκευάσει, να διατηρηθεί, και να λειτουργήσει λόγω της απαίτησης για εξειδικευμένα κομμάτια, ταχύτερα ρολόγια κλπ.

Η περίοδος ενός βρόχου ελέγχου μπορεί να καθοριστεί από την υλοποίησή του. Με μια πρώτη προσέγγιση, η περίοδος είναι ίση με το βραδύτερο τμήμα λειτουργίας ολόκληρου του συστήματος, είτε πρόκειται για μετρητή, τροφοδοτικό, ενεργοποιητή κλπ. Για παράδειγμα, εάν η τροφοδοσία διαρκεί  $100ms$  για να αποκριθεί σε εντολές από έναν κεντρικό υπολογιστή που εκτελεί το λογισμικό που υλοποιεί τον PID ελεγκτή, τότε η περίοδος είναι  $100ms$ , υποθέτοντας ότι ο μετρητής και ο ίδιος ο βρόχος PID μπορούν να ανταποκριθούν ταχύτερα. Το λογισμικό μπορεί σίγουρα να τρέξει γρηγορότερα, αλλά η τροφοδοσία δεν θα αλλάξει περισσότερες από μία φορά κάθε  $100ms$ .

### 3.3.8 Συμπεράσματα

Στις παραπάνω παραγράφους έγινε μια προσπάθεια να περιγραφούν τα κυριότερα προβλήματα που σχετίζονται με τους PID ελεγκτές. Από αυτήν την ανάλυση καθίσταται σαφές ότι η καλή ρύθμιση των PID

ελεγκτών είναι δύσκολη καθώς για να υπάρξει βελτίωση της απόδοσης σε κάποιον τομέα, θα πρέπει να θυσιαστεί η απόδοση σε κάποιον άλλον τομέα. Ο καλύτερος τρόπος για να ρυθμιστεί ένας ελεγκτής PID είναι πρώτα να αποφασιστεί ποιο είδος απόκρισης είναι επιθυμητό. Απαιτείται ένας γρήγορος χρόνος ανόδου; Επιτρέπεται να υπάρχει κάποια ταλάντωση; Είναι η διεργασία υπό έλεγχο πολύ γρήγορη για τον διαθέσιμο εξοπλισμό; Η πιο διαδεδομένη μέθοδος είναι να ξεκινάει η ρύθμιση με τη χρήση μόνο του αναλογικού στοιχείου και στη συνέχεια να προστίθενται οι άλλοι δύο όροι, καθώς εμφανίζονται οι αδυναμίες του αναλογικού όρου. Ένας καλά ρυθμισμένος PID ελεγκτής είναι μια αποτελεσματική και απλή λύση για πολλά προβλήματα ελέγχου.



# Ο ΑΥΤΟ-ΡΥΘΜΙΖΟΜΕΝΟΣ PID ΕΛΕΓΚΤΗΣ

## 4.1 Εισαγωγή

**Σ**ΤΑ προηγούμενα κεφάλαια έγινε ανάλυση των κλασικών μεθόδων ρύθμισης ενός PID ελεγκτή και παρουσιάστηκαν οι τρόποι με τους οποίους ο κάθε όρος του ελεγκτή επηρεάζει την απόκριση του τελικού συστήματος. Σε αυτό το κεφάλαιο παρουσιάζεται ο αυτο-ρυθμιζόμενος PID ελεγκτής που αναπτύχθηκε στα πλαίσια της διπλωματικής μου εργασίας. Αρχικά, γίνεται αναφορά στη θεωρία που αφορά την αυτόματη ρύθμιση του ελεγκτή. Στη συνέχεια, παραθέτονται λεπτομέρειες και επεξηγήσεις για τη δομή του προγράμματος που φτιάχτηκε σε LabVIEW έτσι ώστε ο αναγνώστης να καταλάβει τη λογική λειτουργίας του ελεγκτή και τη δομή του προγράμματος.

## 4.2 Θεωρία

**Μ**Ε τον όρο “αυτο-ρυθμιζόμενος” *PID* ελεγκτής ή αυτόματη ρύθμιση του *PID* ελεγκτή εννοούμε μία μέθοδο στην οποία τα κέρδη του ελεγκτή ρυθμίζονται αυτόματα μετά από απαίτηση του χρήστη. Συνήθως, ο χρήστης θα πατήσει ένα κουμπί ή θα στείλει μια εντολή στον ελεγκτή. Μια αυτόματη ρύθμιση του ελεγκτή περιλαμβάνει τα εξής τρία βήματα:

- Δημιουργία μιας διαταραχής του συστήματος.

- Εκτίμηση της απόκρισης που προκαλεί η διαταραχή.
- Υπολογισμός των παραμέτρων του ελεγκτή.

Αυτά αποτελούν και τα βήματα που εκτελεί και ένας έμπειρος ελεγκτής όταν κάνει χειροκίνητη ρύθμιση του ελεγκτή. Το σύστημα πρέπει να διαταραχθεί με κάποιον τρόπο έτσι ώστε να εκτιμηθούν οι δυναμικές που το διέπουν. Αυτό μπορεί να γίνει με πολλούς τρόπους όπως, για παράδειγμα, εισάγοντας ημιτονοειδή σήματα, παλμούς ή βηματικές αλλαγές στην είσοδο του συστήματος.

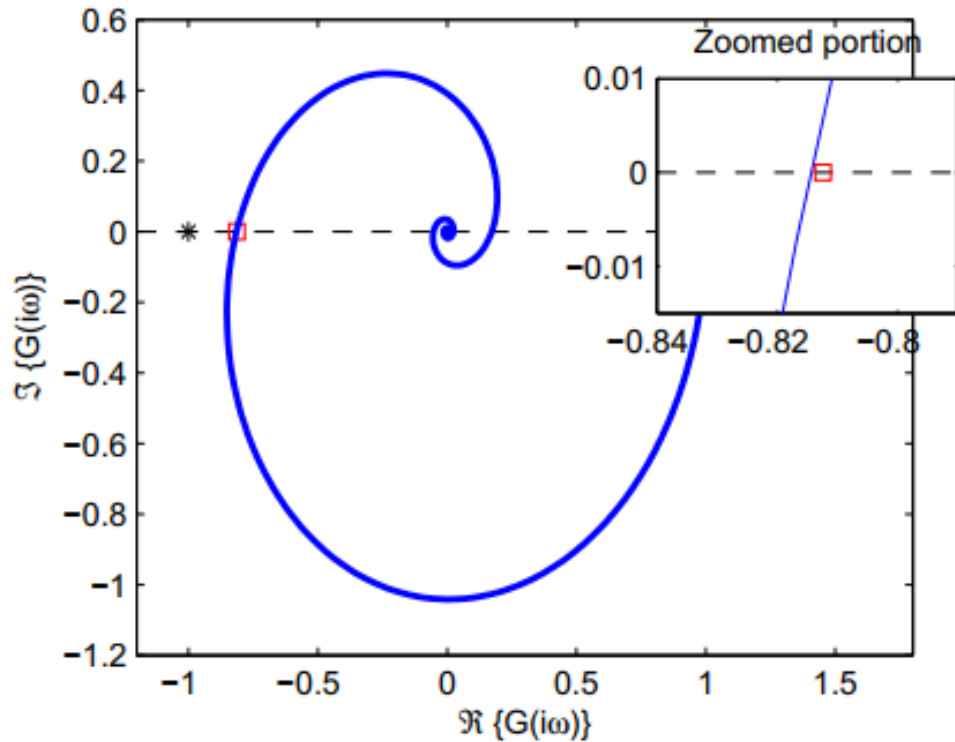
#### 4.2.1 Relay Method

Η μέθοδος που χρησιμοποιήθηκε σε αυτή την εργασία είναι αυτή που στη βιβλιογραφία αναφέρεται ως “*Relay Method*” (δεν υπάρχει ακριβής μετάφραση στα ελληνικά) [18], [24]. Η μέθοδος αυτή βασίζεται, όπως και οι μέθοδοι *Ziegler-Nichols* και *Tyres-Luyben*, στην απόκριση συχνότητας του συστήματος.

Στην πραγματικότητα, η μέθοδος *Ziegler-Nichols*, όπου το κέρδος που φέρνει το σύστημα στην οριακή ευστάθεια βρίσκεται πειραματικά, είναι μια μορφή αναγνώρισης μοντέλου. Όλες οι τεχνικές ρύθμισης στην πραγματικότητα περιέχουν κομμάτια αναγνώρισης μοντέλου, αλλά οι πιο δημοφιλείς απλά βελτιώνουν και αποκρύπτουν αυτά τα κομμάτια καλύτερα. Στην ουσία, όλη η διαδικασία ρύθμισης χρησιμοποιώντας τη μέθοδο *Ziegler-Nichols* γίνεται για να βρεθεί το κέρδος στο οποίο το σύστημα έχει μισό κύκλο καθυστέρηση όταν λειτουργεί σε ανατροφοδότηση. Αυτό είναι που αναφέρθηκε στην Ενότητα 3.2.2 ως απόλυτο κέρδος  $K_u$  και σχετίζεται με το σημείο όπου η καμπύλη Nyquist του συστήματος κόβει πρώτα τον πραγματικό άξονα, όπως φαίνεται στο Σχήμα 4.1.

Το πρόβλημα με τη μέθοδο αυτή είναι ότι για να βρεθεί το απόλυτο κέρδος, το σύστημα θα πρέπει να φτάσει στα όρια του, με ό,τι αρνητικές συνέπειες μπορεί να επιφέρει αυτό, τόσο στη λειτουργία του όσο και στην κατασκευή του. Η μέθοδος “Relay” που περιγράφεται εδώ καταφέρνει να βρίσκει πειραματικά τόσο το απόλυτο κέρδος όσο και την απόλυτη συχνότητα του συστήματος. Τα βήματα που ακολουθούνται για την εφαρμογή της μεθόδου είναι τα ακόλουθα:

1. Προσωρινά, ο ελεγκτής δίνει τη θέση του σε μία μη γραμμική συνάρτηση (στην ουσία είναι on-off έλεγχος) (Σχήμα 4.2) η οποία εναλλάσσει την είσοδο του συστήματος μεταξύ δύο διακριτών τιμών. Κάθε φορά που η τιμή του σφάλματος  $e(t)$  (δηλαδή *setpoint* – *output*) είναι μεγαλύτερη από το μηδέν η έξοδος του relay στοιχείου είναι  $d$ . Κάθε φορά που η τιμή του σφάλματος  $e(t)$  είναι μικρότερη από το μηδέν η έξοδος του relay στοιχείου είναι  $-d$ .
2. Αποφεύγοντας την εις βάθος μαθηματική ανάλυση [18], η αντικατάσταση του ελεγκτή με ένα relay στοιχείο, εξαναγκάζει το σύστημα να εκτελεί ταλαντώσεις σταθερού πλάτους και συχνότητας (Σχήμα 4.3).



Σχήμα 4.1: Διάγραμμα Nyquist ενός ανοιχτού σταθερού συστήματος με κάποια καθυστέρηση (deadtime)

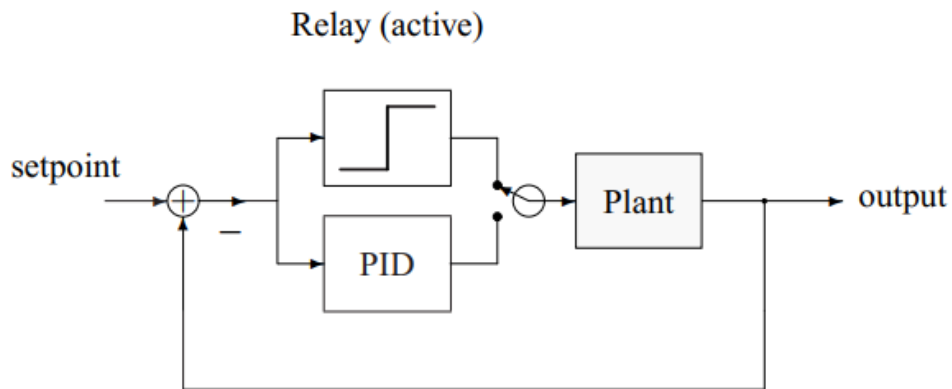
Η συχνότητα αυτή είναι σχεδόν ίδια με την απόλυτη συχνότητα του συστήματος, ενώ το απόλυτο κέρδος είναι αντιστρόφως ανάλογο του πλάτους της παρατηρούμενης ταλάντωσης.

3. Περιμένουμε η απόκριση του συστήματος να σταθεροποιηθεί σε αμείωτες ταλαντώσεις σταθερής συχνότητας και καταγράφουμε το πλάτος  $\alpha$  και την περίοδο  $T$ .
4. Η παρατηρούμενη περίοδος είναι η απόλυτη περίοδος, δηλαδή  $T_u = T$ , ενώ το απόλυτο κέρδος συνδέεται με το πλάτος  $\alpha$  με τη σχέση,

$$K_u = \frac{4d}{\pi\alpha} \quad (4.1)$$

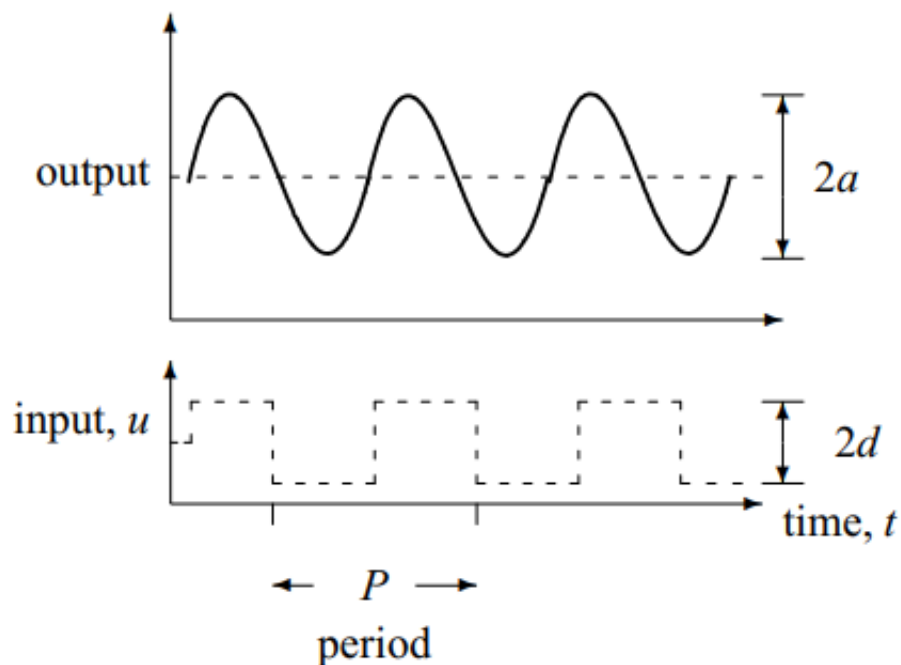
5. Έχοντας υπολογίσει το απόλυτο κέρδος και την απόλυτη περίοδο, μπορούν να οριστούν τα κέρδη του PID ελεγκτή με βάση τους Πίνακες 3.2 και 3.3 των μεθόδων Ziegler-Nichols και Tyreus-Luyben αντίστοιχα.

Η διαδικασία της αντικατάστασης του PID ελεγκτή με το relay στοιχείο, η μέτρηση του πλάτους και της περιόδου των ταλαντώσεων και ο υπολογισμός των κερδών του ελεγκτή μπορεί να αυτοματοποιηθεί αξιόπιστα. Αυτή η μέθοδος χρησιμοποιείται και σε εμπορικά διαθέσιμους PID ελεγκτές. Σημαντικό πλεονέκτημα της μεθόδου αυτής σε σχέση με τη μέ-



Σχήμα 4.2: Δομικό διάγραμμα συστήματος με relay στοιχείο

θοδο Ziegler-Nichols είναι ότι το σύστημα δεν χρειάζεται να διεγείρεται στα όρια του. Επειδή το πλάτος των ταλαντώσεων  $a$  είναι ανάλογο του πλάτους  $d$  του relay, μπορούμε να ορίσουμε το πλάτος  $a$  έτσι ώστε οι ταλαντώσεις να μην είναι υπερβολικά μεγάλες και, κατά συνέπεια, επικίνδυνες για το σύστημα ή τον εξοπλισμό.



Σχήμα 4.3: Ταλαντώσεις ενός συστήματος υπό συνθήκες ανατροφοδότησης μέσω relay στοιχείου αντί για ελεγκτή.



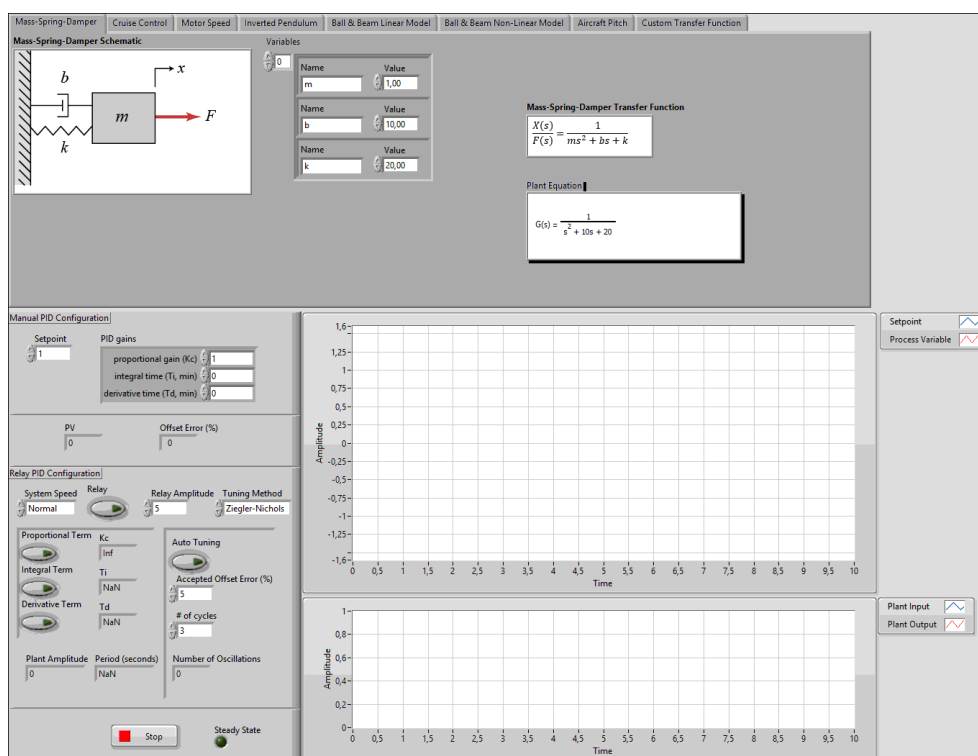
## 4.3 Το Πρόγραμμα

**Σ**ΤΗΝ ενότητα αυτή θα παρουσιαστεί το πρόγραμμα σε LabVIEW που υλοποιεί τον αυτο-ρυθμιζόμενο PID ελεγκτή. Για την καλύτερη οργάνωση του προγράμματος, το κυρίως VI αποτελείται από επί μέρους subVIs που το καθένα επιτελεί διαφορετική λειτουργία του προγράμματος. Σε κάθε υποενότητα λοιπόν θα παρουσιάζεται ένα από αυτά τα subVIs και θα αναλύεται η δομή του καθώς και η λειτουργία που επιτελεί.

### 4.3.1 Το Front Panel και το Block Diagram

#### Front Panel

Στο Σχήμα 4.4 φαίνεται το front panel του προγράμματος LabVIEW, το οποίο αποτελεί το γραφικό περιβάλλον του αυτο-ρυθμιζόμενου PID ελεγκτή και περιλαμβάνονται όλες οι επιλογές για τη ρύθμισή του.



Σχήμα 4.4: Το Front Panel του Αυτο-Ρυθμιζόμενου PID Ελεγκτή

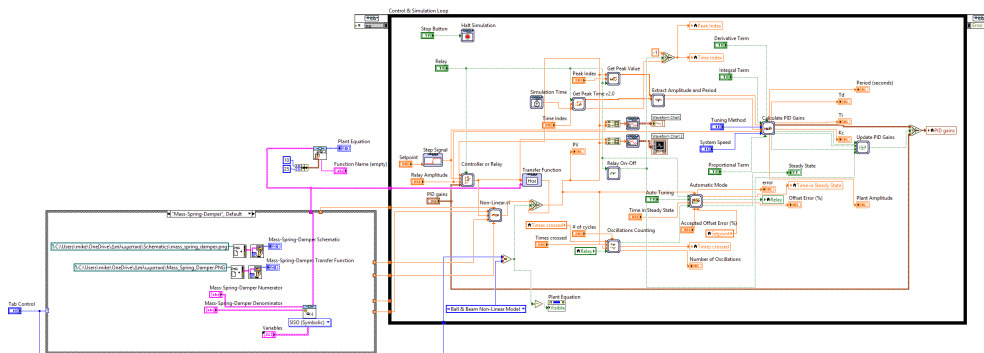
Στην κορυφή του front panel φαίνονται τα διαφορετικά φυσικά συστήματα που έχουν αναλυθεί και μοντελοποιηθεί προκειμένου να φανεί κατά πόσο ο αυτο-ρυθμιζόμενος PID ελεγκτής είναι ικανός να ελέγξει αποδοτικά το καθένα από αυτά.

Πιο κάτω, στα αριστερά, βρίσκονται τα κουμπιά ελέγχου του ελεγκτή καθώς και οι δείκτες που δείχνουν την τρέχουσα τιμή της εξόδου του συστήματος αλλά και το ποσοστό του σφάλματος που υπάρχει μεταξύ αυτής και της επιθυμητής τιμής της. Υπάρχουν επιλογές τόσο για χειροκίνητη ρύθμιση των κερδών του ελεγκτή από κάποιον χειριστή, όσο και επιλογές για την αυτόματη ρύθμιση αυτών, με βάση τη μέθοδο relay που περιγράφηκε προηγουμένως.

Στα δεξιά βρίσκονται τα διαγράμματα που θα παρουσιάσουν με γραφικό τρόπο την απόκριση του συστήματος. Το πάνω γράφημα δείχνει ταυτόχρονα τη βηματική απόκριση του κλειστού συστήματος και την επιθυμητή τιμή που θέλουμε να έχει η έξοδος του συστήματος. Το κάτω γράφημα δείχνει ταυτόχρονα την είσοδο και την έξοδο του συστήματος. Αυτό σημαίνει ότι στην περίπτωση που στο σύστημα είναι συνδεδεμένος ο ελεγκτής, το γράφημα αυτό θα μας δείχνει το σήμα ελέγχου και την απόκριση του συστήματος, ενώ αν είναι συνδεδεμένο το relay στοιχείο θα μας δείχνει τον τετράγωνο παλμό που θα παράγει αυτό σε σχέση με τις ταλαντώσεις του συστήματος.

## Block Diagram

Στο Σχήμα 4.5 φαίνεται ολόκληρο το block diagram. Αυτός είναι ουσιαστικά “ο κώδικας” του προγράμματος. Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, το LabVIEW είναι περιβάλλον γραφικής γλώσσας προγραμματισμού οπότε οι διάφορες συναρτήσεις του συνδέονται μεταξύ τους με καλώδια μέσω των οποίων περνάνε τα δεδομένα από τη μία συνάρτηση στην επόμενη.

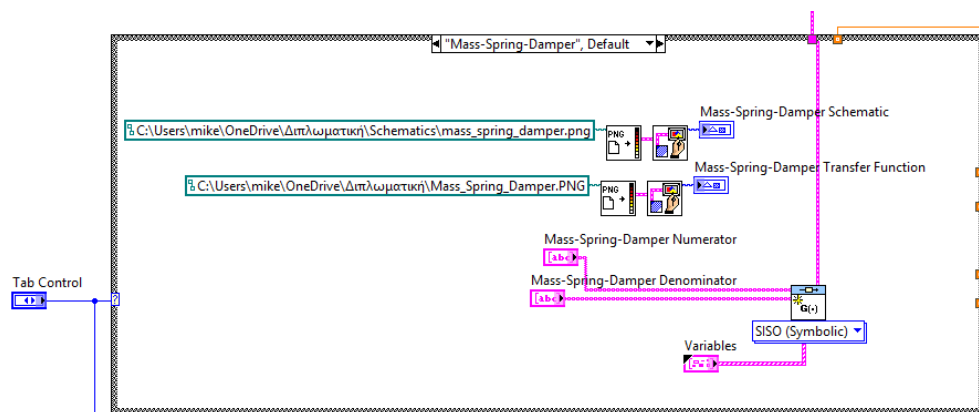


Σχήμα 4.5: Το Block Diagram του Αυτο-Ρυθμιζόμενου PID Ελεγκτή

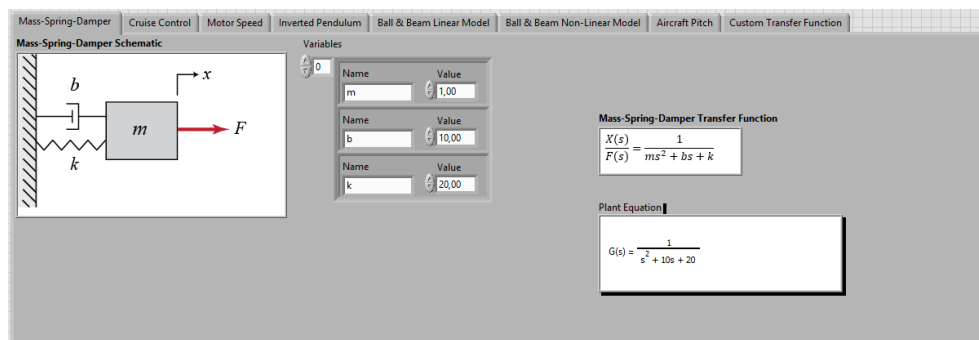
Στη συνέχεια κρίνεται σκόπιμο να αναλυθούν τα επιμέρους κομμάτια του block diagram, το καθένα ξεχωριστά, έτσι ώστε να γίνει κατανοητός ο τρόπος υλοποίησης των διαφόρων λειτουργιών του προγράμματος, καθώς και η δομή αυτού.

### 4.3.2 Tab Control

Στο Σχήμα 4.6 φαίνεται ο κώδικας που χρησιμοποιείται για την επιλογή του συστήματος που πρόκειται να ελεγχθεί. Ο χρήστης επιλέγει ποιο σύστημα από τα διαθέσιμα θέλει να ελέγξει χρησιμοποιώντας τις καρτέλες front panel (Σχήμα 4.7). Ανάλογα με την επιλογή του, αλλάζει η τιμή της μεταβλητής “Tab Control” του block diagram και επιλέγεται η κατάλληλη περίπτωση της δομής “Case Structure”. Κάθε case της δομής έχει παρόμοια διάταξη με αυτό που εμφανίζεται εδώ για το σύστημα “Mass-Spring-Damper”. Όλα περιλαμβάνουν ένα VI που υλοποιεί την εκάστοτε συνάρτηση μεταφοράς και δέχεται ως εισόδους τις τιμές των παραμέτρων, καθώς και δύο VIs που παράγουν το ένα την εικόνα του φυσικού συστήματος και το άλλο την εικόνα του τύπου της συνάρτησης μεταφοράς. Στην συνέχεια, το μοντέλο της συνάρτησης μεταφοράς μαζί με τις τιμές των παραμέτρων, τροφοδοτούνται σε ένα VI που αναλαμβάνει να δείξει στο χρήστη την τελική μορφή της συνάρτησης μεταφοράς έχοντας αντικαταστήσει αριθμητικές τιμές στις μεταβλητές του συστήματος. Αυτή η ένδειξη είναι το “Plant Equation” που φαίνεται στο Σχήμα 4.7.



Σχήμα 4.6: Block diagram του Tab Control

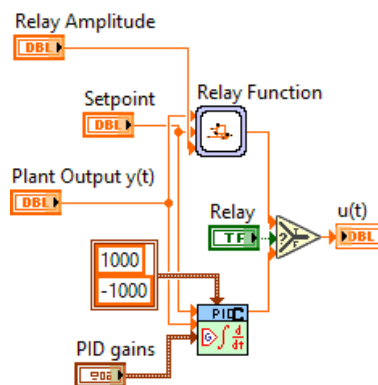


Σχήμα 4.7: Front panel του Tab Control



#### 4.3.4 Controller or Relay subVI

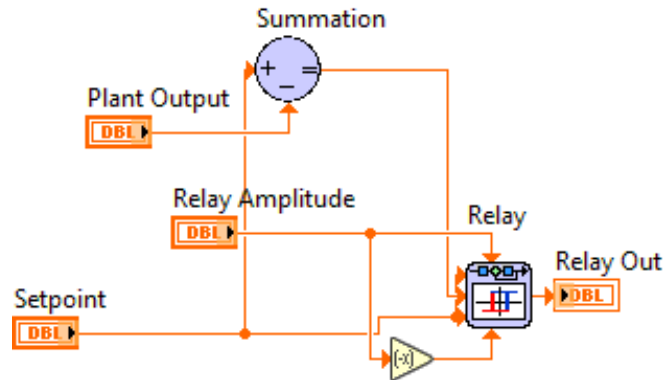
Στο Σχήμα 4.9 παρουσιάζεται το block diagram του subVI που αναλαμβάνει να αντικαταστήσει τον PID ελεγκτή με το relay στοιχείο έτσι ώστε να πραγματοποιηθεί το πείραμα. Το subVI αυτό δέχεται ως εισόδους την αριθμητική τιμή της εξόδου του συστήματος, την επιθυμητή τιμή (*setpoint*) που θέλουμε να έχει το σύστημα, το πλάτος του τετραγωνικού παλμού relay, τα κέρδη του PID ελεγκτή (*PID gains*) καθώς και ένα λογικό (*boolean*) σήμα με το όνομα “Relay”. Το λογικό αυτό σήμα ενεργοποιείται είτε μετά από εντολή του χρήστη πατώντας το αντίστοιχο κουμπί στο front panel, είτε αυτόματα όταν ικανοποιηθούν κάποιες προϋποθέσεις που έχει θέσει ο χρήστης. Το λογικό αυτό σήμα τροφοδοτείται στη δομή “Select”. Αν η τιμή της λογικής μεταβλητής είναι *True* τότε η δομή επιλέγει το πάνω σήμα που τροφοδοτείται σε αυτή, ενώ αν είναι *False* επιλέγει το κάτω. Έτσι λοιπόν καθορίζεται το σήμα ελέγχου  $u(t)$  που έπειτα τροφοδοτείται στο σύστημα. Αν δηλαδή το κουμπί “Relay” είναι πατημένο ο διακόπτης στο Σχήμα 4.2 βρίσκεται στην πάνω θέση, ενώ αν δεν είναι ο διακόπτης βρίσκεται στην κάτω θέση.



Σχήμα 4.9: Controller or Relay block diagram

#### Relay Function

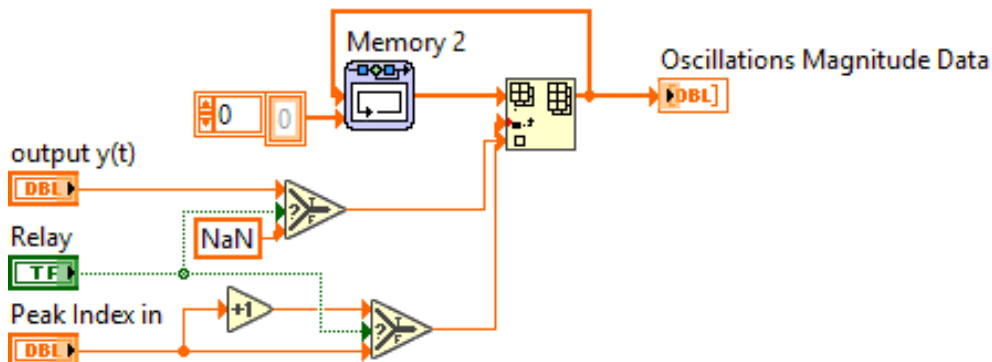
Όπως φαίνεται από το Σχήμα 4.9, μέσα στο “Controller or Relay” subVI υπάρχει ένα ακόμα subVI με όνομα “Relay Function”. Το block diagram αυτού παρουσιάζεται στο Σχήμα 4.10. Η συνάρτηση αυτή δέχεται ως εισόδους το *setpoint*, την έξοδο του συστήματος και το πλάτος του relay. Στη συνέχεια αφαιρεί την έξοδο του συστήματος από το *setpoint* υπολογίζοντας το σφάλμα. Αν το σφάλμα είναι μεγαλύτερο από το *setpoint* τότε η έξοδος του relay (*Relay Out*) είναι  $+Relay\ Amplitude$  ενώ αν είναι μικρότερο από το *setpoint* η έξοδος είναι  $-Relay\ Amplitude$ .



Σχήμα 4.10: Relay Function block diagram

#### 4.3.5 Record Oscillations Magnitude subVI

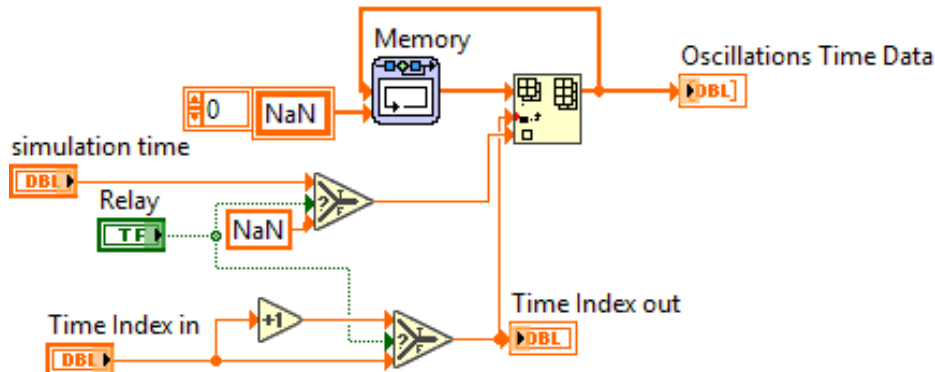
Όπως έχει ήδη αναφερθεί, προκειμένου να χρησιμοποιηθεί η μέθοδος για την αυτόματη ρύθμιση του PID ελεγκτή, θα πρέπει να μετρηθεί το πλάτος και η περίοδος των ταλαντώσεων του συστήματος όταν είναι συνδεδεμένο το relay στοιχείο στην είσοδο του. Προκειμένου να γίνεται καλύτερη διαχείριση μνήμης του προγράμματος και να μην πέφτει η απόδοσή του, τα δεδομένα της απόκρισης καταγράφονται μόνο όσο το λογικό σήμα “Relay” έχει την τιμή *True*. Το subVI αυτό είναι που αναλαμβάνει αυτή τη λειτουργία. Όσο λοιπόν το λογικό σήμα έχει αληθινή τιμή, το subVI κάνει εισχωρεί με χρονολογική σειρά την αριθμητική τιμή της εξόδου του συστήματος  $y(t)$  σε ένα πίνακα. Αυτός ο πίνακας, με όνομα “Oscillations Magnitude Data”, είναι η έξοδος του subVI. Όσο η τιμή του λογικού σήματος είναι μη αληθής, βάζει την τιμή NaN (*Not a Number*). Τα δεδομένα αυτού του τύπου αγνοούνται από το LabVIEW και συνεπώς δεν έχουν επίδραση στη συνολική απόδοση του συστήματος.



Σχήμα 4.11: Record Oscillations Magnitude block diagram

#### 4.3.6 Record Oscillations Time subVI

Το subVI αυτό λειτουργεί σε πλήρη αντιστοιχία με το προηγούμενο, με τη μόνη διαφορά ότι αυτό αντί να καταγράφει τις τιμές της απόκρισης του συστήματος, καταγράφει τις χρονικές στιγμές και τις βάζει στον πίνακα “Oscillations Time Data”. Όπως και πριν, αν η τιμή του “Relay” είναι μη αληθής τότε στον πίνακα περνάει δεδομένα του τύπου NaN.



Σχήμα 4.12: Record Oscillations Time block diagram

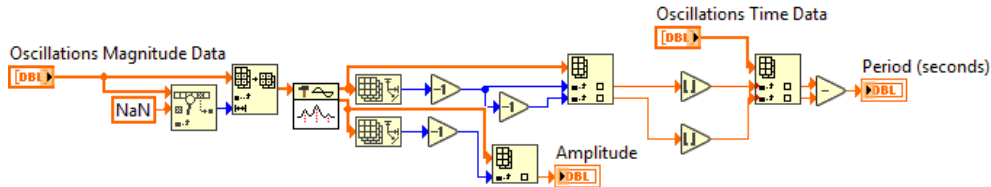
#### 4.3.7 Extract Amplitude and Period subVI

Το συγκεκριμένο subVI είναι υπεύθυνο για τον υπολογισμό του πλάτους και της περιόδου των ταλαντώσεων. Ως εισόδους δέχεται τους πίνακες “Oscillations Magnitude Data” και “Oscillations Time Data” από τις δύο προηγούμενες συναρτήσεις. Αν, κατά την εκτέλεση του προγράμματος, το κουμπί “Relay” πατηθεί πολλές φορές, το σύστημα θα εκτελέσει πολλά σετ ταλαντώσεων, ένα για κάθε ενεργοποίηση του “Relay”. Θα πρέπει λοιπόν το πρόγραμμα να είναι σε θέση να ξεχωρίσει το ένα σετ ταλαντώσεων από το άλλο. Για να γίνει αυτό ψάχνει στον πίνακα “Oscillations Magnitude Data” για την τιμή NaN. Όπως αναφέρθηκε προηγουμένως, όταν η τιμή του “Relay” είναι *False*, στους πίνακες μπαίνει η τιμή αυτή. Οπότε παίζει το ρόλο του διαχωριστικού μεταξύ των ταλαντώσεων που οφείλονται σε ξεχωριστό πάτημα του κουμπιού “Relay”.

Αφού βρεθούν, τα δεδομένα που αντιπροσωπεύουν την ίδια ταλάντωση εισάγονται στο “Peak Detector.vi” του LabVIEW το οποίο βρίσκει τα ακρότατα των δεδομένων καθώς και τις θέσεις του πίνακα που βρίσκονται αυτά τα ακρότατα. Σύμφωνα με τη θεωρία, το σύστημα εκτελεί ταλαντώσεις αμείωτου πλάτους, οπότε για να βρούμε το πλάτος, αρκεί να κρατήσουμε μία, οποιαδήποτε, από τις τιμές που μας δίνει ως ακρότατο το “Peak Detector.vi”. Επίσης, τα δεδομένα στους πίνακες “Oscillations Magnitude Data” και “Oscillations Time Data” είναι εντελώς συμμετρικά. Δηλαδή, στη θέση του πίνακα που υπάρχει η τιμή του πλάτους της ταλάντωσης, στην ακριβώς ίδια θέση του πίνακα που περιέχει τις χρονικές



στιγμές, θα υπάρχει η χρονική στιγμή κατά την οποία το σύστημα είχε το συγκεκριμένο πλάτος. Συνεπώς, για να βρούμε την περίοδο των ταλαντώσεων, το μόνο που χρειάζεται είναι να αφαιρεθούν οι χρονικές στιγμές δύο διαδοχικών μέγιστων ακρότατων.



Σχήμα 4.13: Extract Amplitude and Period block diagram

#### 4.3.8 Calculate PID Gains subVI

Αφού έχουν βρεθεί τα χαρακτηριστικά των ταλαντώσεων, συγκεκριμένα το πλάτος και η περίοδος, το μόνο που απομένει είναι ο αριθμητικός υπολογισμός των παραμέτρων. Αυτό αναλαμβάνει να κάνει το συγκεκριμένο subVI. Το υποσύστημα δέχεται ως εισόδους το πλάτος του τετραγωνικού παλμού, το πλάτος των ταλαντώσεων, την περίοδο των ταλαντώσεων, τη μέθοδο ρύθμισης (Ziegler-Nichols ή Tyreus-Luyben) και την επιθυμητή ταχύτητα απόκρισης που θέλουμε να έχει το σύστημα μετά τον έλεγχο (ισχύει μόνο για τη μέθοδο Ziegler-Nichols). Στο Σχήμα 4.14 έως και το Σχήμα 4.17, φαίνεται ο κώδικας κάθε περίπτωσης και πώς υπολογίζονται τα κέρδη. Αρχικά υπολογίζεται το απόλυτο κέρδος σύμφωνα με τον τύπο 4.1 και μετά ανάλογα με την τιμή που έχει το “Tuning Method” υπολογίζονται οι τιμές των  $K_p$ ,  $T_i$ ,  $T_d$ . Οι τύποι για τον υπολογισμό των παραμέτρων συνοψίζονται στον πίνακα 4.1. Μια μικρή λεπτομέρεια που αξίζει να αναφερθεί για αποφυγή μπερδεμάτων είναι ότι στον κώδικα, οι χρονικές παράμετροι  $T_i$  και  $T_d$  διαιρούνται με το 60. Αυτό γίνεται επειδή το “PID.vi” του LabVIEW που έχει χρησιμοποιηθεί στην εργασία δέχεται τις παραμέτρους σε λεπτά και όχι σε δευτερόλεπτα όπως έχουν υπολογιστεί αρχικά.

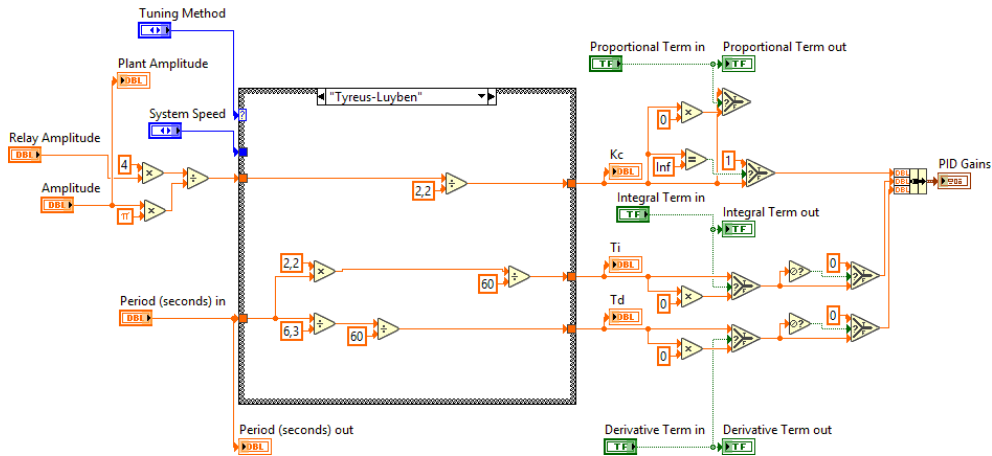
Μέθοδος ελέγχου	Ταχύτητα	$K_p$	$T_i$	$T_d$
Tyreus-Luyben	–	$K_u/2.2$	$2.2T_u$	$T_u/6.3$
Ziegler-Nichols	Fast	$0.6K_u$	$T_u/2$	$T_u/8$
Ziegler-Nichols	Normal	$0.33K_u$	$T_u/2$	$T_u/3$
Ziegler-Nichols	Slow	$0.2K_u$	$T_u/0.1$	$T_u/0.8$

Πίνακας 4.1: Τύποι για την αυτόματη ρύθμιση του ελεγκτή

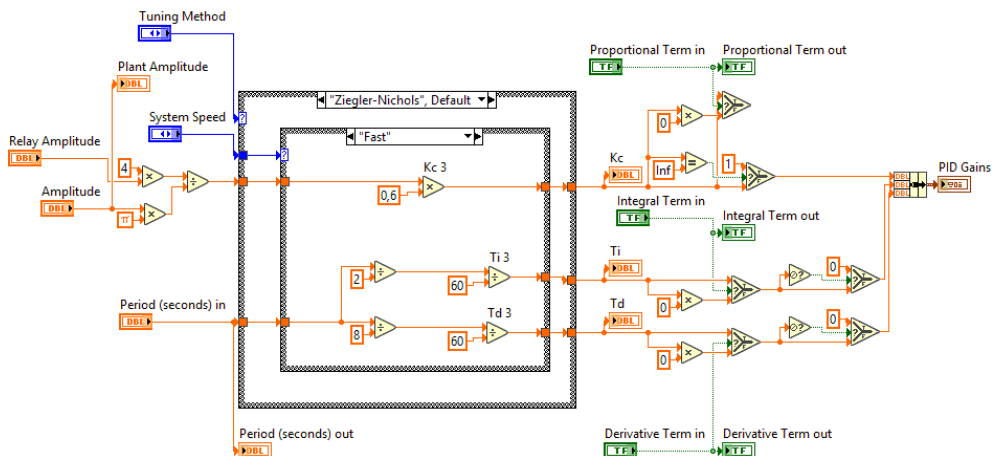
Μετά τον υπολογισμό τους, τα κέρδη εισάγονται στη δομή “PID Gains” που αποτελεί και την έξοδο του συστήματος. Προκειμένου να υπάρχει ομαλή εκτέλεση του προγράμματος, πριν ανανεωθούν τα κέρδη περνάνε



από κάποιους υπολογισμούς. Έτσι, αρχικά, αν τα κουμπιά για την ενεργοποίηση του κάθε όρου είναι απενεργοποιημένα τότε το κέρδος αυτού του όρου θα είναι μηδέν. Εξαιρέση αποτελεί ο αναλογικός όρος ο οποίος όταν είναι απενεργοποιημένο το κουμπί του το κέρδος του είναι μοναδιαίο, δηλαδή έχει ελάχιστη έως μηδαμινή επίδραση στην απόκριση του συστήματος. Αν ήταν μηδέν, τότε στην έξοδο του συστήματος δε θα βλέπαμε τίποτα καθώς όλα τα κέρδη θα ήταν μηδέν.

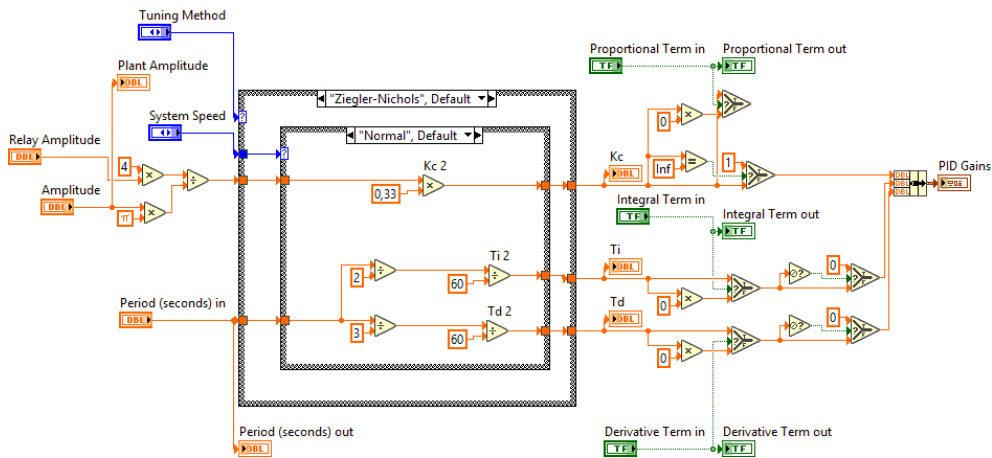


Σχήμα 4.14: Calculate PID Gains block diagram για την περίπτωση χρησιμοποίησης Tyreus-Luyben μεθόδου

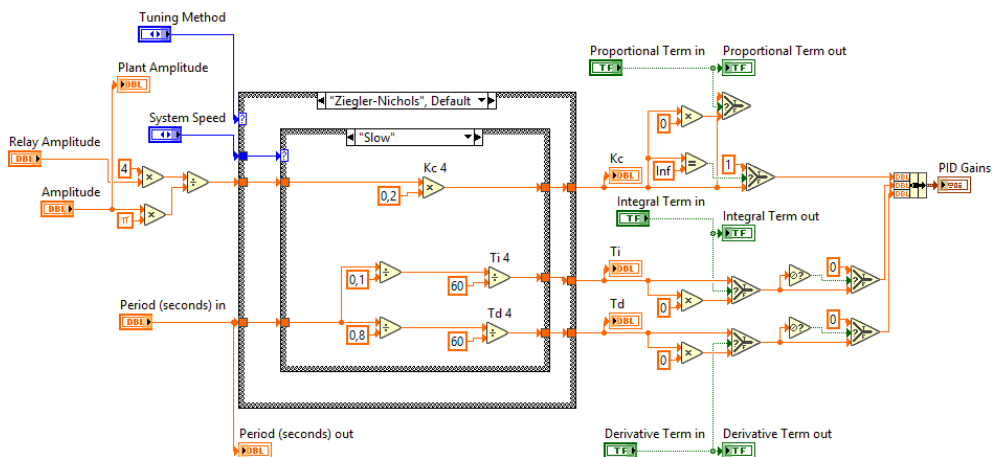


Σχήμα 4.15: Calculate PID Gains block diagram για την περίπτωση χρησιμοποίησης Ziegler-Nichols μεθόδου, γρήγορης ταχύτητας απόκρισης

Επίσης, σε κάθε επανάληψη του προγράμματος γίνεται έλεγχος για την τιμή που έχουν τα κέρδη. Αν οι τιμές τους δεν είναι κάποιος πραγματικός αριθμός αλλά κάποιου άλλου είδους δεδομένα, τότε η προσομοίωση του συστήματος θα αποτύχει και η απόκριση του συστήματος δε θα έχει σχέση με την πραγματική. Λόγω του τρόπου με τον οποίο έχει φτια-



Σχήμα 4.16: Calculate PID Gains block diagram για την περίπτωση χρησιμοποίησης Ziegler-Nichols μεθόδου, μεσαίας ταχύτητας απόκρισης

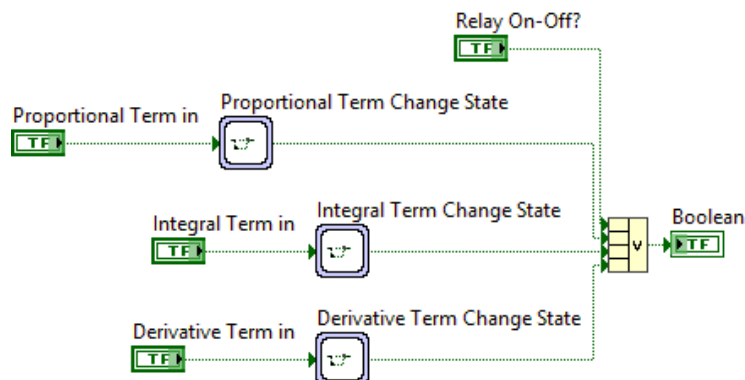


Σχήμα 4.17: Calculate PID Gains block diagram για την περίπτωση χρησιμοποίησης Ziegler-Nichols μεθόδου, αργής ταχύτητας απόκρισης

χτεί το πρόγραμμα, η προεπιλεγμένη τιμή για το αναλογικό κέρδος είναι άπειρο ( $Inf$ ) ενώ για τα άλλα δύο είναι το  $NaN$ . Ένας πραγματικός PID ελεγκτής όμως θα είχε ως αρχικά κέρδη τα  $K_p = 1$ ,  $T_i = 0$ ,  $T_d = 0$ . Συνεπώς, το πρόγραμμα ελέγχει το αναλογικό κέρδος αν έχει την τιμή  $Inf$ . Αν αυτό ισχύει, τότε η τιμή του γίνεται μονάδα, αλλιώς ο αλγόριθμος κρατάει αυτή που έχει. Ομοίως, για τα άλλα δύο κέρδη, ελέγχει αν έχουν την τιμή  $NaN$ . Αν αυτό ισχύει τότε οι τιμές τους γίνονται μηδενικές ενώ σε αντίθετη περίπτωση διατηρούν αυτές που είχαν.

### 4.3.9 Update PID Gains subVI

Στο Σχήμα 4.18 φαίνεται το δομικό διάγραμμα του συγκεκριμένου υποσυστήματος. Αυτό το subVI δέχεται ως εισόδους τα λογικά σήματα που παράγονται από το πάτημα των κουμπιών “Proportional Term”, “Integral Term”, “Derivative Term” και “Relay”. Τα σήματα αυτά περνάνε από κάποια άλλα επιμέρους subVIs τα οποία επιστρέφουν τιμή “Αληθής” αν το λογικό σήμα έχει αλλάξει κατάσταση, δηλαδή εάν από “Αληθής” η κατάσταση του έγινε “Ψευδής” ή το αντίθετο και επιστρέφουν τιμή “Ψευδής” αν το λογικό σήμα δεν έχει αλλάξει κατάσταση. Στο τέλος οι έξοδοι από τα υποσυστήματα αυτά περνάνε από μία πύλη που υλοποιεί τη λογική πράξη “AND” και αποδίδουν μια τιμή στην έξοδο. Αν αυτή η τιμή είναι “Αληθής” τότε τα κέρδη του ελεγκτή θα ανανεωθούν στις καινούριες τιμές, ενώ αν είναι “Ψευδής” θα παραμείνουν ως έχουν. Εν συντομία, κάθε φορά που πατιέται είτε το κουμπί “Relay” είτε κάποιο από τα κουμπιά των όρων του ελεγκτή, τα κέρδη ενημερώνονται έτσι ώστε η αλλαγή στην απόκριση να είναι άμεση χωρίς ο χρήστης να χρειάζεται να πατήσει κάποιο γενικό κουμπί ανανέωσης των κερδών.



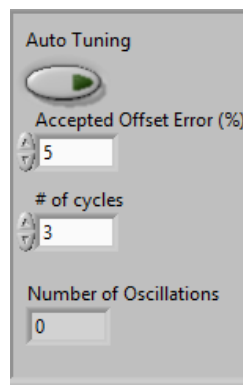
Σχήμα 4.18: Update PID Gains block diagram

### 4.3.10 Automatic Mode subVI

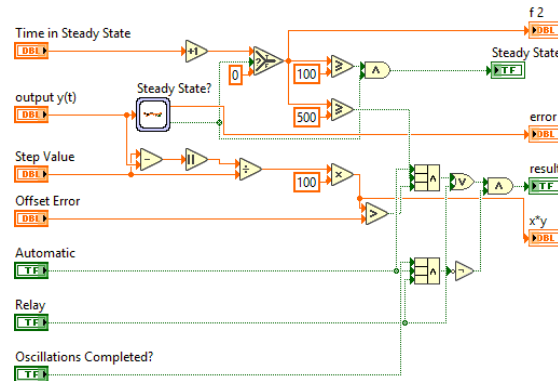
Τα controls που φαίνονται στο Σχήμα 4.19 αφορούν την αυτόματη λειτουργία του ελεγκτή. Το block diagram αυτής φαίνεται στο Σχήμα 4.20. Όταν το κουμπί “Auto Tuning” είναι ενεργοποιημένο, τότε ο ελεγκτής ελέγχει αν το σύστημα βρίσκεται σε μόνιμη κατάσταση. Αν αυτό ισχύει, και το σφάλμα του συστήματος είναι μεγαλύτερο από αυτό που ο χρήστης έχει ορίσει ως επιτρεπτό σφάλμα μόνιμης κατάστασης (Accepted Offset Error %), το κουμπί “Relay” ενεργοποιείται αυτόματα. Ο ελεγκτής επανέρχεται σε λειτουργία όταν το σύστημα εκτελέσει τόσες ταλαντώσεις όσες έχει ορίσει ο χρήστης στο πεδίο “# of cycles”. Προφανώς όσες περισσότερες ταλαντώσεις κάνει το σύστημα, τόσο πιο πιθανόν είναι αυτές να έχουν ξεπεράσει τα μεταβατικά φαινόμενα και να έχουν σταθερό πλάτος και περίοδο. Από τις

δοκιμές που πραγματοποιήθηκαν φάνηκε ότι περίπου τρεις ταλαντώσεις είναι αρκετές για να επέλθει αυτή η κατάσταση.

Με την πάροδο του χρόνου, τα χαρακτηριστικά μιας διεργασίας μπορεί να αλλοιωθούν και να αλλάξουν ελαφρώς. Αυτό έχει ως αποτέλεσμα, ο έλεγχος που στο παρελθόν ήταν ικανοποιητικός πλέον να κρίνεται ανεπαρκής. Η λειτουργία “Auto Tuning” αποσκοπεί στο να λύσει αυτό το πρόβλημα. Αν λοιπόν, για παράδειγμα, μια διεργασία στην οποία ο έλεγχος επέβαλλε μηδενικό σφάλμα μόνιμης κατάστασης αλλάξει και το σφάλμα πλέον έχει μια τιμή μεγαλύτερη από την αποδεκτή, ο ελεγκτής θα το αντιληφθεί και θα προσαρμοστεί αυτόματα στα καινούρια δεδομένα χωρίς την παρέμβαση του χρήστη.



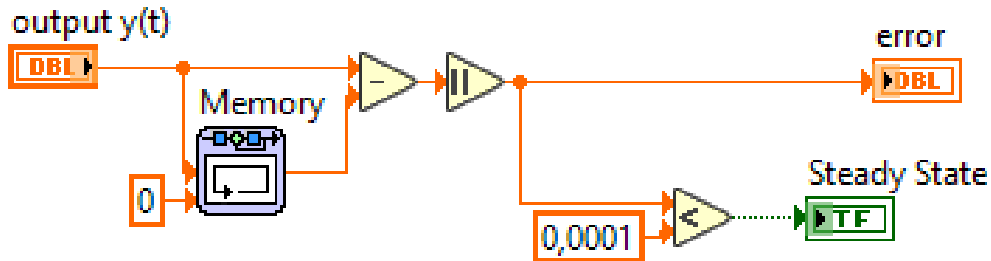
Σχήμα 4.19: Automatic Mode front panel



Σχήμα 4.20: Automatic Mode block diagram

Προκειμένου το σύστημα να μην ανταποκρίνεται σε θόρυβο αλλά μόνο στο σφάλμα του συστήματος, αυτή η λειτουργία ενεργοποιείται μόνο άμα το σύστημα βρίσκεται σε μόνιμη κατάσταση για περισσότερο από πέντε δευτερόλεπτα. Προκειμένου να καταλαβαίνει ο ελεγκτής ότι η απόκριση του συστήματος έχει σταθεροποιηθεί, έχει υλοποιηθεί το υποσύστημα “Steady State” που φαίνεται στο Σχήμα 4.21. Η λογική του είναι αρκετά απλή. Ουσιαστικά ελέγχει αν η τρέχουσα αριθμητική τιμή της εξό-

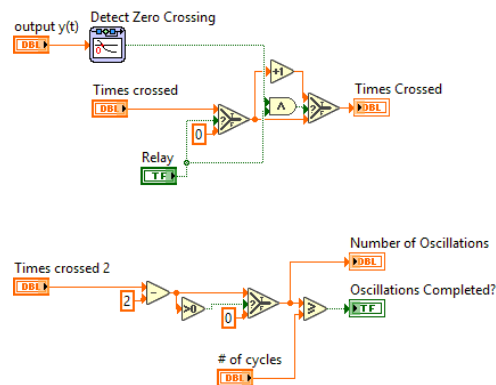
δου του συστήματος διαφέρει κατά 0.0001 από την προηγούμενη. Αν αυτό συμβαίνει τότε είναι ασφαλές να θεωρήσουμε ότι το σύστημα βρίσκεται σε μόνιμη κατάσταση και η έξοδος του subVI παίρνει την τιμή “Αληθής”.



Σχήμα 4.21: Steady State block diagram

#### 4.3.11 Oscillations Counting subVI

Προκειμένου να ξέρει το πρόγραμμα πότε να σταματήσει τις ταλαντώσεις όταν η αυτόματη επιλογή είναι ενεργοποιημένη, θα πρέπει να είναι σε θέση να μετράει τις ταλαντώσεις που έχει εκτελέσει. Ο κώδικας που το κάνει αυτό φαίνεται στο Σχήμα 4.22. Οι ταλαντώσεις που εκτελεί το σύστημα κατά τη διάρκεια του πειράματος είναι συμμετρικές γύρω από το μηδέν. Συνεπώς, σε κάθε επανάληψη του προγράμματος και εφόσον το κουμπί “Relay” είναι πατημένο, το subVI ελέγχει αν η έξοδος του συστήματος πέρασε την τιμή μηδέν πηγαίνοντας από τα αρνητικά στα θετικά. Κάθε φορά που αυτό συμβαίνει, αυξάνει την τιμή της μεταβλητής “Times Crossed” κατά ένα. Αυτή η τιμή ισούται με την τιμή των ταλαντώσεων που έχει εκτελέσει το σύστημα. Επειδή οι πρώτες ταλαντώσεις δεν είναι σταθερές, με την έννοια ότι δεν έχουν σταθερό πλάτος, ο αλγόριθμος ξεκινάει το μέτρημα μετά από τις πρώτες δύο. Τέλος, όταν η λειτουργία “Relay” απενεργοποιηθεί, είτε αυτόματα είτε από τον χρήστη, ο μετρητής επιστρέφει στο μηδέν.



Σχήμα 4.22: Oscillation Counting block diagram

## 4.4 Σύνοψη

Στο κεφάλαιο αυτό παρουσιάστηκε η θεωρία του αυτο-ρυθμιζόμενου PID ελεγκτή καθώς και η δομή του προγράμματος. Η λογική του κάθε υποσυστήματος αναλύθηκε εκτενώς και συμπεριλήφθηκαν εικόνες για την καλύτερη κατανόηση του κώδικα. Μετά από αυτό το κεφάλαιο, ο αναγνώστης θα πρέπει να έχει μια καλή ιδέα των αλγορίθμων του κάθε subVI καθώς και τον τρόπο που όλα μαζί συνδυάζονται προκειμένου να “χτιστεί” το κυρίως πρόγραμμα. Στο επόμενο κεφάλαιο θα παρουσιαστούν τα πειράματα που έγιναν και τα συμπεράσματα σχετικά με την αποτελεσματικότητα του αυτο-ρυθμιζόμενου PID ελεγκτή.

## ΚΕΦΑΛΑΙΟ

### 5

# ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

## 5.1 Εισαγωγή

**Σ**Ε αυτό το κεφάλαιο θα παρουσιαστούν τα πειράματα που έγιναν έτσι ώστε να μελετηθεί κατά πόσο ο ελεγκτής που σχεδιάστηκε είναι ικανός να ελέγξει ικανοποιητικά διάφορα συστήματα ελέγχου. Σε κάθε ενότητα θα παρουσιάζεται ένα από τα διαθέσιμα συστήματα, το μαθηματικό μοντέλο του και τα αποτελέσματα του ελέγχου που παρέχει ο αυτο-ρυθμιζόμενος PID ελεγκτής. Τέλος, η κάθε ενότητα κλείνει με το σχολιασμό των πειραματικών αποτελεσμάτων.





# 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
- [21] Li, Yun; Ang, Kiam Heong; Chong, Gregory C.Y. (February 2006). “Patents, software, and hardware for PID control: An overview and analysis of the current art”. *IEEE Control Systems Magazine*. 26 (1): 42–54. doi:10.1109/MCS.2006.1580153.
- [22] Soltesz, Kristian (January 2012). “On Automation of the PID Tuning Procedure” (Licentiate thesis). Lund university. 847ca38e-93e8-4188-b3d5-8ec6c23f2132.
- [23] Li, Y. and Ang, K.H. and Chong, G.C.Y. (2006) “PID control system analysis and design - Problems, remedies, and future directions”. *IEEE Control Systems Magazine*, 26 (1). pp. 32-41. ISSN 0272-1708
- [24] VanDoren, Vance (1 September 2009). “Relay Method Automates PID Loop Tuning”. *Control Engineering*.

Πανεπιστήμιο Πατρών, Πολυτεχνική Σχολή  
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών  
Μιχαήλ-Άγγελος Τριανταφύλλης του Άγγελου  
(σε ονομαστική πτώση)  
© Ιανουάριος 2018 – Με την επιφύλαξη παντός δικαιώματος.