

2^ο Εργαστήριο στα Συστήματα Ελέγχου

Ευάγγελος Κατσούπης 2017030077
Απόστολος Γιουμερτάκης 2017030142
Α. Ραφαήλ Ελληνιτάκης 2017030118
Κωνσταντίνος Βούλγαρης 2017030125

Ομάδα 37

5 Απριλίου 2021

Κεφάλαιο 1

Υπολογιστικό Μέρος

1.1 Έλεγχος στροφών Κινητήρα

Βασικός μας στόχος είναι να ελέγξουμε τον αριθμό στροφών του κινητήρα ανά λεπτό, με το ανάλογο σήμα ελέγχου κάθε φορά, και επίσης το αντίστοιχο φορτίο. Έχοντας την συνάρτηση μεταφοράς που περιγράφει έναν κινητήρα, προσπαθούμε να υπολογίσουμε την συμπεριφορά του θεωρητικά σε διάφορες καταστάσεις φόρτου και εισόδων ελέγχου. Χρησιμοποιήσαμε το λογισμικό Matlab για όλους τους υπολογισμούς και τις γραφικές παραστάσεις. Η συνάρτηση μεταφοράς που μας δίνεται:

$$H(s) = K_s \frac{1}{T_1 s + 1} \frac{1}{T_2 s + 1} = \frac{b_0}{s^2 + a_1 s + a_0}$$

1)Μας δίνονται οι εξής παράμετροι, που υπολογίστηκαν πειραματικά στο εργαστήριο για τον συγκεκριμένο DC PM κινητήρα:

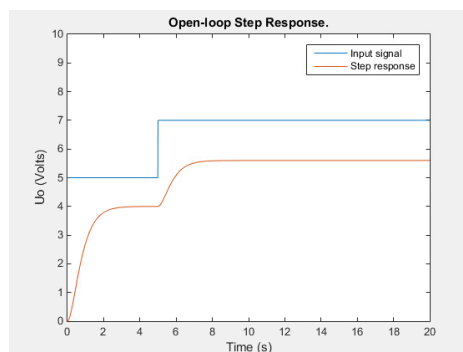
$$K_s = 0.8, T_g = 1.05 \text{sec}, T_u = 0.14 \text{sec}$$

Δεδομένων των παραπάνω, υπολογίσαμε τις παραμέτρους για ένα PI σύστημα ελέγχου με την μέθοδο CHR overshoot 0% . Όπως και στο προηγούμενο εργαστήριο, με την μέθοδο pidstd() ορίσαμε το εικονικό σύστημα ελέγχου μας. Το T_b αντιστοιχεί στο T_g και το T_e αντιστοιχεί στο T_u όπως ξέρουμε από θεωρία.Στον κώδικα 1.1 βλέπουμε την υλοποίηση σε Matlab.

```
Kp = (0.35*Tb)/(Ks*Te);  
Ti = 1.2*Tb;  
figure('Name','PI controller')  
PID_controller = pidstd(Kp, Ti); % definition of the PI controller  
m = feedback(PID_controller*sys,1); % definition of the feedback  
step(m) % step response of the system
```

Listing 1.1: Definition of the PI Controller

Στην εικόνα 1.1 βλέπουμε την απόκριση του συστήματος σε έναν παλμό από τα 5 στα 7 volt , όπως μας δόθηκε από την Matlab .



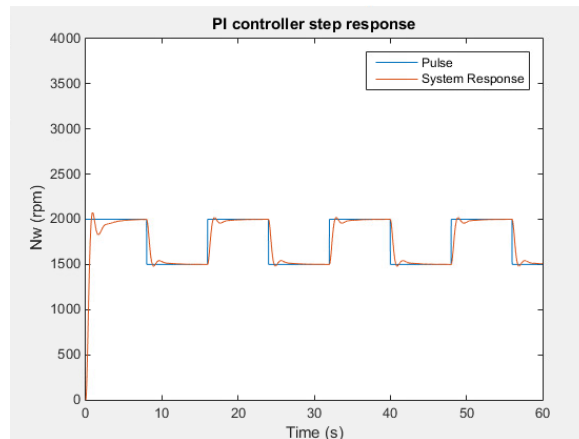
Σχήμα 1.1: Η απόκριση του νέου συστήματος σε ένα step απο τα 5 στα 7 volts.

2) Για το δεύτερο ερώτημα, πρέπει να δημιουργήσουμε τον παλμό που εικονίζεται στην εκφώνηση, με πλάτος peak to peak 500 rpm, θετικό offset 1500, και περίοδο 16 δευτερόλεπτα, ο οποίος θα χρησιμεύσει σαν σήμα ελέγχου στον κινητήρα μας. Με το σήμα αυτό, θα μπορέσουμε να πάρουμε ενδεικτικές μετρήσεις για την απόδοση του συστήματος ελέγχου μας και να το συγκρίνουμε με άλλα συστήματα ελέγχου, όπως στο τρίτο ερώτημα. Τον κατασκευάζουμε όπως φαίνεται στον κώδικα παρακάτω, στον κώδικα 1.2.

```
f = 1/16; % definition of the frequency
t1=0:0.01:60;
pulse = 500*square(2*pi*f*t1)/2 + 1750; % definition of the square pulse
plot(t1, pulse)
axis([0 60 0 4000])
hold on
resp = lsim(m,pulse,t1); % feeding the pulse into the control system m
plot(t1, resp)
e = resp - (pulse.');
```

Listing 1.2: Definition of the square pulse

Ο παλμός που δημιουργήθηκε, δόθηκε ως είσοδος ελέγχου στο σύστημα ρυθμισμένο με την μέθοδο CHR 0% overshoot που δημιουργήσαμε στο προηγούμενο ερώτημα και η απόκριση του κινητήρα φαίνεται στο σχήμα 1.2.



Σχήμα 1.2: Εδώ φαίνεται η απόκριση του συστήματος στο σήμα με CHR μέθοδο.

Για να προχωρήσουμε στο επόμενο βήμα, πρέπει πρώτα να ορίσουμε το σφάλμα συναρτήσεως του χρόνου:

$$e(t) = |response(t) - pulse(t)|$$

Το οποίο υλοποιήθηκε ως $e = \text{resp} - (\text{pulse.}')$; (με όρους Matlab) με την τελεία και την απόστροφο να δείχνουν τον ανάστροφο πίνακα του pulse για να μπορεί να γίνει η πράξη με την απόκριση. Το το σφάλμα συναρτήσεως του χρόνου, μπορεί να ολοκληρωθεί με τις παρακάτω μεθόδους, για να μας δώσει τους αντίστοιχους δείκτες σφάλματος, που είναι ένας καλός τρόπος να συγκρίνουμε μεθόδους ρύθμισης συστημάτων ελέγχου:

$$ISE = \int_0^{\infty} e^2(t) dx, ISE = \int_0^{\infty} |e(t)| dx$$

$$ITSE = \int_0^{\infty} te^2(t) dx, ITAE = \int_0^{\infty} t|e(t)| dx$$

Τα παραπάνω υλοποιήθηκαν με την συνάρτηση trapz() της Matlab, που υλοποιεί ολοκλήρωση χωρίζοντας το επίπεδο σε τραπέζια, όπως φαίνεται στον κώδικα 1.3.

```
iea = trapz(t1,abs(e)); % IAE trapz=numerical integration
ise = trapz(t1,e.^2); % ISE
itae = trapz(t1, t1'.*abs(e)); % ITAE
itse = trapz(t1,t1'.*(e.^2)); % ITSE
```

Listing 1.3: Definition of the Error Indices

Με τα παραπάνω, δηλαδή την ρύθμιση με CHR , καταλήξαμε στην εξής τιμή δείκτη απόδοσης αθροισμάτων τετραγώνων σφαλμάτων(ISE):

$$ISE = 1.5235 \cdot 10^6$$

και στους εξής δείκτες απόδοσης, απο τους οποίους θα χρησιμοποιήσουμε ως μέτρο σύγκρισης τον $ITAE$ καθώς είναι ο καταλληλότερος:

$$IAE = 2.7377 \cdot 10^3$$

$$ITAE = 5.7907 \cdot 10^4 (\text{προτιμητέος δείκτης})$$

$$ITSE = 1.5321 \cdot 10^7$$

3) **Ρύθμιση με την εμπειρική μέθοδο T_{sum} .** Για να ρυθμίσουμε τον ελεγκτή με την συγκεκριμένη μέθοδο, αρκεί να συμβουλευτούμε τον πίνακα που ξέρουμε για την ρύθμιση T_{sum} , όπως φαίνεται παρακάτω:

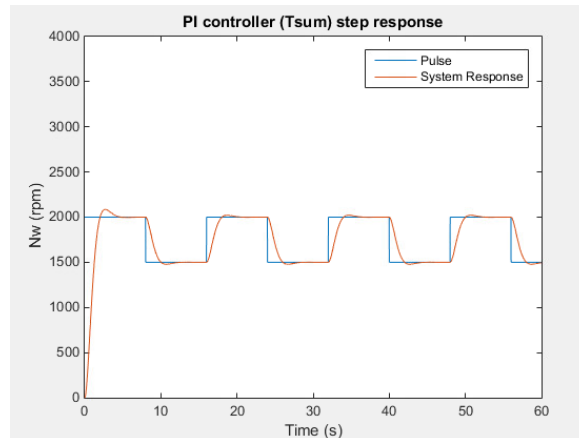
Ελεγκτής	K	T_i	T_d
PI	$\frac{0.5}{K_s}$	$0.5T_s$	-

Και στο παρακάτω πεδίο δίνεται ο κώδικας για την υλοποίηση. Γνωρίζουμε ότι $T_s = T_1 + T_2$ από θεωρία.:

```
Kp = 0.5/Ks; % Tsum Method to find Kp
Ti = 0.5*(T1+T2); % Tsum Method to find Ti
PID_controller = pidstd(Kp, Ti); % definition of the PI system
m = feedback(PID_controller*sys,1); % the feedback of the system
```

Listing 1.4: Νέος ορισμός K_p και T_i για το νέο εμπειρικά ρυθμισμένο σύστημα.

Στην εικόνα 1.3 φαίνεται η απόκριση στον παλμό του κινητήρα. Είναι εμφανώς ομαλότερη από την μέθοδο CHR 0% overshoot , αλλά με πιο αργή απόκριση.



Σχήμα 1.3: Η απόκριση του συστήματος που είναι ρυθμισμένο με την μέθοδο T_{sum} .

Με την μέθοδο T_{sum} , ο δείκτης απόδοσης τετραγώνων του σφάλματος:

$$ISE = 3.6908 \cdot 10^6$$

Ενώ οι υπόλοιποι δείκτες σφάλματος ήταν οι εξής:

$$IAE = 5.3722 \cdot 10^3$$

$$ITAE = 11.347 \cdot 10^4 (\text{προτιμητέος δείκτης})$$

$$ITSE = 3.7717 \cdot 10^7$$

Συμπαιρύνουμε απο την σύγκριση των δεικτών σφάλματος των δύο μεθόδων, CHR με 0% overshoot και T_{sum} , ότι η πρώτη μέθοδος συντελεί σε μικρότερο συνολικό σφάλμα, καθώς ο δείκτης ISE αλλά και ο $ITAE$ είναι φανερά μικρότεροι από ότι με την δεύτερη μέθοδο.

Γεννήτρια

- Όμοια με τα προηγούμενα ερωτήματα κατασκευάζουμε την συνάρτηση μεταφοράς σύμφωνα με τις παραμέτρους της γεννήτριας

$$K_s = 0.87, \quad T_u = 0.10 \text{ sec}, \quad T_g = 1.12 \text{ sec}$$

Η συνάρτηση μεταφοράς μας θα έχει όμοια την εξής μορφή:

$$H(s) = K_s \cdot \frac{1}{T_1 s + 1} \cdot \frac{1}{T_2 s + 1}$$

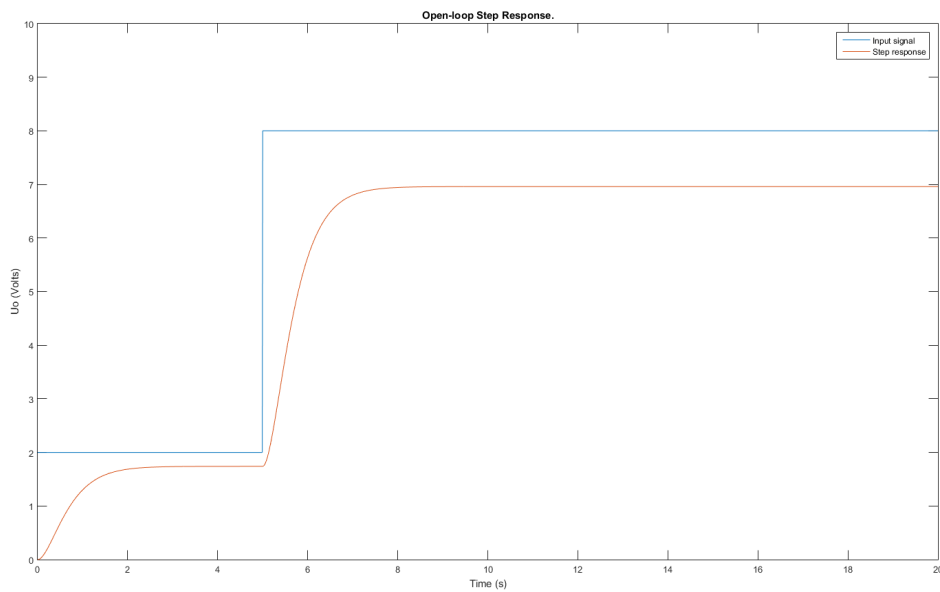
Για τον υπολογισμό των T_1, T_2 κάναμε χρήση των βασικών εξώσεων δευτεροβάθμιου συστήματος όπου:

$$T_1 \simeq 0.37 \cdot T_b, \quad T_2 \simeq 3.33 \cdot T_u$$

Άρα η συνάρτηση μεταφορά του παραπάνω συστήματος είναι η εξής:

$$H(s) = \frac{0.87}{0.138s^2 + 0.7474s + 1}$$

Η απόκριση συστήματος ανοιχτού βρόγχου για την παραπάνω συνάρτηση μεταφοράς είναι η εξής:



Σχήμα 1: Βηματική απόκριση ανοιχτού βρόγχου

Για τον υπολογισμό της συνάρτησης μεταφοράς αλλά και της απεικόνισης της γραφικής χρησιμο-
ποιήθηκε ο εξής κώδικας:

```
Ks = 0.87;
Te = 0.1; %tu
Tb = 1.12; %tg

T1 = 0.37*Tb;
T2 = 3.33*Te;

sys = Ks * tf(1, [T1 1]) * tf(1, [T2 1])

% D.4

figure('Name','Transfer function')
% [y,t] = step(sys);
t=0:0.01:20;
for k=1:length(t)
    if(t(k)<5)
        y(k) = 2;
    else
        y(k) = 8;
    end
end
plot(t, y);
hold on;
st = lsim(sys,y,t);
plot(t, st);
title('Open-loop Step Response. ');
ylabel('Uo (Volts)');
xlabel('Time (s)');
axis([0 20 0 10]);
legend('Input signal', 'Step response');
hold off;
```

Listing 1: TF and plots

Με βάση την παραπάνω και με την μέθοδο CHR για 0% overshoot (setpoint response) υπολο-
γίζουμε τις παραμέτρους για τον PI ελεγκτή.

0% OVERSHOOT			
ΕΛΕΓΚΤΗΣ	K	T_i	T_d
P	$\frac{0.3 T_b}{K T_e}$.	.
PI	$\frac{0.35 T_b}{K T_e}$	$1.2 T_b$.
PID	$\frac{0.6 T_b}{K T_e}$	T_b	$0.5 T_e$

Άρα έχουμε:

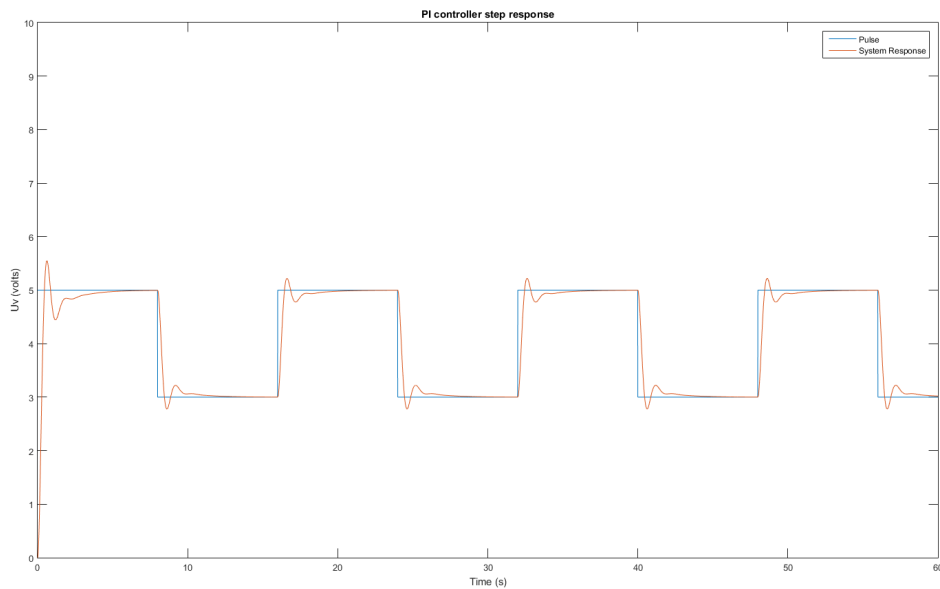
0% OVERSHOOT			
ΕΛΕΓΚΤΗΣ	K	T_i	T_d
PI	4.5057	1.3440	.

Για τον υπολογισμό των παραμέτρων του PI controller αλλά και για την απεικόνιση της γραφικής χρησιμοποιήθηκε ο εξής:

```
% PI controller
%%%Parameters;
Kp = (0.35*Tb)/(Ks*Te);
Ti = 1.2*Tb;
%\\
figure('Name','PI controller');
PID_controller = pidstd(Kp, Ti);
m = feedback(PID_controller*sys,1);
step(m);
title(['PI controller (Kp = ' num2str(Kp) ', Ki = ' num2str(Ti) ')']);
legend('PI_step');
```

Listing 2: Pi Controller

- Για το 5^ο ερώτημα, όμοια με το 2^ο , δημιουργήσαμε το δεδομένο σήμα εισόδου και στην συνέχεια το θέσαμε ως είσοδο στο παραπάνω σύστημα.
Το αποτέλεσμα:



Σχήμα 2: Απόκριση συστήματος με PI Controller με περιοδική είσοδο.

Για τον υπολογισμό των δεικτών σφάλματος, με όμοιο τρόπο, ορίσαμε το σφάλμα των δύο παλμών κάθε χρονική στιγμή:

$$e(t) = |response(t) - pulse(t)|$$

ώστε στην συνέχεια να υπολογίσουμε τους δείκτες απόδοσης:

- $ISE = 9.9818 \cdot 10^2$
- $IAE = 7.6093$
- $ITAE = 185.4254$ (ο προτιμότερος δείκτης)
- $ITSE = 172.3845 \cdot 10^3$

Κώδικας για τον υπολογισμό των δεικτών απόδοσης:

```
f = 1/16;
t1=0:0.01:60;
pulse = 2*square(2*pi*f*t1)/2 + 4;
figure('Name','Step response')
plot(t1, pulse)
axis([0 60 0 10])
hold on
resp = lsim(m,pulse,t1);
plot(t1, resp)
title('PI controller step response')
legend('Pulse', 'System Response')
xlabel('Time (s)');
ylabel('Uv (volts)');
e = resp - (pulse. ');

% Errors
iea = trapz(t1,abs(e)); % IAE trapz=numerical integration
ise = trapz(t1,e.^2); % ISE
itae = trapz(t1, t1'.*abs(e)); % ITAE
itse = trapz(t1,t1'.*(e.^2)); % ITSE
```

Listing 3: Performance indicators

- Για τον υπολογισμό της απόκρισης με την χρήση της εμπειρικής μεθόδου, υπολογίσαμε αρχικά το T_{sum} το οποίο ισούται με:

$$T_{sum} = T_1 + T_2 \simeq 0.37T_u + 3.33T_e = 0.7474$$

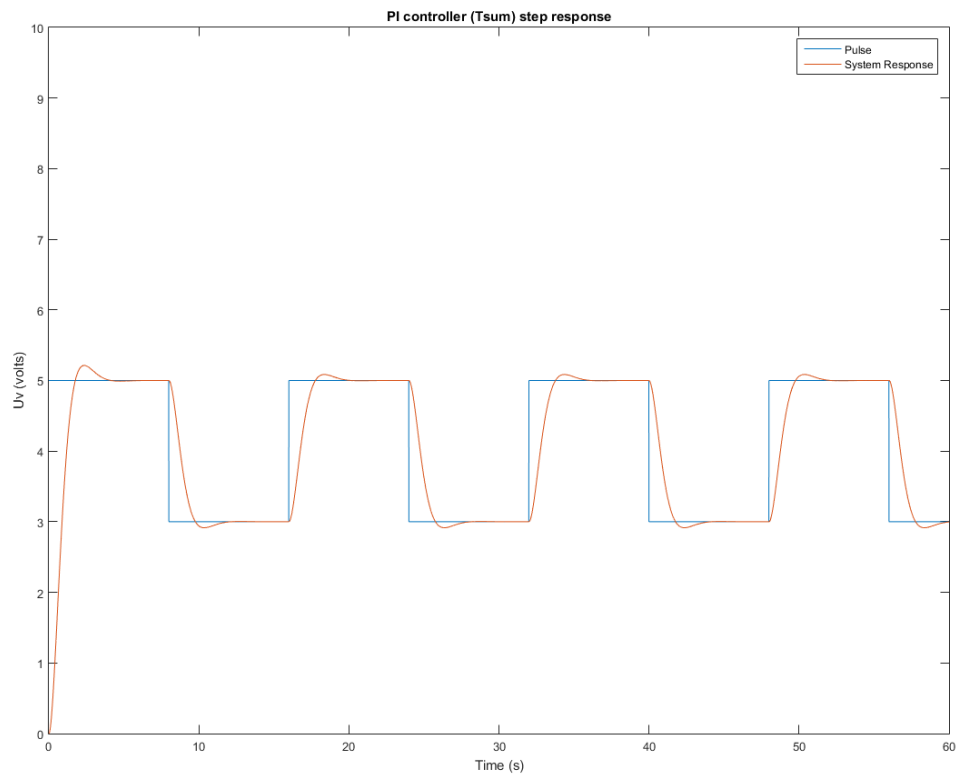
Στην συνέχεια, υπολογίζονται οι παράμετροι για του PI ελεγκτή σύμφωνα με τον πίνακα:

ΕΛΕΓΚΤΗΣ	K	T_i	T_d
P	$\frac{1}{K_s}$.	.
PI	$\frac{0.5}{K_s}$	$0.5T_\Sigma$.
PID	$\frac{1}{K_s}$	$0.66T_\Sigma$	$0.17T_\Sigma$

και έχουμε το εξής:

ΕΛΕΓΚΤΗΣ	K	T_i	T_d
PI	0.5747	0.3737	.

Στην συνέχεια θέσαμε σαν είσοδο στο σύστημα το σήμα που μας ενδιαφέρει και έχουμε τα παρακάτω αποτελέσματα:



Σχήμα 3: Απόκριση συστήματος με PI Controller υπολογισμένος με χρήση εμπειρικής μεθόδου (T_{Σ}).

Τέλος υπολογίσαμε τους δείκτες απόδοσης:

- $ISE = 29.8029$
- $IAE = 16.2469$
- $ITAE = 394.0100$ (ο προτιμότερος δείκτης)
- $ITSE = 517.0379$

Κώδικας για τον υπολογισμό των παραμέτρων αλλά και των δεικτών απόδοσης:

```
% Step response for voltage control with Tsum method
f = 1/16;
t1=0:0.01:60;
pulse = 2*square(2*pi*f*t1)/2 + 4;
figure('Name','Step response')
plot(t1, pulse)
axis([0 60 0 10])
hold on
resp = lsim(m,pulse,t1);
plot(t1, resp)
title('PI controller (Tsum) step response')
legend('Pulse', 'System Response')
xlabel('Time (s)');
ylabel('Uv (volts)');
e = resp - (pulse. ');

% Errors
iea = trapz(t1,abs(e)); % IAE trapz=numerical integration
ise = trapz(t1,e.^2); % ISE
itae = trapz(t1, t1'.*abs(e)); % ITAE
itse = trapz(t1,t1'.*(e.^2)); % ITSE
```

Listing 4: System response and performance indicators

Κεφάλαιο 2

Ε. Επεξεργασία Μετρήσεων και Αξιολόγηση Αποτελεσμάτων

1. Οι DC Permanent Magnet κινητήρες είναι πιο σύνθετοι στην κατασκευή τους -γιατί για παράδειγμα μπορεί να χρειαστούν ψύκτρες ή άλλα εξαρτήματα- από τους εναλλασσόμενου ρεύματος αλλά είναι πιο γραμμικοί, το οποίο τους καθιστά πιο εύκολους στον έλεγχο τους.
2. Ο κινητήρας που ελέγχουμε, ουσιαστικά δέχεται μία τάση A και ανάλογα το μέγεθος της και το φορτίο του, ανταποκρίνεται σε συγκεκριμένο αριθμό στροφών ανά λεπτό. Σαν φορτίο μπορεί να χρησιμοποιηθεί η γεννήτρια που είναι συνδεδεμένη στον άξονα του κινητήρα, είτε με προσθήκη φορτίου στην έξοδο της, είτε με μικρή τάση που την ωθεί να περιστραφεί αντίθετα με τον κινητήρα. Αντίστοιχα, την γεννήτρια την ελέγχουμε δίνοντας της συγκεκριμένο αριθμό στροφών, από τον οποίο παίρνουμε συγκεκριμένη τάση. Ός πηγή της περιστροφικής κίνησης αυτήν την φορά, μπορούμε επίσης να έχουμε έναν κινητήρα ο οποίος έχει τον άξονα του συνδεδεμένο με τον άξονα της γεννήτριας.
Ο ίδιος τύπος κινητήρα επίσης, όταν του δίνουμε περιστροφική κίνηση στον δρομέα του, λειτουργεί ως γεννήτρια και ως εκ τούτου μπορεί να χρησιμοποιηθεί εναλλάξ σε ότη χρήση από τις δύο θέλουμε.
3. Τα δύο βασικά προβλήματα που αντιμετωπίσαμε, είναι αυτά του ελέγχου στροφών ενός κινητήρα, και του ελέγχου τάσης εξόδου της γεννήτριας. Πιο αναλυτικά, στην περίπτωση της **γεννήτριας**, το φορτίο στο ρεύμα που παράγει επηρεάζει την τάση εξόδου της, και αυτό προσπαθούμε να το ελέγξουμε μεταβάλλοντας τις στροφές που δίνουμε στον δρομέα της γεννήτριας. Όσον αφορά τον **κινητήρα**, το φορτίο στον άξονα του επηρεάζει τις στροφές ανά λεπτό στις οποίες δουλεύει ο κινητήρας, και αυτό προσπαθούμε να το κρατήσουμε σταθερό μεταβάλλοντας κατάλληλα την τάση εισόδου.
4. Στα δύο παραπάνω προβλήματα χρησιμοποιήθηκε η μέθοδος CHR , καθώς προσφέρει καλύτερη σταθερότητα στο σύστημα σε σχέση με άλλες μεθόδους όπως η ZN (Ziegler-Nichols). Επίσης , άλλες μέθοδοι σχεδιασμού ελεγκτή όπως η Ziegler-Nichols , βασίζονται στην εύρεση του κέρδους κρίσιμου ελεγκτή K_{crit} , το οποίο προϋποθέτει το σύστημα να μπορεί να ταλαντωθεί χωρίς να προκληθεί ζημιά, το οποίο δεν είναι πάντα εφικτό σε ένα κινητήρα ή μια γεννήτρια. Τέλος, η επιλογή της μεθόδου CHR για την ρύθμιση του συστήματος, κάνει για εμάς αρκετά πιο εύκολο να υπολογίσουμε τις σταθερές T_u και T_g καθώς υπάρχει σημείο καμπής, κάτι που δεν μας διευκολύνει απαραίτητα με την μέθοδο Ziegler Nichols. Επίσης, φαίνεται να είναι αρκετά πιο αποτελεσματική από την μέθοδο T_{sum} , καθώς οι δείκτες σφάλματος είναι εμφανώς μικρότεροι, όπως θα δούμε στο ερώτημα 7.
5. Για τον σχεδιασμό του ελεγκτή, χρησιμοποιήθηκε η μέθοδος CHR με overshoot 0% για setpoint response , η οποία είναι η πιο κατάλληλη για την συγκεκριμένη εφαρμογή καθώς θέλουμε η έξοδος του κάθε συστήματος να πλησιάζει όσο το δυνατόν καλύτερα την είσοδο.
6. Χρησιμοποιείται PI ελεγκτής γιατί το σύστημα είναι αρκετά γρήγορο και δεν χρειάζεται το D στοιχείο για διόρθωση του σφάλματος του ακόμα γρηγορότερα.

7. Tracking είναι το πρόβλημα στο οποίο θέλουμε η έξοδος να ακολουθεί όσο το δυνατόν πιο κοντά την είσοδο που δίνουμε στο σύστημα. Παρατηρούμε ότι με την μέθοδο σχεδιασμού CHR και στα δύο προβλήματα, η έξοδος του συστήματος συγκλίνει περισσότερο στην είσοδο και πιο γρήγορα απ' ό,τι στην μέθοδο T_{sum} , όπως μας δείχνουν και οι δείκτες σφάλματος παρακάτω, αρχικά για τον κινητήρα:

- Στην μέθοδο CHR : $ISE = 1.5235 \cdot 10^6$ και $ITAE = 5.7907 \cdot 10^4$
- Στην μέθοδο T_{sum} : $ISE = 3.6908 \cdot 10^6$ και $ITAE = 11.327 \cdot 10^4$

Και όσον αφορά την γεννήτρια, επίσης ισχύει ότι:

- Στην μέθοδο CHR : : $ISE = 9.9818$ και $ITAE = 185.4254$
- Στην μέθοδο T_{sum} $ISE = 29.8029$ και $ITAE = 394.0100$

Κεφάλαιο 3

Επεξεργασία του υπολογιστικού μέρους και Αξιολόγηση Αποτελεσμάτων

- Αρχικά και στα δύο πειράματα, για τον υπολογισμό των παραμέτρων του PI (το σύστημα motor/-generator είναι ήδη γρήγορο) χρησιμοποιήθηκε η μέθοδος CHR με 0% *overshoot* και *setpointresponse*. Για την σύγκριση αποδοτικότητας της κάθε περίπτωσης μετρήσαμε το σφάλμα μεταξύ της εισόδου και της απόκρισης του κάθε συστήματος.

Αυτό έγινε με την χρήση των δεικτών απόδοσης, που προαναφέρθηκαν. Συγκεκριμένα εστιάσαμε στα:

- ISE που, τετραγωνίζοντας το σφάλμα, αγνοεί τα τα σφάλματα μικρότερα της μονάδας αλλά, αντίστοιχα προωθεί τα ποιο μεγάλα.
- Αντίστοιχα ο ITEA οπού εξαρτάται από το μέτρο του σφάλματος για κάθε χρονική στιγμή, λαμβάνει υπόψιν όλες τις αποκλίσεις στον ίδιο βαθμό τονίζοντας αυτά τα οποία γίνονται κατά την μόνιμη κατάσταση.

Μπορεί εύκολα να διαπιστωθεί πως ο ITEA είναι πιο αποδοτικός για τα παραπάνω πειράματα.

Τέλος συγκρίνοντας τις 2 μεθόδους μπορούμε να πούμε πως η CHR έχει καλύτερη απόδοση συγκριτικά από αυτή της T_{sum} λόγω των χαμηλότερων τιμών που λαμβάνουν οι ISE & ITAE, αλλά περισσότερη πολυπλοκότητα αντίστοιχα, καθώς με την T_{sum} μπορούμε να υπολογίσουμε την παράμετρο T_{Σ} εμπειρικά χωρίς να γνωρίζουμε τις προδιαγραφές του συστήματος. Αναλυτικότερα, οι τιμές των δεικτών φαίνονται στο προηγούμενο κεφάλαιο στην έβδομη ερώτηση.