

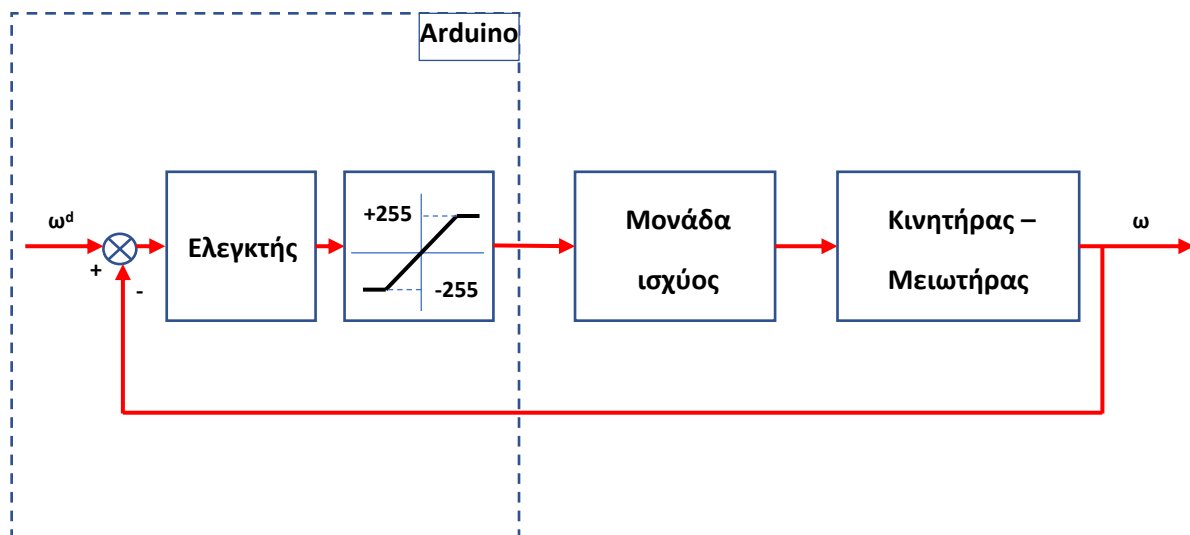
Εργαστηριακή Άσκηση 2

Έλεγχος κλειστού βρόχου γωνιακής ταχύτητας κινητήρα DC με ενσωματωμένο μειωτήρα

Υλοποίηση Αναλογικού Ελέγχου (P)

Στην εργαστηριακή Άσκηση 1 διαπιστώσαμε ότι μπορούμε να δώσουμε κάποιον αριθμό ως εντολή από τον μικροπολογιστή και ο κινητήρας να περιστραφεί. Η συνάρτηση μεταφοράς ανοικτού βρόχου που προσδιορίστηκε μας λέει επίσης πόση γωνιακή ταχύτητα περιμένουμε ανά μονάδα εντολής αλλά και το πόσο γρήγορα φτάνει ο κινητήρας την ταχύτητα αυτή.

Για αν μπορέσουμε να ελέγξουμε πραγματικά (σωστά) την γωνιακή ταχύτητα, θα πρέπει να είμαστε σε θέση να ζητήσουμε από το σύστημα συγκεκριμένη, επιθυμητή τιμή, ω^d και ο κινητήρας να την δώσει. Κάτι τέτοιο μπορεί να γίνει μόνο με σύστημα κλειστού βρόχου. Στο Σχήμα που ακολουθεί φαίνεται το σύστημα αυτό.



Θα προσπαθήσουμε στην συνέχεια να υλοποιήσουμε με την βοήθεια του μικροελεγκτή Arduino Uno κατάλληλους «Ελεγκτές», ώστε να ελέγξουμε την γωνιακή ταχύτητα όσο γίνεται καλύτερα. Το «στοιχείο κορεσμού» που φαίνεται στο σχήμα (μετά τον ελεγκτή), αποτυπώνει το γεγονός ότι η μονάδα ισχύος δέχεται εντολές με τιμές μεταξύ -255 και 255. Πρόκειται για ένα στοιχείο που εισάγει όπως λέμε «μη γραμμικότητες» στο σύστημα.

Απλός αναλογικός ελεγκτής (Proportional Control)

Είναι η απλούστερη μορφή ελεγκτή. Πρόκειται για ένα κέρδος K_p .

Υλοποίηση με την βοήθεια προγράμματος στον Arduino.

```

Encoder Enc(2, 3);

long pos =0;

long newP;

long vel, set_point, Err=0, cmd=0;

float Kp=0.1; (Κέρδος αναλογικού ελέγχου)

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
  digitalWrite(8, LOW); (Απαιτείται !)
  set_point=3600; (Επιθυμητή ταχύτητα σε παλμούς/sec)
  pos=0; vel=0;
  Enc.write(0);
}

void loop()
{
  cmd=Kp*(Err); (Εντολή = Κρ * Σφάλμα ταχύτητας)
  if (cmd<0) {cmd=-cmd; digitalWrite(13, LOW);} (Αν η εντολή είναι αρνητική, άλλαξε εντολή φοράς )
    else {digitalWrite(13, HIGH);}
  if (cmd>255) cmd=255; (Αν η εντολή προκύψει μεγαλύτερη του μεγίστου, περιορίσέ την ..)
  analogWrite(11, cmd); (Στείλε την εντολή στην εγκατάσταση)
  delay(10);
  newP = Enc.read();
  vel = (newP-pos) * 100; (Υπολόγισε την πραγματική ταχύτητα σε παλμούς/sec)
  Serial.println (vel);
  pos = newP;
  Err=set_point-vel; (Υπολόγισε το σφάλμα ταχύτητας)
}

```

Κινήστε τον κινητήρα και δείτε την πραγματική τιμή της ταχύτητας.

Το σφάλμα μόνιμης κατάστασης που θα περιμέναμε, σύμφωνα με την θεωρία, είναι :

$$e_{ss} = \frac{A}{1 + K_p K_{tot}}$$

Όπου A η επιθυμητή τιμή γωνιακής ταχύτητας, 3600 παλμοί/sec εν προκειμένω και K_{tot} ότι έχετε βρει από την ταυτοποίηση του συστήματος.

Συγκρίνετε την παραπάνω τιμή σφάλματος με αυτή που βρίσκετε στην πραγματική εγκατάσταση. Αξιολογήστε την διαφορά.

Αλλάξτε την τιμή του κέρδους K_p σε 0.3 και δείτε ότι το σφάλμα μόνιμης κατάστασης μικραίνει.

Συμπεριφορά του κινητήρα σε επιβολή εξωτερικής ροπής «φρεναρίσματος» - διαταραχή.

α. Κινείστε τον κινητήρα με «ανοικτό βρόχο», χρησιμοποιώντας το «προγραμματάκι» του προηγούμενου εργαστηρίου.

Φρενάρετε τον άξονά του με το χέρι σας. Παρατηρήστε ότι η γωνιακή ταχύτητα μειώνεται – αναμενόμενο.

Αξιολογήστε την μείωση αυτή παρατηρώντας στην οθόνη του Η/Υ την πτώση της γωνιακής ταχύτητας.

Σωστή ποσοτική αξιολόγηση μπορεί να γίνει μόνο με χρήση συσκευής που μπορεί να επιβάλλει ελεγχόμενη ροπή φρεναρίσματος – δεν την έχουμε προς το παρόν.

β. Κινείστε τον κινητήρα με «κλειστό βρόχο», χρησιμοποιώντας το «προγραμματάκι» του σημερινού εργαστηρίου.

Για τρεις διαφορετικές τιμές του κέρδους K_p , (0.1, 0.3, 0.5) αξιολογήσετε και πάλι την μείωση της γωνιακής ταχύτητας, προσπαθώντας (δύσκολο) να επιβάλλετε την ίδια ροπή φρεναρίσματος με το χέρι σας. Παρατηρείστε επίσης ότι το ρεύμα που «τραβά» ο κινητήρας.