



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Eidgenössisches Departement für
Umwelt, Verkehr, Energie und Kommunikation UVEK
Bundesamt für Energie BFE

OPAL-ERWEITERUNG FÜR PERMANENT-MAGNET-MOTOREN

Jahresbericht 2007

Autor und Koautoren	Dr. Ronald Tanner
beauftragte Institution	SEMAFOR Informatik & Energie AG
Adresse	Sperrstrasse 104 B, CH-4057 Basel
Telefon, E-mail, Internetadresse	061 690 9088, tar@semafor.ch
BFE Projekt-/Vertrag-Nummer	102128 / 152677
BFE-Projektleiter	Roland Brüniger
Dauer des Projekts (von – bis)	01.08.2007 - 31.03.2008
Datum	3. Dezember 2007

ZUSAMMENFASSUNG

Das bestehende Programmsystem OPAL für die energieoptimale Auslegung von Pumpen und Lüftersystemen mit ASM-Normmotoren, wurde mit einem Modul für Permanentmagnetmotoren ergänzt, welches es ermöglicht, Einsparungsvergleiche mit diesen beiden Motortypen durchzuführen. Um diese Möglichkeit effektiv zu nutzen, werden jedoch Motordaten benötigt, deren Verfügbarkeit zur Zeit noch unklar ist.

Um die Anwendung einem möglichst breiten Benutzerkreis zugänglich zu machen, ist darüberhinaus vorgesehen, die Software mit einer webfähigen Benutzerschnittstelle auszustatten, die die neuen Möglichkeiten der Web-2.0-Technologie nutzt. Hierfür wurde mit der Evaluation geeigneter Web-Frameworks begonnen.

Projektziele

Mit diesem Projekt soll das bestehende Programmsystem OPAL zur energieoptimalen Auslegung von Pumpen und Lüftersystemen mit ASM-Normmotoren, die bei gegebenem Lastspiel mit konstanter und variabler Drehzahl betrieben werden, mit einem Modul für Permanentmagnetmotoren ergänzt werden, so dass Einsparungsvergleiche durchgeführt werden können.

Darüberhinaus soll die Benutzbarkeit der Software weiter verbessert werden, um so auch den Benutzerkreis zu erweitern. Dazu ist vorgesehen, die Oberfläche webfähig zu machen. Dies ermöglicht es dem Nutzer auf eine Installation der Software auf seinem lokalen Rechner zu verzichten und mit einem gängigen Internet-Browser (Internet Explorer, Mozilla Firefox, ...) darauf zu zugreifen. Zudem kann die Software auch jederzeit auf dem aktuellsten Stand gehalten werden.

Ein weiteres Ziel ist die Unterstützung mehrsprachiger Benutzerführung und -dokumentation (mind. in Deutsch und Französisch).

Durchgeführte Arbeiten und erreichte Ergebnisse

Das Projekt ist in die folgenden Phasen gegliedert:

- Erstellung der Systemspezifikationen
- Entwurf und Implementation
- Test, Verifizierung, Dokumentation

Da sich in wesentlichen Teilen die in OPAL vorhandene Funktionalität als praxistauglich erwiesen hat, ging es bei der Systemspezifikation hauptsächlich darum, sich einen Überblick über den Stand der Technik von Permanentmagnet-Motoren zu verschaffen und sich für ein für die Evaluation der Energieeffizienz geeignetes Berechnungsmodell entscheiden. Dazu wurden mit verschiedenen Anbietern Kontakt aufgenommen (Circle Motor, ABB und Siemens).

Für die Integration des PM-Motor-Moduls wurden an der Softwarestruktur von OPAL die nötigen Anpassungen entworfen und implementiert. Da jedoch keine Motordaten im benötigten Umfang verfügbar sind, ist ein Einsatz dieses Moduls noch nicht sinnvoll.

Begonnen wurde parallel dazu auch mit einer Evaluation von Web-Frameworks für die möglichst einfache Implementierung einer benutzerfreundlichen Web-Schnittstelle. Dieser Teil ist noch in Arbeit.

Nationale Zusammenarbeit

Wie der Titel schon sagt, soll die Zusammenarbeit mit Forschungsinstitutionen (FH, Uni, ETH), der Privatwirtschaft und Förderinstitutionen (SNF, KTI, private Fonds, etc.) beurteilt werden.

1. Verfügbarkeit von Permanentmotor-Daten

Die Abklärung zur Verfügbarkeit von PM-Motordaten ergibt folgendes Bild:

- CircleMotor (Markus Lindegger):
 - verschiedene Simulationen mit Motoren der Leistungen 5.5, 15, 55 und 90 kW im Rahmen einer BFE-Studie. Die Daten liegen im Maxwell-Format vor.
 - PM IEC 56: einphasig, 230V, 150W, 3600 1/min
 - PM IEC 100: dreiphasig, 400V, 3kW, 3000 1/min
- ABB: DriveIT M3BP (Baugrößen 280 - 560, Nennmomente 1257 - 50156 Nm, Polzahlen 10 - 12)
 - 17 - 200 kW, 0 - 220 1/min
 - 25 - 280 kW, 0 - 300 1/min
 - 38 - 400 kW, 0 - 430 1/min
 - 57 - 500 kW, 0 - 600 1/min

Von der ABB-Web-Seite kann die PDF-Datei "DriveIT Permanent Magnet Motors" heruntergeladen werden. Darin sind die Daten Leistung, Strom, Wirkungsgrad, Polpaarzahl, Drehmoment, Leistungsfaktor, Trägheitsmoment, Maximaldrehzahl, Gewicht (Masse) aufgeführt.

- Siemens: SIEMOSYN 1FU8 (Baugrößen 71M bis 160L)

Polzahl	Drehmoment [Nm]
2	1.3 - 7.3
4	2 - 18
6	34 - 60

Im Online-Katalog sind zu verschiedenen Nennspannungen die Maximalfrequenz sowie jeweils die Nennmomente, Nennleistung, Nenn- und Anlaufströme bei 50 Hz und bei Maximalfrequenz angegeben.

Fazit: ein direkter Vergleich von ASM-Normmotoren mit PM-Motoren ist zur Zeit nicht möglich.

2. Modellierung des Permanentmagnet-Motors

Permanentmagnet-Motoren können in unterschiedlichen Varianten gebaut werden. Hier wird nur die Variante "permanentmagnet-erregter Synchronmotor" (PM-SM) betrachtet. Der Motor hat eine Drehfeldwicklung im Stator und Permanentmagnete im Rotor.

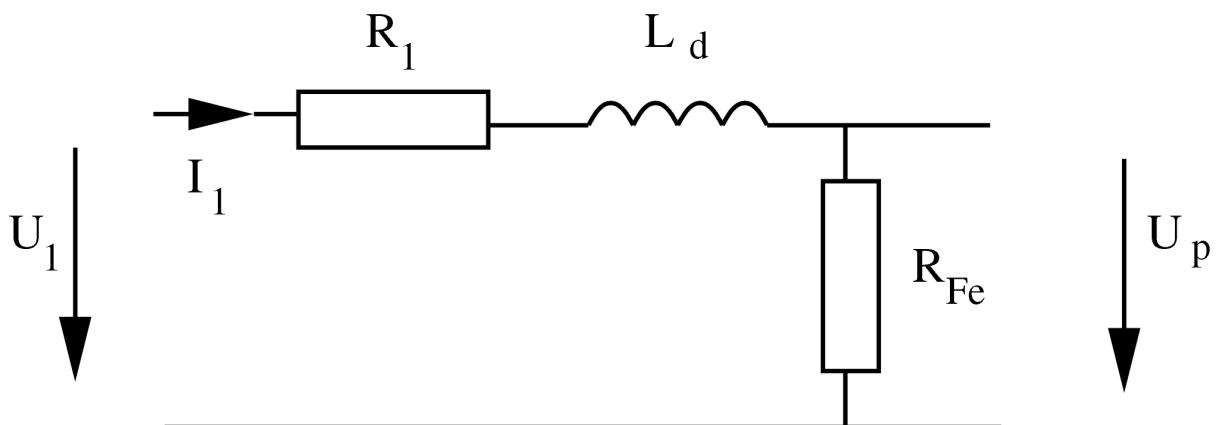


Abbildung 1: Ersatzschaltbild des Permanentmagnet-Motors, mit Eisen- und Kupferverlusten

- Die Spannungsgleichung lautet wie folgt:

$$U_1 = (R_1 + j\omega L_d) \cdot I_1 + (\sin \delta + j \cos \delta) \cdot U_p \quad (1)$$

mit dem Polradwinkel δ

- Das Moment lässt sich aus der Leistungsbilanz bestimmen:

$$\omega_1 \cdot p \cdot M = 3 \cdot U_1 \cdot \operatorname{Re}[I_1] - 3 \cdot I_1 \cdot I_1^* \cdot R_1(\theta) - 3 \cdot U_p^2 / R_{fe} \quad (2)$$

Bei gegebener Spannung, Frequenz und Moment M lässt sich aus diesen beiden Gleichungen der komplexe Strom und der Polradwinkel δ bestimmen.

Der Wirkungsgrad berechnet sich wie folgt:

$$\eta = 1 - \frac{\omega_1 \cdot p \cdot M}{3 \cdot U_1 \cdot \operatorname{Re}[I_1]} \quad (3)$$

3. Evaluation eines Web-Frameworks

Bis vor kurzem zeichneten sich die webbasierten Applikationen nicht durch eine hohe Benutzerfreundlichkeit aus. Viel zu träge war ihr Reaktionsverhalten auf Benutzereingaben, viel zu stark eingeschränkt die verwendbaren Schnittstellenelemente. Infolge technologischer Fortschritte, zu der die sogenannte Ajax-Technologie einen wesentlichen Beitrag liefert, ist es heute jedoch möglich Web-Applikationen zu entwickeln, die sich bezüglich Bedienkomfort kaum mehr von einer Desktop-Variante unterscheiden lassen. Diese Entwicklung wird nebst der Microsoft-ASP.Net-Technologie ganz besonders auch von zahlreichen Open-Source-Paketen erheblich gefördert und beeinflusst.

Aus Gründen der Plattformunabhängigkeit und der Integrationsmöglichkeit interessieren vor allem Frameworks, die auf den Sprachen Java, Python, C++ und Ruby basieren und eine Ajax-Integration anbieten. Wichtige Vertreter sind:

JBoss Seam	www.jboss.com/products/seam
Grails/Groovy	grails.codehaus.org
Ruby/JRuby on Rails	www.rubyonrails.org
Google Web Toolkit	code.google.com/webtoolkit
Django	www.djangoproject.com
Pylons	pylonshq.com
Tntnet	www.tntnet.org
Click	click.sourceforge.net

Die wesentlichen Kriterien anhand darer diese Vertreter verglichen werden sollen, sind:

- Qualität und Umfang der Dokumentation
- Möglichkeit zur Einbindung von C++-Programmen
- Schnittstellenelemente und Features: Tabellen, 2D-Plots, Drag&Drop
- Community, Entwicklungs- und Wartungsaufwand

Um sich jedoch ein genaueres Bild von den Vor- und Nachteilen der einzelnen Vertreter machen zu können, muss eine kleine Beispielapplikation erstellt werden.

Die Frameworks werden im Folgenden kurz charakterisiert.

3.1. JBoss Seam

JBoss Seam ist ein von JBoss entwickeltes Webframework für Java EE, welches das Zusammenspiel zwischen der Business-Logik und der Präsentation vereinfacht. Die Business-Logik läuft dabei in Regel in einem EJB 3.0 Container, während die Präsentationsschicht mit JavaServer Faces (JSF) realisiert wird. Es ist jedoch auch möglich, reine POJOs (Plain Old Java Objects) anstelle der EJBs zu verwenden.

Durch den Einsatz von Annotations und der Registrierung aller Komponente im Seam-Kontext, ist es einerseits möglich auf die umfangreichen XML-Konfigurationsdateien zu verzichten und anderseits die EJB-Komponenten direkt in der Präsentation zu referenzieren. Motiviert ist dieses Vorgehen durch das Prinzip sich bei der Programmierung und der Konfiguration möglichst wenig wiederholen zu müssen (vgl. Don't repeat yourself).

Bestandteil von JBoss Seam ist das Hilfsprogramm seam-gen mit welchem es unter anderem möglich ist eine CRUD1-Applikation von einer bestehenden Datenbank zu erzeugen. In Kombination mit ICEfaces oder RichFaces kann eine JBoss Seam Applikation mit Ajax-Funktionalität ausgestattet werden.

JBoss Seam erlaubt durch die Reduktion der Konfigurationsdateien eine schnelle Entwicklung von Webapplikationen unter Verwendung der skalierbaren und standardisierten Java EE Architektur.

3.2. Ruby/JRuby on Rails

Ruby on Rails, abgekürzt Rails, ist ein von David Heinemeier Hansson in Ruby geschriebenes und quelloffenes Web-Framework, das im Juli 2004 zum ersten Mal der Öffentlichkeit vorgestellt wurde. Es basiert auf dem Prinzip "Don't Repeat Yourself" (DRY) und stellt Programmierkonventionen über die Anwendungskonfiguration. Durch seinen Aufbau ermöglicht es rasche Umsetzung von Anforderungen und unterstützt damit agile Softwareentwicklung.

Rails folgt der Model View Controller-Architektur (MVC) und besteht aus fünf Modulen:

- Active Support: Ruby-Erweiterungen von Rails
- Active Record: Objektabstraktionsschicht basierend auf dem objekt-relationalen Muster oder Model (das "M" aus MVC) von Martin Fowler

- Action Pack: Request-Behandlung und Response-Ausgabe. Die Anfragen werden durch eine öffentliche Methode des Controllers (Action Controller, das "C" aus MVC) behandelt. Die Ausgabe wird mittels eines Templates (Action View, das "V" aus MVC) vorgenommen.
- Action Mailer: E-Mail-Versand und -Empfang
- Action Web Service: Web-Service-Programmierung, Unterstützung für SOAP, XML-RPC und REST.

Rails bietet auch Methoden zur einfachen Entwicklung von Ajax-Applikationen.

3.3. Google Web Toolkit

Das **Google Web Toolkit (GWT)** ist ein Framework zur Entwicklung von Webanwendungen. Es wurde am 17. Mai 2006 von Google veröffentlicht. Seine Besonderheit ist ein Java-nach-Javascript-Compiler, so dass nahezu die gesamte Entwicklung von Client und Server auf Basis von Java realisiert werden kann. Weiterhin ist das GWT mit einem XML-Parser, Internationalisierungs-Unterstützung, einer Schnittstelle für Remote Procedure Calls, Integration von JUnit, und einem reichhaltigen Widget-Paket zur Gestaltung der graphischen Oberfläche (GUI) ausgestattet. Diese kann dabei ähnlich wie mit Swing erstellt werden.

Was das GWT im Wesentlichen von anderen Frameworks dieser Art unterscheidet ist die Tatsache, dass auch der client-seitige Code komplett in Java erstellt werden kann, was erhebliche Vorteile in der Entwicklung mit sich bringt, da bewährte Entwicklungsumgebungen benutzt werden können.

3.4. Django

Django ist ein in Python geschriebenes quelloffenes Web-Framework, welches dem Model-View-Controller-Schema folgt. Es wurde ursprünglich entwickelt, um die News-Seite "Lawrence Journal-World" zu verwalten, und wurde im Juli 2005 unter einer BSD-Lizenz veröffentlicht. Benannt ist es nach dem Jazz-Gitarristen Django Reinhardt.

3.5. Pylons

Pylons ist ein Open-Source Web-Framework auf der Basis von Python. Es macht ausgiebig Gebrauch des WSGI-Standards. Durch die Gliederung der Funktionalität in eigenständige Module will es die Wiederverwendbarkeit und Wartbarkeit fördern.

Pylons ist der aktuellste Vertreter der Web-Applikations-Frameworks, Es ist stark beeinflusst von Ruby on Rails: zwei seiner Hauptkomponenten, Routes und WebHelpers, sind Python-basierte Re-Implementationen von Rails-Komponenten.

3.6. Tntnet

Tntnet ist ein in C++ geschriebener Webserver welcher andere C++-Programme zur Bearbeitung von Anfragen als Modul einbinden kann. Damit lassen sich Webapplikationen in C++ entwickeln. Die entsprechenden Schnittstellen werden von tntnet bereitgestellt, etwa für den Import der per GET oder POST übergebenen Daten, die Kommunikation über das Netzwerksocket und die Parallelverarbeitung der eintreffenden Anfragen.

Durch die Verwendung der Programmiersprache C++ können moderne Programmietechniken wie Objektorientierung, Vererbung und Polymorphismus mit der Geschwindigkeit von nativ im Maschinencode laufenden Programmen kombiniert werden. Im Gegensatz zu Skriptsprachen, die für Webapplikationen sehr häufig eingesetzt werden, wird der Programmcode von kompilierten Programmen nicht zur Laufzeit übersetzt, was deren Ausführung beschleunigt.

3.7. Click

Click ist ein einfaches Java EE-Web-Applikations-Framework für Java. Click ist ein Open-Source-Projekt basierend auf einer Apache-Lizenz.

Click verwendet ein Event-gesteuertes Programmiermodell zur Verarbeitung der Browser-Requests und Velocity für das Rendering Ausgabe.

Das Framework stützt sich auf ein einzelnes Servlet, ClickServlet genannt, ab, welches die Browser-Anfragen entgegennimmt. Es erzeugt jeweils ein Page-Objekt, welches anschliessend von Velocity in HTML-Format konvertiert und zum Browser zurückgeschickt wird.

Bewertung und Ausblick

Die für 2007 geplanten Arbeiten konnten in wesentlichen Teilen abgeschlossen werden. Die Bewertung und Evaluation der Web-Frameworks soll bis Ende Dezember abgeschlossen werden, so dass im Januar 2007 mit der Umsetzung begonnen werden kann. Ein nützlicher Kontakt konnte vor kurzem mit Thomas Heldstab von der Firma Hematik geschaffen werden. Er befasst sich im Auftrag von S.A.F.E mit der Evaluation von Software für energie-effiziente Antriebe. Seine bis Ende Jahr zusammengetragenen Erkenntnisse werden sicherlich noch in die Detailspezifizierung der OPAL-Erweiterungen einfließen können.

Ein weiterer wichtiger Input wird vom Topmotors-Workshop erwartet, der am 11. Januar 2008 in Zürich stattfinden wird.