Univerzitet u Novom Sadu
Fakultet tehničkih nauka
Novi Sad
Departman za računarstvo i automatiku
Odsek za računarsku tehniku i računarske komunikacije

Električni dinamometar

Predmet: **Logičko projektovanje računarskih sistema 2**

Autori:
Žarko Ostojić- RA 192/2020
Mladen Bazina – RA194/2020
Bogdan Petrić – RA 113/2021

Mentor:
Miloš Subotić

Novi Sad, Jun, 2024.

Uvod

Ovaj kod implementira osnovnu kontrolu za trofazni BLDC motor koristeći trapezoidno upravljanje. Kod je napisan za Arduino platformu i koristi digitalne pinove za kontrolu motora. Trapezoidno upravljanje podrazumeva komutaciju faza motora kako bi se postiglo kontinualno rotiranje rotora. Ovaj sistem može biti korišćen u raznim aplikacijama, uključujući testiranje motora pomoću električnog dinamometra.

Trapezoidno upravljanje

Trapezoidno upravlanje je metoda kontrole BLDC motora koja koristi šest različitih stanja za komutaciju faza kako bi se postiglo rotiranje motora. Ovo upravljanje naziva se trapezoidnim zbog talasnog oblika napona primenjenog na faze koji podseća na trapez.
Važniji pojmovi:

Komutacija - Proces prebacivanja između različitih faza kako bi se održalo kontinuirano rotiranje rotora.

Faze - BLDC motor ima tri faze (U,V,W), svaka sa dva izlaza (EN i IN).

Stanja faza - Svaka faza može biti u jednom od tri stanja - pozitivno (1), negativno (0) ili u stanju visoke impedanse (HiZ).

Rotacija - Rotacija rotora se postiže promenom stanja faza u tačno određenom redosledu.

Trapezoidalno upravljanje je jednostavno za implementiaciju i efikasno za mnoge aplikacije koje zahtevaju visoku efikasnost pri konstantim brzinama, a zbog svoje jednostavnosti smanjuje troškove sistema.

Definicije i makroi

#define T_MS (500): Definiše konstantu za kašnjenje u milisekundama.
#define DELAY() do{ delay(T_MS); }while(0): Makro za kašnjenje.
#define ENU 2, #define ENV 4, #define ENW 7 : Definišu pinove za omogućavanje faza U, V i W.
#define INU 3, #define INV 5, #define INW 6 : Definišu pinove za ulazne signale faza U, V i W.

Promenljive

int currentStep = 0 : Trenutni korak u periodi komutacije
unsigned long commutationInterval = 10 : Interval između komutacija u milisekundama.
unsigned long lastCommutationTime = 0 : Vreme poslednje komutacije
unsigned long currentRunTime = 0 : Trenutno vreme rada motora u milisekundama.
unsigned long maxRunTime = 15000 : Maksimalno dozvoljeno vreme rada motora u milisekundama.
unsigned long totalRunTime = 0 : Ukupno vreme rada motora

Funkcije

setPhaseState - Postavlja stanja pinova za sve tri faze motora

```
void setPhaseState(int enU, int inU, int enV, int inV, int enW, int inW) {
  digitalWrite(ENU, enU);
  digitalWrite(INU, inU);
  digitalWrite(ENV, enV);
  digitalWrite(INV, inV);
  digitalWrite(ENW, enW);
  digitalWrite(INW, inW);
}
```

setPhase - Postavlja stanje za pojedinačne faze motora (U, V, W) na osnovu unetih parametara.

```
void setPhase(int phase, int state)
{
  switch(phase){
    case U:
      switch(state){
        case 0:
          digitalWrite(ENU, 1);
          digitalWrite(INU, 0);
          break;
        case 1:
          digitalWrite(ENU, 1);
          digitalWrite(INU, 1);
          break;
        case Z:
          digitalWrite(ENU, 0);
          digitalWrite(INU, 0);
          break;
      }
    case V:
      switch(state){
        case 0:
          digitalWrite(ENV, 1);
          digitalWrite(INV, 0);
          break;
        case 1:
          digitalWrite(ENV, 1);
          digitalWrite(INV, 1);
          break;
        case Z:
          digitalWrite(ENV, 0);
          digitalWrite(INV, 0);
          break;
      }
```

```
    case W:
      switch(state){
        case 0:
          digitalWrite(ENW, 1);
          digitalWrite(INW, 0);
          break;
        case 1:
          digitalWrite(ENW, 1);
          digitalWrite(INW, 1);
          break;
        case Z:
          digitalWrite(ENW, 0);
          digitalWrite(INW, 0);
          break;
      }
  }
}
```

setPhaseState2 - Postavlja upravljanje za sve tri faze motora istovremeno.

```
void setPhaseState2(int u, int v, int w)
{
  setPhase(U, u);
  setPhase(V, v);
  setPhase(W, w);
}
```

updateMotorPosition - Ažurira položaj motora na osnovu trenutnog koraka komutacije.

```
void update_motor_position(int step) {
  switch (step) {
    case 0:
      setPhaseState(HIGH, LOW, HIGH, HIGH, LOW, LOW);
      break;
    case 1:
      setPhaseState(HIGH, LOW, LOW, LOW, HIGH, HIGH);
      break;
    case 2:
      setPhaseState(LOW, LOW, HIGH, LOW, HIGH, HIGH);
      break;
    case 3:
      setPhaseState(HIGH, HIGH, HIGH, LOW, LOW, LOW);
      break;
    case 4:
      setPhaseState(HIGH, HIGH, LOW, LOW, HIGH, LOW);
      break;
    case 5:
      setPhaseState(LOW, LOW, HIGH, HIGH, HIGH, LOW);
      break;
  }
}
```

setup - Inicijalizuje serijsku komunikaciju i postavlja režime rada pinova.

```
void setup() {
  Serial.begin(115200);
  Serial.println("setup() running...");
  delay(1);

  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, 0);

  pinMode(ENU, OUTPUT);
  digitalWrite(ENU, 0);
  pinMode(INU, OUTPUT);
  digitalWrite(INU, 0);
  pinMode(ENV, OUTPUT);
  digitalWrite(ENV, 0);
  pinMode(INV, OUTPUT);
  digitalWrite(INV, 0);
  pinMode(ENW, OUTPUT);
  digitalWrite(ENW, 0);
  pinMode(INW, OUTPUT);
  digitalWrite(INW, 0);
}
```

loop - Glavna petlja koja kontroliše motor, ažurira položaj motora u redovnim intervalima.

```
void loop() {
  unsigned long currentTime = millis();
  if (currentTime - lastCommutationTime >= commutationInterval) {
    lastCommutationTime = currentTime;
    update_motor_position(currentStep);
    currentStep = (currentStep + 1) % 6;
    totalRunTime += commutationInterval;
  }
}
```

Zaključak

Trapezoidno upravljanje predstavlja efikasan i jednostavan metod za kontrolu BLDC motora, omogućavajući precizno upravljanje brzinom i položajem rotora. Pogodan je za aplikacije gde su jednostavnost i efikasnost ključni. Implementacija trapezoidnog upravljanja na Arduino platformi, kao što je prikazano u ovom kodu, pruža osnovu za širok spektar primena, uključujući robotiku, industrijsku automatizaciju i električna vozila.

Korišćenje trapezoidnog upravljanja u kombinaciji sa električnim dinamometrom omogućava detaljno testiranje i karakterizaciju BLDC motora. Električni dinamometar meri ključne parametre kao što su snaga i obrtni moment pod različitim opterećenjima, pružajući dragocene povratne informacije za optimizaciju dizajna i performansi motora. Ovaj sistem omogućava inženjerima da simuliraju različite radne uslove, analiziraju ponašanje motora i identifikuju potencijalne oblasti za poboljšanje.