

Musterlösung Laborübung Gleichstrommotor und Motorkennlinie

SchülerInnen der Gruppe: Erharder Leonhard

Verfasser: Erharder Leonhard

Datum der Durchführung: 03.17.2021

- [Musterlösung Laborübung Gleichstrommotor und Motorkennlinie](#)
 - [Teil 1](#)
 - [Gleichstrommotoren](#)
 - [Betriebsspannung](#)
 - [Kalibrierung](#)
 - [Messungen](#)
 - [Genauigkeit](#)
 - [Teil 2](#)
 - [Drehzahlmessung](#)
 - [Kombination](#)
 - [Messung](#)
 - [Auswerten der Messdaten](#)
 - [Interpretation](#)
-

Teil 1

Gleichstrommotoren

Bei dem für die Übung verwendeten Elektromotor handelt es sich um einen permanterregten Gleichstrommotor.

Da dieser mit Graphitbürsten betrieben wird kann er den Kommutator(Stromwender)-Maschinen zugeordnet werden.

Diese Art von Elektromotoren ist hauptsächlich bei kleinen Motoren gebräuchlich.

Anwendungsfälle wären:

- Spielzeug
- Fahrradantriebe

Ihr größter Vorteil liegt in dem, in Fällen von kleinen Motoren, verbesserten Wirkungsgrad.

Auch größere Motoren sind möglich, jedoch sind diese aufgrund der Kosten großer Permanentmagnete sehr selten.

Der Nachteil findet sich in der geringeren möglichen Variation der Drehzahl, da das Magnetfeld nicht geschwächt werden kann.

Betriebsspannung

Trotz einer Nominalspannung von unter 20 Volt kann ihm Rahmen der Übung mit 20 Volt gearbeitet werden, da die Nominalspannung nur für den Dauerbetrieb aussagekräftig ist, und die über die Dauer der Versuche, durch die höhere Spannung, entstehende Hitze den Motor nicht beschädigt.

Kalibrierung

Die Kalibrierung der Wägezelle erfolgt mit folgenden Code.

```
1  #include <HX711.h>
2  #define LOADCELL_DOUT_PIN  2
3  #define LOADCELL_SCK_PIN   3
4
5  HX711 scale;
6
7  float calibration_factor = 433; //Working for the 5kg full bridge load cell
8
9  void setup() {
10     Serial.begin(9600);
11     Serial.println("After readings begin, place known weight on scale");
12     Serial.println("Adjust Factor by pressing + or -");
13
14     scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
15     scale.set_scale();
16     scale.tare(); //Reset the scale to 0
17
18 }
19
20 void loop() {
21
22     scale.set_scale(calibration_factor); //Adjust to this calibration factor
23     Serial.print("Reading: ");
24     Serial.print(scale.get_units(), 1); //Read load from HX711
25     Serial.print("g");
26     Serial.print(" calibration_factor: ");
27     Serial.print(calibration_factor); //print current factor
28     Serial.println();
29
30     if(Serial.available()) //checks for input
31     {
32         char temp = Serial.read();
33         if(temp == '+' || temp == 'a')
34             calibration_factor += 10;
35         else if(temp == '-' || temp == 'z')
36             calibration_factor -= 10;
37     }
38 }
```

Durch Eingabe von + und - in die serielle Schnittstelle wird der Faktor so lange angepasst bis der HX711 das zum kalibrieren platzierte Gewicht ausgibt.

Messungen

Für die Messungen muss das Programm wie folgt modifiziert werden.

```

1  #include <HX711.h>
2  #define LOADCELL_DOUT_PIN  2
3  #define LOADCELL_SCK_PIN  3
4
5  HX711 scale;
6
7  float calibration_factor = 433; //433 for 5kg full bridge load cell in grams
8
9  void setup() {
10     Serial.begin(9600);
11     scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
12     scale.set_scale(calibration_factor); //adjust to calibration factor
13     scale.tare(); //Reset the scale to 0
14
15 }
16
17 void loop() {
18
19     Serial.print(millis());
20     Serial.print("\t");
21     Serial.print(scale.get_units(), 1);
22     Serial.print("\n");
23
24 }

```

Genauigkeit

Nach Testen der Wägezelle mit bekannten Gewichten konnten Schwankungen im Bereich von 1-2g festgestellt werden.

Bei Testen mit höheren Gewichten konnten Abweichungen von bis zu 20g festgestellt werden.

Dies ist für die Messungen jedoch nicht relevant, da dies einem Fehler von weniger als 0.1% entspricht.

Die minimalen Schwankungen sind durch das unglaublich kleine Spannungsniveau der Wägezelle zu erklären.

Somit können Änderungen von wenigen mV, etwa durch Veränderung der Luftfeuchtigkeit bei nicht ganz abgedecktem DMS (Dehnungsmessstreifen) durch die Verstärkung des HX711 zu solchen Schwankungen führen.

Teil 2

Drehzahlmessung

Die Drehzahl kann mit folgendem Programm gemessen werden:

```

1  #include <Wire.h>
2  #define IR_AOUT_PIN A0
3  unsigned long cutime;
4  boolean state;
5  boolean prestate = false;
6  void setup() {
7    Serial.begin(9600);
8    pinMode(IR_AOUT_PIN, INPUT);
9  }
10
11 void loop() {
12
13   int in = analogRead(IR_AOUT_PIN);
14   if(in < 200) state = false;
15   else state = true;
16
17   if(state!=prestate){ //hat sich die Scheibe gedreht?
18     prestate = state;
19     Serial.println(millis());
20   }
21 }

```

Der digitale Ausgang wird nicht verwendet, da dieser bei Versuchen als fehlerhaft empfunden wurde. Alternativ wird hier das Übertreten der Schwelle per Software überprüft.

Kombination

Die Kombination beider Programme erfolgte über das parallele Auslesen von zwei Schnittstellen.

Somit können beide der vorhergehenden Programme wiederverwendet werden.

Um das zeitgleiche Öffnen zweier Schnittstellen zu ermöglichen wurde der Installationsordner der Arduino IDE kopiert und in beiden Ordnern die Datei arduino.exe ausgeführt.

Folglich mussten nur die millis() Variablen synchronisiert werden.

Dies erfolgt anhand des Zeitpunktes zu welchem der Motor die niedrigste Drehzahl hat, da hier zeitgleich auch das Lastmoment am höchsten sein muss.

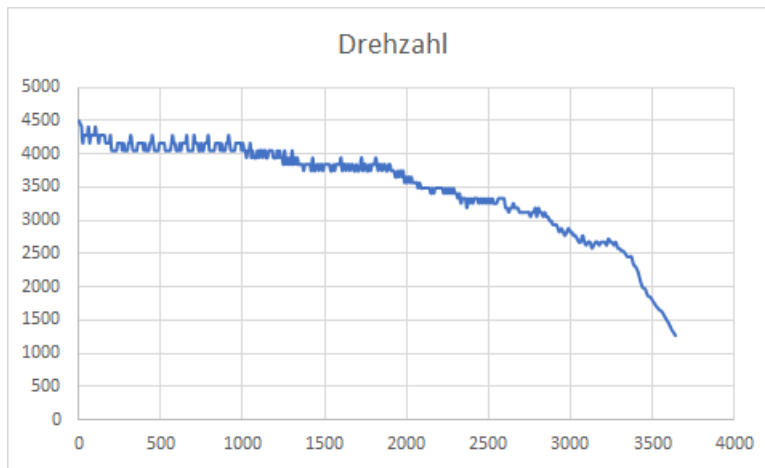
Messung

Die Messung verlief bis auf leichte Schwankungen bei der Drehzahlmessung problemlos.

Diese könnten durch das parallele Verwenden von zwei IR-Sensoren und folglich mitteln der Daten verringert werden.

Auswerten der Messdaten

Nach dem Auswerten der Messdaten konnten folgende Grafiken ermittelt werden.

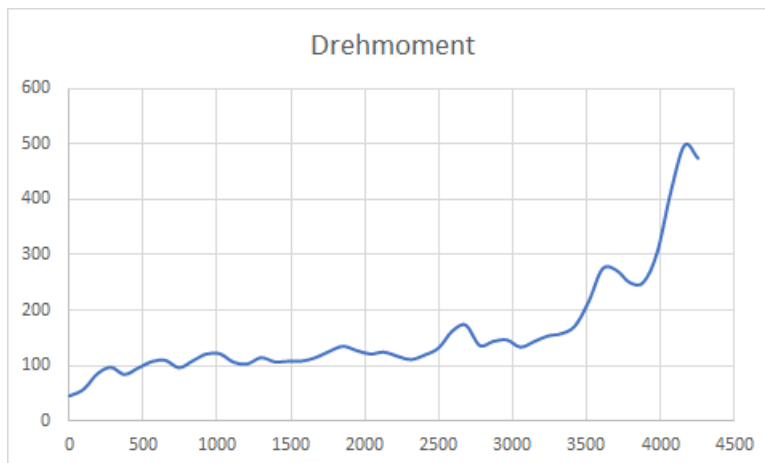


Das leichte Schwanken der Drehzahl ist durch Ungenauigkeiten bei der Messung dieser zu erklären, jedoch kann auch ein ungleichmäßiger Bremsvorgang Grund hierfür sein.

Ansonsten kann eine klare Abnahme der Drehzahl erkannt werden, welche jedoch nie 0 erreicht.

Dies ist damit zu erklären, dass um die Linie zu glätten immer die Drehzahl über mehrere Messpunkte ermittelt wurde.

Würde nun die das Stehenbleiben auch einberechnet werden, hätte dies aufgrund der Natur des verwendeten Codes eine Verfälschung der Messwerte zur Folge.



Hier ist klar ein Anstieg des Drehmoments mit der Zeit zu erkennen.

Die Schwankungen erhärten den Verdacht auf einen ungleichmäßigen Verlauf des Bremsvorganges.

Interpretation

Durch Betrachten beider vorhergehenden Grafiken kann festgestellt werden, dass mit abnehmender Drehzahl eine Zunahme des Drehmoments geschieht.

Dies entspricht dem Fakt, dass Elektromotoren bei niedriger Drehzahl am leistungsfähigsten sind.