

Inbetriebnahme und Modellierung einer Asynchronmaschine

Lukas Tetzlaff, Robby Kozok, Nic Fränky Siebenborn und Pascal Kahlert

Abstract—Dieser Anwendungshinweis beschäftigt sich mit der einfachen Inbetriebnahme einer Asynchronmaschine vom Typ MDFMAIG071-12D von Lenze. Außerdem wird in diesem Dokument die Benutzung eines passenden Modells behandelt.

Index Terms—IEEE, IEEEtran, journal, L^AT_EX, paper, template.

I. INTRODUCTION

THIS

Einleitung schreiben

IEEEtran.cls version 1.8b and later. I wish you the best of success.

Nic Siebenborn, Robby Kozok, Lukas Tetzlaff, Pascal Kahlert

17 Februar, 2016

II. MODELLIERUNG DER ASYNCHRONMASCHINE

Dieser Abschnitt befasst sich mit der Benutzung der beigefügten Simulation. Es soll beschrieben werden welche Parameter angepasst werden können. Es sollen weiterhin verschiedene Simulationen dargestellt werden um darzustellen wie sich ein richtig eingestellter Motor verhalten sollte. Die Modellierung wurde in drei Abschnitte unterteilt.

- Feldorientierte Regelung
- Transformation
- Motormodell

A. Konfiguration des Motormodells

Dieses Modell kann grundsätzlich an jeden Asynchronmotor angepasst werden. Es muss dabei nur beachtet werden, dass alle Parameter im *Model Explorer* richtig eingestellt werden. Es folgt nun eine Auflistung von Motorparametern, welche bekannt sein müssen.

Bezeichnung	Beschreibung	Model Explorer
Elektrische Kenngrößen		
I_{μ}^{+}	Magnetisierungsstrom	I_magnet
Ψ_R^{+}	Läuferfluss	PsiR_plus
L_R^{+}	Läuferinduktivität	I_magnet
L_{σ}^{+}	Streuinduktivität	Lsigma_plus
R_R^{+}	Läuferwiderstand	RR_plus
R_S	Ständerwiderstand	Rs
I_A	Start-Strom	StartStrom
M_A	Start-Moment	StartMoment
p	Polpaaranzahl	p
Mechanische Kenngrößen		
μ	Reibungskoeffizient	Rreib
J	Trägheitsmoment	J

Diese Parameter müssen in den *Model Explorer* eingetragen werden. Fehlt einer oder mehrere Parameter kann es passieren, dass das Model kaum oder garnicht funktioniert, also sich nicht wie erwartet verhält.

Um nun eine Simulation sinnvoll durchzuführen, müssen noch Sollvorgaben und Lastsituationen eingetragen werden. Dafür können die die Blöcke Drezahlvorgabe und Lastvorgabe genutzt werden. Diese Blöcke verwenden die Parameter MLast und wSoll, diese sollten ebenfalls an realistische Last-Situationen angepasst werden.

1) *Bestimmung von dem Reibungskoeffizienten μ* : Sollte μ nicht bekannt sein, lässt sich dieser leicht über die Simulation ermitteln. Dafür sind folgende Schritte notwendig.

a) *Anpassung der Simulation*: Im ersten Schritt muss der Momentregler vom Rest der Simulation getrennt werden und durch das konstante Nennmoment ersetzt werden, dies ist in fig:reibComp zu sehen.

b) *Bestimmung des Parameters*: Nun muss μ solange verändert werden, bis die Drehzahl gleich der Nenndrehzahl wird. Dabei müssen alle MLast gleich null sein. Dies ist im *Model Explorer* einzustellen. In fig:reibChart ist Beispielhaft zu erkennen, wie sich die Drehzahl der Nenndrehzahl nähert.

B. Simulation der Asynchronmaschine

Im folgenden sollen verschiedene Simulations-Szenarien dargestellt und erläutert werden.

1) *Sollwertvorgabe - ohne Last*: Dieses Szenario kann genutzt werden um die grundsätzliche Funktion des Modell sicherzustellen.

a) *Einstellung der Parameter*: Im ersten Schritt müssen alle MLast Parameter auf 0 gesetzt werden(siehe II-A1b). Außerdem sollten sinnvolle WSoll Vorgaben gemacht werden, so könnte zum Beispiel folgender Ablauf gewählt werden:

Zeitpunkt t	Solldrehzahl n_{Soll} in n/min
1s	500
6s	1370

b) *Durchführung der Simulation*: Im nächsten Schritt sollte eine sinnvolle Zeit für die Simulation gewählt werden. In diesem Beispiel haben wir 10s gewählt.

c) *Analyse der Ergebnisse*: In fig:KeineLastMoment ist zu sehen, dass der Motor auf Vorgabeänderungen der Drehzahl mit starken Schwankungen des Drehmomentes reagiert. Bei einer Erhöhung der Drehzahl wird der Motor versuchen das Drehmoment innerhalb seiner Grenzen(Strom, Spannung, etc.) zu erhöhen um die gewünschte Drehzahl zu erreichen. Bei einer Überschreitung wird er ruckartig das Moment verringern. Bei einer Verringerung der Drehzahl reagiert der Motor damit, das Moment zu verringern, dort ist zu sehen wie der Moment des Motors negativ wird, da dem Motor fast keine Energie mehr zugeführt wird und damit auch kein Antrieb im Motor steckt. In fig:KeineLastDrehzahl sind keine Besonderheiten zu erkennen, der Motor versucht nach seinen Möglichkeiten der Vorgabe zu folgen und schafft dies in einer relativ kurzen Zeit.

2) *Sollwertvorgabe - mit Last*: Diese Simulation zeigt das Lastverhalten des Motors. Es soll gezeigt werden, dass dieser bei seinem angegebenen Haltemoment stehen bleibt und bei zu großer Last sogar rückwärts dreht.

a) *Einstellung der Parameter*: Für diese Simulation sollten sinnvolle Werte gewählt werden. In unserem Fall wurden die Werte so gewählt, dass im ersten Zeitfenster(1s-6s) zu sehen ist wie die Maschine durch eine Last gehalten wird und im zweiten Zeitfenster(6s-10s) zu sehen ist wie die Last die Maschine dreht.

Zeitpunkt t	n_{Soll} in n/min	M in Nm
1s	1370	3,8
6s	500	10

b) *Analyse der Ergebnisse*: In fig>LastMoment is zu sehen, dass das Drehmoment des Motors und der Drehmoment der Last sich angleichen, da der Motor versucht die eingestellte Drehzahl zu erreichen. Ab Sekunde 6 ist zu sehen, dass das Drehmoment des Motors deutlich unter dem der Last ist. Auch wenn der Motor im ersten Moment versucht, gegen die Last anzukommen bringt ihn dann die simulierte Begrenzung des Stromes wieder auf sein Haltemoment. Einen passenden Verlauf können wir in fig>LastDrehzahl sehen. Im Zeitraum von 1s - 6s ist zu sehen, wie der Motor nahezu steht. Sobald

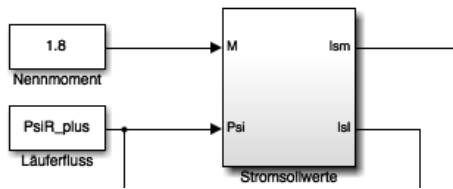


Fig. 1. Aufbau der Simulation zur Bestimmung des Reibkoeffizient

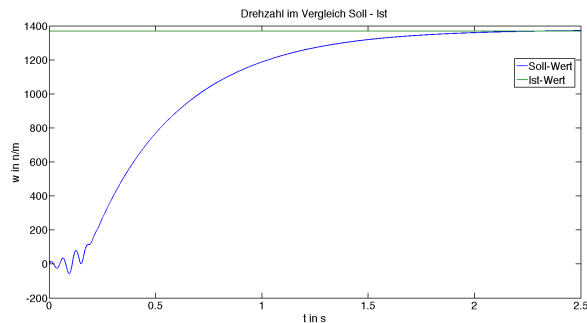


Fig. 2. Zeitlicher Verlauf der Drehzahl im Vergleich zur Nenndrehzahl(hier Solldrehzahl)

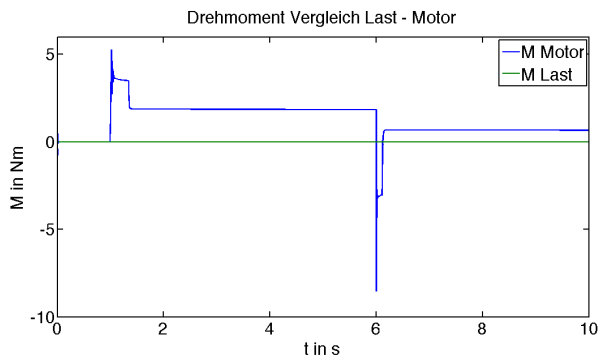


Fig. 3. Zeitlicher Verlauf des Drehmomentes

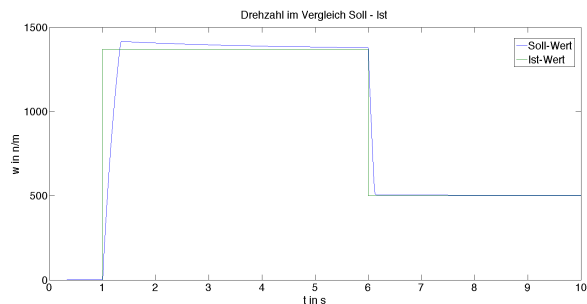


Fig. 4. Zeitlicher Verlauf der Drehzahl

dann ab Sekunde 6 das Moment der Last größer als sein Haltemoment wird, fängt der Motor an, sich rückwärts zu drehen.

C. Schwächen des Modells

Dieses Modell stellt keine reale Abbildung des Motors dar, kann aber einen Eindruck vermitteln, welches Verhalten von diesem Asynchronmotor zu erwarten ist.

Vor allem in der Begrenzung von Strom und Spannung ist deutliches Verbesserungspotential zu sehen.

So wird die Spannung im gesamten Modell nicht begrenzt und steigt teilweise auf bis zu 1500V, dies ist mit einer Versorgungsspannung von 230V-AC aber nicht möglich. Änderungen dieser Parameter können an den Reglerbausteinen vorgenommen werden, dies führt allerdings zu einer Neueinstellung der Parameter.

Es ist zu erwarten, dass der Motor nicht ganz so schnell reagieren kann.

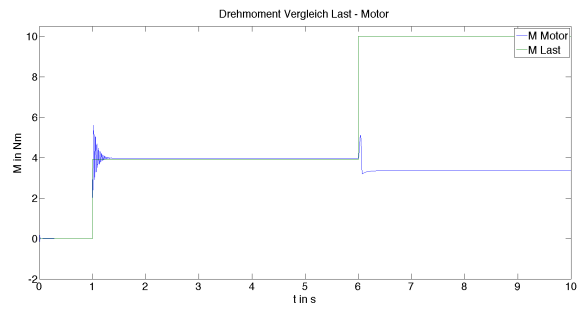


Fig. 5. Zeitlicher Verlauf des Drehmomentes

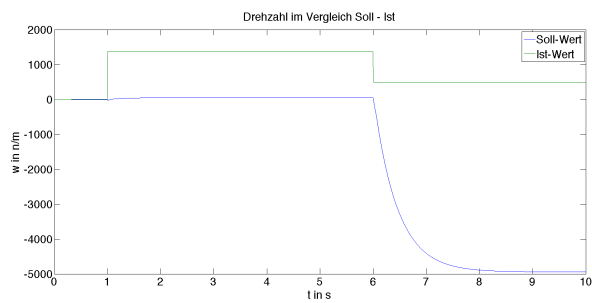


Fig. 6. Zeitlicher Verlauf der Drehzahl

III. INBETRIEBNAHME DER ASYNCHRONMASCHINE

Robbys

Teil
fi-
nal-
isieren
-
zB
Ver-
strip-
pung
evtl
um-
be-
nen-
nen

A. Verkabelung

Da der in diesem Projekt verwendete Motor, schon einmal von einer anderen Gruppe in Betrieb genommen wurde, mussten wir lediglich die schon verdrahteten Anschlusskabel, anschließen. Folgende Verbindungen müssen gesteckt werden:

- Die Leitungen U,V und W der Asynchronmaschine werden an das TI Kit geklemmt
- Die Leitungen des Drehimpulsgebers werden an den Anschluss "QEP Connector" angeschlossen.
- Die 12V Spannungsversorgung wird an das TI Kit angeschlossen
- Das USB Kabel ist an das TI Kit und den PC anzuschliessen
- Die Kaltgerätestecker des TI Kits und des Lüfters sind zu stecken

B. Software-Realisierung

1) *erster Ansatz*: - Matlab importieren - gescheitert an elektr. Modell und dependency-Problemen - wie in Simulink feldorientiert regeln mit Drehzahl als input und Ausgabe der sechs PWM-Duty-Cycles die die Mosfets beschalten

2) *tatsliche Realisierung*: - Inkrementieren einer Zvariable - Nutzen der Zvariable als Index für ein vorgeneriertes Array mit Sinuswerten - diese Sinuswerte werden jeweils um ein Drittel verschoben in Form von Duty-Cycles auf die sechs PWMs gegeben, bei denen jeweils zwei zu Schalter-Paaren zusammengefasst und sich zueinander entgegengesetzt verhalten - die Duty-Cycle-Werte sind Dezimal bezogen auf eine festgelegte Periode von 2000 ausgerechnet - Beim Ertragen der Werte wird so der Reload-Wert des jeweiligen ePWM-Registers gesetzt, der damit die Zeit für beide logischen Zustände festlegt - unmodifiziert wird das der Betrieb auf fester Drehzahl mit Stellreserve (nicht bermoduliert)

3) *Hinweise zum CCS-Projekt*: TI's Fehler ausbaden

Das Makefile des Demoprogramms nutzt teilweise fehlerhaft (s. <http://e2e.ti.com/support/microcontrollers/c2000/f/171/p/326227/1136098>) absolute Pfade, zum Beheben haben wir einen symbolischen Link auf das "angebliche" Verzeichnis gesetzt:

“cmd mklink /D "C:\2_DMCRev" "C:\controlSUITE"“

09:28] 7b kann mir bitte jemand sagen was ich in der doku beschreiben soll? wie immer wrde ich fr ecuh alles tun :heart: [09:36] lt Du knntest unsere Messungen und Signale vom letzten Stand da beschreiben.. Also was man da sieht, modulationsgrad, wie verhalten die sich zueinander etc [09:36] Ich schreib so: wie wollten wir erst implementieren, warum ging das nicht, was haben wir stattdessen gemacht [09:37] 7b jawohl [09:38] wir haben doch keinen modulationsgrad mschen [09:41] lt Kp gib iwelche ablesbaren Parameter an [09:51] 7b hat jemand bock nochmal in die beuth zu hucken alles anzuschlien und die phasenverschiebung zwischen strom und spannung an einer leitung zu messen? [11:50] lt :upside_{down}face :: -1 : [15 : 14]robby1990Ne[15 : 21]pkahlertihstrom[15 : 21]7bDannwohlnicht

P = 0.5 ; 2408 ; 1451

P Wert ab jetzt negativ P = 0.05; 2373 ; 1447

Faktor laser berechnung 2,89

REFERENCES

[1]