



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ



Σχολή  
Μηχανικών  
Παραγωγής &  
Διοίκησης

# Εφαρμογή και σύγκριση μεθόδων συντονισμού παραμέτρων ενός PID ελεγκτή

ΕΛΕΥΘΕΡΙΟΥ ΣΤΕΛΙΟΣ

A.M.:2016010171

Επιβλέπων καθηγητής:

Κ. Ιψάκης Δημήτριος

Χανιά, Ιούνιος 2023

## Περίληψη

Στην παρούσα διπλωματική θα μελετήσουμε την σημασία των ελεγκτών P, PI, PID και θα εξετάσουμε ορισμένες μεθοδολογίες συντονισμού (tuning methods). Πιο συγκεκριμένα θα μελετηθεί και θα συγκριθεί η αποτελεσματικότητα τριών δημοφιλών μεθόδων συντονισμού παραμέτρων ενός ελεγκτή PID (Proportional-Integral-Derivative): Ziegler-Nichols, Cohen-Coon και Tyreus-Luyben. Η μελέτη στοχεύει να αξιολογήσει την απόδοση και την ευστάθεια του συστήματος ελέγχου χρησιμοποιώντας κάθε μέθοδο συντονισμού και να καθορίσει τη κατάλληλη μέθοδο για διαφορετικά συστήματα ελέγχου. Τα αποτελέσματα της μελέτης δείχνουν ότι κάθε μέθοδος συντονισμού έχει τα δικά της δυνατά και αδύνατα σημεία. Για τη διεξαγωγή των προσομοιώσεων, χρησιμοποιείται ένα μοντέλο N-τάξης και εκτελούνται προσομοιώσεις χρησιμοποιώντας κάθε μέθοδο συντονισμού.

Στο πρώτο κεφάλαιο θα αναλύσουμε την ιστορική αναδρομή και το εννοιολογικό πλαίσιο ενός PID ελεγκτή. Θα εξηγήσουμε την φυσική σημασία των τριών όρων του ελεγκτή και τους λόγους για τους οποίους ο συντονισμός των παραμέτρων είναι σημαντικός. Στην συνέχεια, στα επόμενα κεφάλαια θα εξετάσουμε μεθοδολογίες συντονισμού και τους τρόπους με τους οποίους υλοποιούνται θεωρητικά και έπειτα θα αναπτυχθούν κώδικες για αυτές τις μεθοδολογίες στο προγραμματιστικό περιβάλλον Matlab. Μέσα από την διεξοδική βιβλιογραφική μελέτη και τα παραδείγματα με την χρήση Matlab εντοπίζουμε τις διαφορές και την χρησιμότητα των μεθόδων.

Λέξεις-κλειδιά: ελεγκτής PID, μέθοδοι συντονισμού, συστήματα ελέγχου, απόδοση, ευστάθεια

## Abstract

In this thesis we will study the importance of P, PI, PID controllers and we will examine some tuning methods. More specifically, we will study and compare the effectiveness of three popular methods of tuning parameters of a PID (Proportional-Integral-Derivative) controller: Ziegler- Nichols, Cohen-Coon, and Tyreus-Luyben. The study aims to evaluate the performance and stability of the control system using each tuning method and determine the appropriate method for different control systems. The results of the study show that each coordination method has its own strengths and weaknesses. To conduct the simulations, an N-order model is used and simulations are performed using each tuning method.

In the first chapter we will analyze the historical background and conceptual framework of a PID controller. We will explain the physical meaning of the three controller terms and why parameter tuning is important. Then, in the following chapters we will examine tuning methodologies and how they are theoretically implemented and then develop codes for these methodologies in the Matlab programming environment. Through the thorough bibliographic study and the examples using Matlab we identify the differences and the usefulness of the methods.

Keywords: PID controller, tuning methods, control systems, , performance, stability

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα μου, επίκουρο καθηγητή μου Κ. Ιψάκη Δημήτριο Πολυτεχνείου Κρήτης, για την καθοδήγηση που μου προσέφερε και το χρόνο που διέθεσε δίνοντας μου χρήσιμες οδηγίες και συμβουλές για την ολοκλήρωση της διπλωματικής μου εργασίας. Επιπλέον οφείλω ένα μεγάλο ευχαριστώ στους γονείς και τον αδελφό μου για την ηθική υποστήριξη σε όλο το διάστημα των σπουδών μου, καθώς και τους συγγενείς και του φίλους μου.

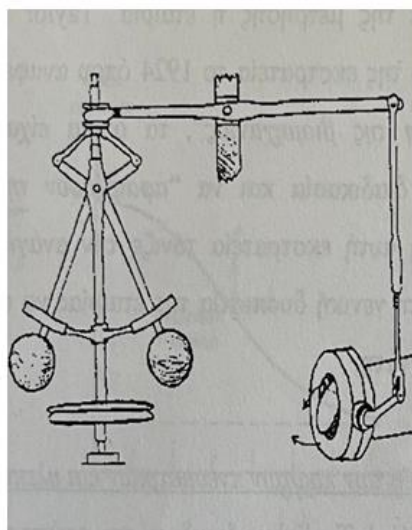
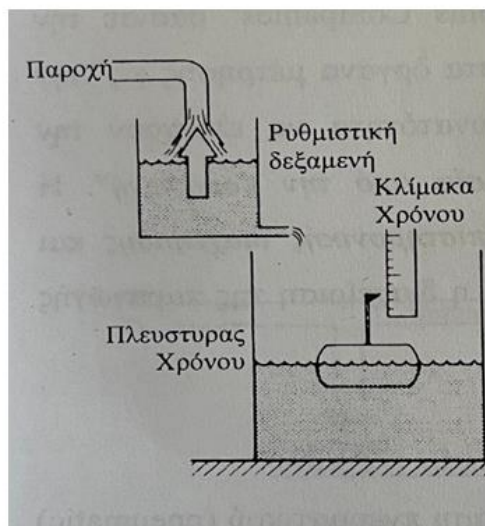
## Περιεχόμενα

Περίληψη.....	2
Abstract .....	3
Ευχαριστίες.....	4
Κεφάλαιο 1 <sup>ο</sup> Ιστορική Αναδρομή και τι είναι ένας PID ελεγκτής .....	6
1.1 Σύστημα με ανατροφοδότηση .....	9
1.2 Οι τρεις όροι του ελεγκτή .....	11
1.3 Η σημασία και η αξία του συντονισμού παραμέτρων (tuning).....	15
1.4 Βιβλιογραφική ανασκόπηση των μεθοδολογιών tuning.....	16
Κεφάλαιο 2: Μεθοδολογίες Συντονισμού Παραμέτρων PID Ελεγκτών .....	17
Κεφάλαιο 3 .....	38
3.1 Ανοικτός Βρόγχος – Φούρνος υψηλών θερμοκρασιών .....	38
Συμπεράσματα .....	49
Βιβλιογραφία .....	52

## Κεφάλαιο 1<sup>ο</sup>

### Ιστορική Αναδρομή και τι είναι ένας PID ελεγκτής

Ένας PID ελεγκτής αποτελεί τον πρώτο ελεγκτή και ίσως τον μοναδικό βιομηχανικό ελεγκτή ευρείας γενικής χρήσης. Μπορούμε να πούμε ότι οι αυτοματισμοί και ο αυτόματος έλεγχος έχει τις ρίζες του στην αρχαιότητα. Χαρακτηριστικά παραδείγματα αποτελεί ο αυτοματισμός του σχήματος παρακάτω που χρησιμοποίησε ο Κτησίβιος τον 3ο αιώνα π.Χ. για την ρύθμιση της ροής του νερού σε μια κλεψύδρα και επίσης ο ρυθμιστής του Watt το 1788, ο οποίος χρησιμοποιήθηκε για τον έλεγχο στροφών των ατμομηχανών. Ο τελευταίος αποτελεί ίσως τον πρώτο ελεγκτή με ανάδραση που είχε εκτεταμένη χρήση. Ο ελεγκτής αυτός κάνοντας χρήση της φυγόκεντρης δύναμης υλοποιούσε πρακτικά ένα αναλογικό ελεγκτή στροφών [1].



Εικόνα 1.1: Ο αυτοματισμός του Κτησίβιου[1]      Εικόνα1.2 :Ο ρυθμιστής του Watt [1]

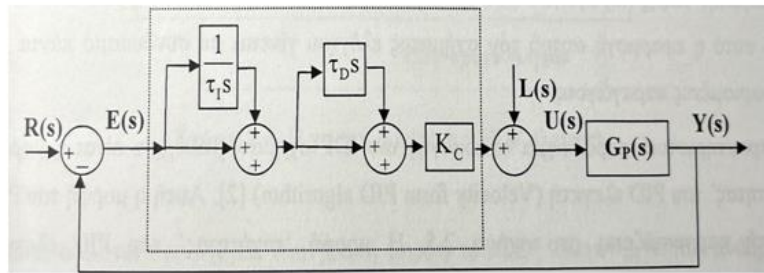
Γενικά ένας PID ελεγκτής αποτελείται από τρεις όρους και ειδικότερα Proportional (Αναλογικό), Integral (Ολοκληρωτικό) και Derivative (Διαφορικό ή παραγωγικό) όρο. Ο κάθε όρος έχει την δικιά του σημασία και υλοποίηση του PID ελεγκτή όπου θα εξηγήσουμε στην συνέχεια. Τα βασικότερα χαρακτηριστικά ενός PID ελεγκτή είναι ότι

έχει την ιδιότητα να επιτυγχάνει καλή απόδοση στο να απορρίπτει διαταραχές και να διατηρεί το σήμα αναφοράς[1].

Παράλληλα, διακρίνεται από ιδιαίτερη προσαρμοστικότητα καθώς μπορούμε να ρυθμίσουμε τις τρεις παραμέτρους με προκαθορισμένες μεθοδολογίες. Σημαντικό είναι να αναφέρουμε ότι βρίσκει ιδιαίτερες εφαρμογές στην βιομηχανία σε πλήθος εργασιών και μας επιτρέπει γρήγορους υπολογισμούς, ενώ συνήθως αποφεύγει και τυπικά προβλήματα αριθμητικής σύγκλισης. Στις πρώτες 3 δεκαετίες του 20<sup>ου</sup> αιώνα όλες οι ανθρώπινες δραστηριότητες όπως το εμπόριο και η βιομηχανία έτειναν προς την ιδέα της συστηματοποίησης.

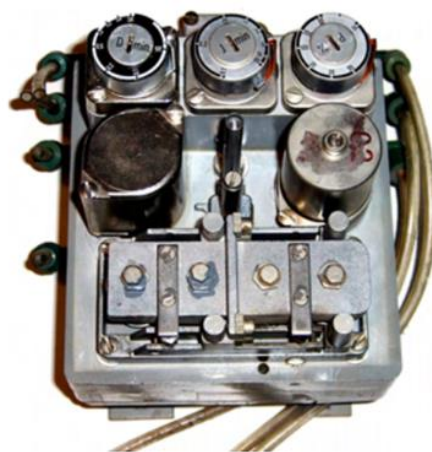
Ένα παράδειγμα ελεγκτή τύπου PID για αυτοματισμό πλοήγησης πλοίων αναπτύχθηκε το 1911 από τον Elmer Sperry. Η πρώτη θεωρητική ανάλυση πάνω σε αυτό αναπτύχθηκε στις αρχές της δεκαετίας του 1920 από τον μηχανικό Nicolas Minorsky, ο οποίος ερευνούσε και σχεδίαζε αυτόματα καθοδήγηση πλοίων για το πολεμικό ναυτικό των ΗΠΑ. Στόχος του ήταν η σταθερότητα του πηδαλιούχου και όχι ο γενικός έλεγχος όπου αυτό πρακτικά απλοποίησε σημαντικά το πρόβλημα [1].

Το πρόβλημα αυτό μοντελοποιήθηκε από τον ίδιο ως εξής. Ο αναλογικός έλεγχος παρείχε σταθερότητα έναντι μικρών διαταραχών, ήταν όμως ανεπαρκής για την αντιμετώπιση μιας σταθερής διαταραχής η οποία αυτή η διαταραχή απαιτούσε την προσθήκη του ολοκληρωτικού όρου. Τέλος ο παραγωγικός όρος προστέθηκε για να βελτιώσει την ευστάθεια και τον έλεγχο. Έπειτα στο USS New Mexico κάνανε δοκιμές πάνω σε αυτό, με τον ελεγκτή να ελέγχει την γωνιακή ταχύτητα και όχι την γωνία του πηδαλίου. Διεξάγοντας τους ελέγχους παρατηρήσεων παρατήρησαν ότι ο PI ελεγκτής απέδιδε σταθερή στροφή  $\pm 2^\circ$  και η προσθήκη του παραγωγικού όρου οδήγησε σε σφάλμα εκτροπής  $\pm 1/6^\circ$ , καλύτερα από ότι θα μπορούσε να πετύχει και ο καλύτερος πηδαλιούχος. Τελικά η μέθοδος αυτή δεν υλοποιήθηκε από το πολεμικό ναυτικό για το λόγο ότι το προσωπικό και οι χειριστές αντιστάθηκαν για την εγκατάσταση του συστήματος [2].



Εικόνα 1.3:Κλασσική μορφή του PID ελεγκτή εν-σειρά[1]

Μέχρι την δεκαετία του 40 η αξία του PID ελεγκτή είχε αποδειχθεί σε πολλές απαιτητικές εφαρμογές. Τα τρία βασικά θέματα που ζητούσαν λύση ήταν α) απλοποίηση επιλογής των παραμέτρων, β) πιο ανθεκτικός ο μηχανισμός των ελεγκτών και γ) να είναι πιο ανεξάρτητες οι δράσεις τους. Η λύση για το πρώτο πρόβλημα που αφορούσε την επιλογή των παραμέτρων ήρθε το 1942 από τους Ziegler and Nichols στην γνωστή εργασία «Optimum settings for automatic controllers». Όπου αυτήν την μέθοδο θα την εξετάσουμε αναλυτικά στο 2<sup>ο</sup> κεφάλαιο. Το δεύτερο πρόβλημα ήταν αρκετά δύσκολο να αντιμετωπιστεί και εξελίχθηκε σταδιακά μέχρι τις αρχές της δεκαετίας του 1950 οπότε ο R.E. Clarridge πρότεινε τον σχεδιασμό πνευματικού ελεγκτή με δύο «flapper-nozzle» ενισχυτών και έτσι με αυτό τον τρόπο υλοποιείται η γνωστή μορφή εν-σειρά PID ελεγκτής όπως φαίνεται παρακάτω:



Εικόνα 1.4:Πνευματικός ελεγκτής PID όπου οι συντελεστές P, I και D ρυθμίζονται από τους διακόπτες στην κορυφή[1]



Με το τέλος της δεκαετίας 1930 είχαν ήδη κατασκευαστεί ηλεκτρονικοί ελεγκτές. Μέχρι και το 1957 οι ηλεκτρονικοί ελεγκτές παρείχαν την δυνατότητα εφαρμογής μαθηματικών πράξεων. Είναι σημαντικό να αναφέρουμε σε μία αναφορά που έγινε στο DSIR (Department of Scientific and Industrial Research) το 1956, το 90% των ελεγκτών που χρησιμοποιούσαν ήταν ακόμη πνευματικοί ελεγκτές. Ο λόγος που ακόμη χρησιμοποιούσαν ελεγκτές πνευματικού τύπου ήταν η έλλειψη αξιοπιστίας στις λυχνίες. Έτσι ανέπτυξαν τρανζίστορ για να μπορέσουν οι ηλεκτρονικοί ελεγκτές να ανταγωνιστούν τους πνευματικούς. Ο πρώτος ελεγκτής «all solid state controller» προωθήθηκε από την Foxboro το 1957 [1].

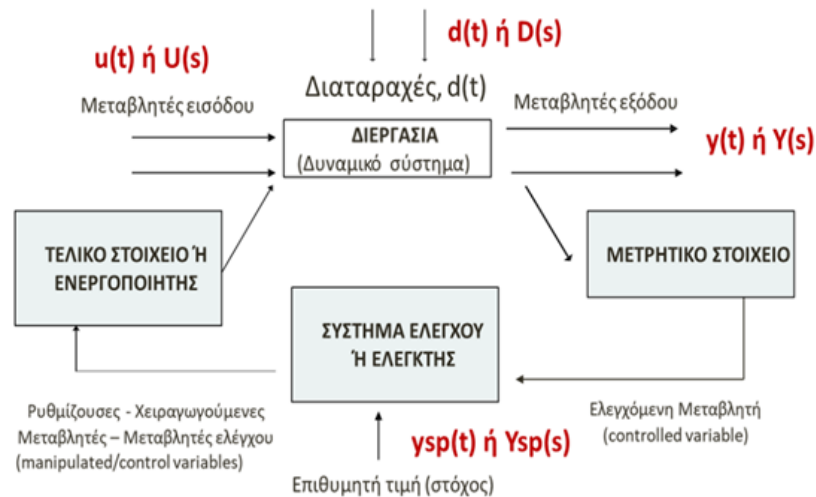
Η πρώτη χρήση ψηφιακού ελεγκτή σε βιομηχανική εφαρμογή έγινε το 1959 σε ένα σύστημα πολυμερισμού της Texaco. Μέχρι και το 1960 όλες οι μεγάλες βιομηχανίες και εταιρίες κατασκεύαζαν ψηφιακούς ελεγκτές. Παρατηρώντας την ραγδαία εξέλιξη και την ταχύτατη εξάπλωση των ψηφιακών ελεγκτών είναι ότι το 1960 υπήρχαν 100 βιομηχανίες που χρησιμοποιούσαν εγκαταστάσεις ψηφιακών ελεγκτών ενώ το 1965 ο αριθμός αυτός έφτανε μέχρι και τις 1000 βιομηχανικές εγκαταστάσεις.

Σήμερα η εφαρμογή του αυτόματου ελέγχου γίνεται σχεδόν αποκλειστικά από μικροεπεξεργαστές ενώ υπολογίζεται ότι το 95% όλων των κλειστών βρόγχων υλοποιούνται με PID ελεγκτές. Παρόλη όμως αυτή την ευρεία χρήση των ελεγκτών στις διάφορες βιομηχανικές εφαρμογές σε αρκετές περιπτώσεις οι βρόγχοι ελέγχου δεν ανταποκρίνονται στις αποκτήσεις που έχουν τεθεί αρχικά. Συγκεκριμένα υπολογίζεται ότι στις βιομηχανικές εφαρμογές μόνο το 20% των κλειστών βρόγχων λειτουργούν καλά [3].

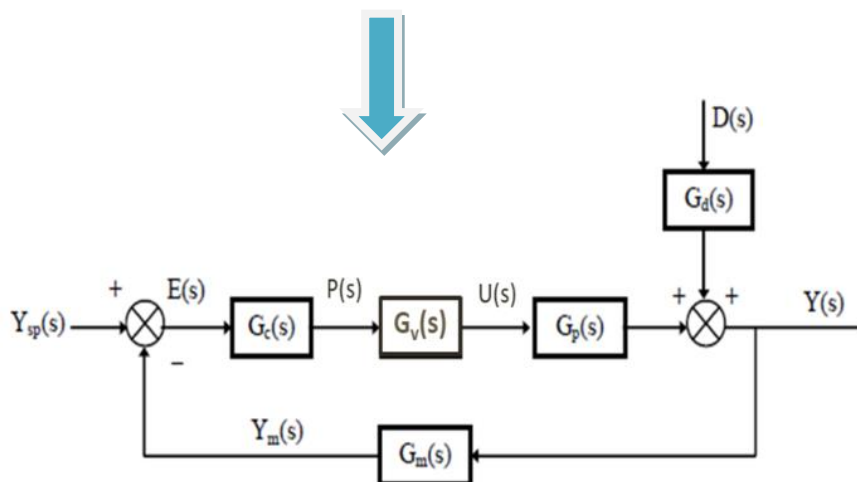
### **1.1 Σύστημα με ανατροφοδότηση**

Οι περισσότεροι ελεγκτές όπως και οι PID βασίζονται στην έννοια της ανάδρασης ή ανατροφοδότησης. Σε ένα σύστημα κλειστού βρόγχου υπάρχει η θετική ανατροφοδότηση ή αρνητική. Αρνητική ανατροφοδότηση είναι όταν η μεταβλητή ελέγχου μιας διαδικασίας κινείται αντίθετα από την έξοδο της διαδικασίας. Από την άλλη μεριά θετική ανατροφοδότηση είναι όταν οι κατευθύνσεις των δυο μεταβλητών είναι ταυτόσημες. Ιδιαίτερο ενδιαφέρον έχει η αρνητική ανατροφοδότηση για το λόγο ότι μειώνει την επιρροή των μεταβολών των παραμέτρων και ότι αναγκάζει την

διαδικασία να επανέλθει στο αρχικό σημείο λειτουργίας της και καταφέρνει τον μηδενισμό του σφάλματος μεταξύ της επιθυμητής και της πραγματικής τιμής της μεταβλητή παρά τις εξωτερικές διαταραχές.



Εικόνα 1.5: Διάγραμμα βαθμίδων κλειστού βρόχου με αρνητική ανατροφοδότηση



Εικόνα 1.6: Σύστημα κλειστού βρόχου με τις συναρτήσεις μεταφοράς

Όπου τα παραπάνω σημαίνουν τα εξής:

$Y_{sp}(s)$ : Επιθυμητό σήμα αναφοράς

$E(s)$ : Σφάλμα,  $E(s) = Y_{sp}(s) - Y_m(s)$

$G_c(s)$ : Ελεγκτής

$P(s)$ : Σήμα εξόδου από τον ελεγκτή

$Y(s)$ : Μεταβλητή εξόδου από το σύστημα

$G_v(s)$ : Τελικό στοιχείο ενεργοποίησης

$U(s)$ : Σήμα εξόδου από το τελικό στοιχείο ενεργοποίησης

$G_m(s)$ : Μετρητικό στοιχείο

$Y_m(s)$ : Μεταβλητή εξόδου από το μετρητικό

$G_p(s)$ : Σύστημα/Διεργασία/Διαδικασία

$G_d(s)$ : Διαταραχή

$D(s)$ : Μεταβλητή του συστήματος

## 1.2 Οι τρεις όροι του ελεγκτή

### Proportional Action (Αναλογική δράση):

Ο μαθηματικός τύπος του αναλογικού όρου δίνεται από την σχέση:

$$P(t) = p(t) - p(s) = Kc * E(t) \quad (1.1)$$

Όπου  $Kc$  η σταθερά ενίσχυσης του ελεγκτή και αφού όμως το  $P(t)$  είναι ανάλογο του  $E(t)$  τότε έχουμε το εξής :

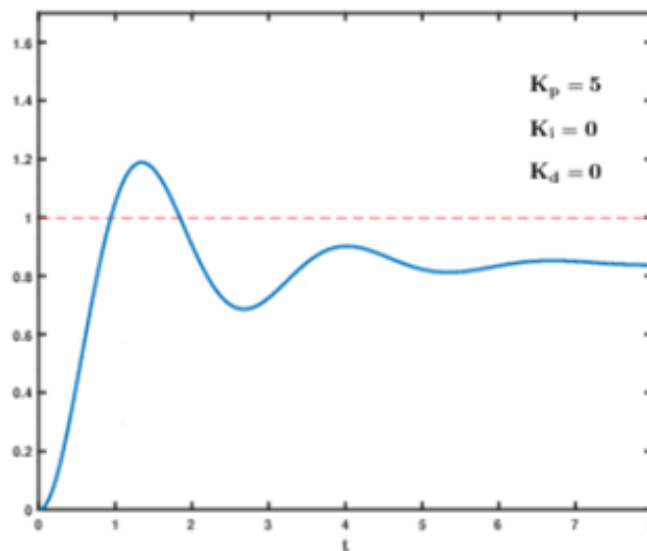
$$P(t) = Kc * E(t) \quad (1.2)$$

Στην συνέχεια με την χρήση μετασχηματιστών Laplace και μετασχηματίζοντας την πιο πάνω εξίσωση θα πάρουμε τον τελικό τύπο του αναλογικού όρου :

$$L\{P(t)\} = L\{Kc * E(t)\} \Leftrightarrow P(s) = Kc * E(s) \quad (1.3)$$

Ο αναλογικός όρος όπως φαίνεται και στον τύπο ασκεί αρνητική ανάδραση. Αν πάρουμε ένα ελατήριο και θέσουμε ότι όπου το  $Kc$  θα είναι η σκληρότητα του ελατηρίου. Είναι προφανές ότι όσο μεγαλύτερη είναι η απόκλιση από την θέση ισορροπίας τόσο μεγαλύτερη θα είναι η δύναμη επαναφοράς που ασκείται. Θα παρατηρήσουμε ότι όσο αυξάνεται το  $Kc$  τόσο αυξάνεται και η δύναμη επαναφοράς και αυτό μας λέει με λίγα λόγια ότι ο αναλογικός όρος του ελεγκτή μπορεί να κάνει το κλειστό σύστημα γρήγορο ενώ ταυτόχρονα μειώνει το σφάλμα στην μόνιμη κατάσταση [1]. Σημαντικό είναι να προσθέσουμε εδώ ότι για τα συστήματα δεύτερης τάξης και άνω τα πράγματα δεν είναι τόσο εύκολα. Σε μεγάλες τιμές του  $Kc$  και δεύτερης τάξης και άνω είναι πιθανόν ότι θα οδηγήσουν το σύστημα σε αστάθεια. Συμπεραίνοντας, ο αναλογικός ελεγκτής όταν το σύστημα υπόκειται σε εξωτερικές

διαταραχές θα οδηγηθεί σε μια ταλαντωτή συμπεριφορά και δεν είναι αρκετός πλέον να δώσει μηδενική τιμή στο σφάλμα στην μόνιμη κατάσταση. Για την εξάλειψη του σφάλματος στην μόνιμη κατάσταση χρησιμοποιούμε τον Integral (διαφορικό) όρο όπου θα εξηγήσουμε στην συνέχεια. Δίνοντας γραφικά ενός συστήματος με χρήση αναλογικού όρου [1].



Εικόνα 1.7: Επίδραση του αναλογικού όρου σε ένα σύστημα [1]

#### IntegralAction (Ολοκληρωτική δράση):

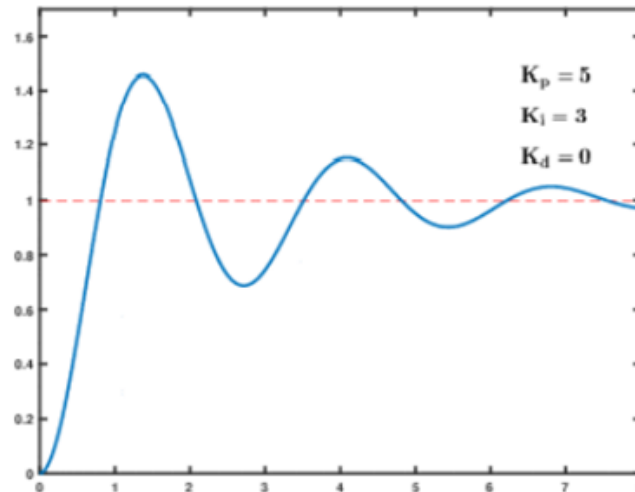
Ο μαθηματικός τύπος του ολοκληρωτικού όρου δίνεται από την σχέση :

$$P(t) = p(t) * p(s) = \left(\frac{Kc}{ti}\right) * \int E(t) (1.4)$$

Όπως αναφέραμε και πριν για την μαθηματική σχέση ισχύουν και τώρα. Η μόνη διαφορά είναι ότι προσθέσαμε το  $ti$  όπου είναι η ολοκληρωτική χρονική σταθερά ή αλλιώς ο χρόνος επαναφοράς. Με την ίδια μεθοδολογία όπως και πριν χρησιμοποιούμε τον μετασχηματισμό Laplace και έχουμε τα εξής :

$$L\{P(t)\} = L\left\{\left(\frac{Kc}{ti}\right) * \int E(t)\right\} \Leftrightarrow P(s) = \left(\frac{Kc}{ti}\right) * \left(\frac{1}{s}\right) * E(s) (1.5)$$

Ουσιαστικά ο ολοκληρωτικός όρος του ελεγκτή χρησιμοποιείται γιατί μπορεί να μηδενίσει το σφάλμα στην μόνιμη κατάσταση.



Εικόνα 1.8: Επίδραση του ολοκληρωτικού όρου στην προηγούμενη απόκριση με ίδια τιμή του αναλογικού κέρδους [1]

#### DerivativeAction (Διαφορικός όρος):

Ο διαφορικός όρος του ελεγκτή περιγράφεται από την σχέση:

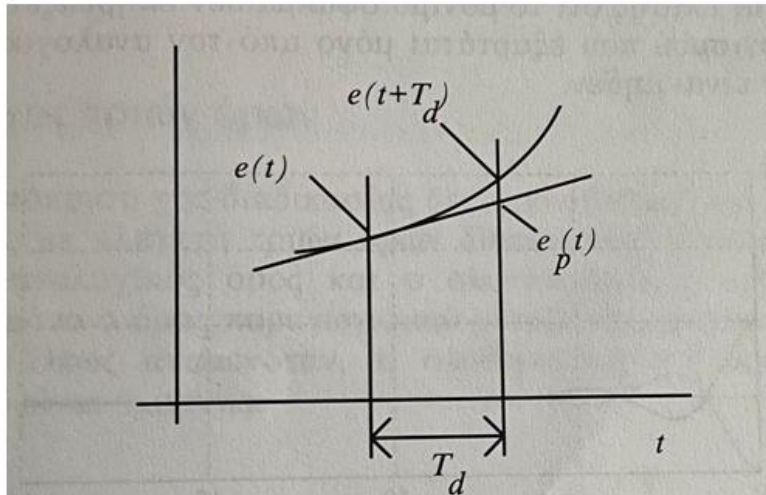
$$P(t) = p(t) - p(s) = (Kc * \tau D) * \frac{dE(t)}{dt} \quad (1.6)$$

Οτι προαναφέραμε και πιο πάνω ισχύει και εδώ με τον προσθετικό όρο  $\tau_d$  και το  $\tau_f$  που είναι η τιμή του φίλτρου. Μετασχηματίζοντας κατά τα γνωστά πλέον έχουμε το εξής:

$$L\{P(t)\} = L\left\{(Kc * \tau D) * \frac{dE(t)}{dt}\right\} \Leftrightarrow P(s) = Kc * \left(\frac{\tau D * s}{\tau_f * s + 1}\right) * E(s) \quad (1.7)$$

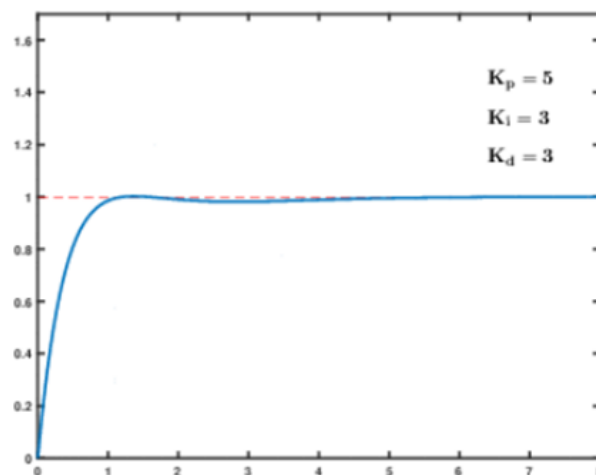
Ο διαφορικός όρος έχει την ιδιότητα να δρα προβλεπτικά δηλαδή μεταβάλλει την είσοδο του συστήματος ανάλογα με τον ρυθμό μεταβολής του σφάλματος ή της εξόδου του συστήματος. Ο διαφορικός όρος όμως δεν επιδρά καθόλου στο σφάλμα για τον προφανή λόγο ότι στην μόνιμη κατάσταση είναι ίση με το μηδέν. Αλλά όμως η χρήση του διαφορικού όρου τείνει να μειώνει αρκετά το σφάλμα που προκύπτει που προκύπτει από εξωτερικές διαταραχές. Ουσιαστικά η ιδιότητα του έρχεται από την

πληροφορία που μας δίνει η κλίση μιας μελλοντικής κατάστασης και έτσι αντιλαμβάνεται τις μεγάλες μεταβολές στο σφάλμα πριν το σφάλμα να γίνει αρκετά μεγάλο όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 1.9: Προβλεπτικός χαρακτήρας του διαφορικού όρου [1]

Ακόμη μία ιδιότητα είναι ότι ο διαφορικός όρος δίνει στο σύστημα αύξηση της ευρωστίας και με αυτό επιτυγχάνεται αύξηση της αναλογικής δράσης και συνεπώς το σύστημα μας γίνεται γρηγορότερο. Όταν όμως χρησιμοποιούμε αυτό τον όρο συνηθίζεται να χρησιμοποιούμε κάποιο φίλτρο για το λόγο ότι ο διαφορικός όρος προέρχεται από τις χαμηλές συχνότητες θορύβου της μέτρησης.



Εικόνα 1.10: Εφαρμογή του διαφορικού όρου με τα προηγούμενα σταθερά. [1]

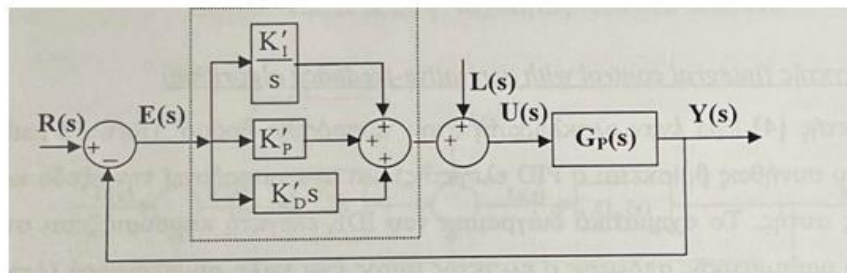
Συνοψίζοντας όλα αυτά που αναλύσαμε και εξηγήσαμε για τους τρεις όρους του ελεγκτή μας, είναι αυτοί που δίνουν στον ελεγκτή το όνομά του. Αθροίζοντας λοιπόν τις σχέσεις (1.1), (1.2) και (1.3) μετά από τον μετασχηματισμό Laplace παίρνουμε την τελική σχέση του PID ελεγκτή μας.

$$P(s) = Kc * E(s) + \left(\frac{Kc}{\tau i}\right) * \left(\frac{1}{s}\right) * E(s) + Kc * \left(\frac{\tau D * s}{\tau f * s + 1}\right) * E(s) \quad (1.8)$$

Βγάζοντας κοινό παράγοντα το  $E(s)$  και απλοποιώντας την πιο πάνω εξίσωση έχουμε

$$P(s) = Kc * \left(1 + \left(\frac{1}{\tau i * s}\right) + \left(\frac{\tau D * s}{\tau f * s + 1}\right)\right) * E(s) \quad (1.9)$$

Αυτή η μορφή PID χρησιμοποιείται συνήθως στην βιομηχανία που είναι γνωστή ως η παράλληλη μορφή όπως φαίνεται στην πιο κάτω εικόνα :



Εικόνα 1.11: Κλασική μορφή του PID ελεγκτή εν-παράλληλων μορφή

Εκτός από τις δύο μορφές που σχολίασαμε την εν-σειρά και παράλληλη μορφή υπάρχουν και άλλες με τις δικές τους ιδιότητες αλλά αυτές οι δυο είναι οι πιο γνωστές.

### 1.3 Η σημασία και η αξία του συντονισμού παραμέτρων (tuning)

Αρχικά ο συντονισμός παραμέτρων ενός PID ελεγκτή έχει να κάνει με την απόδοση των τιμών στο συντελεστή κάθε όρου, έτσι ώστε ο κάθε όρος ξεχωριστά να επηρεάσει θετικά την απόκριση του συστήματος και ταυτόχρονα να ισοζυγίζουν όσο γίνεται τα αρνητικά του κάθε όρου. Σημαντικό ρόλο στον συντονισμό είναι το τι επιθυμούμε να ελέγξει ο ελεγκτής μας. Αντιθέτως με την υλοποίηση ενός PID που είναι σχετικά απλή και εύκολη διαδικασία, ο σωστός συντονισμός είναι πολύ πιο περίπλοκο ζήτημα. Αυτό συμβαίνει για το λόγο ότι ο κάθε όρος επηρεάζει το συνολικό μας αποτέλεσμα.

Ένας κακός συντονισμός των παραμέτρων θα εμφανίσει αρκετά προβλήματα στο σύστημα μας όπως ταλαντώσεις, μη επαρκής απόσβεση, αργούς χρόνους ανύψωσης και άλλα πολλά. Το αναλογικό κομμάτι επιδρά στην σταθερά ενίσχυσης του ελεγκτή και αύξηση της σταθερά ενίσχυσης δεν επιτυγχάνεται η απόκριση κλειστού βρόγχου. Στο νολοκληρωτικό όρο αυξάνοντας το χρόνο επαναφοράς τείνει να επιβραδύνει την απόκριση κλειστού βρόγχου ενώ όταν μειώνεται την επιταχύνει. Τέλος με αύξηση του παραγόμενου χρόνου τείνει να βοηθήσει το σύστημα κλειστού βρόγχου να σταθεροποιηθεί και αυτό όμως προκαλεί ευαισθησία στις μετρήσεις μας. Με λίγα λόγια ο συντονισμός είναι το παν σε ένα PID ελεγκτή και πάντα η ρύθμιση γίνεται ανάλογα με το επιθυμητό αποτέλεσμα που θέλουμε να έχουμε. Άρα ένας καλός συντονισμός ενός PID ελεγκτή αν αυτό γίνεται στις βιομηχανίες την διασφάλιση της ποιότητας του προϊόντος, την μεγιστοποίηση της παραγωγής και ελαχιστοποίηση της ενέργειας και αξιοποίηση του εξοπλισμού.

#### **1.4 Βιβλιογραφική ανασκόπηση των μεθοδολογιών tuning**

Όπως είχαμε αναφέρει στα επόμενα κεφάλαια θα εξετάσουμε μεθοδολογίες συντονισμών των παραμέτρων και έπειτα από αναλυτική περιγραφή και τον τρόπο υλοποίησης της κάθε μεθόδου θα τις κωδικοποιήσουμε και θα πάρουμε κάποια αποτελέσματα και θα δούμε που εφαρμόζονται και αν στην βιομηχανία.

Οι μεθοδολογίες αυτές είναι οι εξής : Ziegler and Nichols, Tyreus – Luben, Cohen and Coon.



## Κεφάλαιο 2: Μεθοδολογίες Συντονισμού Παραμέτρων PIDΕλεγκτών

Το πρόβλημα ελέγχου μιας διαδικασίας είναι συχνά μη γραμμικό. Οι ελεγκτές PID, έχουν χρησιμοποιηθεί ευρέως για βιομηχανικές διεργασίες λόγω της απλότητας και της αποτελεσματικότητάς τους για γραμμικά συστήματα, ειδικά για συστήματα πρώτης και δεύτερης τάξης. Είναι γνωστό ότι ένα από τα κύρια πλεονεκτήματα των ελεγκτών Proportional Integral Derivative (PID) είναι ότι δεν απαιτείται μαθηματική μοντελοποίηση αφού οι κανόνες του ελεγκτή βασίζονται ειδικά στη γνώση της συμπεριφοράς του συστήματος και στην εμπειρία του μηχανικού ελέγχου.

Οι πιο κοινές κλασικές μέθοδοι συντονισμού ελεγκτών είναι οι μέθοδοι Ziegler-Nichols και Cohen-Coon. Αυτές οι μέθοδοι χρησιμοποιούνται συχνά όταν το μαθηματικό μοντέλο του συστήματος δεν είναι διαθέσιμο. Η μέθοδος Ziegler-Nichols μπορεί να χρησιμοποιηθεί τόσο για συστήματα κλειστού όσο και ανοιχτού βρόχου, ενώ η μέθοδος Cohen-Coon χρησιμοποιείται συνήθως για συστήματα ανοιχτού βρόχου.

Αξίζει να αναφέρουμε βασικές έννοιες που θα χρησιμοποιηθούν παρακάτω στο κείμενό μας, πιο συγκεκριμένα,

$K_p$ : αναλογικό κέρδος

$K_{max}$ : η τιμή του  $K_p$  στο σημείο της αστάθειας

$f_0$ : η συχνότητα της ταλάντωσης είναι  $f_0$ .

$K_c$ : ο κέρδος του ελεγκτή,  $K_c$

$K_u$ : το τελικό κέρδος

$P_u$ : περίοδος ταλάντωσης

$T_i$ : Ολοκληρωτικός χρόνος

$T_d$ : Παραγωγικός χρόνος.

Επιπλέον το ελάχιστο ποσό των κερδών του ελεγκτή που είναι απαραίτητο για τη διατήρηση των ημιτονοειδών ταλαντώσεων ονομάζεται τελική ευαισθησία ( $S_u$ ) ή τελικό κέρδος ( $K_u$ ) του συστήματος, ενώ η χρονική περίοδος μεταξύ διαδοχικών κορυφών ταλάντωσης ονομάζεται τελική περίοδος ( $P_u$ ) του συστήματος[1]. Τα παραπάνω βέβαια ίσως διαφοροποιούνται ανάλογα με την μέθοδο εφαρμογής.

## 2.1 Η μέθοδος Ziegler–Nichols

Μια δημοφιλής μέθοδος για τον συντονισμό ελεγκτών P, PI και PID είναι η μέθοδος Ziegler–Nichols. Η μέθοδος αυτή πήρε το όνομά της από τους John G. Ziegler και Nathaniel B. Nichols. Εισήχθη το 1942 και χρησιμοποιείται μέχρι σήμερα. Την προσέγγιση αυτή καθορίζουν δύο παράμετροι που σχετίζονται με την καθυστέρηση απόκρισης και τη μέγιστη ταχύτητα απόκρισης. Ξεκινά μηδενίζοντας τους όρους I, D και στη συνέχεια αυξάνει το αναλογικό κέρδος μέχρι το σύστημα να είναι ασταθές. Η τιμή του  $K_p$  στο σημείο αστάθειας αποτελεί τη συχνότητα της ταλάντωσης. Στη συνέχεια, η μέθοδος αφαιρεί το αναλογικό κέρδος σε ένα προκαθορισμένο ποσό και ορίζει τα ολοκληρωτικά και διαφορικά κέρδη συναρτήσει του πίνακα 2.1 [4].

Η μέθοδος Ziegler - Nichols στηρίζεται στον πειραματικό προσδιορισμό της βηματικής απόκρισης του συστήματος. Αποτελεί έναν ευρετικό κανόνα συντονισμού PID που επιχειρεί να παράγει καλές τιμές για τις τρεις παραμέτρους απολαβής PID:

$K_p$  - το αναλογικό κέρδος του ελεγκτή

$T_i$  - η σταθερά χρόνου του ολοκληρώματος του σφάλματος

$T_d$  - η χρονική σταθερά της παραγώγου του σφάλματος

Επίσης δίνονται δύο παράμετροι που προέρχονται από μετρήσεις που αφορούν (βλέπε Πίνακα 2.1):

- την περίοδο  $T_{max}$  της συχνότητας ταλάντωσης στο όριο ευστάθειας
- το περιθώριο κέρδους  $K_{max}$  για την ευστάθεια βρόχου με στόχο την επίτευξη καλής ρύθμισης (απόρριψη διαταραχών)[5].

Από την μία μεριά οι κανόνες ρύθμισης Ziegler – Nichols συνήθως οδηγούν σε συστήματα με ιδιαίτερα «επιθετική» απόκριση. Δεν παρέχουν δηλαδή μια έτοιμη λύση ρύθμισης, αλλά τις περισσότερες φορές προσφέρουν ένα αρκετά ικανοποιητικό σημείο εκκίνησης από το οποίο ο χρήστης μπορεί να ξεκινήσει να τροποποιεί τα κέρδη έτσι ώστε να έχει την επιθυμητή απόκριση. Από την άλλη, οι κανόνες συντονισμού λειτουργούν αρκετά καλά όταν έχει κανείς έναν αναλογικό ελεγκτή, ένα σύστημα που είναι γραμμικό, μονοτονικό και αργό, και μια απόκριση που κυριαρχείται από μια εκθετική "υστέρηση". Οι πραγματικές εγκαταστάσεις είναι απίθανο να έχουν ένα τέλειο χαρακτηριστικό υστέρησης πρώτης τάξης, αλλά αυτή η προσέγγιση είναι λογική για να περιγράψει την απόκριση συχνότητας στην πλειονότητα των περιπτώσεων.

### **Πλεονεκτήματα – Μειονεκτήματα μεθόδου**

Η προσέγγιση Ziegler-Nichols έχει το πλεονέκτημα ότι παρέχει πολύ φιλικούς προς τον χρήστη κανόνες συντονισμού. Τα μειονεκτήματα περιλαμβάνουν το ότι μπορεί να είναι χρονοβόρα. Επίσης ενώ δοκιμάζεται ο ελεγκτής μπορεί να οδηγηθεί σε ασταθείς περιοχές, κάτι που θα μπορούσε να προκαλέσει το σύστημα να βγει εκτός ελέγχου, οπότε θέλει ιδιαίτερη προσοχή η εφαρμογή της.

Η μέθοδος αποτελείται ουσιαστικά από δύο βήματα:

1. Προσδιορισμός των δυναμικών χαρακτηριστικών του βρόχου ελέγχου
2. Εκτίμηση των παραμέτρων συντονισμού του ελεγκτή που παράγουν μια επιθυμητή απόκριση για το δυναμικό χαρακτηριστικό που καθορίστηκε στο πρώτο βήμα, με άλλα λόγια, αντιστοίχιση του ελεγκτή με τα άλλα στοιχεία του βρόχου.

Πίνακας 2.1 Ρυθμίσεις για τα κέρδη P, I και D σύμφωνα με τη μέθοδο Ziegler–Nichols

	$K_p$	$T_I$	$T_D$
P controller	$0.5K_{MAX}$	0	0
PI controller	$0.45K_{MAX}$	$(1/1.2)T_{max}$	0
PID controller	$0.6K_{MAX}$	$0.5T_{max}$	$0.125T_{max}$

Με την χρήση του υπολογιστικού περιβάλλοντος Matlab υλοποιήσαμε την μέθοδο Ziegler&Nichols, αποτυπώνοντας προγραμματιστικά τις παραμέτρους και τις προϋποθέσεις που αναφέρουμε παραπάνω σε ένα σύστημα κλειστού βρόχου. Κάθε κλειστό σύστημα έχει την παρακάτω δομή, όπου:

- Η  $G_p(s)$  είναι η συνάρτηση μεταφοράς της υπό έλεγχο διαδικασίας
- Η  $G_c(s)$  είναι η συνάρτηση μεταφοράς του ελεγκτή τριών όρων (P,PI,PID)

Πιο συγκεκριμένα, η συνάρτηση μεταφοράς έχει την μορφή (τυχαίο παράδειγμα 3<sup>ης</sup> τάξης)

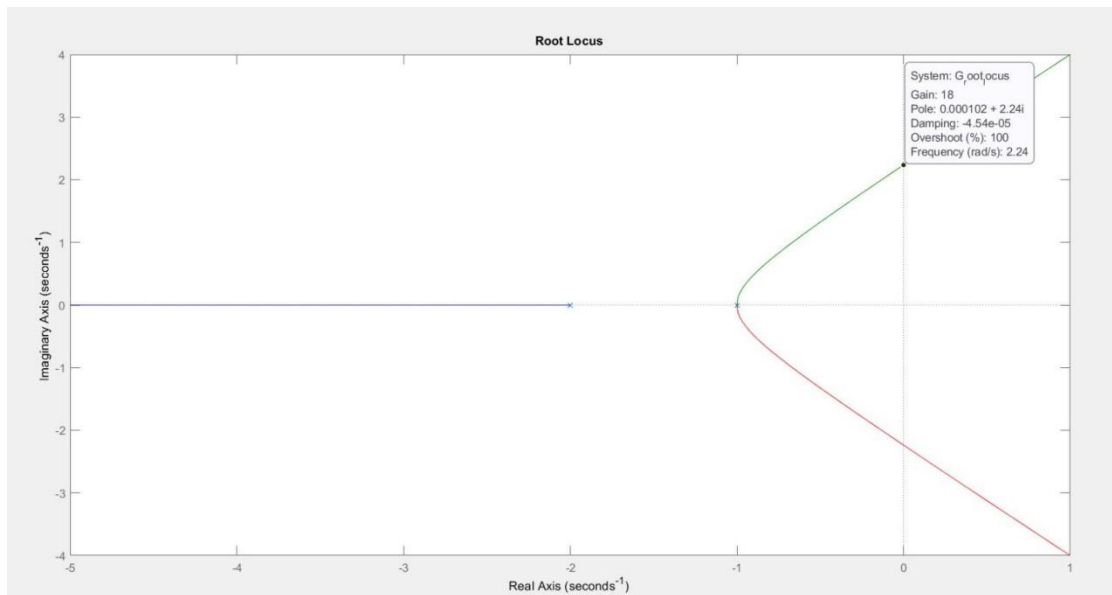
$$G_p = 1/((s + 1) * (s + 1) * (s + 2)) \quad (2.2)$$

Στόχος μας είναι ο υπολογισμός του μέγιστου κέρδους  $K_{max}$  και της περιόδου  $T_{max}$ . Πιο συγκεκριμένα, ονομάζεται  $K_{max}$  η τιμή του  $K_p$  στο σημείο της αστάθειας.

Οι υπολογισμοί μας πραγματοποιούνται για τους ελεγκτές P,PI , PID με τον εξής κώδικα και οι τιμές που προκύπτουν παρουσιάζονται παρακάτω. Ο κώδικας που εκτελείται έχει το όνομα ZN.m.

Μελετώντας την συνάρτηση μεταφοράς καταλήξαμε στα  **$K_{max} = 18.00$  &  $f_o = T_o = 2.8099$**  όπως φαίνεται στο παρακάτω διάγραμμα ταλαντώσεων στο οποίο παρατηρούμε το κρίσιμο σημείο με τον γεωμετρικό τόπο των ριζών. Το  $K_{max}$  ισούται με 18.00 και έπειτα κάνοντας την πράξη  $\frac{(2 \cdot p_i)}{2.24}$  λαμβάνουμε και την τιμή 2.8099.

Ο τύπος είναι ο εξής :  $T = \frac{(2 \cdot \pi)}{\omega}$ .

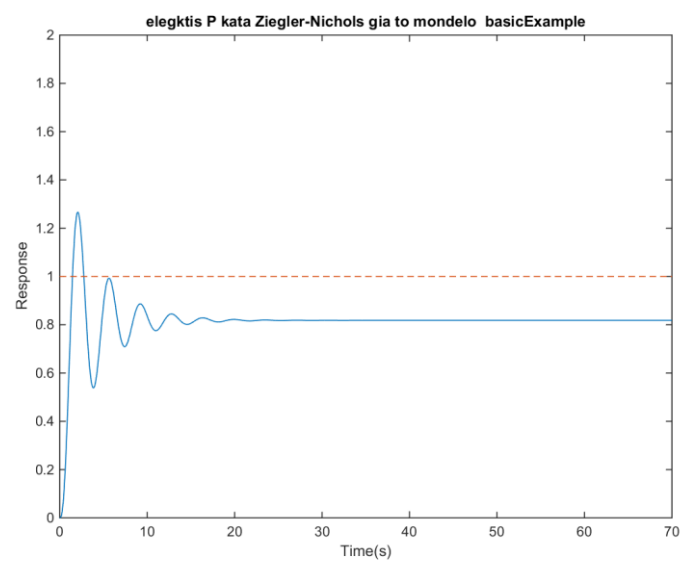


Στον παρακάτω πίνακα παρουσιάζονται τα αποτελέσματα του συντονισμού παραμέτρων.

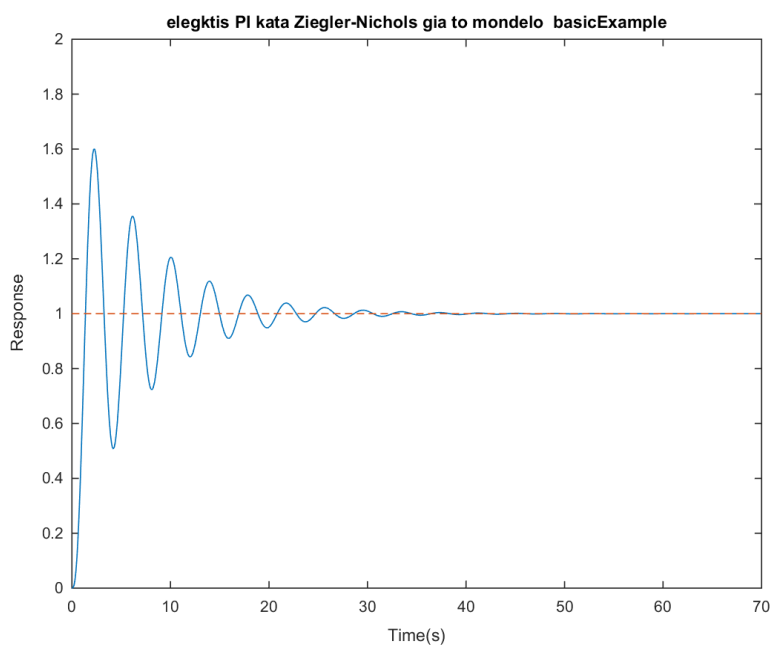
Πίνακας 2.2 Παράδειγμα Ziegler&Nichols

ΕΛΕΓΚΤΗΣ	$K_p$	$T_I$	$T_D$
P	9	0	0
PI	8.1	2.3416	0
PID	10.8	1.4050	0.3512

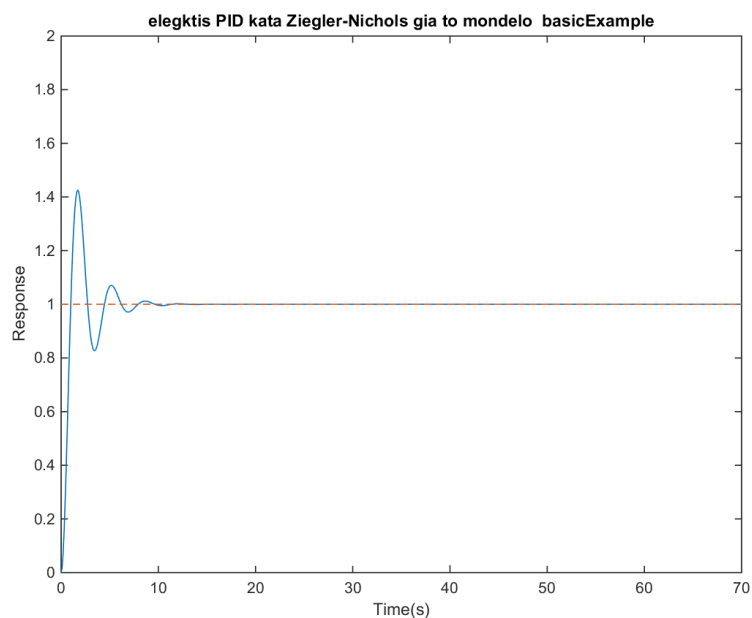
Επιπλέον, προκύπτουν τα παρακάτω διαγράμματα :



Εικόνα 2.1: Μέθοδος Ziegler&Nichols& ελεγκτής P



Εικόνα 2.2: Μέθοδος Ziegler&Nichols& ελεγκτής PI

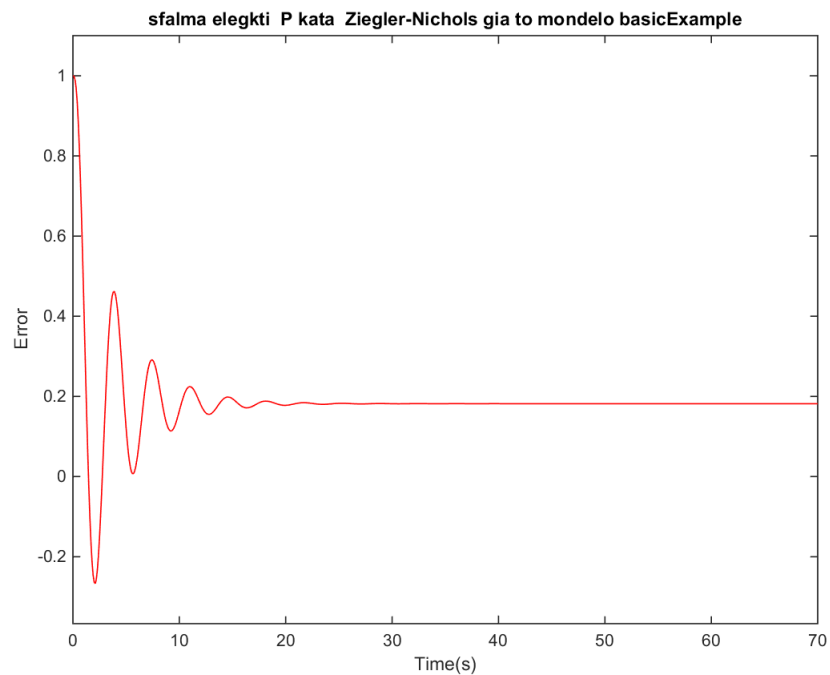


Εικόνα 2.3: Μέθοδος Ziegler&Nichols& ελεγκτής PID

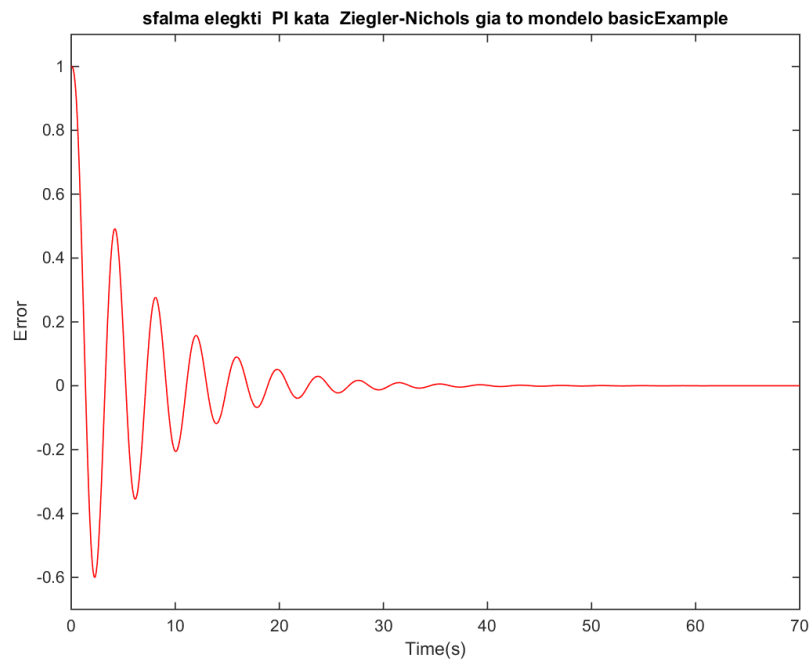
Όπως βλέπουμε από τις εικόνες 2.1-2.3, ο P ελεγκτής δεν καταφέρνει να φτάσει το σύστημα στην τελική κατάσταση, έχοντας ένα σφάλμα. Από την άλλη, οι ελεγκτές PI και PID οδηγούν επιτυχώς το σύστημα προς την επιθυμητή τελική κατάσταση, ενώ

μάλιστα ο ελεγκτής με ολοκληρωτικό παράγοντα (PID) το επιτυγχάνει αυτό στο περίπου 1/3 του χρόνου που χρειάζεται ο PI, καθώς χάρη στον διαφορικό όρο αποφεύγει τις υπερβολικές ταλαντώσεις γύρω από τη τελική κατάσταση που εμφανίζει το σύστημα με τον PI ελεγκτή.

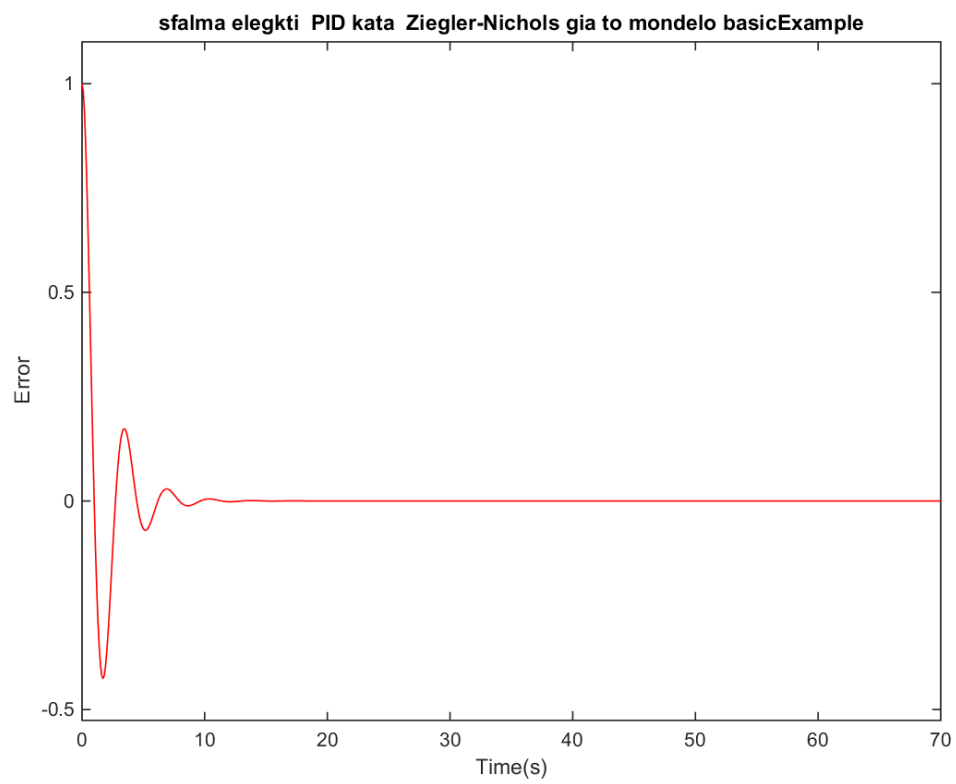
Παρακάτω παραθέτουμε τα διαγράμματα σφάλματος:



Εικόνα 2.4 Μέθοδος Ziegler & Nichols & σφάλμα ελεγκτής P



Εικόνα 2.5: Μέθοδος Ziegler&Nichols & σφάλμα ελεγκτής PI



Εικόνα 2.6: Μέθοδος Ziegler&Nichols & σφάλμα ελεγκτής PID



## 2.2 Tyreus- Luyben

Η μέθοδος Tyreus-Luyben ακολουθεί την ίδια διαδικασία με τη μέθοδο Ziegler-Nichol μέχρι το σημείο λήψης του τελικού κέρδους και της τελικής περιόδου. Αυτή η μέθοδος προτείνει μόνο ρυθμίσεις για τον ελεγκτή PI και PID. Η μέθοδος συντονισμού των Tyreus και Luyben εισήχθη το 1997 [6,7] και φέρει τροποποιήσεις στους τύπους για τις παραμέτρους του ελεγκτή ώστε να επιτευχθεί καλύτερη ευστάθεια στον βρόχο ελέγχου σε σύγκριση με τη μέθοδο του Ziegler-Nichols [8].

**Πίνακας 2.2 Ρυθμίσεις για τους ελεγκτές PI και PID σύμφωνα με τη μέθοδο Tyreus-Luyben**

ΤΥΠΟΣ ΕΛΕΓΚΤΗ	$K_p$	$T_i$	$T_D$
PI	$K_{\max}/3.2$	$2.2 T_o$	-
PID	$K_{\max}/2.2$	$2.2 T_o$	$T_o/6.3$

Η διαδικασία της μεθόδου Tyreus-Luyben όπως αναφέραμε παραπάνω περιλαμβάνει πιο συντηρητικές ρυθμίσεις (υψηλότερος συντελεστής απόσβεσης κλειστού βρόχου) έτσι είναι πιο κατάλληλη για εφαρμογές ελέγχου χημικών διεργασιών. Τα μειονεκτήματα της αφορούν στο ότι διακρίνεται από υποτονική αντιστάθμιση διαταραχών, μπορεί να μην είναι γρήγορη στη χρήση επειδή απαιτεί δοκιμή και σφάλμα και ότι είναι απαραίτητο ο χρήστης να πρέπει να βεβαιωθεί ότι το σήμα ελέγχου δεν φτάνει τις μέγιστες και ελάχιστες τιμές του κατά τη διάρκεια του πειράματος [9].

Η διαδικασία δοκιμής και σφάλματος είναι η εξής:

**Στάδιο 1.** Όταν το σύστημα φτάσει σε κατάσταση ισορροπίας (τουλάχιστον κατά προσέγγιση), απαλείφεται η ολοκληρωτική και διαφορική δράση.

**Στάδιο 2.** Το  $K_c$  γίνεται ίσο με μια μικρή τιμή (π.χ. 0.1) και τοποθετείται ο ελεγκτής στον αυτόματο τρόπο λειτουργίας.

**Στάδιο 3.** Εισάγεται μια μικρή στιγμιαία αλλαγή στην επιθυμητή τιμή έτσι ώστε η ελεγχόμενη μεταβλητή να απομακρυνθεί από την επιθυμητή τιμή. Αυξάνεται σταδιακά το  $K_{\max}$  μέχρις ότου πραγματοποιηθεί συνεχής ταλάντωση με σταθερό

πλάτος. Η τιμή του  $K_{max}$  που προκαλεί την συνεχής ταλάντωση ονομάζεται οριακή ενίσχυση. Η περίοδος της αντίστοιχης συνεχούς ταλάντωσης ονομάζεται ως οριακή περίοδος,  $P_u$ .

**Στάδιο 4.** Υπολογίζεται η βαθμονόμηση του ελεγκτή PID κάνοντας χρήση των σχέσεων βαθμονόμησης των Tyreus – Luyben (Πιν.2.3).

**Στάδιο 5.** Αξιολογούνται οι παράμετροι ελέγχου όπως ορίζονται από τους Tyreus και Luyben.

### 2.3 Υλοποίηση μεθόδου σε MATLAB

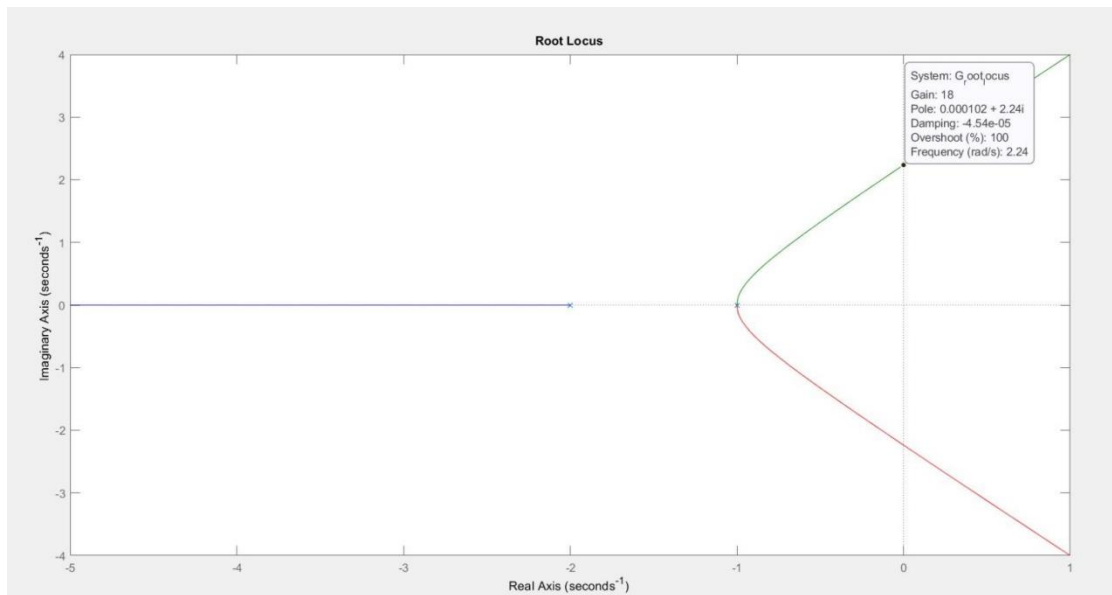
Με την χρήση του υπολογιστικού περιβάλλοντος Matlab υλοποιήσαμε την μέθοδο Tyreus – Luyben, αποτυπώνοντας προγραμματιστικά τις παραμέτρους και τις προϋποθέσεις που αναφέρουμε παραπάνω. Πιο συγκεκριμένα, η συνάρτηση μεταφοράς έχει την μορφή

$$G_p = 1/((s + 1) * (s + 1) * (s + 2)) \quad (2.3)$$

Στόχος μας είναι ο υπολογισμός του μέγιστου κέρδους  $K_{max}$ .

Οι υπολογισμοί μας πραγματοποιούνται για τους ελεγκτές P,PI , PID με τον εξής κώδικα και οι τιμές που προκύπτουν παρουσιάζονται παρακάτω: Ο κώδικας που εκτελείται έχει το όνομα TL.m.

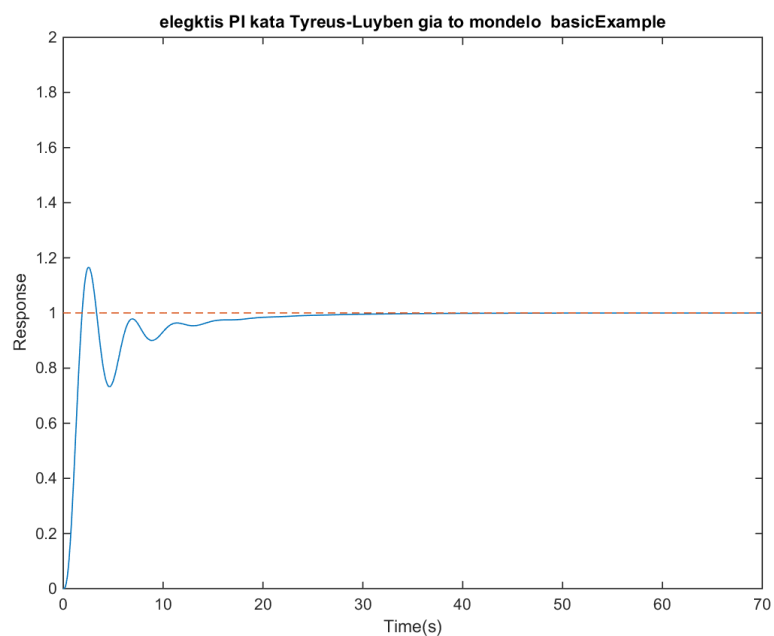
Μελετώντας την συνάρτηση μεταφοράς καταλήξαμε στα  **$K_{max} = 18.00$  &  $f_o = T_o = 2.8099$**  όπως φαίνεται στο παρακάτω διάγραμμα ταλαντώσεων στο οποίο παρατηρούμε το κρίσιμο σημείο με τον γεωμετρικό τόπο των ριζών. Το  $K_{max}$  ισούται με 18.00 και έπειτα κάνοντας την πράξη  $\frac{(2 * \pi)}{2.24}$  λαμβάνουμε και την τιμή 2.8099. Ο τύπος είναι ο εξής :  $T = \frac{(2 * \pi)}{\omega}$ .



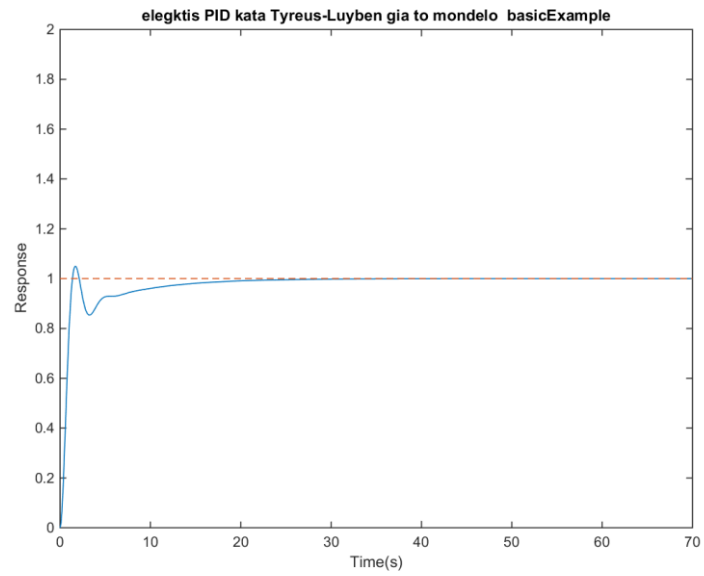
Πίνακας 2.3 Παράδειγμα Tyreus – Luyben

ΤΥΠΟΣ ΕΛΕΓΚΤΗ	$K_p$	$T_i$	$T_D$
PI	5.625	6.18178	-
PID	8.18	6.18178	0.446

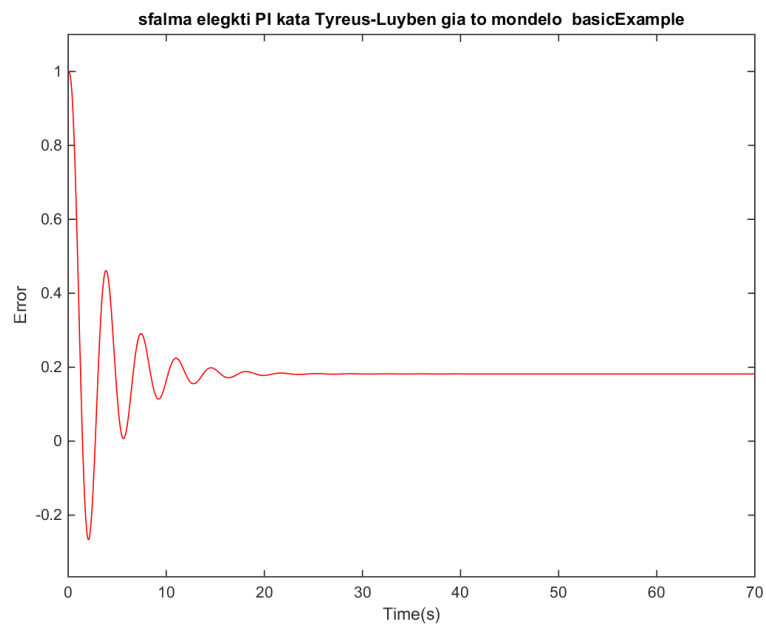
Επιπλέον, προκύπτουν τα παρακάτω διαγράμματα :



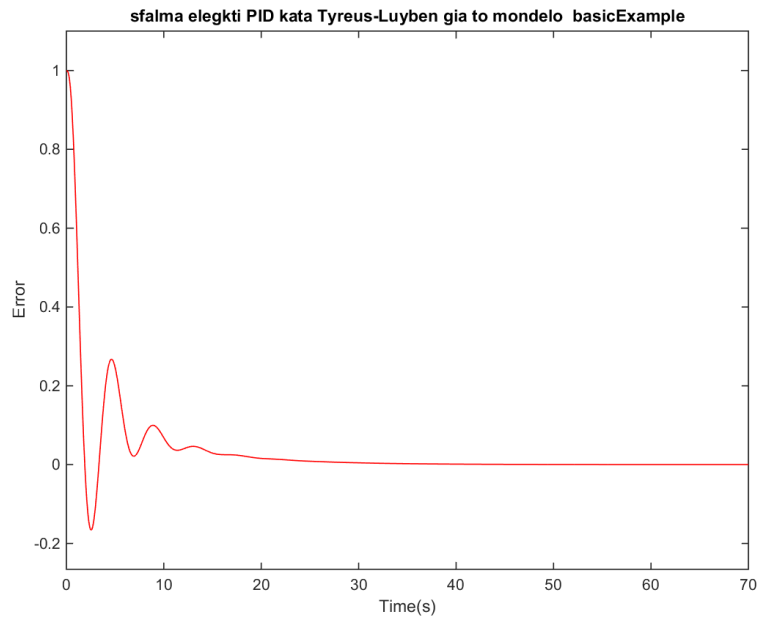
Εικόνα 2.7 Μέθοδος Tyreus - Luyben και Ελεγκτής PI



Εικόνα 2.8 Μέθοδος Tyreus - Λuybenκαι Ελεγκτής PID



Εικόνα 2.9 Tyreus–Λuyben& σφάλμα ελεγκτής PI



Εικόνα 2.10 Tyreus–Luyben & σφάλμα ελεγκτής PID

## 2.4 Cohen – Coon

Η μέθοδος συντονισμού ελεγκτή Cohen-Coon διορθώνει την αργή απόκριση σταθερής κατάστασης που δίνεται από τη μέθοδο Ziegler-Nichols όταν υπάρχει μεγάλος «νεκρός χρόνος» δηλαδή μεγάλη καθυστέρηση του συστήματος. Λόγω της καθυστερημένης απόκρισης του ελεγκτή στη διαταραχή, αυτή η προσέγγιση είναι κατάλληλη μόνο για μοντέλα με χρονική καθυστέρηση πρώτης τάξης. Δεδομένου ότι η είσοδος μπορεί να αλλάξει αφού φτάσει σε σταθερή κατάσταση, η προσέγγιση Cohen-Coon κατηγοριοποιείται ως τρόπος συντονισμού «offline». Η απόκριση μπορεί στη συνέχεια να χρησιμοποιηθεί για την αξιολόγηση των αρχικών παραμέτρων ελέγχου. Η έξοδος μπορεί στη συνέχεια να αξιολογηθεί ως προς τη χρονική σταθερά και τη χρονική καθυστέρηση[4].

### Πλεονεκτήματα Μεθόδου

- Χρησιμοποιείται για συστήματα με χρονικές καθυστερήσεις
- Ταχύτερος χρόνος απόκρισης κλειστού βρόχου.

### Μειονεκτήματα και Περιορισμοί Μεθόδου

- Ασταθή συστήματα κλειστού βρόχου.

- Μπορεί να χρησιμοποιηθεί μόνο για μοντέλα πρώτης τάξης που χαρακτηρίζονται από μεγάλες καθυστερήσεις διεργασιών.
- Μέθοδος «offline».

Οι υπολογισμοί της Cohen – Coonπραγματοποιούνται ως εξής[10]:

**Βήμα 1:** Εκτελείται μια δοκιμή βήματος για να ληφθούν οι παράμετροι ενός μοντέλου πρώτης τάξης συν την χρονική καθυστέρηση

- i. Ελέγχουμε ότι η διαδικασία βρίσκεται σε αρχική κατάσταση.
- ii. Εισαγάγουμε μια αλλαγή βήματος στη μεταβλητή που έχει μεταβληθεί.
- iii. Περιμένουμε έως ότου η διαδικασία βρεθεί σε μια νέα σταθερή κατάσταση.

Βήμα 2: Υπολογίζουμε τις παραμέτρους του συστήματος και χρησιμοποιούμε τις προδιαγεγραμμένες τιμές που διέπουν την μέθοδο όπως παρουσιάζονται στον πίνακα 2.5.

Οι τύποι για τον υπολογισμό των παραμέτρων είναι οι εξής[11]:

$$t_1 = \frac{t_2 - (\ln(2))t_3}{1 - \ln(2)} \quad (2.4)$$

$$\tau = t_3 - t_1 \quad (2.5)$$

$$\tau_{del} = t_1 - t_0 \quad (2.6)$$

$$K = \frac{B}{A} \quad (2.7)$$

$$r = \frac{\tau_{del}}{\tau} \quad (2.8)$$

Οι παράμετροι ορίζονται ως εξής:

$t_1$ : χρόνος στο βήμα εισόδου σημείο έναρξης

$t_2$  :χρόνος όταν φτάνει στο 50%

$t_3$  : χρόνος όταν φτάνει το 63,2% σημείο

$\tau$  : ο χρόνος που απαιτείται για τη μέτρηση του σφάλματος.

$\tau_{del}$ : καθυστέρηση μεταφοράς ή νεκρός χρόνος, δηλαδή ο χρόνος από τη στιγμή που εισήχθη η διαταραχή έως το πρώτο σημάδι αλλαγής στο σήμα εξόδου

$r$  : ποσοστό διαταραχής

$K$  : η τελική τιμή που φτάνει η απόκριση σε σταθερή κατάσταση

$A$  :Η είσοδος του πλάτους της διαταραχής

$B$  : δεύτερη απόκριση που έχει ταλάντωση 1/4 του πλάτους της πρώτης ταλάντωσης

**Πίνακας 2.4** Ρυθμίσεις για τα κέρδη P, I και D σύμφωνα με τη μέθοδο Cohen - Coon

	$K_p$	$T_I$	$T_D$
P controller	$\frac{1}{rK} \left(1 + \frac{r}{3}\right)$	0	0
PI controller	$\frac{1}{rK} \left(0.9 + \frac{r}{12}\right)$	$\tau_{del} \frac{30 + 3r}{9 + 20r}$	0
PID controller	$\frac{1}{rK} \left(\frac{4}{3} + \frac{r}{4}\right)$	$\tau_{del} \frac{32 + 3r}{13 + 8r}$	$\tau_{del} \frac{4}{11 + 2r}$

#### 2.4.1 Υλοποίηση μεθόδου σε MATLAB

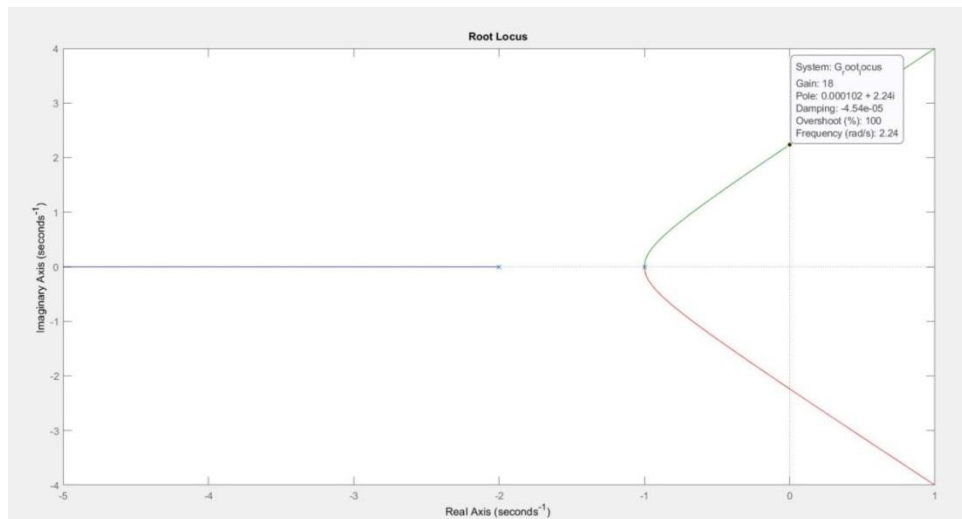
Με την χρήση του υπολογιστικού περιβάλλοντος Matlab υλοποιήσαμε την μέθοδο Cohen - Coon, αποτυπώνοντας προγραμματιστικά τις παραμέτρους και τις προϋποθέσεις που αναφέρουμε παραπάνω.

Πιο συγκεκριμένα, η συνάρτηση μεταφοράς έχει την μορφή[12]:

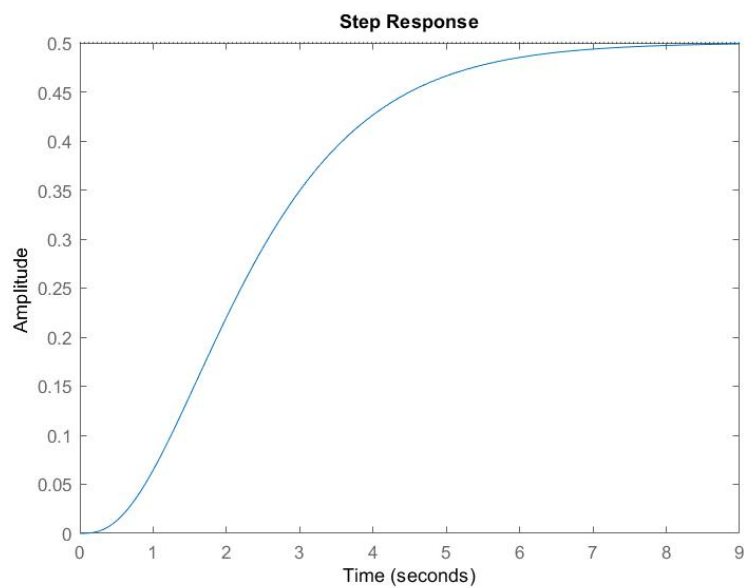
$$G_p = 1 / ((s+1) * (s+1) * (s+2)) \quad (2.9)$$

Μελετώντας την συνάρτηση μεταφοράς καταλήξαμε στα  $K_{max} = 18.00$  &  $f_o = T_o = 2.8099$  όπως φαίνεται στο παρακάτω διάγραμμα ταλαντώσεων στο οποίο παρατηρούμε το κρίσιμο σημείο με τον γεωμετρικό τόπο των ριζών. Το  $K_{max}$  ισούται με 18.00 και έπειτα κάνοντας την πράξη  $((2 * p_i)) / 2.24$  λαμβάνουμε και την τιμή 2.8099.

Ο τύπος είναι ο εξής :  $T = \frac{(2*\pi)}{\omega}$ .



Η εφαρμογή της αλλαγής βήματος φαίνεται στο παρακάτω διάγραμμα και παρουσιάζονται τα αποτελέσματα που λαμβάνουν οι μεταβλητές εξαιτίας αυτής.



$$t = 1.6509$$

$$t_1 = 1.0201$$

$$t_3 = 2.6710$$

$$t_2 = 2.1644$$

$$t_{del} = 1.0201$$



$$k = 0.4982$$

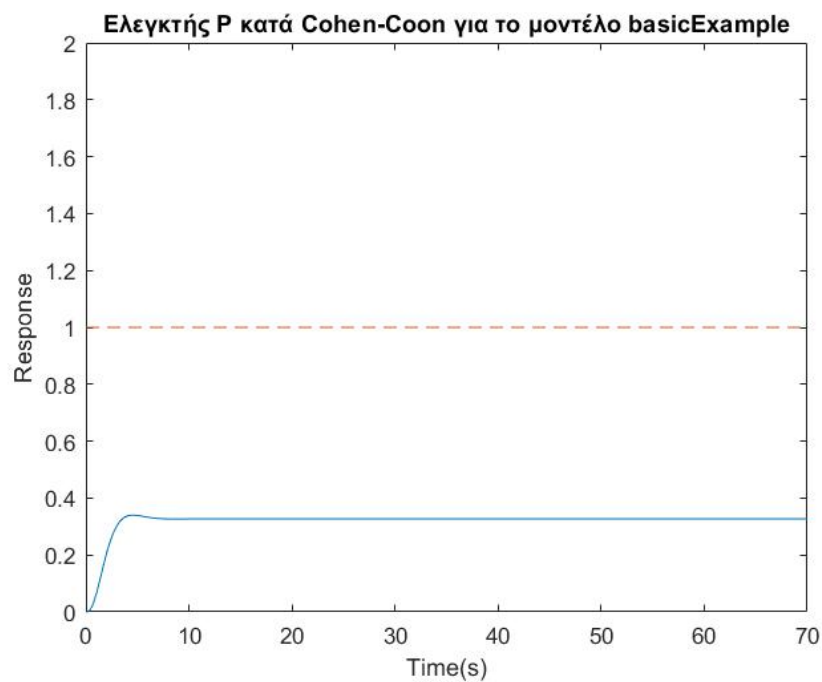
$$r = 0.6180$$

Οι υπολογισμοί μας πραγματοποιούνται για τους ελεγκτές P,PI , PID με τον εξής κώδικα και οι τιμές που προκύπτουν παρουσιάζονται παρακάτω. Ο κώδικας ονομάζεται CC.m

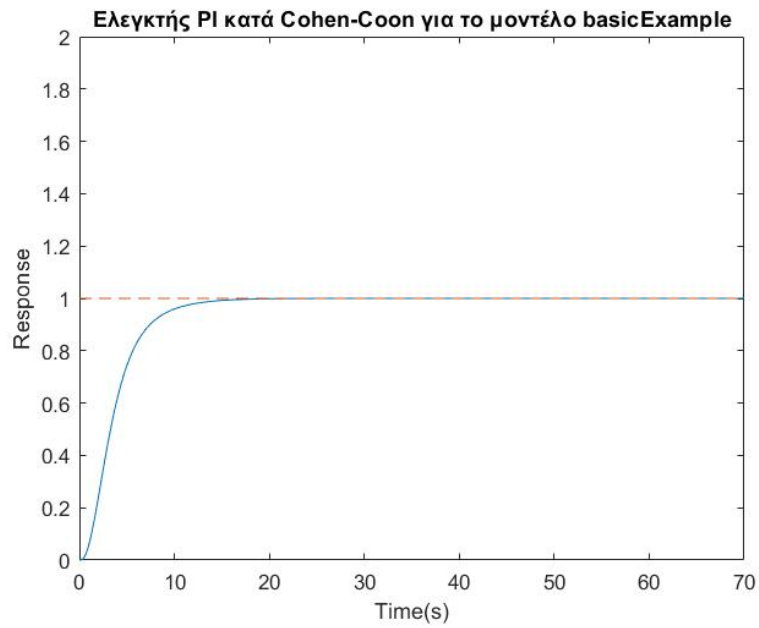
**Πίνακας 2.5 Παράδειγμα Cohen Coon**

ΕΛΕΓΚΤΗΣ	$K_p$	$T_I$	$T_D$
P	0.9723	0	0
PI	0.7672	1.5214	0
PID	1.1996	2.0301	0.3335

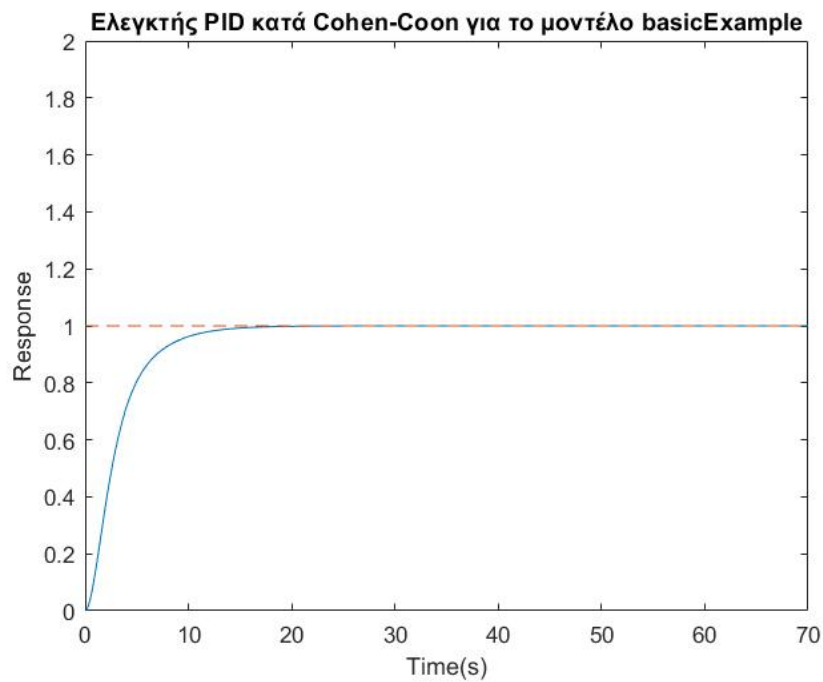
Επιπλέον προκύπτουν τα παρακάτω διαγράμματα,



**Εικόνα 2.11 Ελεγκτής P κατά Cohen -Coon**



Εικόνα 2.12 Ελεγκτής PI κατά Cohen -Coon

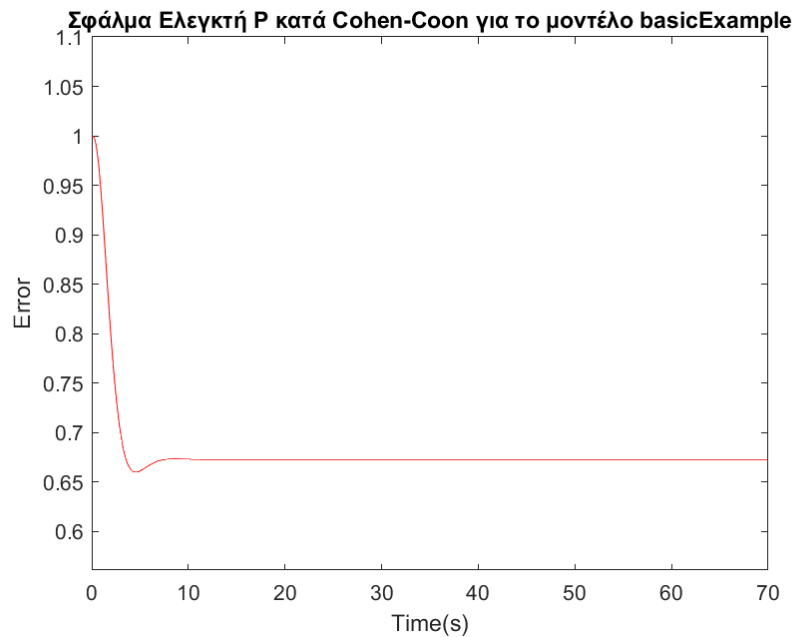


Εικόνα 2.13 Ελεγκτής PID κατά Cohen–Coon

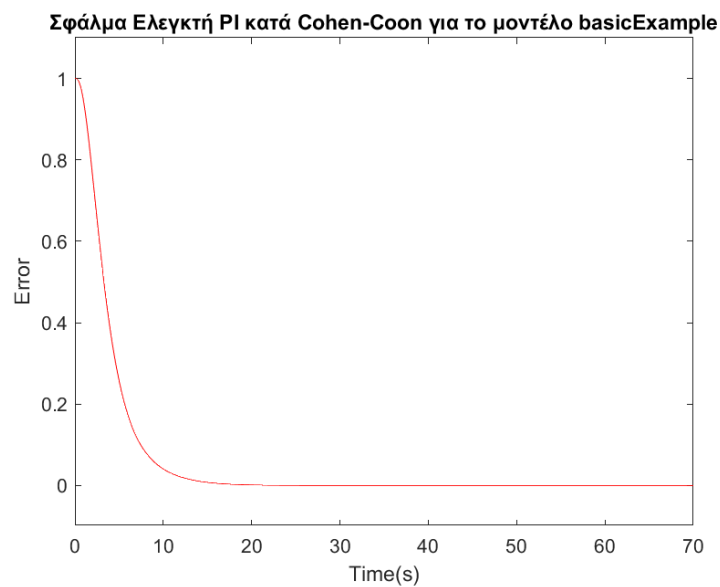
Όπως βλέπουμε από τις εικόνες 2.1-2.3, ο P ελεγκτής δεν καταφέρνει να φτάσει το σύστημα στην τελική κατάσταση, έχοντας ένα σφάλμα. Από την άλλη, οι ελεγκτές PI και PID οδηγούν επιτυχώς το σύστημα προς την επιθυμητή τελική κατάσταση, ενώ μάλιστα ο ελεγκτής με ολοκληρωτικό παράγοντα (PID) το επιτυγχάνει αυτό στο

περίπου  $1/3$  του χρόνου που χρειάζεται ο PI, καθώς χάρη στον διαφορικό όρο αποφεύγει τις υπερβολικές ταλαντώσεις γύρω από τη τελική κατάσταση που εμφανίζει το σύστημα με τον PI ελεγκτή.

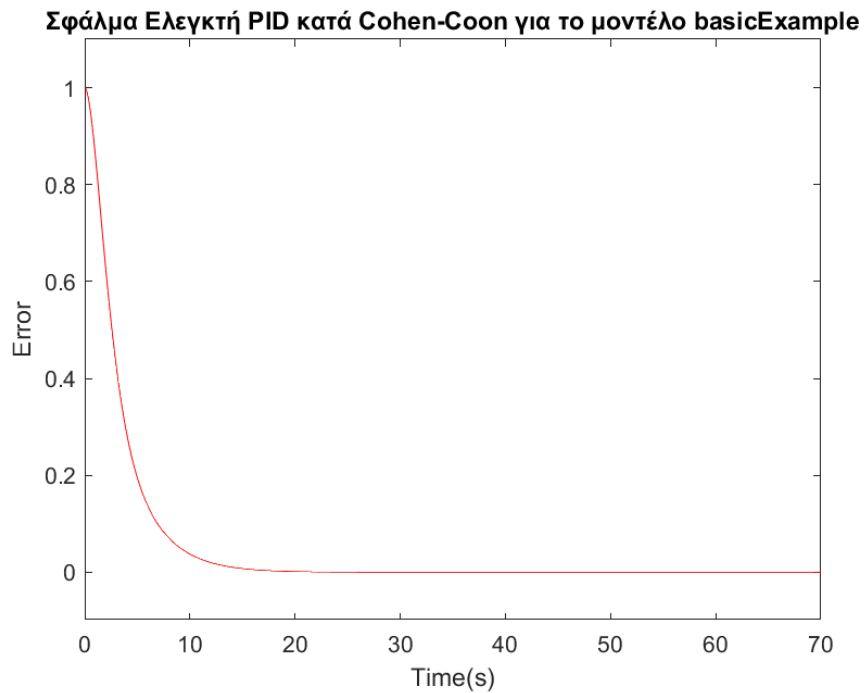
Παρακάτω παραθέτουμε τα διαγράμματα σφάλματος.



Εικόνα 2.14 CohenCoon& σφάλμα ελεγκτής P



Εικόνα 2.15 CohenCoon& σφάλμα ελεγκτής PI



Εικόνα 2.16 CohenCoon& σφάλμα ελεγκτής PI

Συμπερασματικά τα errors ανά ελεγκτή για όλες τις μεθόδους για το basicExample είναι τα εξής:

Μέθοδος	P	PI	PID
Ziegler – Nichols	90.667	41.422	14.134
Cohen-Coon	309.13	40.163	34.347
Tyreus - Luyben	-	25.77	16.096

#### Ελεγκτής P:

- Ziegler-Nichols: Αυτή η μέθοδος έχει το μικρότερο σφάλμα για τον ελεγκτή P σε σύγκριση με τις άλλες μεθόδους.
- Cohen-Coon: Το σφάλμα για τον ελεγκτή P είναι σημαντικά υψηλότερο σε σύγκριση με το Ziegler-Nichols.

- Tyreus-Luyben: Δεν παρέχονται αρκετά στοιχεία για την αξιολόγηση της απόδοσής του.

Με βάση τις διαθέσιμες πληροφορίες, η μέθοδος Ziegler-Nichols φαίνεται να είναι καλύτερη για τον ελεγκτή P.

#### Ελεγκτής PI

- Ziegler-Nichols: Το σφάλμα για τον ελεγκτή PI είναι μικρότερο σε σύγκριση με το Cohen-Coon.
- Cohen-Coon: Το σφάλμα για τον ελεγκτή PI είναι ελαφρώς υψηλότερο σε σύγκριση με το Ziegler-Nichols.
- Tyreus-Luyben: Το σφάλμα για τον ελεγκτή PI είναι μικρότερο σε σύγκριση με το Cohen-Coon.

Με βάση τα δεδομένα που δίνονται, η μέθοδος Tyreus-Luyben φαίνεται να είναι καλύτερη για τον ελεγκτή PI.

#### Ελεγκτής PID

- Ziegler-Nichols: Το σφάλμα για τον ελεγκτή PID είναι το μικρότερο από όλες τις μεθόδους.
- Cohen-Coon: Το σφάλμα για τον ελεγκτή PID είναι υψηλότερο σε σύγκριση με το Ziegler-Nichols.
- Tyreus-Luyben: Το σφάλμα για τον ελεγκτή PID είναι μικρότερο σε σύγκριση με το Cohen-Coon.

Με βάση τα δεδομένα που παρέχονται, η μέθοδος Ziegler-Nichols φαίνεται να είναι καλύτερη για τον ελεγκτή PID.

## Κεφάλαιο 3

### Ανοικτός Βρόγχος – Φούρνος υψηλών θερμοκρασιών

Σε αυτό το κεφάλαιο, θα συζητήσουμε τη σχεδίαση και την ανάλυση ενός συστήματος αρνητικής ανάδρασης κλειστού βρόχου για έλεγχο θερμοκρασίας σε φούρνο υψηλής θερμοκρασίας. Θα διερευνήσουμε τρεις διαφορετικές μεθόδους για τον συντονισμό των παραμέτρων του συστήματος, δηλαδή τις μεθόδους ελέγχου Proportional (P), Proportional-Integral (PI) και Proportional-Integral-Derivative (PID). Κάθε μέθοδος θα αξιολογηθεί αναλύοντας τον γεωμετρικό τόπο των ριζών και παρουσιάζοντας τις παραμέτρους που προκύπτουν σε πίνακες.

Το πρόβλημα αφορά ένα σύστημα ελέγχου θερμοκρασίας που να μπορεί να ρυθμίζει τη θερμοκρασία μέσα σε έναν φούρνο υψηλής θερμοκρασίας. Το σύστημα θα πρέπει να μπορεί να διατηρεί τη θερμοκρασία σε ένα επιθυμητό σημείο ρύθμισης, ενώ παράλληλα αντισταθμίζει τις διαταραχές και τους εξωτερικούς παράγοντες που μπορεί να επηρεάσουν τη θερμοκρασία του φούρνου. Στόχος μας είναι να επιτύχουμε ακριβή και σταθερό έλεγχο θερμοκρασίας μέσω ενός συστήματος αρνητικής ανάδρασης κλειστού βρόχου.

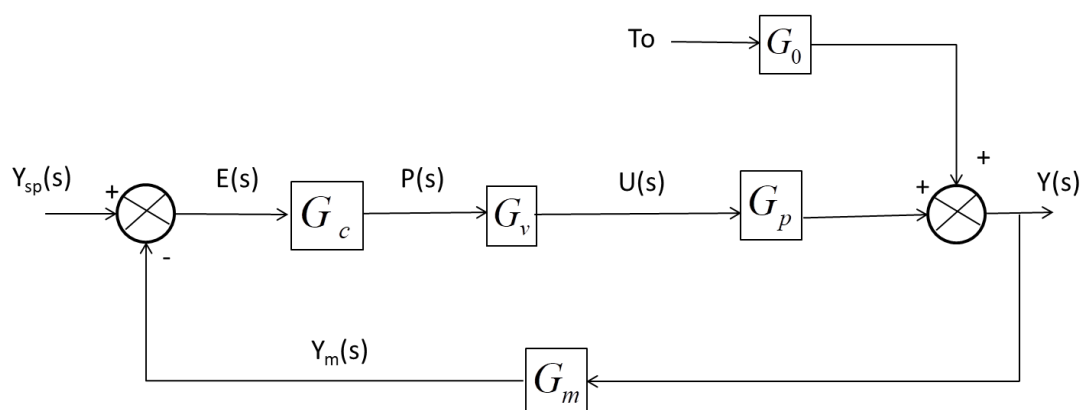
Οι συναρτήσεις μεταφοράς που περιγράφουν τη δυναμική του συστήματος είναι οι εξής:

$G_m(s)$ : Συνάρτηση μεταφοράς από τον μηχανισμό θέρμανσης του φούρνου στη μετρούμενη θερμοκρασία.

$G_v(s)$ : Συνάρτηση μεταφοράς από την είσοδο ελέγχου στον μηχανισμό θέρμανσης του φούρνου.

Η συνάρτηση μεταφοράς  $G_m(s)$  αντιπροσωπεύει τη δυναμική του αισθητήρα του φούρνου, το  $G_v(s)$  αντιπροσωπεύει τη δυναμική του ενεργοποιητή και το  $G_d(s)$  καταγράφει την επίδραση των διαταραχών στη μετρούμενη θερμοκρασία.

Για να αναλύσουμε τη ευστάθεια και την απόδοση του συστήματος, μπορούμε να εξετάσουμε τον γεωμετρικό τόπο των ριζών για τη συνάρτηση μεταφοράς κλειστού βρόχου  $T_r(s)$ . Μεταβάλλοντας το αναλογικό κέρδος, μπορούμε να παρατηρήσουμε την κίνηση των ριζών στο μιγαδικό επίπεδο. Με βάση τον τόπο των ριζών, μπορούμε να προσδιορίσουμε κατάλληλες τιμές για το  $K_r$  που εξασφαλίζουν ευστάθεια και επιθυμητή απόδοση. Επιπλέον, μπορούμε να συντονίσουμε τον ελεγκτή P επιλέγοντας κατάλληλες τιμές για το αναλογικό κέρδος. Αυτό μπορεί να γίνει παρατηρώντας την απόκριση του συστήματος σε εισόδους βήματος ή ράμπας και ρυθμίζοντας το  $K_r$  μέχρι να επιτευχθεί η επιθυμητή απόκριση. Οι παράμετροι που προκύπτουν για τον ελεγκτή P μπορούν να παρουσιαστούν σε έναν πίνακα, συνοψίζοντας την επιλεγμένη τιμή  $K_r$  και τις αντίστοιχες μετρήσεις απόδοσης συστήματος.



**Εικόνα 3.1 Διάγραμμα διαδικασίας**

*Συναρτήσεις μεταφοράς που σχετίζονται με το σύστημα ανοιχτού βρόχου*

<b>Μετρητικό Στοιχείο</b>	$G_m = \frac{1}{s+1}$
<b>Φούρνος Υψηλών Θερμοκρασιών</b>	$G_p = \frac{1}{2s+1}$
<b>Τελικό Στοιχείο Ενεργοποίησης</b>	$G_v = \frac{1}{3s+1}$

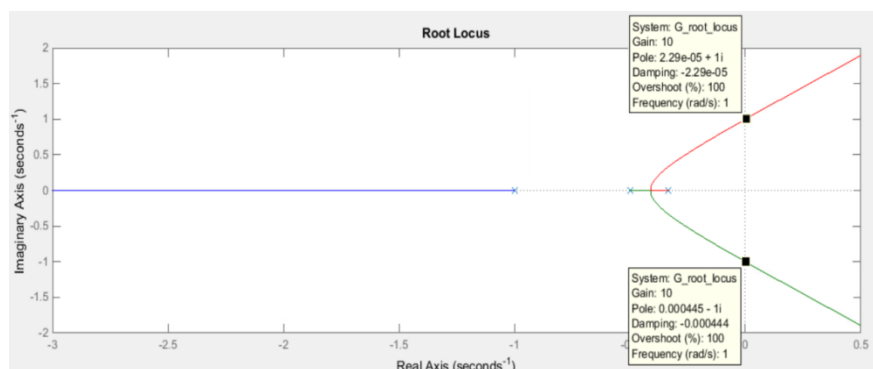
Η συνάρτηση μεταφοράς που θα μελετήσουμε είναι η εξής:

$$G_{ClosedLoop} = g_p * G_m * G_v * G_c$$

$$G_{ClosedLoop} = \frac{1}{2s + 1} * \frac{1}{s + 1} * \frac{1}{3s + 1}$$

### 3.2 Υλοποίηση μεθόδου σε MATLAB

Με την χρήση του υπολογιστικού περιβάλλοντος Matlab υλοποιήσαμε το παραπάνω παράδειγμα. Πρωταρχικά υπολογίσαμε τον γεωμετρικό τόπο των πόλων. Βάση της γραφικής αυτής απεικόνισης και στο σημείο όπου τέμνεται ο άξονας των πραγματικών αριθμών μπορεί να βρεθεί η τιμή της ενίσχυσης  $K_{cr}$  για την οποία το σύστημα παρουσιάζει οριακή ευστάθεια.



Εικόνα 3.2 Γεωμετρικός τόπος ριζών

Παρουσιάζουμε με πίνακες τα αποτελέσματα για κάθε υπό μελέτη μέθοδο ως εξής:

#### Ziegler – Nichols

Οι τιμές των παραμέτρων είναι οι εξής :

$$K_{max} = 10$$

$$T_o = 6.2832$$



Πίνακας 3.1 Συντονισμός παραμέτρων Ziegler – Nichols

ΕΛΕΓΚΤΗΣ	$K_c$	$T_I$	$T_D$
P	5	0	0
PI	4.5	5.2360	0
PID	6	3.1416	0.7854

#### Tyreus-Luyben

Οι παράμετροι λαμβάνουν τις παρακάτω τιμές:

$K_{max} = 10$

$T_o = 6.2832$

Πίνακας 3.2 Συντονισμός παραμέτρων Tyreus – Luyben

	$K_p$	$T_i$	$T_D$
PIcontroller	3.125	13.8230	-
PID controller	4.5455	13.8230	0.9973

#### CohenCoon

Οι παράμετροι λαμβάνουν τις παρακάτω τιμές:

$t_1 = 2.4531$

$t_2 = 5.1578$

$t_3 = 6.3551$

$t = 3.9020$

$t_{Del} = 2.4531$

$r = 0.6287$

$K = 0.9973$

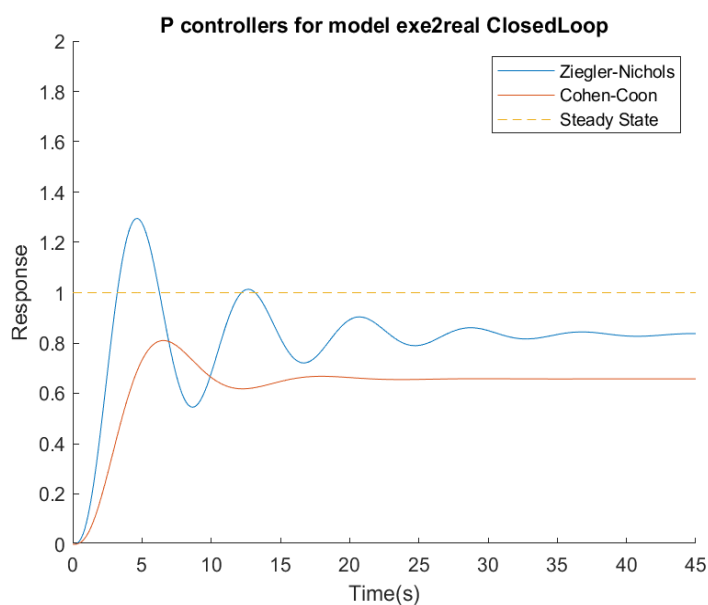
$A = 1$

B =0.9973

Πίνακας 3.3 Συντονισμός παραμέτρων Cohen Coon

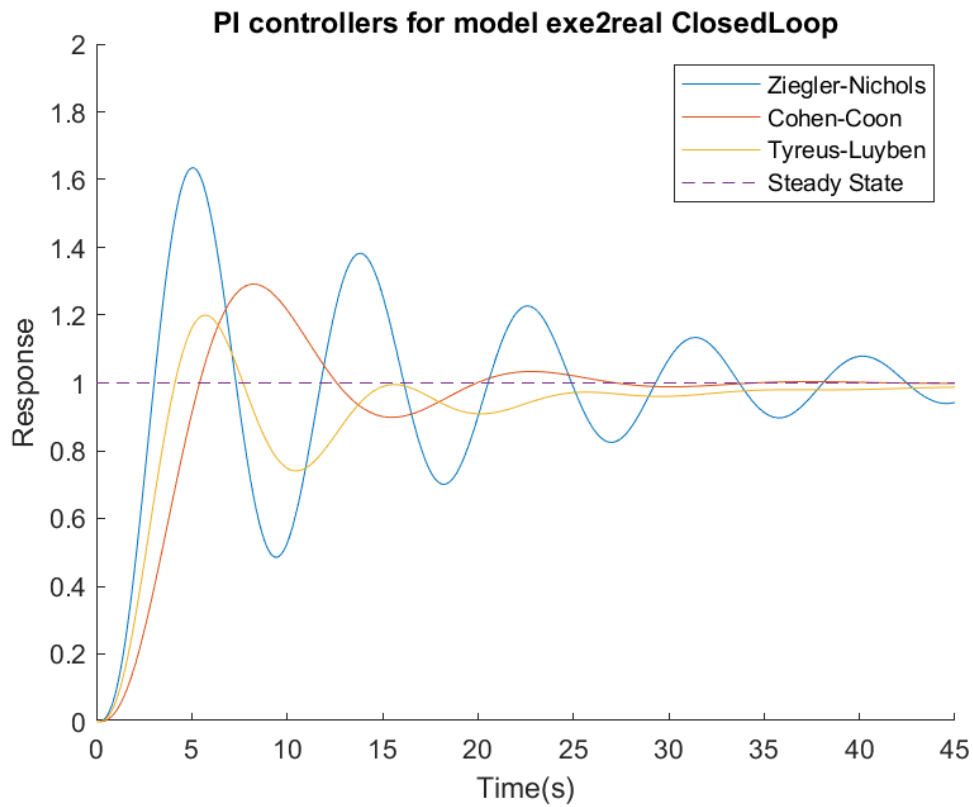
ΕΛΕΓΚΤΗΣ	$K_C$	$T_I$	$T_D$
P	1.9188	0	0
PI	1.5108	3.6257	
PID	2.3645	4.8672	0.8005

Επομένως προκύπτουν τα παρακάτω διαγράμματα απόκρισης :



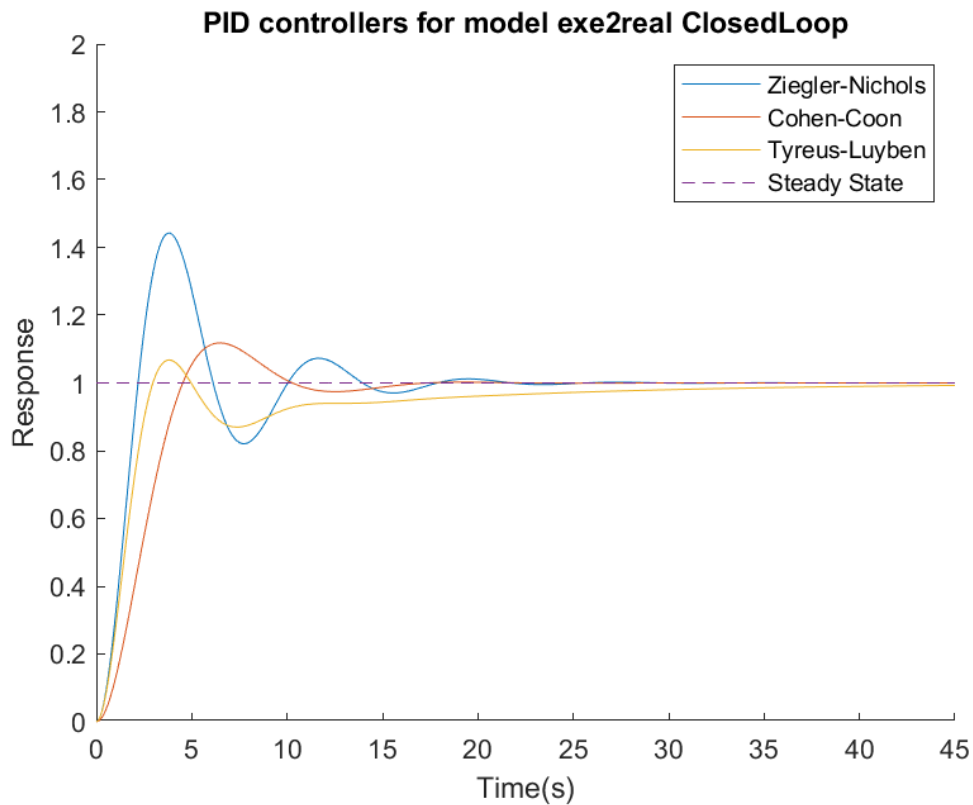
Εικόνα 3 Εφαρμογή ελεγκτή P και απόκριση σήματος εξόδου

Όπως βλέπουμε στην παραπάνω εικόνα, για τον P ελεγκτή δεν μπορεί να πραγματοποιηθεί η Tyreus-Luyben, ενώ η Cohen-Coon δεν πλησιάζει την επιθυμητή κατάσταση, σε αντίθεση με την Ziegler-Nichols που προσεγγίζει περισσότερο την τιμή ένα (1) συγκριτικά με τις άλλες μεθόδους.



**Εικόνα 4** Εφαρμογή ελεγκτή PI και απόκριση σήματος εξόδου

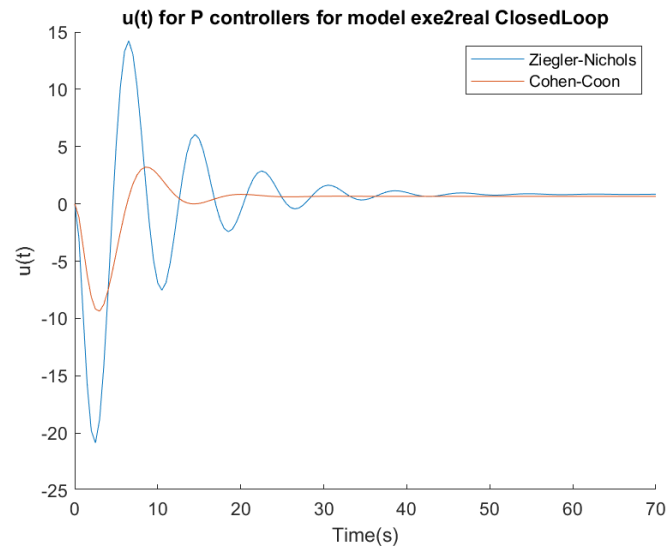
Όπως βλέπουμε στην παραπάνω εικόνα, για τον PI ελεγκτή και οι τρεις μέθοδοι προσεγγίζουν την επιθυμητή κατάσταση, με την Tyreus-Luyben να έχει μικρότερη ταλάντωση συγκριτικά με τις άλλες δύο μεθόδους.



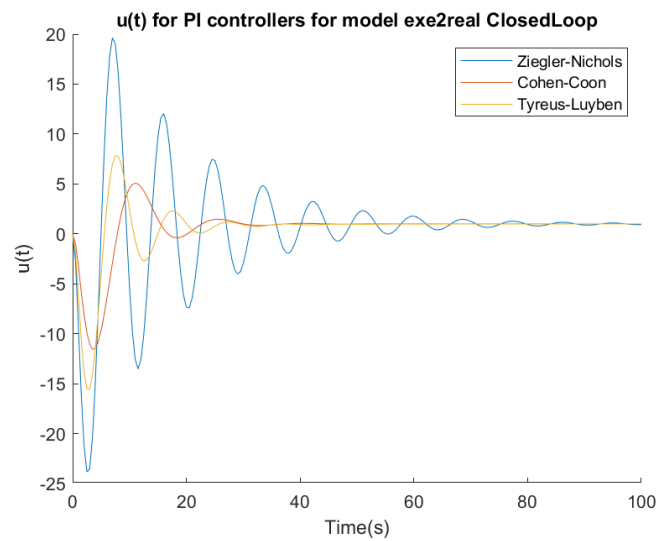
**Εικόνα 5 Εφαρμογή ελεγκτή PID και απόκριση σήματος εξόδου**

Όπως βλέπουμε στην παραπάνω εικόνα, για τον PID ελεγκτή και οι τρεις μέθοδοι προσεγγίζουν την επιθυμητή κατάσταση, με την CohenCoon να έχει μικρότερη ταλάντωση συγκριτικά με τις άλλες δύο μεθόδους.

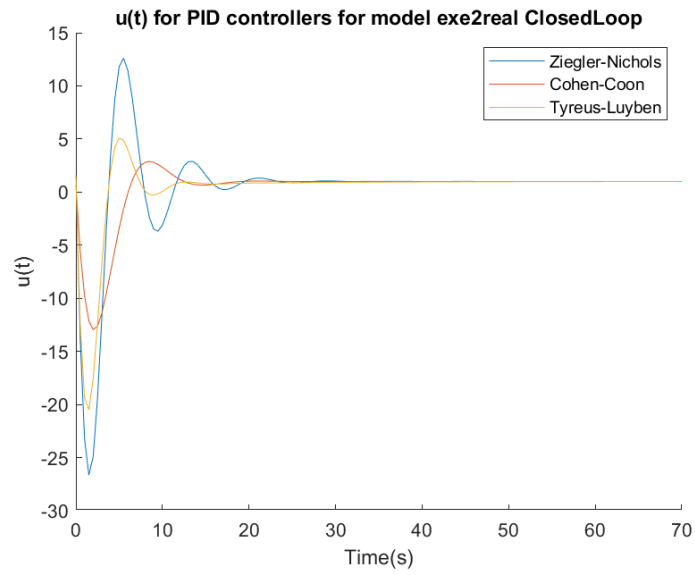
Παρακάτω παραθέτουμε τα διαγράμματα απόκρισης του σήματος  $u(t)$  :



Εικόνα 6: Διάγραμμα αποκρίσης σήματος  $u(t)$  για P ελεγκτές

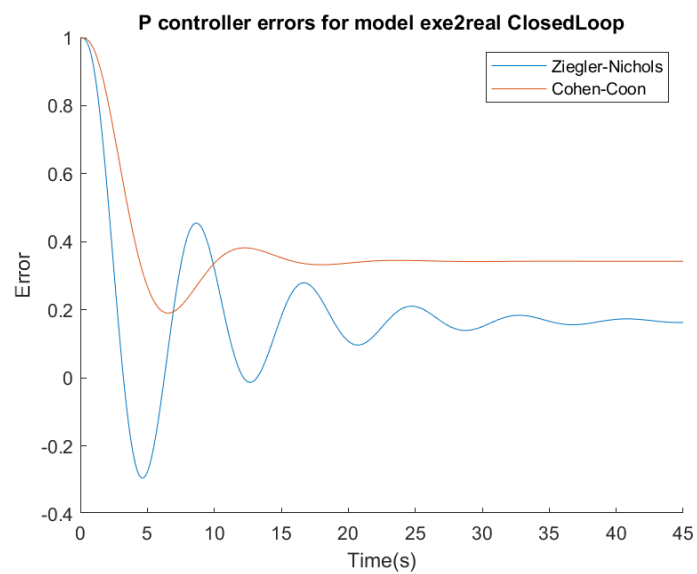


Εικόνα 7: Διάγραμμα αποκρίσης σήματος  $u(t)$  για PI ελεγκτές

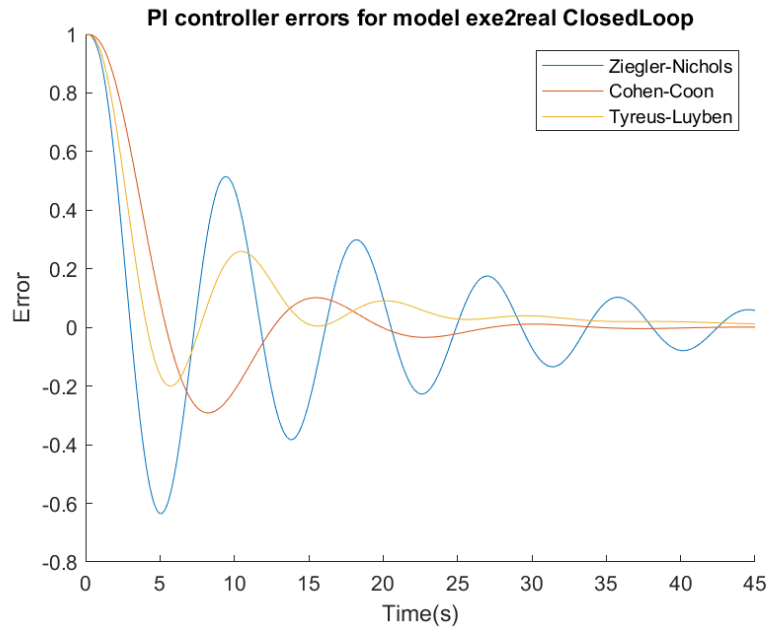


Εικόνα 8: Διάγραμμα αποκρίσης σήματος  $u(t)$  για PID ελεγκτές

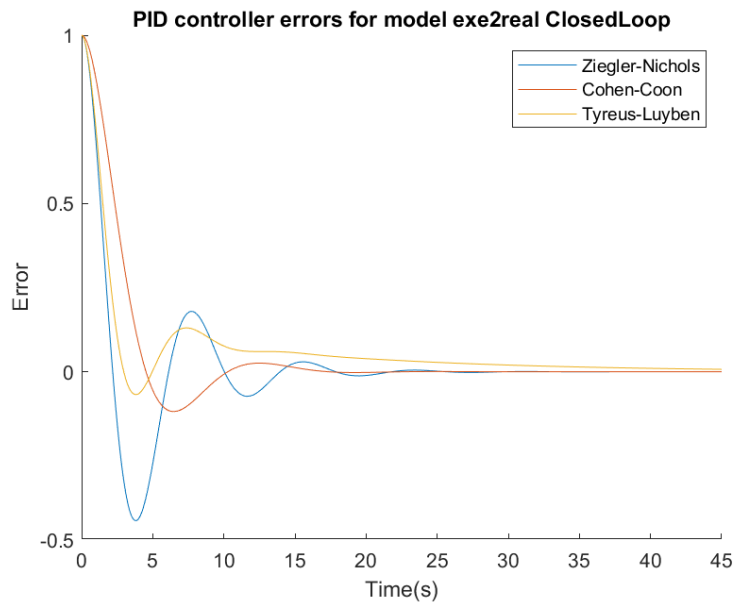
Παρακάτω παραθέτουμε τα διαγράμματα σφάλματος.



Εικόνα 9 Εφαρμογή ελεγκτή P και σφάλμα



**Εικόνα 3.10 Εφαρμογή ελεγκτή PI και σφάλμα**



**Εικόνα 3.11 Εφαρμογή ελεγκτή PID και σφάλμα**

Συμπερασματικά τα errors ανά ελεγκτή για όλες τις μεθόδους για το παράδειγμα OpenLoop είναι τα εξής:

Μέθοδος	P	PI	PID
Ziegler – Nichols	95.777	92.741	31.674
Cohen-Coon	168.36	53.791	29.304
Tyreus - Luyben	-	51.725	31.597

#### Ελεγκτής P:

- Ziegler-Nichols: Το σφάλμα για τον ελεγκτή P είναι το χαμηλότερο μεταξύ όλων των μεθόδων.
- Cohen-Coon: Το σφάλμα για τον ελεγκτή P είναι ψηλότερο σε σύγκριση με το Ziegler-Nichols.
- Tyreus-Luyben: Δεν βαθμονομεί P ελεγκτές .

Με βάση τις διαθέσιμες πληροφορίες, η μέθοδος Ziegler-Nichols φαίνεται να είναι καλύτερη για τον ελεγκτή P.

#### Ελεγκτής PI:

- Ziegler-Nichols: Το σφάλμα για τον ελεγκτή PI είναι το υψηλότερο μεταξύ όλων των μεθόδων.
- Cohen-Coon: Το σφάλμα για τον ελεγκτή PI είναι χαμηλότερο σε σύγκριση με το Ziegler-Nichols.
- Tyreus-Luyben: Το σφάλμα για τον ελεγκτή PI είναι χαμηλότερο σε σύγκριση με το Ziegler-Nichols και με την Cohen-Coon .

Με βάση τα δεδομένα που δίνονται, η μέθοδος Tyreus-Luyben φαίνεται να είναι καλύτερη για τον ελεγκτή PI.

#### Ελεγκτής PID:

- Ziegler-Nichols: Το σφάλμα για τον ελεγκτή PID είναι ελαφρώς ψηλότερο από την Tyreus-Luyben .
- Cohen-Coon: Το σφάλμα για τον ελεγκτή PID είναι χαμηλότερο από όλες τις μεθόδους .
- Tyreus-Luyben: Το σφάλμα για τον ελεγκτή PID είναι ελαφρώς χαμηλότερο σε σύγκριση με το Ziegler-Nichols.



Με βάση τα δεδομένα που παρέχονται, η μέθοδος Cohen-Coon φαίνεται να είναι καλύτερη για τον ελεγκτή PID.

## Συμπεράσματα

Η βέλτιστη μέθοδος συντονισμού μπορεί να εξαρτάται από τη συγκεκριμένη εφαρμογή και τις απαιτήσεις του συστήματος που ελέγχεται. Κάθε μία από τις μεθόδους έχει τα δικά της πλεονεκτήματα και μειονεκτήματα και η επιλογή της μεθόδου θα πρέπει να βασίζεται στην προσεκτική εξέταση των χαρακτηριστικών του συστήματος και της επιθυμητής απόδοσης ελέγχου.

Η μεθοδολογία Ziegler-Nichols είναι μια ευρέως χρησιμοποιούμενη και δημοφιλής μέθοδος συντονισμού, αλλά μπορεί να μην παρέχει πάντα τη βέλτιστη απόδοση για όλα τα συστήματα. Συγκεκριμένα, μπορεί να μην είναι κατάλληλη για συστήματα με εξαιρετικά μη γραμμική ή χρονικά μεταβαλλόμενη δυναμική. Οι μέθοδοι Cohen-Coon και Tyreus-Luyben βασίζονται περισσότερο σε μοντέλα που μπορεί να απαιτούν περισσότερες πληροφορίες σχετικά με το σύστημα που ελέγχεται, αλλά μπορεί να παρέχουν καλύτερη απόδοση σε ορισμένες καταστάσεις, ιδιαίτερα για συστήματα με πολύπλοκη ή μη γραμμική δυναμική.

Μπορεί να χρειαστεί να πειραματιστεί κανείς με διαφορετικές μεθόδους συντονισμού και παραμέτρους για να προσδιορίσει την καλύτερη προσέγγιση για ένα δεδομένο σύστημα. Η προσομοίωση και ο πειραματισμός μπορούν να αποτελέσουν χρήσιμα εργαλεία για τη σύγκριση της απόδοσης διαφορετικών μεθόδων και τη λεπτομερή ρύθμιση των παραμέτρων για την επίτευξη της επιθυμητής απόδοσης ελέγχου. Η παρούσα μελέτη που συγκρίνει αυτές τις διαφορετικές μεθόδους συντονισμού μπορεί να προσφέρει πολύτιμες πληροφορίες για τα δυνατά και αδύνατα σημεία κάθε προσέγγισης και να βοηθήσει στην επιλογή της καλύτερης μεθόδου για μια συγκεκριμένη εφαρμογή.

Εκτός από αυτές τις τρεις μεθόδους, υπάρχουν πολλές άλλες μέθοδοι και παραλλαγές που μπορεί να είναι πιο κατάλληλες για ορισμένους τύπους συστημάτων

ή στόχους ελέγχου. Είναι σημαντικό να παραμένουμε ενημερωμένοι με την πιο πρόσφατη έρευνα σε αυτόν τον τομέα και να λάβετε υπόψη

Το κείμενο που παρουσιάζεται παρέχει πολύτιμες γνώσεις σχετικά με την εφαρμογή και σύγκριση μεθόδων συντονισμού για τις παραμέτρους ενός ελεγκτή PID στο πλαίσιο του ελέγχου θερμοκρασίας σε έναν κλίβανο υψηλής θερμοκρασίας. Αναλύοντας τρεις διαφορετικές μεθόδους - Ziegler-Nichols, Tyreus-Luyben και Cohen-Coon - το κείμενο υπογραμμίζει την απόδοση και την καταλληλότητά τους για την επίτευξη ακριβούς και σταθερού ελέγχου θερμοκρασίας.

Μετά την αξιολόγηση των αποτελεσμάτων, γίνεται φανερό ότι η μέθοδος Ziegler-Nichols δείχνει πολλά υποσχόμενη απόδοση για τον ελεγκτή P, καθώς δείχνει εγγύτητα στην επιθυμητή κατάσταση σε σύγκριση με τις άλλες μεθόδους. Προχωρώντας στον ελεγκτή PI, και οι δύο μέθοδοι Tyreus-Luyben και Cohen-Coon παρουσιάζουν ικανοποιητική σύγκλιση προς την επιθυμητή κατάσταση, με το Tyreus-Luyben να εμφανίζει μειωμένες ταλαντώσεις. Στην περίπτωση του ελεγκτή PID, και οι τρεις μέθοδοι εμφανίζουν μια αξιοσημείωτη προσέγγιση στην επιθυμητή κατάσταση, με τον Cohen-Coon να παρουσιάζει ελαφρώς μικρότερη ταλάντωση. Ωστόσο, είναι σημαντικό να σημειωθεί ότι η επιλογή της καταλληλότερης μεθόδου συντονισμού εξαρτάται από διάφορους παράγοντες, συμπεριλαμβανομένων των ειδικών απαιτήσεων εφαρμογής και συστήματος. Κάθε μέθοδος έχει τα δικά της πλεονεκτήματα και μειονεκτήματα, που απαιτούν προσεκτική εξέταση των χαρακτηριστικών του συστήματος και της επιθυμητής απόδοσης ελέγχου.

Η μέθοδος Ziegler-Nichols, γνωστή για την απλότητά της, χρησιμοποιείται ευρέως και ευνοείται στο πεδίο. Ωστόσο, μπορεί να μην παρέχει σταθερά βέλτιστη απόδοση, ειδικά όταν αντιμετωπίζουμε συστήματα που παρουσιάζουν μη γραμμική ή χρονικά μεταβαλλόμενη δυναμική. Σε τέτοιες περιπτώσεις, οι μέθοδοι Cohen-Coon και Tyreus-Luyben, οι οποίες βασίζονται περισσότερο στη μοντελοποίηση του συστήματος και ενδέχεται να απαιτούν πρόσθετες πληροφορίες συστήματος, ενδέχεται να προσφέρουν βελτιωμένη απόδοση, ιδιαίτερα για συστήματα με περίπλοκη ή μη γραμμική δυναμική. Για να εξακριβωθεί η πιο αποτελεσματική προσέγγιση συντονισμού για ένα δεδομένο σύστημα, συνιστάται να πειραματιστείτε

με διάφορες μεθόδους και ρυθμίσεις παραμέτρων. Η χρήση εργαλείων προσομοίωσης και η διεξαγωγή πρακτικών πειραμάτων επιτρέπει μια ολοκληρωμένη σύγκριση της απόδοσης και διευκολύνει τη λεπτομερή ρύθμιση των παραμέτρων του ελεγκτή για την επίτευξη της επιθυμητής απόδοσης ελέγχου.

Επιπλέον, η παρακολούθηση της τελευταίας έρευνας και των εξελίξεων στον σχεδιασμό και τη ρύθμιση του συστήματος ελέγχου είναι υψίστης σημασίας. Συνεχίζουν να εμφανίζονται νέες μέθοδοι και παραλλαγές, οι οποίες θα μπορούσαν να ταιριάζουν καλύτερα σε συγκεκριμένους τύπους συστημάτων ή στόχους ελέγχου. Ως εκ τούτου, το να παραμένει κανείς ενημερωμένος με το εξελισσόμενο τοπίο της θεωρίας ελέγχου είναι ζωτικής σημασίας για τη λήψη τεκμηριωμένων αποφάσεων σχετικά με τις μεθόδους συντονισμού. Στο ευρύτερο πεδίο εφαρμογής και σύγκρισης μεθόδων συντονισμού για ελεγκτές PID, είναι απαραίτητη μια διεξοδική προσέγγιση ανάλυσης και πειραματισμού. Η κατανόηση της δυναμικής του συστήματος, οι στόχοι απόδοσης και οι διαθέσιμες πληροφορίες είναι βασικά στοιχεία για την αξιολόγηση της αποτελεσματικότητας των διαφορετικών στρατηγικών ελέγχου. Η αξιολόγηση της απόκρισης, της σταθερότητας και των μετρήσεων σφάλματος του συστήματος παρέχει πολύτιμες πληροφορίες για την επιλογή της καταλληλότερης μεθόδου συντονισμού.

Συμπερασματικά, η εφαρμογή και η σύγκριση μεθόδων συντονισμού για ελεγκτές PID απαιτούν μια σχολαστική εξέταση της δυναμικής του συστήματος, των στόχων απόδοσης και των διαθέσιμων πληροφοριών. Ενώ η παρεχόμενη ανάλυση προσφέρει ένα πολύτιμο σημείο εκκίνησης για την κατανόηση των πλεονεκτημάτων και των αδυναμιών διαφόρων μεθόδων, απαιτείται περαιτέρω έρευνα και πειραματισμός για τον προσδιορισμό της βέλτιστης μεθόδου συντονισμού για μια συγκεκριμένη εφαρμογή.

## Βιβλιογραφία

- [1]Πασγιανός, Γ. (2003). Διατριβή: ΒαθμονόμησηPID ελεγκτών σε πραγματικό χρόνο για τον έλεγχο ασταθών συστημάτων με καθυστέρηση χρόνου - Κωδικός: 16295. Retrieved October 20, 2022, from thesis.ekt.gr website: <https://thesis.ekt.gr/thesisBookReader/id/16295?lang=el#page/1/mode/2u>
- [2] S., A. (2014). *COMPARISON OF TUNING METHODS OF PID CONTROLLER* (pp. 1–8).
- [3]How Does a PID Controller Work? - Structure & Tuning Methods. (2018, November 10). Retrieved from ElProCus - Electronic Projects for Engineering Students website: <https://www.elprocus.com/the-working-of-a-pid-controller/>
- [4]Woolf, P. (2020, May 19). 9.3: PID Tuning via Classical Methods. Retrieved from Engineering LibreTexts website: [https://eng.libretexts.org/Bookshelves/Industrial\\_and\\_Systems\\_Engineering/Book%3A\\_Chemical\\_Process\\_Dynamics\\_and\\_Controls\\_\(Woolf\)/09%3A\\_Proportional-Integral-Derivative\\_\(PID\)\\_Control/9.03%3A\\_PID\\_Tuning\\_via\\_Classical\\_Methods](https://eng.libretexts.org/Bookshelves/Industrial_and_Systems_Engineering/Book%3A_Chemical_Process_Dynamics_and_Controls_(Woolf)/09%3A_Proportional-Integral-Derivative_(PID)_Control/9.03%3A_PID_Tuning_via_Classical_Methods)
- [5 ]Kushwah, M., &Patra, A. (2014).*PID Controller Tuning using Ziegler-Nichols Method for Speed Control of DC Motor.*
- [6] Ibrahim, O., Yahaya, N. Z. B., &Saad, N. (2016). PID Controller Response to Set-Point Change in DC-DC Converter Control.*International Journal of Power Electronics and Drive Systems (IJPEDS)*, 7(2), 294. <https://doi.org/10.11591/ijpeds.v7.i2.pp294-302>

- [7] Luyben, W. and Luyben, M. Essentials of Process Control. McGraw-Hill, New York, 1997.
- [8] Sharmila, B., & Vidhyanandhan, L. (2016). *Modeling and Designing of Controllers for pH Process*.
- [9] Nikita S., Chidambaram M., Tuning of PID Controllers for First Order Plus Time Delay Unstable Systems. In: Regupathi I., Shetty K V., Thanabalan M. (eds) Recent Advances in Chemical Engineering. Springer, Singapore, 2016, pp 277-283.
- [10] Standard PID Tuning Methods. (2012).
- [11] Giriraj Kumar, S. M., Sivasankar, R., Radhakrishnan, T. K., Dharmalingam, V., & Anantharaman, N. (2008). Particle Swarm Optimization Technique Based Design of Pi Controller for a Real-Time Non-Linear Process. *Instrumentation Science & Technology*, 36(5), 525–542.  
<https://doi.org/10.1080/10739140802234980>
- [12] Sharmila, B., & Vidhyanandhan, L. (2016). *Modeling and Designing of Controllers for pH Process*.

## Παραρτήματα

### Παράρτημα 1 – ΚΩΔΙΚΑΣΖΝ.m

```
function [Gc] = ZieglerNichols(Gp)
syms s;
G_ClosedLoop = Gp;
G_ClosedLoop = simplifyFraction(G_ClosedLoop);
[num, den] = numden(G_ClosedLoop);

num=sym2poly(num);
den=sym2poly(den);

G_root_locus = tf(num,den);

[Kmax,~,W,~] = margin(G_root_locus);
if (isnan(W) || isinf(W))
    warning(['Model ',char(Gp),' is not suitable for Ziegler-Nichols
method, as it does not oscilate.']);
    Gc=[0,0,0];
    return;
end
T0 = (2*pi)/W;

Gc = {};

Kp = 0.5*Kmax;
P = Kp;
Gc{end+1} = P;

Kp = 0.45*Kmax;
Ti = (1/1.2)*T0;

PI = Kp*(1+(1/(Ti*s)));
Gc{end+1} = PI;

Kp=0.6*Kmax;
Ti=0.5*T0;
Td=0.125*T0;
PID=Kp*(1+(1/(Ti*s))+Td*s);
Gc{end+1} = PID;
end
```

## Παράρτημα2 – ΚΩΔΙΚΑΣ TL.m

```
Function [Gc] = TyreusLuyben(Gp)
syms s;
G_ClosedLoop = Gp;
G_ClosedLoop = simplifyFraction(G_ClosedLoop);
[num, den] = numden(G_ClosedLoop);

num=sym2poly(num);
den=sym2poly(den);

G_root_locus = tf(num,den);

[Kmax,~,W,~] = margin(G_root_locus);
if (isnan(W) || isinf(W))
    warning(['Model ',char(Gp),' is not suitable for Tyreus-Luyben',
method, as it does not oscilate.']);
    Gc=[0,0,0];
    return;
end
T0 = (2*pi)/W;

Gc = {};

% PI
Kp = Kmax/3.2;
Ti = T0*2.2;

PI = Kp*(1+(1/(Ti*s)));
Gc{end+1} = PI;

% PID
Kp = Kmax/2.2;
Ti = T0*2.2;
Td = T0/6.3;
PID = Kp*(1+(1/(Ti*s))+Td*s);
Gc{end+1} = PID;
end
```

## Παράρτημα – ΚΩΔΙΚΑΣCC.m

```
function [Gc] = CohenCoon(Gp)
syms s;
G_ClosedLoop = Gp;
G_ClosedLoop = simplifyFraction(G_ClosedLoop);
[num, den] = numden(G_ClosedLoop);

num=sym2poly(num);
den=sym2poly(den);
G_tf = tf(num,den);

A = 1;
opt = stepDataOptions('StepAmplitude',A);
[Y,T] = step(G_tf,opt);

t0 = T(1);
B = max(Y) - min(Y);

Y_t2 = Y<=(B/2);
Y_t2 = max(Y(Y_t2));
t2 = T(Y == Y_t2);

Y_t3 = Y<=(B*0.632);
Y_t3 = max(Y(Y_t3));
t3 = T(Y == Y_t3);

t1 = (t2-log(2)*t3)/(1-log(2));
t = t3 - t1;
tDel = t1 - t0;
K = B/A;
r = tDel/t;

Gc = {};

% P
Kp = (1/r*K)*(1 + r/3);
P = Kp;
Gc{end+1} = P;

% PI
Kp = (1/r*K)*(0.9 + r/12);
Ti = tDel * ( (30 + 3*r) / (9 + 20*r));
PI = Kp*(1+(1/(Ti*s)));

Gc{end+1} = PI;

% PID
Kp = (1/r*K)*(4/3 + r/4);
Ti = tDel * ( (32 + 6*r) / (13 + 8*r));
Td = tDel * ( 4 / (11 + 2*r));
PID = Kp*(1+(1/(Ti*s))+Td*s);
Gc{end+1} = PID;
end
```



## Παράρτημα – ΚΩΔΙΚΑΣ ΓΙΑ ΔΙΑΓΡΑΜΜΑΤΑ ΑΠΟΚΡΙΣΗΣ

```
function [] =  
createResponsePlotsPerController(Gp, controllers, c_names, c_type, model_  
name, totalTime)  
syms s;  
Ysp=1/s;  
time=0:0.1:totalTime;  
figure();  
hold on;  
title([c_type, ' controllers for model ', model_name]);  
for i=1:length(controllers)  
    Y=(Gp*controllers{i})/(1+Gp*controllers{i})*Ysp;  
    y=ilaplace(Y);  
    y=vpa(y);  
    y=subs(y,time);  
    y=double(y);  
    ysp1=ilaplace(Ysp);  
    ysp1=vpa(ysp1);  
    ysp1=subs(ysp1,time);  
    ysp1=double(ysp1);  
    ysp=ysp1;  
    plot(time,real(y));  
    axis([0 max(time) 0 2]);  
    xlabel('Time(s)');  
    ylabel('Response');  
end  
plot(time,ysp,'--');  
c_names{end+1} = 'Steady State';  
legend(c_names);  
saveas(gcf,[model_name, '_', c_type, '.png']);  
end
```

## Παράρτημα – ΚΩΔΙΚΑΣ ΓΙΑ ΔΙΑΓΡΑΜΜΑΤΑ ΣΦΑΛΜΑΤΟΣ ΓΙΑ ΚΑΘΕ ΕΛΓΚΤΗ

```
function [] =  
createErrorPlotsPerController(Gp,controllers,c_names,c_type,model_name,totalTime)  
syms s;  
Ysp=1/s;  
time=0:0.1:totalTime;  
figure();  
hold on;  
title([c_type, ' controller errors for model ',model_name]);  
for i=1:length(controllers)  
    Y=(Gp*controllers{i}/(1+Gp*controllers{i}))*Ysp;  
    y=ilaplace(Y);  
    y=vpa(y);  
    y=subs(y,time);  
    y=double(y);  
    ysp1=ilaplace(Ysp);  
    ysp1=vpa(ysp1);  
    ysp1=subs(ysp1,time);  
    ysp1=double(ysp1);  
    ysp=ysp1;  
    error = ysp - y;  
    plot(time,real(error));  
    %axis([0 max(time) 0 2]);  
    xlabel('Time(s)');  
    ylabel('Error');  
end  
legend(c_names);  
saveas(gcf,[model_name,'_',c_type,'_error.png']);  
end
```

ΠΑΡΑΡΤΗΜΑ Κώδικας εύρεση συνολικού σφάλματος για κάθε ελεγκτή:

```
function [total_errors] =
```

```

calculateTotalErrorPerController(Gp,controllers,c_names,c_type,model_
name,totalTime)
syms s;
Ysp=1/s;
time=0:0.1:totalTime;
total_errors = [];
for i=1:length(controllers)
    Y=(Gp*controllers{i}/(1+Gp*controllers{i}))*Ysp;
    y=ilaplace(Y);
    y=vpa(y);
    y=subs(y,time);
    y=double(y);
    ysp1=ilaplace(Ysp);
    ysp1=vpa(ysp1);
    ysp1=subs(ysp1,time);
    ysp1=double(ysp1);
    ysp=ysp1;
    error = ysp - y;
    total_errors(i) = sum(abs(real(error)));
end

end

```

## Παράρτημα : Κώδικας για διαγράμματα απόκρισης σήματος $u(t)$

```
function [] =  
createUtDiagramsPerController(controllers,Gv,Gp,Gm,Go,c_names,c_type,  
model_name>Total_Time)  
syms s t  
Ysp = 24/s;  
To = 24;  
time = 0:0.5>Total_Time;  
figure();  
hold on;  
title(['u(t) for ',c_type,' controllers for model ',model_name]);  
for i=1:length(controllers)  
    Gc = controllers{i};  
    U = (Gv*Gc/(1+Gc*Gv*Gp*Gm))*Ysp -  
(Gc*Gv*Gm*Go/(1+Gc*Gv*Gm*Gp))*To;  
    u = ilaplace(U);  
    u = vpa(u);  
    u = subs(u,time);  
    u = double(u);  
    plot(time,u);  
    xlabel('Time(s)');  
    ylabel('u(t)');  
end  
legend(c_names);  
saveas(gcf,[model_name,'_',c_type,'_u(t).png']);  
end
```

## Παράρτημα –MAIN ΚΩΔΙΚΑΣ

```
%% Clear Workspace
clc;
clear all;
close all;
%% Define System
addpath(' ../Models');
model_name = 'exe2real_ClosedLoop'; % exe2real_ClosedLoop
basicExample
run([model_name, '.m']);

%% Plot the root locus
[num, den] = numden(Gp);
num=sym2poly(num);
den=sym2poly(den);
G_root_locus = tf(num,den);
rlocus(G_root_locus)
%% Get controllers (P,PI, PID) from each method
Controllers_ZN = ZieglerNichols(Gp);
Controllers_CC = CohenCoon(Gp);
Controllers_TL = TyreusLuyben(Gp);

P_controllers = {Controllers_ZN(1),Controllers_CC(1)};
PI_controllers =
{Controllers_ZN(2),Controllers_CC(2),Controllers_TL(1)};
PID_controllers =
{Controllers_ZN(3),Controllers_CC(3),Controllers_TL(2)};

controller_names = {'Ziegler-Nichols','Cohen-Coon','Tyreus-Luyben'};
%% Produce Plots
model_name(strfind(model_name, "_")) = ' ';
% P controllers
createResponsePlotsPerController(Gp,P_controllers,controller_names(1:
2),'P',model_name,45);
createErrorPlotsPerController(Gp,P_controllers,controller_names(1:2),
'P',model_name,45);

% PI controllers
createResponsePlotsPerController(Gp,PI_controllers,controller_names, '
PI',model_name,45);
createErrorPlotsPerController(Gp,PI_controllers,controller_names,'PI'
,model_name,45);

% PID controllers
createResponsePlotsPerController(Gp,PID_controllers,controller_names,
'PID',model_name,45);
createErrorPlotsPerController(Gp,PID_controllers,controller_names,'PI
D',model_name,45);

if strcmp(model_name,'exe2real_ClosedLoop')

createUtDiagramsPerController(P_controllers,Gv,gp,Gm,Go,controller_na
mes,'P',model_name,70);

createUtDiagramsPerController(PI_controllers,Gv,gp,Gm,Go,controller_n
ames,'PI',model_name,70);

createUtDiagramsPerController(PID_controllers,Gv,gp,Gm,Go,controller_
names,'PID',model_name,70);
```

```

end
%% Get Controller Errors
errors = NaN*ones(3,3);
% P errors
errors(1:2,1) =
calculateTotalErrorPerController(Gp,P_controllers,controller_names(1:
2),'P',model_name,45);
% PI errors
errors(:,2) =
calculateTotalErrorPerController(Gp,PI_controllers,controller_names,'
PI',model_name,45);
% PID errors
errors(:,3) =
calculateTotalErrorPerController(Gp,PID_controllers,controller_names,
'PID',model_name,45);

errors = array2table(errors);
errors.Properties.RowNames = controller_names;
errors.Properties.VariableNames = {'P','PI','PID'};
errors.Properties.DimensionNames = {'Method','Controller Type'};
errors.Properties.Description = 'Total Errors of all controllers';
errors

```