

# Bauanleitung eines Raumenergie-Konverters nach realitätsnaher Berechnung DFEM-Berechnung

von Claus W. Turtur

Wolfenbüttel, den 3. April 2011

Ostfalia Hochschule für Angewandte Wissenschaften, Braunschweig-Wolfenbüttel  
Salzdahlumer Straße 46 – 48 38302 Wolfenbüttel Germany  
Email: c-w.turtur@ostfalia.de Tel.: (+49) 5331 / 939 – 42220  
Internet-Seite: <http://www.ostfalia.de/cms/de/pws/turtur/FundE>

(veröffentlicht in PHILICA.COM, ISSN 1751-3030, Article no. 233)

PACS numbers: 88.05.Ec, 88.05.Gh, 88.05.Np, 88.90.+t

## Zusammenfassung

In [Tur 11] wurde die Theorie eines leistungsstarken Raumenergie-Konverters entwickelt, und mit der Dynamischen Finite-Elemente-Methode (DFEM) simuliert. Es ergibt sich eine Maschine, die aus theoretischer Sicht für technische Anwendungen geeignet ist.

Aufgrund der Vielzahl von Anfragen zum praktischen Aufbau einer solchen Maschine im Experiment, hat sich der Autor entschlossen, mit Hilfe seines DFEM-Rechenverfahrens einen realitätsnahen konkreten Raumenergie-Motor auf dem Computer zu simulieren, und diesen mit praktischen Erläuterungen zum Aufbau der Allgemeinheit zur Verfügung zu stellen.

Die theoretischen Hintergründe der Berechnung werden eingehend erläutert, so dass jeder Fachkollege in der Lage sein sollte, in den im Anhang dargestellten Algorithmus eine geometrische Anordnung eines Raumenergie-Magnetmotors nach eigenen Vorstellungen einzugeben und zu berechnen.

## Inhaltsverzeichnis

1. Physikalische Vorarbeit und Grundlagen
2. Bewegungen der Komponenten des Raumenergie-Konverters
3. Auswertung der Rechenergebnisse am Raumenergie-Konverter
4. Rechenbeispiel eines konkreten Raumenergie-Motors
5. Ein konkreter EMDR-Raumenergie-Konverter
6. Der EMDR-Konverter mit mechanischer Leistungsentnahme
7. Praktische Hinweise, Anleitung zum Aufbau eines EMDR-Konverters
8. Resumée
9. Literatur-Hinweise
10. Anhang: Quell-Code des DFEM-Algorithmus

### 1. Physikalische Vorarbeit und Grundlagen

Die DFEM-Berechnungsmethode, auf der die hier vorliegenden Berechnungen basieren, ist ein allgemeines Rechenverfahren zur Berechnung von Raumenergie-Motoren die sich im speziellen sehr gut dafür eignet, elektrische und magnetische Raumenergie-Konverter zu simulieren. Gezielt für die Raumenergie-Konverter der letztgenannten Art werden in der hier vorliegenden Arbeit die Grundlagen, ein Berechnungsverfahren und ein konkretes Beispiel bis hin zur Erstellung einer praktischen Bauanleitung vorgestellt. Grundlage des DFEM-Berechnungsverfahrens ist es, die

Bewegung der einzelnen Komponenten (mechanisch bewegte Komponenten ebenso, wie elektrische und magnetische Felder, die sich im Inneren des Raumenergie-Motors bewegen) auf fundamentale physikalische Größen zurückzuführen, deren Verhalten dann in Differentialgleichungen simuliert wird.

Um die Einsatzmöglichkeiten des hier vorgestellten Rechenverfahrens so allgemein wie möglich zu gestalten, ist es vorgesehen, eine beliebige Anzahl von Spulen ebenso wie eine beliebige Anzahl von Dauermagneten zu emulieren. Desweiteren ist natürlich auch eine Wechselwirkung mit der Außenwelt erforderlich, alleine schon, um die aus Raumenergie gewandelte Leistung entnehmen zu können. Dabei werden z.B. Spulen über externe elektrische Schaltkreise beschaltet, die dann ebenfalls über Differentialgleichungen simuliert werden. Dauermagneten werden durch Stromverteilungen in Stromschleifen emuliert, wodurch die Möglichkeit geschaffen wird, die auf die Dauermagneten wirkenden Lorentzkräfte sinnvoll zu berechnen.

Um diese abstrakte Erläuterung für alle Leser und Leserinnen greifbar und begreifbar zu machen, wurde die Rechenmethode an einem konkreten Beispiel vorgeführt, welches in einem Computerprogramm umgesetzt wurde, dessen Quellcode im Anhang der vorliegenden Publikation zu finden ist. Für das genannte konkrete Beispiel sind zwei Spulen und ein Dauermagnet vorgesehen, so wie sie in der Darstellung von Abb.1 modelliert sind.

Anzumerken sei aber noch, dass sich mit den DFEM-Verfahren auch beliebige andere Raumenergie-Konverter simulieren lassen, so dass man auch bereits bestehende Konverter aus der Theorie heraus nachrechnen kann.

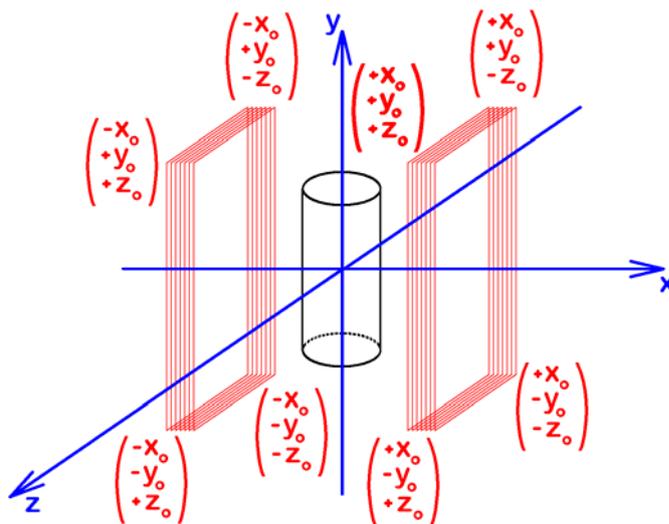


Abb.1:

In einem dreidimensionalen kartesischen Koordinatensystem (blaue Farbe) sind zwei Spulen parallel zur yz-Ebene angeordnet (rote Farbe). Deren Koordinaten-Eckpunkte sind ebenso in kartesischen Koordinaten angegeben.

Ein Dauermagnet wird durch zwei Stromschleifen an dessen oberem und unterem Ende simuliert und steht für die Rotation um die z-Achse zur Verfügung.

Die Anzahl und Anordnung der Spulen und Magnete ist im DFEM-Algorithmus frei wählbar. Das hier gezeigte Beispiel entspricht dem Quellcode im Anhang der Publikation.

Um die Eingaben für die Lösung der Differentialgleichungen eines solchen Systems für den DFEM-Algorithmus vorzubereiten, benötigen wir:

- (a.) eine Berechnung der induzierten Spannung, die der rotierende Dauermagnet in die Spule einbringt
- und
- (b.) eine Berechnung der Magnetkraft, die auf den Dauermagneten wirkt.

Auf diese Weise wird die Ankopplung der beiden Teile des Systems aneinander realisiert. Dies ist einerseits die mechanische Rotation des Dauermagneten und andererseits die elektrische Schwingung in einem LC-Schwingkreis (siehe [Tur 11]), in dem die rot gezeichnete Spule eingebunden ist. Dabei erfasst die induzierte Spannung, in welcher Weise die mechanische Bewegung in das elektrische System einwirkt, und andererseits die Magnetkraft, in welcher Weise die Ströme des Schwingkreises auf die mechanische Bewegung einwirken.

Die beiden Berechnungen unter Punkt (a.) und Punkt (b.) wollen wir nun im Detail betrachten:

Zu a:

Die Berechnung der in den Spulen induzierten Spannung beruht auf einer zeitlichen Veränderung des magnetischen Flusses  $\psi$ , die durch Bewegung (in unserem Beispiel eine Rotation um die z-Achse) des oder der Dauermagneten in den Spulen zustande kommt (siehe Gleichung (1), [Jac 81]).

$$\psi = \int \vec{B} \cdot d\vec{A} \Rightarrow U_{ind} = -N \cdot \frac{d\psi}{dt} \quad (N = \text{Windungszahl}) \quad (1)$$

Zu dieser Berechnung wird das Magnetfeld des (oder der) Dauermagneten als Vektorfeld berechnet und in einem Array abgespeichert. Im Algorithmus ist dafür das Unterprogramm „Magnetfeld\_zuweisen“ verantwortlich, das mit einer Nummer versehen wurde, um verschiedene mögliche Dauermagneten zu emulieren. Das Vektorfeld wird nun starr an den Dauermagneten (bzw. an die Dauermagneten) angeoppelt, so dass es sich mit dem Magneten aufgrund dessen Rotation mitdreht. Die Rotation um die z-Achse wird dann mit der üblichen Drehtransformation nach Gleichung 2 realisiert. Darin ist  $\varphi$  der Drehwinkel gegenüber der y-Achse, und  $(x,y)$  sind die Koordinaten im ungedrehten Bezugssystem, in dem der Magnet entlang der y-Achse ausgerichtet ist, wohingegen  $(x', y')$  die Koordinaten im gedrehten Bezugssystem sind, in dem der Magnet um den Winkel  $\varphi$  gegenüber der y-Achse gedreht ist. Verantwortlich dafür ist im Algorithmus das Unterprogramm „Magnet\_drehen“.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(-\varphi) & \sin(-\varphi) \\ -\sin(-\varphi) & \cos(-\varphi) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{Matrixmultiplikation} \quad (2)$$

Auf diese Weise lässt sich die Feldstärke ausrechnen, die der Dauermagnet an jedem beliebigen Ort im Raum erzeugt, und zwar als Funktion seines Drehwinkels  $\varphi$ . Verantwortlich dafür ist im Algorithmus das Unterprogramm „Feldstaerke\_am\_Ort\_suchen“.

Am Ende dieses Unterprogramms wird der magnetische Fluss durch die Spule bestimmt, wobei eine Unterteilung der Spule in finite Elemente (mit endlicher Ausdehnung) vorhanden ist. Einerseits ist der Spulendraht in einzelne Leiterstücke endlicher Länge unterteilt, andererseits ist auch die Spulen-Querschnittsfläche in Flächenstücke endlicher Größe untergliedert. Für jedes einzelne Spulen-Flächen-Element bestimmt das genannte Unterprogramm dann den zugehörigen magnetischen Fluss, dessen Änderung als Funktion der Zeit, und daraus schließlich die induzierte Spannung in der Spule. In unserem speziellen Beispiel kommt uns dabei die Tatsache gelegen, dass die Flächen-Normalen-Vektoren dieser Spulen-Flächen-Elemente in Richtung der x-Achse zeigen, was die Bestimmung des Skalarproduktes des Feldes mit dem Flächen-Normalen-Vektor vereinfacht, wie in Gleichung (3) zu sehen (sie ist nur einfach die x-Koordinate).

$$\psi_{SFE} = \vec{B} \cdot d\vec{A} = B_x \cdot |d\vec{A}| \quad (3)$$

Der Fluss durch die gesamte Spule wird einfach als Summe über den jeweiligen Fluß durch alle Spulen-Flächenelemente (einer Spule) aufsummiert, so dass sich der Gesamtfluß durch die Spule ergibt gemäß Gleichung (4).

$$\psi_{GES} = \sum \psi_{SFE} \quad (4)$$

Anmerkung zu den Spulen-Flächen-Elementen (Index „SFE“):

Um die Variationsmöglichkeiten der Spulengeometrie so umfangreich wie möglich zu gestalten, muss die Spule als Polygonzug zwischen beliebig vorgebbaren Stützpunkten modelliert werden. Jeder Polygonzug definiert eine Fläche mit beliebiger Gestalt, so dass der Spulendraht aus Leiterschleifen-Elementen von Stützpunkt zu Stützpunkt besteht. Die Spulen-Flächen-Elemente füllen nun die von jedem Polygonzug umrundete Fläche aus, so dass der magnetische Fluss durch die gesamte Spule als Summe der finiten magnetischen Flüsse durch die einzelnen Spulen-Flächen-Elemente bestimmt werden kann.

Mit der zeitlichen Ableitung gemäß Gleichung (1) ist dann die Berechnung der induzierten Spannung fertig. Dazu wird natürlich die Abhängigkeit des Drehwinkels  $\varphi$  von der Zeit benötigt, also die Winkelgeschwindigkeit des Magneten zum jeweiligen Zeitpunkt. Wenn wir unsere Rechenmethode mit einer konstanten Winkelgeschwindigkeit kontrollieren, kommen wir zu einem Ergebnis wie z.B. dargestellt in Abb.2.

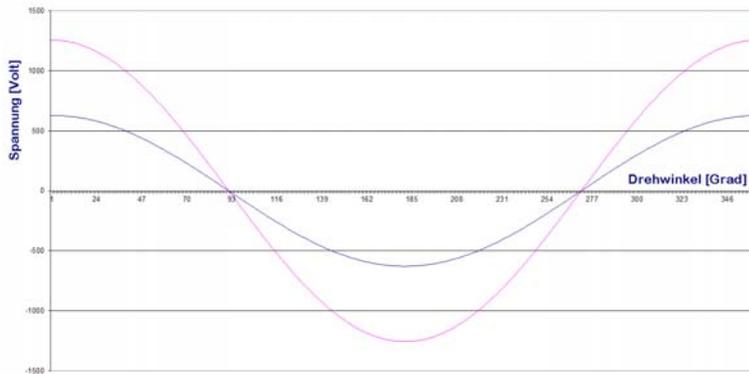


Abb.2:

Zur Verifikation der Rechenmethode wurde ein Magnet, der ein homogenes Magnetfeld erzeugt, mit konstanter Winkelgeschwindigkeit gedreht und die Berechnung der induzierten Spannung in den beiden Spulen dargestellt.

Dass die Spannung in beiden Spulen unterschiedlich ist, liegt an der Tatsache, dass die eine Spule doppelt so viele Windungen hat wie die andere.

Da wir für die spätere Berechnung des Drehmoments, welches der Magnet aufgrund der Spulenströme erfährt (siehe Punkt (b.)), den Dauermagneten durch eine Konfiguration aus Leitern emulieren müssen, ist bei komplizierteren magnetischen Anordnungen (also nicht einfach bei einem simplen homogenen Magnetfeld, wie es Abbild 2 zu Grunde liegt), die Berechnung des magnetischen Flusses von der räumlichen Auflösung der Berechnung (oder der Messung und Emulation) des Magnetfeldes abhängig. Aus diesem Grunde ergibt sich für den magnetischen Fluss komplizierter Anordnungen verschiedener Leiter ein numerisches Rauschen, das naturgemäß beim Ableiten, und somit bei der Bestimmung der induzierten Spannung drastisch verstärkt wird. Ein Beispiel für ein solches Rauschen sehen wir in Abbild 3, bei dem ein Magnetfeld gemäß dem Unterprogramm „Magnetfeld\_zuweisen\_02“ zugrunde liegt. Noch wesentlich stärker wird das numerische Rauschen später bei der Emulation eines echten Stabmagneten ausfallen.

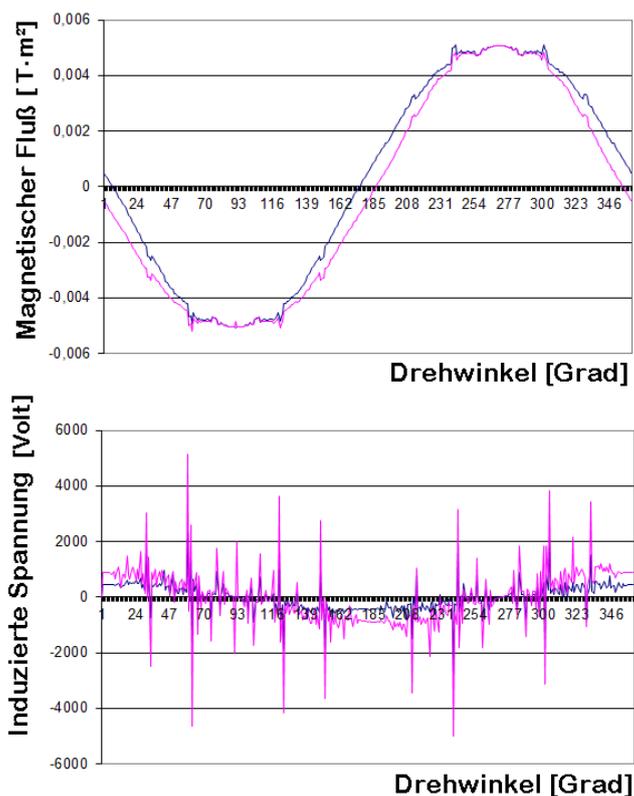


Abb.3:

Während der magnetische Fluss bei konstanter Winkelgeschwindigkeit des rotierenden Magneten ein gewisses numerisches Rauschen zeigt, erfordert die Berechnung der induzierten Spannung eine zusätzliche (nachträgliche) numerische Glättung, damit überhaupt ein sinnvoller Wert für die Weiterverarbeitung im DFEM-Algorithmus zur Verfügung gestellt werden kann.

Um das numerische Rauschen des Spannungs-Signals zu glätten, wurde eine Fourier-Entwicklung programmiert (im gleichnamigen Unterprogramm), bei der allerdings gezielt die Komponenten nur bis zu einer nicht all zu hohen Ordnung (z.B. bis zur fünften Harmonischen) mitgenommen wurden, um sicherzustellen, dass hochfrequente Anteile nicht die numerische Glättung verringern.

Im übrigen erfüllt die Entwicklung einer Fourier-Reihe die dringende Notwendigkeit der CPU-Zeit-Ersparnis bei der Berechnung des magnetischen Flusses. Da die eigentliche Berechnung des magnetischen Flusses eine Summe (nach Gleichung 4) über zeitaufwändige Rechenoperationen (Gleichungen 2 und 3) erfordert, ist deren Ausführung wesentlich langwieriger als die Berechnung von nur fünf Summanden einer Fourierreihe mit ungerader Funktionssymmetrie nach Gleichung (5). In Anbetracht der Tatsache, dass zum Lösen der Differentialgleichung eine sehr feine Zeit Diskretisierung erforderlich sein wird, mit Zeitschritten im Bereich weniger Nanosekunden, muss die Berechnung des magnetischen Flusses bei jedem Lauf des Algorithmus einige  $10^8 \dots 10^9$  mal ausgeführt werden, so dass hier die CPU-Zeit-Ersparnis eine durchaus wichtige Rolle hat. Um das Ergebnis der Lösung der Differentialgleichung abzusichern, werden wir die Größe der Zeitschritte sukzessive verfeinern müssen, so dass eine Beschleunigung der innersten Rechenschleifen (dazu gehört auch die Berechnung des magnetischen Flusses) von essenzieller Bedeutung ist.

$$\psi_{GES} = \sum_{v=1}^{N_0} A_v \sin(v \omega t) \quad \begin{array}{l} \text{Näherung mit Fourier-Reihe in 5.Ordnung} \\ A_v = \text{Fourier-Koeffizienten} \end{array} \quad \text{(nach [Bro 08])} \quad (5)$$

Aufgrund der geringen Anzahl der tatsächlich verwendeten Fourier-Koeffizienten ( $N_0 \leq 5$ ) ist deren einfache und rasche Bestimmung mittels der Gauß'schen Methode der kleinsten Abweichungsquadrate zwischen der Näherung mit Fourier-Reihe und dem eigentlichen Signal ein sehr bequemes Verfahren zur Bestimmung der Koeffizienten  $A_v$ .

Zu b:

Zur Berechnung der Lorentz-Kraft, die die Spulenströme auf den Dauermagneten ausüben, wurde wie bereits erwähnt, eine Emulation des Dauermagneten durch eine geeignet wählbare Anzahl von Stromschleifen vorgesehen. In unserem Demonstrationsbeispiel begnügen wir uns mit einem zylindrischen Stabmagneten, da dieser auch für eine spätere Realisation in einem realen Gerät einfach zu bekommen ist. Der Einfachheit halber sei hier der Stabmagnet durch zwei kreisförmige Leiterschleifen an den beiden Enden des zylindrischen Stabes repräsentiert. Das Magnetfeld des Stabmagneten wird also als Magnetfeld eines Spulenpaares emuliert. Als Parameter hierfür benötigt der Algorithmus lediglich die Länge des Zylinderstabes, dessen Durchmesser, und das Magnetfeld an den Enden des Stabes. Letzteres wird mithilfe eines fiktiven Emulationsspulen-Stroms so eingestellt, dass die Emulation die tatsächlich gemessenen Feldstärken an einem vorhandenen Stabmagneten wiedergibt.

Das Magnetfeld dieses Emulations-Spulen-Paares berechnen wir mithilfe des Biot-Savart'schen Gesetzes für einen beliebigen Aufpunkt, so wie es in Abbild 4 dargestellt ist.

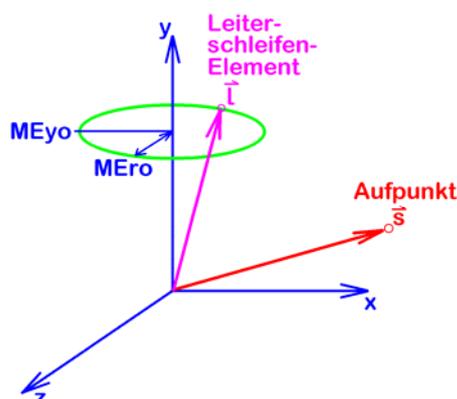


Abb.4: Veranschaulichung der Geometrie einer Leiterschleife (grün) deren Elemente durch einen Ortsvektor  $\vec{l}$  parametrisiert sind.

Nach Biot-Savart berechnet wird das Magnetfeld, das diese Leiterschleife an einem beliebigen Aufpunkt am Ort  $\vec{s}$  erzeugt.

Da die besagte Leiterschleife das obere Ende eines in Richtung der y-Achse orientierten Stabmagneten bildet, liegt sie parallel zur xz-Ebene, hat den Radius „MEro“ und befindet sich bei der y-Koordinate „MEyo“.

Eine bequeme Parametrisierung der Leiterschleife kann realisiert werden durch Gleichung 6.

$$\vec{l}(t) = \begin{pmatrix} MEro \cdot \cos(\omega t + \varphi) \\ MEyo \\ MEro \cdot \sin(\omega t + \varphi) \end{pmatrix} \Rightarrow \vec{v}(t) = \frac{d}{dt} \vec{l}(t) = \begin{pmatrix} -\omega \cdot MEro \cdot \sin(\omega t + \varphi) \\ 0 \\ +\omega \cdot MEro \cdot \cos(\omega t + \varphi) \end{pmatrix} \quad (6)$$

Geben wir den Aufpunkt  $\vec{s} = \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix}$  durch dessen kartesische Koordinaten an, so können wir mit

Gleichung (7) in das Biot-Savart'sche Gesetz einsetzen:

$$d\vec{H} = \frac{q_1 \cdot \vec{v} \times (\vec{l} - \vec{s})}{4\pi \cdot |\vec{l} - \vec{s}|^3} \cdot \frac{d\varphi}{2\pi} \quad (7)$$

Das Kreuzprodukt im Zähler lautet dort

$$\vec{v} \times (\vec{l} - \vec{s}) = \begin{pmatrix} -\omega \cdot MEro \cdot \cos(\omega t + \varphi) \cdot [MEyo - s_y] \\ \omega \cdot MEro \cdot \cos(\omega t + \varphi) \cdot [MEro \cdot \cos(\omega t + \varphi) - s_x] + \omega \cdot MEro \cdot \sin(\omega t + \varphi) \cdot [MEro \cdot \sin(\omega t + \varphi) - s_z] \\ -\omega \cdot MEro \cdot \sin(\omega t + \varphi) \cdot [MEyo - s_y] \end{pmatrix} \quad (8)$$

Damit bestimmen wir den Betrag im Nenner von Gleichung (7) zu

$$|\vec{l} - \vec{s}|^3 = \left( [MEro \cdot \cos(\omega t + \varphi) - s_x]^2 + [MEyo - s_y]^2 + [MEro \cdot \sin(\omega t + \varphi) - s_z]^2 \right)^{3/2} \quad (9)$$

Damit können wir zwar im Prinzip die Ausdrücke (8) und (9) in das Kreuzprodukt von (7) einsetzen, aber es fehlt uns noch die Ladung  $q_1$  in (8). Diese ist zu bestimmen aus dem Spulenstrom  $I$  und der Laufgeschwindigkeit der Ladung  $\omega$  ständig, denn diese beiden Größen müssen kombiniert werden, um einen sinnvollen Wert für die Ladung und deren Bewegung zu erhalten. Da der Strom bekanntlich als die pro Zeiteinheit fließende Ladung zu verstehen ist, können wir schreiben:

$$\left. \begin{array}{l} I = \frac{q_1}{T} \\ T = \frac{2\pi}{\omega} \end{array} \right\} \Rightarrow I = \frac{\omega}{2\pi} \cdot q_1 \quad (10)$$

Von den beiden Größen  $q_1$  und  $\omega$  können wir also willkürlich die Eine festlegen und die andere dann geeignet anpassen, so dass sich der richtige Spulenstrom  $I$  für die Emulationsspulen ergibt. Setzen wir also willkürlich  $q_1 = 1 \text{ Ampere}$  an und passen wir dann  $\omega$  so an, dass sich der gewünschte Spulenstrom einstellt. Dazu lösen wir Gleichung (10) auf nach  $\omega$ :

$$\omega = \frac{2\pi \cdot I}{q_1} \quad (11)$$

Zur Summierung der infinitesimalen Feldstärken-Elemente nach Gleichung (7) könnten wir das entsprechende Ringintegral über die Leiterschleife lösen. Da im Algorithmus beliebig geformte Leiterschleifen und Stromkonfiguration vorgesehen sind, machen wir dies über eine Diskretisierung mit  $N$  finiten Feldstärke-Elementen gemäß Gleichung (12). Ebenso wie in Abb.3 hat die Diskretisierung auch hier ein numerisches Rauschen ist eine Folge.

$$\vec{H}_{GES} = \oint_{\text{Leiter-}} d\vec{H} \approx \sum_{i=0}^N d\vec{H} = \sum_{i=0}^N \frac{q_1 \cdot \vec{v} \times (\vec{l} - \vec{s})}{4\pi \cdot |\vec{l} - \vec{s}|^3} \cdot \frac{\Delta\varphi_i}{2\pi} \quad (12)$$

Diese Summierung wurde im Unterprogramm „Magnetfeld\_zuweisen\_03“ praktisch umgesetzt, wobei die Summationsvariable  $I=0 \dots N$  dazu dient, das Argument der Parametrisierung über eine Periode laufen zu lassen, also  $t=0 \dots T = \frac{2\pi}{\omega}$ .

Die Kontrolle der Ergebnisse anhand der klassischen Formel der Feldstärkeberechnung (siehe Gleichung 13) entlang der Spulenachse (hier die y-Achse) bestätigt die Tauglichkeit des Verfahrens.

$$\vec{H}_{Klass} \approx \frac{I \cdot a^2}{2 \cdot (a^2 + r^2)^{3/2}} \tag{13}$$

Damit ist die reine Simulation des Magnetfeldes eines zylinderförmigen Stabmagneten mit Hilfe zweier Leiterschleifen an den beiden Enden des Zylinders erfolgreich abgeschlossen.

Die Bestimmung des Magnetfelds des Dauermagneten ist aber nicht das Ziel der Berechnung, sondern nur ein Schritt auf dem Weg dorthin. Ziel ist letztlich die Bestimmung des Drehmoments, mit dem die von den Spulen erzeugten Magnetfelder den Dauermagneten drehen. Dazu müssen wir die Lorentz-Kraft zu der dem Magnetfeld zugrunde liegenden Stromverteilung in den Dauermagnet-Emulationsspulen im Feld der äußeren Spulen bestimmen (welche in Abb. 1 in roter Farbe gezeichnet sind). Mit einfachen Worten: Welche Lorentz-Kraft übt der Spulenstrom in der rot gezeichnet Spule auf den Dauermagneten aus? Diese Lorentzkraft bildet dann die Grundlage, aus der das Drehmoment berechnet wird.

Dazu verwenden wir abermals das Biot-Savart'sche Gesetz, diesmal nun, um das Magnetfeld der rot gezeichneten Spulen an beliebigen Orten zu entwickeln, und damit eben auch an den Orten, an denen sich die Leiterschleifen-Elemente befinden, die die Stromverteilung der Magnet-Emulationsspulen wiedergeben.

Die felderzeugende Leiterschleifen-Anordnung ist also der Polygonzug der rot gezeichneten Spule, der Aufpunkt (an dessen Ort das Feld bestimmt werden soll) liegt an den Orten der Leiterschleifen-Elemente, die den Dauermagneten emulieren, so wie wir dies in Abb.5 sehen.

In Rechenbeispiel (siehe Anhang) könnte im Prinzip zu jeder der beiden Spulen (in rot) je ein Polygonzug mit je vier Stützpunkten verwendet werden. Tatsächlich sind aber zwischen den Eckpunkten der Rechteckspulen noch weitere Stützpunkte eingebracht, da diese Stützpunkte gleichzeitig verwendet werden, um die Eckpunkte der Spule-Flächen-Elemente festzulegen. Eine Erhöhung der Zahl der Stützpunkte führt also zu einer feineren Vernetzung der Spulenflächen, die der Berechnung des magnetischen Flusses durch die Spulen zugrundegelegt werden (vgl. Glg. 4).

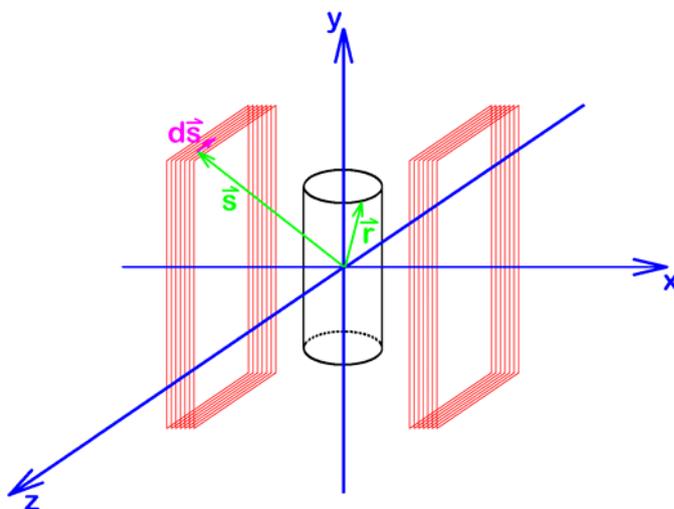


Abb.5: Veranschaulichung der Vektoren, für die Berechnung des Feldes der Spulen am Ort des Dauermagneten nach dem Biot-Savart'schen Gesetz.

$$\vec{s} = \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix} \quad \text{Leiterschleifen-Element}$$

$$\vec{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \quad \text{Aufpunkt}$$

$$d\vec{s} = \begin{pmatrix} d\vec{s}_x \\ d\vec{s}_y \\ d\vec{s}_z \end{pmatrix} \quad \text{Ladungsbewegung am Leiterschleifen-Element}$$

Bezogen auf die Parameter nach Abb.5 können wir das Biot-Savart'sche Gesetz formulieren gemäß Gleichung (14).

$$d\vec{H} = \frac{I \cdot d\vec{s} \times (\vec{s} - \vec{r})}{4\pi \cdot |\vec{s} - \vec{r}|^3} \quad (14)$$

Das Kreuzprodukt im Zähler lautet dort

$$d\vec{s} \times (\vec{s} - \vec{r}) = \begin{pmatrix} d\vec{s}_x \\ d\vec{s}_y \\ d\vec{s}_z \end{pmatrix} \times \begin{pmatrix} s_x - r_x \\ s_y - r_y \\ s_z - r_z \end{pmatrix} = \begin{pmatrix} d\vec{s}_y \cdot (s_z - r_z) - d\vec{s}_z \cdot (s_y - r_y) \\ d\vec{s}_z \cdot (s_x - r_x) - d\vec{s}_x \cdot (s_z - r_z) \\ d\vec{s}_x \cdot (s_y - r_y) - d\vec{s}_y \cdot (s_x - r_x) \end{pmatrix} \quad (15)$$

Wie gewohnt setzen wir den Ausdruck nach Gleichung (15) in Gleichung (14) ein und erhalten so die finiten Feldelemente  $d\vec{H}$ , die jedes Leiterschleifen-Element der rot gezeichneten Spulen am Aufpunkt erzeugt, also an demjenigen Ort der Magnet-Emulations-Spulen erzeugt, die den Dauermagneten emulieren. Diese Berechnung wird für jedes einzelne Leiterschleifen-Element der Dauermagnet-Emulations-Spulen einzeln durchgeführt, um die Lorentz-Kraft auf jedes einzelne dieser Leiterschleifen-Elemente zu bestimmen (gemäß Gleichung (16)).

$$d\vec{F} = q \cdot (\vec{v} \times d\vec{B}) = I \cdot (\vec{l} \times d\vec{B}) \quad (16)$$

mit den Feldelementen  $d\vec{B} = \mu \cdot d\vec{H}$

Die finiten Lorentz-Kraft-Elemente  $d\vec{F}$  werden dann benutzt, um finite Drehmoment-Elemente zu berechnen (siehe Gleichung (16)), die auf die Drehbewegung des Dauermagneten einwirken. Deren Summationen zum Gesamtdrehmoment wird über eine diskrete Summe ausgeführt (siehe Gleichung (17)), denn die DFEM-Methode ist eine Berechnungsmethode der finiten Elemente.

$$d\vec{M} = \vec{r} \times d\vec{F} \quad (\text{finites Drehmoment-Element})$$

$$\vec{M}_{ges} = \sum d\vec{M} \quad (\text{Summation zum Gesamt-Drehmoment, Näherung durch diskrete Summe}) \quad (17)$$

Durchgeführt werden die Berechnungen im Unterprogramm „Drehmoment“.

Auch hier wird ein numerisches Rauschen des Drehmoments als Funktion des Drehwinkels festgestellt, das eine Glättung durch eine Fourier-Entwicklung sinnvoll erscheinen lässt. Wir begnügen uns abermals mit einer nicht all zu hohen Ordnung (z.B. bis zur fünften Harmonischen), um hochfrequente Anteile sicher auszuschließen (vgl. Gleichung (5)).

Auch hier ist wieder die verbrauchte CPU-Zeit für die Berechnung eines einzelnen Drehmoments zu groß, als dass man die Drehmoments-Berechnung zur iterativen Lösung der Bewegungs-Differentialgleichung all zu oft ausführen kann. Deshalb ist auch hier die Fourier-Reihe mit wenigen Summanden ein probates Mittel zur CPU-Zeit-Ersparnis im Hinblick auf die später sehr häufige Benutzung bei der iterativen Lösung der Differentialgleichung in sehr feiner Zeit-Diskretisierung.

Deshalb werden bei Gleichung (5) ebenso wie bei Gleichung (17) die Fourier-Koeffizienten der Näherungen während der Initialisierungsphase des DFEM-Programms bestimmt, um später im Hauptteil der Berechnungen (dies ist die Lösung der Bewegungs-Differentialgleichungen) auf die sehr schnell laufenden Fourier-Näherungen mit wenigen Summanden zurückgreifen zu können.

Damit sind die physikalischen Vorarbeiten abgeschlossen. Dies sind zwei Schritte:

- die Berechnung der induzierten Spannung, die die Drehung des Dauermagneten in den Spulen erzeugt, und
- die Berechnung des Drehmoments, die die Spulenströme in Dauermagneten erzeugen.

Die durch Fourier-Reihen schnell gemachten Berechnungen sind in dem im Anhang gezeigten Quellcode dokumentiert in den Unterprogrammen „Schnell\_Drehmoment“, „Fluss\_T“ und „Fluss\_I“.

Wir fassen also das Ergebnis von Abschnitt 1 zusammen in Abb.6 und Abb.7, die exemplarisch zu der in Abb.5 skizzierten Geometrie gehört.

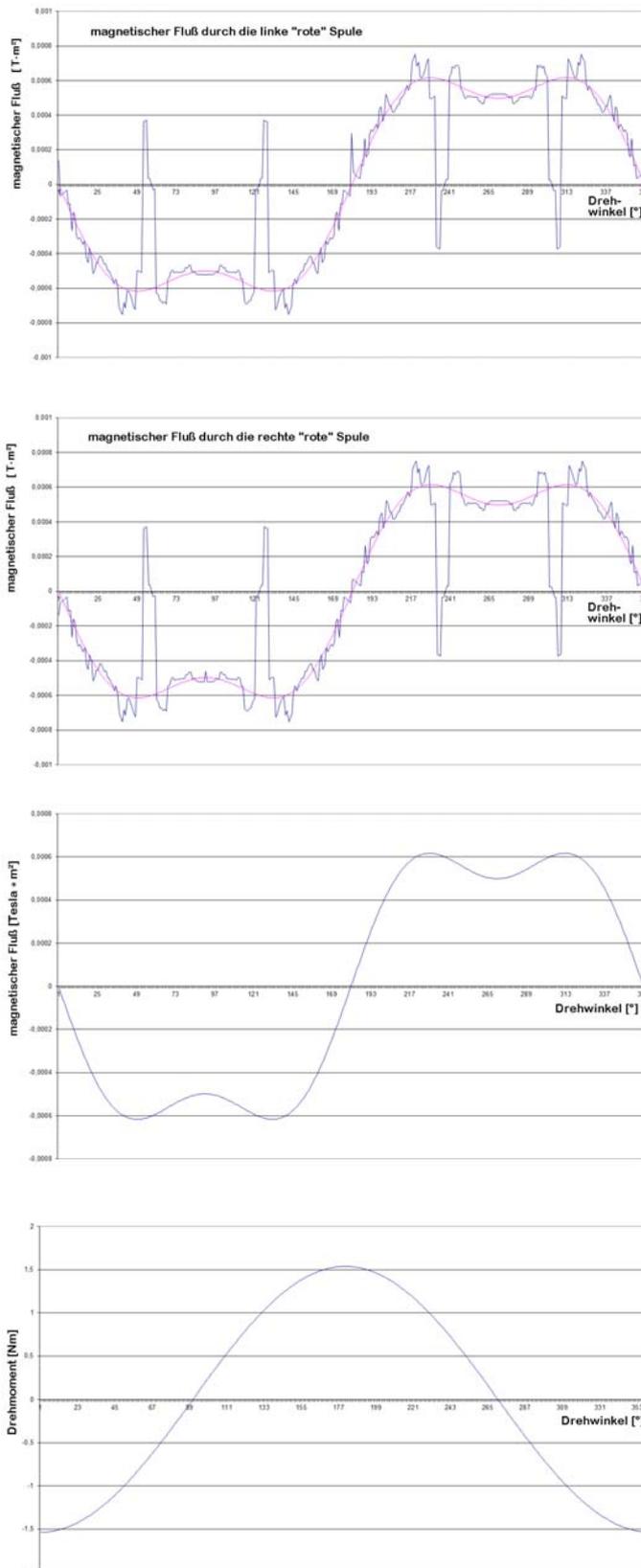


Abb.6:

Der magnetische Fluss, den der Dauermagnet nach Abb.5 durch die beiden rot gezeichneten Spulen fließen lässt ist abhängig von der Orientierung des Dauermagneten (Drehwinkel).

Aufgrund der Symmetrie der Anordnung ist dieser Fluss durch die beiden eingezeichneten Spulen gleich groß.

Das blaue Signal gibt das numerische Rauschen der Berechnung mit finiten Elementen endlicher Größe wieder. Besonders stark ist das Rauschen in dem Moment, in dem die Windungen der Magnet-Emulation-Spulen sich sehr nahe an den Spulendrähten der roten Spulen befinden.

Das violette Signal ist eine Glättung des blauen Signals, die mittels einer Fourierreihe berechnet wurde, die nach dem fünften Summanden abgebrochen wurde.

Abb.7:

Der magnetische Fluss erlaubt mittels zeitlicher Ableitung eine Berechnung der in den roten Spulen induzierten Spannung. (Es wird allerdings die Winkelgeschwindigkeit des Magneten dafür benötigt.)

Die Drehmoment-Kurve gibt wieder, wie groß das Drehmoment ist, welches die beiden roten Spulen bei konstantem Strom auf die Dauermagneten ausüben. Logischerweise ist der Wert des Drehmoments von der Orientierung des Magneten abhängig.

Werte der Parameter für Abb.6 und Abb.7:

- Dauermagnet, zylindrisch, 4 cm dick, 8 cm lang, Feldstärke 1 Tesla am Zylinderende.
- „Rote“ Spulen, rechteckig, 6 cm breit, 12 cm hoch, bei  $x=-2$  cm und  $x=+2$ cm.

Anmerkung: Wir werden später auch etwas abweichende Abmessungen untersuchen.

Dabei sei ausdrücklich betont, dass die Zahl der Spulen ebenso wie die Zahl der Dauermagneten frei wählbar ist, so dass sich mit dem hier vorgestellten DFEM-Algorithmus jeder beliebige Raumenergie-Konverter auf magnetischer Basis berechnen läßt:

- Emuliert man zum Beispiel mehrere Dauermagneten durch geeignete Stromverteilungen, so kann man deren Wechselwirkung in analoger Weise mit dem DFEM-Algorithmus analysieren. Dies wird interessant zur Berechnung rein magnetischer Selbstläufer, wie man sie zum Beispiel findet in [Hoh 11], [Jeb 06].
- Emuliert man zum Beispiel mehrere Spulen, die gegebenenfalls auch über ein Joch miteinander in Verbindung stehen können, um den magnetischen Fluss in geeigneter Weise zu leiten, so lässt sich ein bewegungsloser Raumenergie-Konverter rechnen, wie etwa nach derjenige S. Marinov [Mar 88-98] (oder auch bei [Bea 02]).
- Emuliert man zum Beispiel eine Spule und einen Dauermagneten, so lässt sich der „Elektro-Mechanische Doppel-Resonanz“ Konverter (EMDR-Konverter) berechnen, den der Autor des vorliegenden Artikels ersonnen hat. [Tur 11]
- Emuliert man zum Beispiel zwei Spulen und dazu mehrere kleinere Dauermagneten, die in geeigneter Weise zueinander angeordnet sind, um eine spezielle Geometrie des magnetischen Permanentfeldes zu erreichen, so lässt sich der Keppe-Motor berechnen. [Kep 10]
- Ordnet man zum Beispiel sechs zylindrische Stabmagneten im Inneren von sechs Spulen in einem Hexagon an, so lässt sich ein Coler-Apparat simulieren. Das Verhalten dieser Elemente wird dann in einem geeigneten Differentialgleichung-System simuliert, die sonstigen Schaltelemente gehen in Form von Anfangsbedingungen in dieses Differentialgleichung-System ein. (Für Differentialgleichung: Siehe Abschnitt 2) Vielleicht lässt sich auf diese Weise endlich klären, ob der Coler-Apparat funktionieren kann oder nicht. [Hur 40], [Mie 84], [Nie 83]
- Auch dynamische Energiezufuhr und Energieentnahme bereitet kein Problem, da man in der Bewegungs-Differentialgleichung (siehe Abschnitt 2 der vorliegenden Arbeit) mühelos Spannungs-Versorgungen oder Lastwiderstände in den Spulen einbringen kann, ebenso bequem aber auch mechanische Kräfte und Drehmomente an den Bewegungen der Dauermagneten. Um dies zu veranschaulichen, wird in dem im Anhang wiedergegebenen Rechenbeispiel ein Spulenpaar nach Abb.5 vorgesehen. Die linke der beiden Spulen, die wir als Input-Spule bezeichnen wollen, kann über eine externe Spannung versorgt werden (siehe Unterprogramm „U7“). Der rechten der beiden Spulen, die wir als Turbo-Spule bezeichnen wollen, kann über einen Lastwiderstand („ $R_{Last}$ “) elektrische Leistung entnommen werden. Der Rotationsachse des Dauermagneten wird über eine Start-Drehzahl einmalig mechanische Energie zugeführt, um den Magnetmotor in Bewegung zu setzen. Da es sich in dem im Anhang gezeigten Rechenbeispiel um einen Selbstläufer handelt, muss diese Energie nur einmalig zugeführt werden, um die Drehbewegung zu initialisieren. Desweiteren konnte die Input-Spule im Rechenprogramm auskommentiert werden, da der Raumenergie-Motor als typischer Selbstläufer keine Input-Energie benötigt. Allerdings könnte man auf Wunsch diese Input-Spannungspulse benutzen, um zum Beispiel die Drehzahl des Motors zu steuern oder zu regeln.
- Weiterhin ist im Algorithmus eine mechanische Leistungsentnahme vorgesehen. In die Differentialgleichung wurde sie als Geschwindigkeits-proportionales Drehmoment eingebracht, wie es zum Beispiel bei manchen Reibungs-Mechanismen der Fall sein kann. (Wir werden dies in Abschnitt 6 detailliert besprechen.)
- Um einen Raumenergie-Konverter sicher in einem dauerhaft stabilen Betriebszustand zu halten, ist es eine bequeme Methode, die Leistungsentnahme zu steuern und zu regeln. In dem im Anhang wiedergegebenen DFEM-Algorithmus wird die mechanische Leistungsentnahme anhand der Drehzahl des rotierenden Magneten gesteuert. Dabei wird ein Zielwert für die Drehzahl vorgegeben, der mit einer gewissen Schalt-Hysterese eingehalten wird. Übersteigt die Drehzahl eine einstellbare Schwelle, so wird die mechanische Energieentnahme erhöht (um den Rotor abzubremsen); unterschreitet die Drehzahl hingegen eine

andere ebenfalls einstellbare Schwelle, so wird die Energieentnahme verringert (damit der Rotor nicht zu stark abgebremst wird).

Die Vielfältigkeit des DFEM-Rechenverfahrens ist groß genug, dass sich die Rechenmethode nicht nur auf alle Arten magnetischer Raumenergie-Konverter beschränkt, sondern zum Beispiel ebenso gut auf elektrostatische Systeme angewendet werden kann. Man muss dazu lediglich die Lorentz-Kraft aus Gleichung (16) durch die Coulomb-Kraft aus Gleichung (18) ersetzen.

$$d\vec{F}_{12} = \frac{Q_1 \cdot dQ_2}{4\pi \epsilon_0} \cdot \frac{\vec{r}_1 - \vec{r}_2}{|\vec{r}_1 - \vec{r}_2|^3} \quad \text{mit finiten Kraft-Elementen,} \quad (18)$$

die die Ladung  $Q_1$  auf finite  
Ladungselemente  $dQ_2$  ausübt.

Die Genauigkeit der Berechnung hängt hauptsächlich von der präzisen Eingabe der verwendeten mechanischen und elektrischen Komponenten und deren Wechselwirkungs-Felder ab, über die diese Komponenten miteinander in Beziehung stehen.

## 2. Bewegungen der Komponenten des Raumenergie-Konverters

Damit sind die grundsätzlichen Vorarbeiten erledigt, und wir können uns der eigentlichen Lösung der Bewegungs-Gleichungen zuwenden, die die Bewegungen der (mechanischen und elektrischen) Komponenten des Raumenergie-Konverters beschreiben.

Die Funktion eines jeden Raumenergie-Konverters kann durch die Bewegungen seiner Komponenten beschrieben werden. Das probate physikalische Mittel dazu ist die Aufstellung und die Lösung der entsprechenden Bewegungs-Differentialgleichungen.

Da ein Raumenergie-Konverter natürlich nicht aus einer einzigen Komponente besteht, sondern aus mehreren, die miteinander in Wechselwirkung stehen, handelt es sich dabei grundsätzlich um gekoppelte Differentialgleichungs-Systeme höherer Ordnung. Die Tatsache, dass Energie abgeführt werden muss (bei manchen Raumenergie-Konvertern, die nicht als Selbstläufer arbeiten, sondern nur als over-unity System muss auch Energie zugeführt werden), erfordert die Berücksichtigung geeigneter Störfunktionen, die diese Wechselwirkung mit der Umwelt wiedergeben. Dadurch werden die Differentialgleichungs-Systeme zur Berechnung eines Raumenergie-Konverters inhomogen.

Mathematisch bedeutet dies, dass wir relativ komplizierte inhomogene gekoppelte Differentialgleichungs-Systeme höherer Ordnung zu lösen haben, was natürlich für den beliebigen Allgemeinfall nicht einfach analytisch möglich ist. Daher besteht der zentrale Rechenkern des DFEM-Algorithmus aus einem numerisch-iterativen Differentialgleichung-Löser, dessen Beschreibung und Erläuterung Ziel und Inhalt von Abschnitt 2 der vorliegenden Publikation ist. Auf diesem Hintergrund wird auch klar, warum die Benutzung des DFEM-Algorithmus mit höherer Rechen-Präzision eine gewisse CPU-Zeit erfordert, und nicht nach wenigen Sekunden bereits zu einem Ergebnis führen kann.

Das Verständnis, sowie das Aufstellen und das Lösen dieses Bewegungs-Differentialgleichungs-Systems ist der entscheidende zentrale Punkt, den jeder Benutzer des hier präsentierten DFEM-Algorithmus bewältigen muss, um nach diesem Verfahren einen Raumenergie-Konverter berechnen zu können.

Beginnen wir also nun mit dem Aufstellen der Differentialgleichungen (exemplarisch am Beispiel des EMDR-Konverters), die den eigentlichen Rechenkern des DFEM-Algorithmus bedeuten und somit die Berechnung des Raumenergie-Konverters ausführen. Dies geschieht in Analogie zu [Tur 11].

Da der Algorithmus vollständig dynamisch arbeiten muß (unter Berücksichtigung der Bewegungen sowohl der mechanischen Bauteile, als auch der Propagationsgeschwindigkeiten der Wechselwirkungs-Felder), müssen alle physikalischen Größen auf die Grundgrößen der schwingenden Ladungen bzw. der rotierenden Magneten zurückgeführt werden, somit also auf die Grundgrößen  $q$ ,  $\frac{d}{dt}q = \dot{q}$ ,  $\frac{d^2}{dt^2}q = \ddot{q}$  und auf  $\varphi$ ,  $\frac{d}{dt}\varphi = \dot{\varphi}$ ,  $\frac{d^2}{dt^2}\varphi = \ddot{\varphi}$ , wo  $q$  die Ladung ist und  $\varphi$  der Drehwinkel des Magneten.

Um die Zeitschritte beliebig fein vorgeben und steuern zu können, wird im Unterschied zu [Tur 11] auf eine vollständige Abspeicherung der Berechnungsdaten in einem Daten-Array verzichtet, damit für die Feinheit und für die Anzahl der Zeitschritte keine Speicherplatz-Begrenzung relevant bleibt. Einzig und allein nur maximal 35.000 Datenpunkte werden für den Daten-Export nach Excel abgespeichert, um eine graphische Darstellung der Ergebnisse zu ermöglichen. (Das sollte gut ausreichen, da kaum ein Bildschirm eine horizontale Auflösung von 35.000 Pixeln hat.)

Werte, wie etwa die Induktivität der zylindrischen Spules (siehe Gleichung 19) oder das Trägheitsmoment des rotierenden Magneten als Vollzylinder (siehe Gleichung 20) werden Standard-Lehrbüchern oder Formelsammlungen der Physik bzw. der Ingenieurwissenschaften entnommen.

Induktivität  $L = \mu \cdot \frac{N^2 A}{s}$ , mit  $N = \text{Windungszahl}$   
 $A = \text{Spulenquerschnittsfläche}$  [Ger 95]  
 $s = \text{Spulenkörper-Länge}$  (19 a)

Oder genauer mit Korrektur für kurze Spulenlänge:

Induktivität  $L = \mu \cdot \frac{N^2 A}{\sqrt{s^2 + \frac{A}{\pi/4}}}$ , mit  $N = \text{Windungszahl}$   
 $A = \text{Spulenquerschnittsfläche}$  Herleitung mit Hilfe [Stö 07]  
 $s = \text{Spulenkörper-Länge}$  (19 b)

$J_y = \frac{m}{4} \cdot \left( r_a^2 + r_i^2 + \frac{h^2}{3} \right)$  Trägheitsmoment eines Vollzylinders bei  
 Rotation um die quer zur Längsachse stehende Symmetrieachse [Dub 90] (20)

Damit ergibt sich die Möglichkeit, z.B. den Aufbau von Abb.8 zu simulieren.

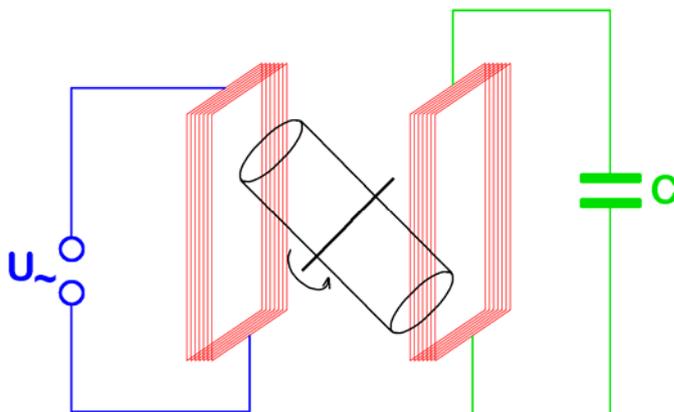


Abb.8:  
 Dies ist die Anordnung eines Raumenergie-Konverters, wie er mit dem Anhang gezeigten DFEM-Differentialgleichungssystem simuliert werden kann.  
 Zwei Spulen sind in roter Farbe eingezeichnet, ein rotierender Dauermagnet in schwarz, eine mögliche Input-Spannung in blau und ein Kondensator in grün.

Da der Raumenergie-Konverter unseres konkreten Rechenbeispiels erwartungsgemäß als Selbstläufer betrieben werden kann, wurde auf ein Anlegen einer Input-Spannung vollständig verzichtet, so dass die linke Spule zwar zur Veranschaulichung für die Leser und Leserinnen der Publikation mitmoduliert wurden, aber für die praktische Benutzung unserer Berechnungen nicht benötigt wurde, und daher mitsamt der Spannungsquelle im Programm auskommentiert wurde, an der Berechnung also nicht teilnimmt. Daher ist der benötigte Platz vorhanden, um die rechte Spule in die

Mitte des Dauermagneten zu schieben, so dass die auf den nachfolgenden Seiten berichteten Rechenergebnisse (siehe Abschnitt 3) auf eine Anordnung gemäß Abb.9 zurückgehen.

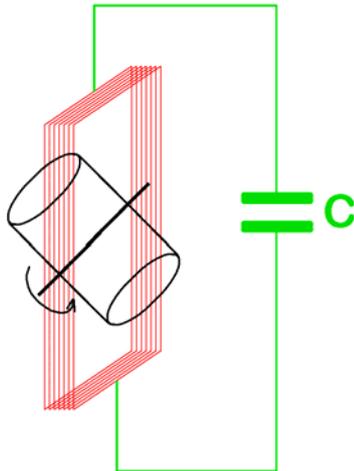


Abb.9:  
Dieser Konverter genügt bereits für eine leistungsstarke Wandlung von Raumergie. Dessen Differentialgleichungssystem wurde im hier präsentierten DFEM-Algorithmus aufgestellt und gelöst, sodaß sich die in Abschnitt 3 dargestellten Ergebnisse auf diese Variante beziehen.

Zum Aufstellen des Differentialgleichungs-Systems gehen wir wieder in ähnlicher Weise vor, wie wir dies bereits aus [Tur 11] kennen:

(a.)

Die Differentialgleichung einer harmonischen Schwingung im elektrischen Schwingkreis sehen wir in Gleichung (21) zu Vorbereitungs-Zwcken.

$$-L \cdot \ddot{Q} + \frac{1}{C} \cdot Q = 0 \Rightarrow \ddot{Q} = \frac{1}{LC} \cdot Q \tag{21}$$

(b.)

Die Differentialgleichung einer gedämpften elektrischen Schwingung sehen wir in Gleichung (22).

$$\ddot{Q} = -\frac{1}{LC} \cdot Q + \frac{R}{L} \cdot \dot{Q} \tag{22}$$

Ihre numerisch-iterative Lösung wurde nach Gleichung (23) in Analogie zu [Tur 11] ausgeführt, ein Test-Ergebnis sieht man in Abb.10 zu Kontroll-Zwecken.

$$Q_i = \dot{Q}_{i-1} + \ddot{Q}_i \cdot \Delta t - \frac{R}{L} \cdot \dot{Q}_{i-1} \cdot \Delta t \tag{23}$$

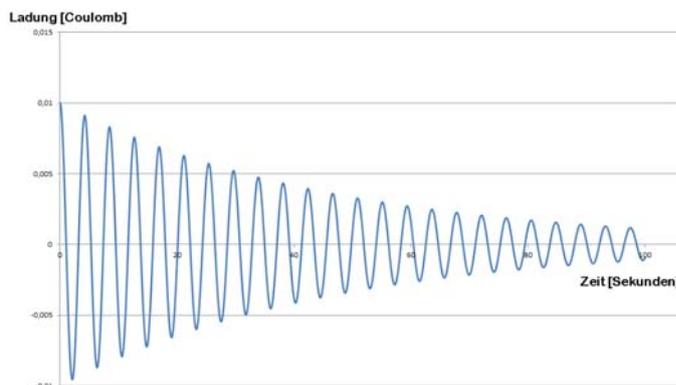


Abb.10:  
Verifikationstest einer gedämpften Schwingung im elektrischen LC-Schwingkreis zur Überprüfung der Differentialgleichung (22) und der Lösung (23).

(c.)

Im Gegensatz zum elektrischen Schwingkreis gibt es in der mechanischen Bewegung keine Rückstellkraft, sondern nur eine Rotation (des Dauermagneten). Zu berücksichtigen sind also zwei Beiträge zum hier auftretenden Drehmoment  $\vec{M}$ . Der eine Beitrag kommt durch die Einwirkung der Spule auf den Magneten zustande, da die Spulenströme ihrerseits Magnetfelder verursachen, die mit dem rotierenden Dauermagneten in Wechselwirkung stehen. Der andere Beitrag kommt durch die mechanische Leistungs-Entnahme zustande, die wir in unserem Rechenbeispiel später als Geschwindigkeits-proportional voraussetzen wollen. Den erstgenannten Beitrag kennen wir bereits aus Abschnitt 1. Den als Zweites genannten Beitrag betrachten wir im Detail weiter unten, nämlich in Abschnitt 6.

Damit wird die Differentialgleichung des mechanischen Anteils des Systems zu einer relativ simplen Integration gemäß Gleichungen (24). Dabei enthält Gleichung (24a) zugleich die Einkopplung des elektrischen System-Anteils in das mechanische System.

$$\ddot{\varphi}(t) = \frac{M(t)}{J} \quad \text{mit } M(t) = \text{Drehmoment} \quad \text{und } J = \text{Trägheitsmoment} \quad (24a)$$

$$\Rightarrow \dot{\varphi}(t) = \int_0^t \ddot{\varphi}(\tau) d\tau \Rightarrow \text{numerische Iteration} \quad \dot{\varphi}(t) = \underbrace{\ddot{\varphi}(t) \cdot dt}_{\text{Integrations-schritt}} + \underbrace{\dot{\varphi}(t-dt)}_{\text{Integrationskonstante}} \quad (24b)$$

$$\Rightarrow \varphi(t) = \int_0^t \dot{\varphi}(\tau) d\tau \Rightarrow \text{numerische Iteration} \quad \varphi(t) = \underbrace{\dot{\varphi}(t) \cdot dt}_{\text{Integrations-schritt}} + \underbrace{\varphi(t-dt)}_{\text{Integrationskonstante}} \quad (24c)$$

(d.)

Was noch in die Differentialgleichung des elektrischen Systems aufgenommen werden muss, ist die Einkopplung des mechanischen System-Anteils in das elektrische System. Dies ist vorzunehmen in Form der induzierten Spannung, die die Rotation des Dauermagneten in die Spule einbringt. Die daraus resultierende Erweiterung der Differentialgleichung des elektrischen Systems sehen wir in Gleichung (25), die sich aus Gleichung (22) durch Ergänzung des erwähnten Terms für die induzierte Spannung ergibt.

$$\ddot{Q} = -\frac{1}{LC} \cdot Q + \frac{R}{L} \cdot \dot{Q} - \frac{U_{ind}}{L} \quad (25)$$

(e.) Damit lässt sich die numerisch-iterative Lösung des elektrischen Anteils des Differentialgleichungssystems zurückführen auf die Integrationsschritte gemäß Gleichung (26), übrigens in Analogie zu [Tur 11].

$$\ddot{Q}(t) = \underbrace{-\frac{1}{LC} \cdot Q}_{\text{Spule und Kondensator}} + \underbrace{\frac{R}{L} \cdot \dot{Q}}_{\text{Ohm'scher Widerstand}} - \underbrace{\frac{U_{ind}}{L}}_{\text{Induzierte Spannung}} \quad (26a)$$

$$\Rightarrow \dot{Q}(t) = \int_{t-\Delta t}^t \ddot{Q}(\tau) d\tau \Rightarrow \text{numerische Iteration} \quad \dot{Q}(t) = \underbrace{\ddot{Q}(t) \cdot dt}_{\text{Integration}} - \underbrace{\dot{Q}(t-dt)}_{\text{Integrationkonstante}} \quad (26b)$$

$$\Rightarrow Q(t) = \int_{t-\Delta t}^t \dot{Q}(\tau) d\tau \Rightarrow \text{numerische Iteration} \quad Q(t) = \underbrace{\dot{Q}(t) \cdot dt}_{\text{Integration}} + \underbrace{Q(t-dt)}_{\text{Integrationkonstante}} \quad (26c)$$

Damit wird im Hauptprogramm des DFEM-Algorithmus die Bewegung des Raumenergie-Konverters berechnet, und zwar einerseits die Bewegung seiner elektrischen Ladungen und andererseits die Bewegung der in ihm enthaltenen beweglichen Magneten.

#### Allgemeine Anmerkung:

Für die Berechnungen dürfen natürlich nur solche Formeln und solche physikalischen Größen verwendet werden, die der Forderung nach einer echten Dynamik genügen. Ein gemittelter Strom, der sich zum Beispiel an Effektivwerten orientiert, oder der sich auf eine bestimmte Signalform bezieht, ist da dieser Stelle völlig ungeeignet. All diejenigen Leser und Leserinnen des vorliegenden Artikels, die gerne den DFEM-Algorithmus an ihre eigenen Apparaturen und Experimente anpassen möchten, müssen hierauf gewissenhaft aufpassen.

#### Grundsätzliche philosophische Anmerkung zur Ausbreitung und zur endlichen Propagationsgeschwindigkeit der Felder im elektrischen Schwingkreis:

Bei einem elektrischen Schwingkreis (wie zum Bsp. in Abb.9 zu sehen) tritt eine elektrische Schwingung in einer Spule und einen Kondensator auf. Die Frage ist nur, welche physikalischen Entitäten dort schwingen ?

Sind es Ladungen, die im elektrischen Schwingkreis hin- und her- laufen ?

Dies ist mit Sicherheit nicht der Fall. Man kann dies sehr einfach einsehen, wenn man die Betrachtung des Schwingkreises zu demjenigen Zeitpunkt beginnt, zu dem die Spule stromfrei ist. In diesem Augenblick ist nämlich der Kondensator maximal geladen, so dass die gesamte Energie des Schwingkreises als elektrostatische Feldenergie zwischen den Kondensatorplatten zu verstehen ist. Nun entlädt sich der Kondensator, so dass die elektrischen Felder (und Spannungen) durch den Draht der Spule laufen. Von der positiv geladenen Platte läuft ein Feld zur negativ geladenen Platte, und umgekehrt läuft von der negativ geladenen Platte ein Feld zur positiv geladenen Platte. Ladungsträger (wie zum Beispiel Elektronen) können es nicht sein, die da von der positiven zur negativen Platte (und umgekehrt) laufen, denn einerseits können diese Ladungsträger nicht mit der Propagationsgeschwindigkeit  $v = \frac{1}{\sqrt{LC}}$  laufen, und andererseits würden sich diese Ladungsträger, sobald sie sich in der Spulenmitte treffen, gegenseitig kompensieren. Würden positive Ladungsträger (wie zum Beispiel Elektronenlöcher/Elektronenmangel) von der positiv geladenen Kondensatorplatte zur negativ geladenen Kondensatorplatte laufen wollen und gleichzeitig negative Ladungsträger (wie zum Beispiel Elektronen) den umgekehrten Weg wählen, so würden sich die Beiden in der Spule treffen und gegenseitig auslöschen. Die Elektronen würden mit den Löchern rekombinieren (den Elektronenmangel ausgleichen), und so wäre nach bereits einer viertel Periode die Schwingung des LC-Schwingkreises vollständig beendet.

Sind es Felder, die im elektrischen Schwingkreis hin- und her- laufen ?

Ja, dies ist der Fall. Es gehört zu den typischen Eigenschaften der Felder (ebenso wie der Wellen) ungestört zu superponieren, d.h. sie überlagern und durchdringen sich, ohne einander zu beeinflussen. Ein Feld läuft (in Form eines Wellenberges) in die eine Richtung, wenn es die positiv geladene Kondensatorplatte verlässt, ein anderes Feld läuft (ebenfalls in Form eines Wellenberges) in

die entgegengesetzte Richtung, wenn es die negative Kondensatorplatte verlässt. Durchlaufen beide die Spule, so tun sie dies, ohne sich gegenseitig zu stören oder zu beeinflussen. Deshalb können sie der typischen Propagationsgeschwindigkeit  $v = \frac{1}{\sqrt{LC}}$  von Feldern in elektrischen Leitungen folgen.

Aufgrund der ungestörten Superposition erreichen die Felder nach einer halben Schwingungsperiode die jeweils gegenüberliegende Kondensatorplatte, so dass im Laufe einer halben Schwingungsperiode die positiv geladene Kondensatorplatte negativ geladen wird, und die negativ geladene Kondensatorplatte positiv.

Die Verschiebung der Ladungsträger, die diese Felder im Leiter verursachen existiert, aber sie ist nur ziemlich gering. Wir können dies sehr gut verstehen, wenn wir das akustische Analogon betrachten. Läuft Schall durch ein Rohr (die eindimensionale Sichtweise dient der Veranschaulichung), kann von der einen Seite ein Überdruck und gleichzeitig von der gegenüberliegenden Seite ein Unterdruck (als einmaliger Wellenberg bzw. einmaliges Wellental) losgeschickt werden, die war als Druckfeld betrachten, wie es in der Akustik durchaus üblich ist (siehe Kontinuitätsgleichung und Eulergleichung). Die beiden Druckfelder durchdringen sich aufgrund der ungestörten Superposition, ohne sich gegenseitig auszulöschen. Die Gasatome der Luft im Inneren des Rohres füllen dabei kleine (finite) gedachte Volumenelemente, die vom Druckfeld ein wenig bewegt werden, aber nicht sehr viel bewegt werden. Wollen wir die Situation auf den Draht der Spule übertragen, so korrespondieren die frei beweglichen Volumenelemente (welche Elektronen enthalten) im Draht mit den frei beweglichen Volumenelementen der Luft (welche Gasatome enthalten). Die Luftatome bewegen sich sehr schnell, ebenso wie die Elektronen im Draht. Betrachtet man aber Finite (kleine) Volumenelemente, die Elektronen bzw. Luftatome enthalten, so bewegen sich diese mit einer recht moderaten Geschwindigkeit, die wir im Falle der Elektronen als „Driftgeschwindigkeit“ bezeichnet, und im Falle der Akustik als „Schallschnelle“. Ändern wir nun an einer Stelle die elektrische Feldstärke bzw. den Druck (die Feldstärke des Druckfeldes), so verursacht dies eine Welle, die sich mit der typischen Propagationsgeschwindigkeit der Felder und Wellen ausbreitet, die wir in der Akustik als „Schallgeschwindigkeit“ bezeichnen und in der Elektrik als Signal-Laufgeschwindigkeit  $v = \frac{1}{\sqrt{LC}}$ .

Diese Signal-Laufgeschwindigkeit ist es, die die Ausbreitungsgeschwindigkeit der Felder und Wellen im Spulendraht unseres LC-Schwingkreises ausmacht. Somit ist es also tatsächlich die Propagationsgeschwindigkeit der Wechselwirkungsfelder, die wir über die Induktivität „L“ und die Kapazität „C“ des Schwingkreises steuern. Und diese Propagationsgeschwindigkeit der Wechselwirkungsfelder ist es, die für die Konversion von Raumenergie verantwortlich ist, so wie dies in [Tur 10a] und [Tur 10b] erläutert wurde.

Diese Erklärung dient dazu, den Lesern und Leserinnen verständlich zu machen, dass die hier gezeigte Arbeit mit einem LC-Schwingkreis, tatsächlich auf die Propagationsgeschwindigkeit der Felder zurückgeht (die die Wechselwirkung vermitteln). Somit wird die hier entwickelte Maschine also tatsächlich auf die bisher dargestellten Grundlagen der Raumenergie-Konversion zurückgeführt.

### 3. Auswertung der Rechenergebnisse am Raumenergie-Konverter

Eine allererste Beurteilung des Verhaltens des Systems wird interaktiv, während des Ablaufs des Programms, auf dem Bildschirm angezeigt. Die Daten, die man dabei im Programm sieht, sind in Tab. 1 wiedergegeben. Dies sind erste wichtige Größen zur Begutachtung des Raumenergie-Konverters.

Physikalische Größe	Erläuterung
$U_{cap,I,max} = \frac{q_{I,max}}{C_I}$	Maximum der Spannung am Input-Kondensator
$U_{cap,T,max} = \frac{q_{T,max}}{C_T}$	Maximum der Spannung im Turbo-Kondensator
$\dot{q}_{I,max}$	Maximum des Stroms in der Input-Spule
$\dot{q}_{T,max}$	Maximum des Stroms in der Turbo-Spule
$L_I \cdot \ddot{q}_{I,max}$	Maximum der Spannung über der Input-Spule
$L_T \cdot \ddot{q}_{T,max}$	Maximum der Spannung über der Turbo-Spule
$\dot{\phi}_{max}$	Maximum der Winkelgeschwindigkeit des Magneten (in rad/sec)
$\frac{\dot{\phi}_{max}}{2\pi}$	Maximum der Winkelgeschwindigkeit des Magneten
$E_{Anf} = E_{ges}(t=0)$	Anfangs-Energie im System
$E_{End} = E_{ges}(t=Ende)$	End-Energie im System
$E_{End} - E_{Anf}$	Energie-Zunahme im System
$\frac{E_{End} - E_{Anf}}{T_{ges}}$	Leistungs-Änderung im System
$P_{ent} = \int_0^{T_{ges}} R_{Last} \cdot \dot{q}_T^2 \cdot dt$	Am Lastwiderstand entnommene Gesamtenergie
$\frac{P_{ent}}{T_{ges}}$	Am Lastwiderstand entnommene mittlere entnommene Leistung
$E_{in} = \int_0^{T_{ges}} \dot{q}_I \cdot U_{in} dt$	Über Input-Spannung zugeführte Gesamt-Energie
$\frac{E_{in}}{T_{ges}}$	Über Input-Spannung zugeführten mittlere Leistung
$P_{mech} = M_{mech} \cdot \dot{\phi}$	Mechanische Leistungsentnahme, mit dem Drehmoment $M_{mech} = c_r \cdot \dot{\phi}$ und $c_r =$ Reibkoeffizient für geschwindigkeitsproportionale Leistungsentnahme.
$T_{ges}$	Betrachtungs-Dauer der Berechnung
Tab. 1: Übersicht über die am Bildschirm angezeigten Ergebnis-Daten.	

Weitere interessante Größen, vor Allen deren echtes dynamisches Verhalten (!) als Funktion der Zeit, sind in Tab. 2 wiedergegeben. Da diese Größen zur grafischen Darstellung der Daten in ein von Excel lesbares File exportiert werden, ist in der Tabelle nicht nur die Größe und ihre Berechnung angegeben, sondern auch die Excel-Kolumne, in der wir die Ergebnisdaten zur späteren Auswertung wiederfinden.

Excel-Kolumne	Physikalische Größe	Angaben zur Berechnung
A	$t$	Zeitskala
B,C,D	$q_T, \dot{q}_T, \ddot{q}_T$	Ladung und deren zeitliche Ableitungen im Schwingkreis der Turbo-Spule
E,F,G	$q_I, \dot{q}_I, \ddot{q}_I$	Ladung und deren zeitliche Ableitungen im Schwingkreis der Input-Spule
H,I,J	$\varphi, \dot{\varphi}, \ddot{\varphi}$	Drehwinkel und dessen zeitliche Ableitungen des rotierenden Magneten
K,L	$\psi_I, \psi_T$	Magnetischer Fluß durch die Spulen
M,N	$U_{ind,I}, U_{ind,T}$	In den Spulen induzierte Spannung
O,P	$E_{mag,I}, E_{mag,T}$	Energie in den Spulen: $E_{mag} = \frac{1}{2} \cdot L \cdot \dot{Q}^2$
Q,R	$E_{cap,I}, E_{cap,T}$	Energie in den Kondensatoren: $E_{cap} = \frac{1}{2} \cdot C \cdot U^2 = \frac{1}{2} \cdot \frac{Q^2}{C}$
S	$E_{rot}$	Energie der mechanischen Magnet-Rotation: $E_{rot} = \frac{1}{2} \cdot J \cdot \omega^2 = \frac{1}{2} \cdot J \cdot \dot{\varphi}^2$
T	$E_{ges}$	Gesamt-Energie im System: $E_{ges} = E_{mag,I} + E_{mag,T} + E_{cap,I} + E_{cap,T} + E_{rot}$
U	$E_{Last}$	Am Lastwiderstand entnommene Leistung, Turbo-seitig: $E_{Last} = R_{Last} \cdot \dot{q}_T^2$
V	U7	Input-Spannung
W	$P_{zuf}$	Zugeführte Leistung über die Input-Spannung
X	$c_r$	Reibkoeffizient (variiert im Falle einer Regelung)
Y	$P_{mech}$	Über Reibung entnommene Leistung $P_{mech} = P_{reib} = M_{reib} \cdot \dot{\varphi} = c_r \cdot \dot{\varphi}^2$
Z	NULL	Leerspalte in Reserve

Tab. 2: Übersicht über die nach Excel exportierten Daten.

Nun haben wir unseren DFEM-Algorithmus so weit ergänzt und erweitert, dass eine realitätsnahe Berechnung beliebig geformter elektrischer und magnetischer Raumenergie-Konverter möglich ist. Wir wollen dies im nachfolgenden Abschnitt an einer Beispiel-Geometrie vorführen.

#### 4. Rechenbeispiel eines konkreten Raumenergie-Motors

Die vorliegende Arbeit und der zugehörige DFEM-Algorithmus wurde letztlich entwickelt, um den vom Autor ersonnen an EMDR-Konverter („Elektro-Mechanischer Doppel-Resonanz“ Konverter) realitätsnah und mit brauchbarer Genauigkeit für praktisch realisierbare Experimente zu berechnen. Aus diesem Grunde ist die Geometrie dieses Raumenergie-Motors diejenige, die für ein praktisches Rechenbeispiel gewählt werden soll. Der Aufbau entspricht Abb.9.

Im hier vorgestellten Beispiel wird die Geometrie durch einen Satz von 32 Input-Parametern definiert, dazu kommen einige Naturkonstanten und einige abgeleitete Parameter zur Kontrolle der Eingabe (hier 18 Stück), die am Bildschirm angezeigt werden. Eine sinnvolle Vorgabe des Input-Parametersatzes benötigt eine genaue Abstimmung der Werte aufeinander und erfordert selbst vom Autor des Programms einige Stunden/Tag Zeit.

Achtung: Der unten gezeigte Parametersatz ist an die spezielle Geometrie des Aufbaus nach Abb.9 angepasst. Für andere geometrische Anordnungen sind andere Parametersätze zu erstellen. Im Programm selbst gibt es außerdem noch einige Unterprogramme, die der automatischen Vernetzung der Anordnung nach Abb.9 in Finite Elemente dienen. Auch diese Unterprogramme müssen für einen abgeänderten geometrischen Aufbau ggf. entsprechend abgeändert werden.

(a.) Definition der Geometrie des Aufbaus

Wir beginnen mit der Erläuterung der Eingabeparameter, die der realitätsnahe DFEM-Algorithmus benötigt. Dabei lassen wir aus didaktischen Gründen zunächst noch die mechanische Leistungsentnahme entfallen, weil dadurch das systematische Verständnis des Raumentriebsmotors Schritt für Schritt erleichtert wird.

{Naturkonstanten, keine Input-Parameter:}

- $\epsilon_0 = 8.854187817 \cdot 10^{-12}$  {As/Vm};      {Magnetische Feldkonstante, Naturkonstante}
- $\mu_0 = 4 \cdot \pi \cdot 10^{-7}$  {Vs/Am};      {Elektrische Feldkonstante, Naturkonstante}
- $c = \sqrt{1/\mu_0/\epsilon_0}$  {m/s};      {Lichtgeschwindigkeit, Naturkonstante}

{ Zum Lösen der Differentialgleichung und zur Darstellung der Ergebnisse:}

- AnzP:=5000000;      {Anzahl der tatsächlich berechneten Zeit-Schritte bei der iterativen Lösung}
- dt:=1E-6; {Sec. }      {Dauer der einzelnen Zeitschritte zur iterativen Lösung der Dgl.}
- Abstd:=1;      {Nur für die Vorbereitung, nicht zum Lösen der Dgl., Wert nicht verändern}
- PlotAnfang:=0000;      {Zum Daten-Exports nach Excel: Erster Plot-Punkt}
- PlotEnde:=5000000;      {Zum Daten-Exports nach Excel: Letzter Plot-Punkt}
- PlotStep:=200;      {Zum Daten-Exports nach Excel.: Schrittweite des Daten-Exports}

{Anmerkung: Da Excel nur maximal 32.767 Daten-Gruppen verarbeitet, können bei feiner Zeitdiskretisierung nicht alle tatsächlich berechneten Daten exportiert werden. In unserem Zahlenbeispiel wird nur jeder 200ste Punkt nach Excel exportiert.}

{Zur Geometrie-Definition der Spulen (DFEM-Vernetzung geschieht automatisch):}

- Spsw:=0.01;      {Angabe in Metern: Schrittweite der Spulen-Vernetzung}
- xo:=0; yo:=6; zo:=5;      {Geometrieparameter nach Abb.1, Angaben in Vielfachen von Spsw}
- Ninput:=80;      {Zahl der Wicklungen der Input-Spule, linke Spule in Abb.1}
- Nturbo:=12;      {Zahl der Wicklungen der Turbo-Spule, rechte Spule in Abb.1}
- nebeninput:=8;      {Windungen nebeneinander in der Input-Spule}
- ueberinput:=10;      {Windungen uebereinander in der Input-Spule}
- nebenturbo:=3;      {Windungen nebeneinander in der Turbo-Spule}
- ueberturbo:=3;      {Windungen uebereinander in der Turbo-Spule}

{Anmerkung: Hier wurden Spulen mit rechteckiger Geometrie nach Abb. 1 gewählt. Der Spulenquerschnitt der Input-Spule besteht dann aus 8 Windungen nebeneinander und 10 solchen Lagen übereinander. Der Spulenquerschnitt der Turbo-Spule besteht aus 3 Windungen nebeneinander und 3 solchen Lagen übereinander. Die Bezeichnung „übereinander“ bedeutet: radial nach außen angeordnet.}

{Zur Emulation des Dauermagneten:}

- Bsw:=1E-2;      {Angabe in Metern} {Magnetfeld-Speicherung in Zentimeter-Schritten}
- MEyo:=0.05;      {Halbe Länge des zylindrischen Dauermagneten}
- MEro:=0.01;      {Radius des zylindrischen Dauermagneten}
- MEI:=15899.87553475;      {Strom des Magnetfeld-Emulationsspulenpaares, Angabe in Ampere}

{Anmerkung: Wir verwenden hier einen zylindrischen Dauermagneten gemäß Abb. 8 und Abb.9. Auch diese Form kann bei Bedarf beliebig geändert werden.}

{Erläuterung zur Magnetfeld-Speicherung: Das Magnetfeld wird in einem hinreichend großen Volumen starr an den Magneten gekoppelt, wobei die Feldstärke-Werte in finiten Schritte in einem Daten-Array abgespeichert werden. Bei Bewegungen des Magneten wird das Feld starr mit dem Magneten mitgeführt. Die Schrittweite der Feldstärke-Speicherung im Daten-Array heißt „Bsw“.

{Anmerkung: Der krumme Wert bei der Eingabe des Emulationsstroms, der zur Emulation des Dauermagneten gehört, hat den Grund, die Feldstärke geeignet einzustellen. Zu den Kontrolldaten während der Dateneingabe-Phase gehört u.a. auch die vom Dauermagneten erzeugte Feldstärke. Diese kann abgelesen werden, um den Strom MEI so einzustellen, dass eine gewünschte Feldstärke erreicht wird. In unserem speziellen Rechenbeispiel wird ein

zylindrischer Dauermagnet emuliert, der an seinen beiden Enden jeweils eine Feldstärke von 1.0 Tesla erzeugt.}

{Weitere benötigte technische Größen:}

- DD:=0.10; {Meter} {Durchmesser des Spulendrahtes zur Angabe der Drahtstärke}
  - rho:=1.35E-8; {Ohm\*m} {Spezifischer elektrischer Widerstand von Kupfer, [Koh 96]}
  - rhoMag:=7.8E3; {kg/m^3} {Dichte des Magnet-Materials, Eisen, [Koh 96]}
  - CT:=36.61E-6; {Farad} {Kapazität des Kondensators, im Schwingkreis der Turbo-Spule}
  - Cl:=100E-6; {Farad} {Kapazität des Kondensators, im Schwingkreis der Input-Spule}
- {Anmerkung: In unserem Rechenbeispiel wird die Input-Spule zwar modelliert, um möglichen Anwendern des DFEM-Algorithmus die spätere Aufnahme einer zweiten Spule möglichst einfach zu machen. Aber diese Input-Spule (mitsamt den dazugehörigen Input-Schwingkreis) wird hier im Hauptprogramm nicht benutzt. Sie ist dort auskommentiert.}
- Rlast:=0.0111; {Ohm} {Ohm'scher Lastwiderstand im Turbo-Schwingkreis zur Energie-Entnahme}
  - UmAn:=50000; {U/min} {Anfangsbedingung mechanisch - Rotierender Magnet: Startdrehzahl}
  - Uc:=0; {Volt} Il:=0; {Ampere} {Anfangsbedingung elektrisch – Kondensatorspannung, Spulenstrom}
- {Anmerkung: Auch Selbstläufer benötigen eine Startenergie, um den Lauf der Maschine zu initiieren. Diese Startenergie kann mechanisch zugeführt werden, wie im vorliegenden Beispiel durch eine anfängliche von außen zugeführte Rotation des Magneten. Sie kann aber auch elektrische Energie zugeführt werden, zum Beispiel durch anfängliches Aufladen des Kondensators oder durch einen anfänglichen Spulenstrom. Auch Selbstläufer laufen nicht einfach so „von alleine“ los, sondern man muß sie starten.}
- U7(t)=0;
- {Anmerkung: Ist der Raumentriebs-Konverter kein Selbstläufer, sondern nur eine over-unity-Maschine, so muss während des Betriebs permanent Energie zugeführt werden. Dies kann zum Beispiel auf mechanische Weise über die Rotationsachse geschehen, ebenso gut aber auch auf elektrischem Wege. Letztere Variante wurde im vorliegenden Programm-Beispiel über eine Input-Spannung mit dem Unterprogramm „U7“ demonstriert. Der Lösungsteil der Differentialgleichung zeigt auch, wie man eine solche Input-Spannung in die Differentialgleichung aufnehmen kann. Allerdings wurde diese Spannung im vorliegenden Rechenbeispiel nicht eingesetzt, weil der EMDR-Konverter ein Selbstläufer ist und somit keine Input-Spannung benötigt.}
- {Abgeleitete Parameter, zu Kontrollzwecken, aus obigen Parametern berechnet, keine Eingabe:}
- DLI:=4\*(yo+zo)\*Spw\*Ninput; {Meter} {Länge des Spulendrahtes, Input-Spule}
  - DLT:=4\*(yo+zo)\*Spw\*Nturbo; {Meter} {Länge des Spulendrahtes, Turbo-Spule}
  - RI:=rho\*(DLI)/(pi/4\*DD\*DD); {Ohm} {Ohm'scher Widerstand des Spulendrahtes, Input-Spule}
  - RT:=rho\*(DLT)/(pi/4\*DD\*DD); {Ohm} {Ohm'scher Widerstand des Spulendrahtes, Turbo-Spule}
  - Breitel:=nebeninput\*DD; HoeheI:=ueberinput\*DD; {Breite und Höhe des Input-Spulenlörpers}
  - BreiteT:=neberturbo\*DD; HoeheT:=ueberturbo\*DD; {Breite und Höhe des Turbo-Spulenlörpers}
  - fkl:=Sqrt(HoeheI\*HoeheI+4/pi\*2\*yo\*2\*zo)/HoeheI; {Korrekturfaktor zur Induktivität der kurzen Input-Spule nach \*1 von S.16}
  - fkT:=Sqrt(HoeheT\*HoeheT+4/pi\*2\*yo\*2\*zo)/HoeheT; {Korrekturfaktor zur Induktivität der kurzen Turbo-Spule nach nach \*1 von S.16}
  - LI:=muo\*(2\*yo+Bretel)\*(2\*zo+Bretel)\*Ninput\*Ninput/(HoeheI\*fkl);  
{Geometrische Mittelung => Induktivität der Input-Spule}
  - LT:=muo\*(2\*yo+Bretel)\*(2\*zo+Bretel)\*Nturbo\*Nturbo/(HoeheT\*fkT);  
{Geometrische Mittelung => Induktivität der Turbo-Spule}
  - omT:=1/Sqrt(LT\*CT); {Kreis-Eigenfrequenz des Turbo-Spulen-Schwingkreises aus LT & CT}
  - TT:=2\*pi/omT; {klassische Schwingungsdauer des Turbo-Spulen-Schwingkreises aus LT & CT}
  - Mmag:=rhoMag\*(pi\*MEro\*MEro)\*(2\*MEyo); {Masse des Magneten}
  - J:=Mmag/4\*(MEro\*MEro+4\*MEyo\*MEyo/3); {Trägheitsmoment des Magneten bei  
Rotation um Querachse, siehe \*2 von S.13 und Dubbel S.B-32}

- $\omega_{An} = \frac{U_{mAn}}{60} \cdot 2 \cdot \pi$ ;      {Start-Winkelgeschwindigkeit (rad/sec.) des rotierenden Magneten}
  - $U_{mSec} = \frac{U_{mAn}}{60}$ ;      {Start-Winkelgeschwindigkeit, rot. Magn. in Umdrehungen pro Sekunde}
- {Anmerkung: Manche dieser Größen werden nicht nur zur Kontrolle benötigt, sondern auch für die weitere Verarbeitung im DFEM-Algorithmus.}

Mit den gezeigten Parametern wird ein Raumenergie-Motor modelliert, wie er in Abb. 11 zu sehen ist. Die Zeichnung ist sogar maßstabsgetreu. Man sieht nur die Turbo-Spule, da die Input-Spule weggelassen wurde, weil diese Letztgenannte beim EMDR-Konverter, den der Autor ersonnen hat, nicht benötigt wird.

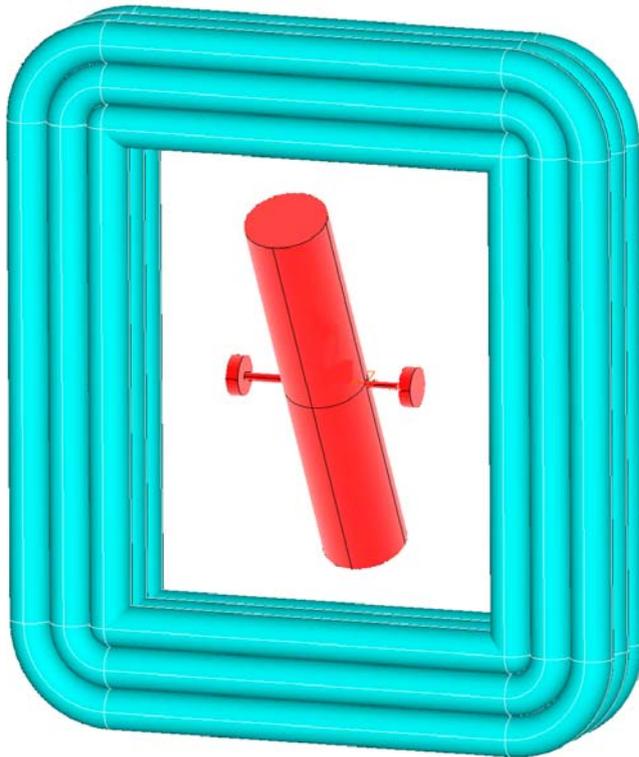


Abb.11:

(gezeichnet unter Verwendung [Ans 08], gerechnet wurde mit dem Ansys allerdings nicht, weil es die Dynamik nicht wie gefordert beherrscht.)

EMDR-Konverter mit einem rotierenden Magneten (in roter Farbe: Länge 10 cm, Dicke 2 cm). Ebenfalls in roter Farbe wurde eine dünne Rotationsachse eingezeichnet, um die der zylindrische Stabmagnet sich dreht. An den beiden Enden der Rotationsachse befindet sich jeweils ein Lager zur Halterung der Achse (ebenfalls in roter Farbe).

Die Windungen der Turbo-Spule sind in blauer Farbe gezeichnet, so dass die vom Magneten erzeugten Feldlinien bei der Rotation des Magneten die Spulendrähte senkrecht schneiden.

Aufgrund der praktischen Dimensionierung des EMDR-Konverters (siehe unten) ist es nötig, die Spule aus sehr wenigen sehr dicken Drähten herzustellen. Der Spulenquerschnitt beträgt  $5 \times 6 \text{ cm}^2$ , die Dicke des Drahtes liegt bei 10 mm.

Die Zeitschritte bei der numerisch-iterativen Lösung der Differentialgleichung müssen fein genug gewählt werden, dass jede einzelne Periode aus hinreichend vielen einzelnen Rechenschritten zusammengesetzt wird. Dazu sind die im obigen Datensatz gezeigte Zeitintervalle von  $dt = 1 \mu\text{sec}$ . bei einer Drehzahl von 50.000 Umdrehungen/Minute bei weitem nicht ausreichend. In der praktischen Benutzung des DFEM-Algorithmus beginnt man mit solch groben Zeitschritten nur, um ein erstes Gefühl für das Verhalten der Maschine zu bekommen. Mit diesen groben Zeitschritten lässt sich dann zum Beispiel die Einstellung der Systemparameter für die Maximierung der gewandelten Leistung aus Raumenergie in einer groben Abschätzung ermitteln. Hat man dieses Ziel erreicht, so müssen die Zeitschritte sukzessive verkleinert werden, um die Konvergenz des Algorithmus zu untersuchen. Diese ist erreicht, sobald eine weitere Verfeinerung der Zeitschritte keine wesentliche Veränderung der Zahlenwerte in den Ergebnissen mehr nach sich zieht. Sobald die Zeitschritte so weit verfeinert sind, dass die Konvergenz zufriedenstellend ist, beginnt die Fein-Abstimmung der Systemparameter zu einer erneuten abermaligen Maximierung der gewandelten Leistung aus Raumenergie.

Um die Optimierung der Systemparameter zu erleichtern, wurde dem Programm eine Bildschirm-Ausgabe angefügt, die eine rasche Evaluierung der Tauglichkeit der Systemparameter ermöglicht. Dabei werden folgende Daten angezeigt:

- Die Startenergie im System, die in die Differenzialgleichungen über die Anfangsbedingungen eingebracht wird. Sie errechnet sich u.a. aus den oben genannten Parametern  $UmAn$ ,  $Uc$ ,  $Il$ .
- Die Energie im System am Ende des Zeitraums der Berechnung. Hierzu werden alle im System enthaltenen Energiebeträge addiert, also die Bewegungsenergie des/der Magneten, die Energien aller Spulen und auch die Energien aller Kondensatoren.
- Die Energie-Zunahme des Systems während des Berechnungs-Zeitraums, die angegeben wird als Differenz zwischen der Startenergie und der End-Energie. Ist die Energie-Zunahme zuzüglich der entnommenen Energie positiv, so konvertiert das System Raumenergie in klassische Energie. Ist die Energie-Zunahme plus die entnommene Energie hingegen negativ, so konvergiert das System umgekehrt klassische Energie in Raumenergie, oder es geht Energie durch Reibungsverluste verloren (wobei wir allerdings Reibung in der Berechnung berücksichtigen). Ein idealer klassischer Elektromotor würde zum Beispiel (wenn er reibungsfrei laufen könnte), als Differenz zwischen der End-Energie und der Anfangs-Energie genau die während des Betriebs zugeführte Energie abzüglich der entnommenen Energie anzeigen.
- Die Leistungs-Änderung im System, die sich ganz simpel aus der Energie-Zunahme im Bezug auf die gesamte Berechnungs-Dauer ergibt.
- Die während des Betriebs am Ohm'schem Widerstand entnommen der Energie, und die zugehörige dort entnommene Leistung. Nicht angezeigt werden Ohm'sche Verluste an den Spulendrähten; diese werden zwar in die Berechnung einbezogen und tatsächlich als Leistungs- bzw. Energie- Verluste berücksichtigt, aber sie werden nicht angezeigt, denn man kann sie nicht beeinflussen.
- Die bei mechanischer Belastung entnommene mittlere mechanische Leistung und die mechanische entnommene Gesamtenergie. Sie kann durch geeignete Maßnahmen an der drehenden Welle abgegriffen werden, die durch die Mitte des rotierenden Dauermagneten verläuft. Jetzt, in den Abschnitten 4 und 5 wird sie noch nicht betrachtet, aber später in Abschnitt 6 wird sie im Detail besprochen.
- Außer der mechanisch entnommenen Leistung und Energie gibt es aber auch noch eine mechanische Energiezunahme im System, die dort verbleibt.
- Die über die Input-Spannung in die Input-Spule zugeführte Leistung wird zwar angezeigt, aber sie ist im vorliegenden Rechenbeispiel NULL, weil keine Energie zugeführt wird. Der EMDR-Konverter ist ein Selbstläufer. Die Anzeige diese Energie wird nur exemplarisch vorgeführt, um den Lesern und Leserinnen der Publikation die Benutzung des DFEM-Algorithmus zu erleichtern.
- Die gesamte Dauer der Simulations-Berechnung, die sich als die Summe aller Zeitschritte „dt“ ergibt. Gemeint ist nicht die CPU-Zeit der Berechnung, sondern die der Simulation zugrunde liegende Betriebsdauer des Raumenergie-Konverters.

Die vorangehend beschriebene kurze Bildschirm-Evaluation einiger ausgewählter Ergebnisse hilft, um eine einfache und rasche Kontrolle des Betriebszustands des Konverters zur Verfügung zu stellen, damit man Aussagen zur weiteren Variation und Optimierung der Systemparameter machen kann. Durch wiederholtes Variieren der Systemparameter und wiederholt erneutes Berechnen des veränderten Konverters lässt sich ein Aufbau ermitteln, der als Raumenergie-Konverter funktionieren sollte (zumindest theoretisch).

Hat man diesen Aufbau ermittelt, so empfiehlt sich die vollständige Evaluierung des Systems, und zwar unter grafischer Kontrolle der Systemvariablen als Funktion der fortlaufenden Zeit. Dies geschieht am bequemsten, indem man die in Tab. 2 aufgezählten Excel-Kolumnen direkt in Excel graphisch darstellt. Dabei müssen zuallererst und vor Allem Fragen beantwortet werden wie etwa diese:

- Hat das System schon seinen stabilen Betriebszustand bereits erreicht ?  
Dies erkennt man zum Beispiel an der zeitlichen Entwicklung der Winkelgeschwindigkeit  $\dot{\varphi}$  des Systems, die im stabilen Betriebszustand um einen konstanten Wert oszilliert. Auch der Spulenstrom und die Kondensatorspannung müssen um jeweils einen konstanten Wert oszillieren. Ist dies nicht der Fall, so ist der Konverter entweder noch in der Anlaufphase, oder er läuft überhaupt nicht in einem stabilen Betriebszustand. Um zwischen diesen beiden Möglichkeiten unterscheiden zu können, empfiehlt sich eine Verkürzung der Zeitschritte „dt“ bei gleichzeitiger Vergrößerung der gesamten Beobachtungsdauer. (Letzere Maßnahme ist wichtiger als die erstgenannte.)
- Wird beim Anlauf der Maschine genug mechanische Leistung erzeugt, dass die Maschine nicht aufgrund der Reibung zum Stillstand kommt ?  
Hierzu kontrolliert man die mechanische Rotationsenergie (des Dauermagneten) als Funktion der Zeit. Im Falle einer mechanischen Leistungsentnahme (siehe Abschnitt 6) kann man gezielt die Reibungskoeffizienten vorgeben und die zugehörige entnommene Leistung anzeigen lassen.
- Man kontrolliert auch die zeitlichen Verläufe der Ströme und Spannungen im LC-Schwingkreis graphisch, um sich nicht auf die simple Abschätzung des Maximums der einfachen Bildschirm-Ausgabe verlassen zu müssen. Bei einem ordentlichen Betrieb der Maschine als Raumenergie-Konverter sind die Spannungen und Ströme meistens ziemlich groß, und man muss für den praktischen Aufbau sicher sein, die Spulendrähte und die Kondensatoren nicht zu überlasten.

Allerdings sei erwähnt, dass die Laufzeit des DFEM-Algorithmus bei feinen Zeitschritten (z.B. im Bereich einiger Nanosekunden) durchaus CPU-Zeiten im Bereich einiger Minuten bis hin zu einigen Stunden (je nach Gesamtdauer der der Berechnung zu Grunde liegenden Zeitspanne) erfordern kann.

## 5. Ein konkreter EMDR-Raumenergie-Konverter

Die Inbetriebnahme des im Anhang befindlichen DFEM-Programms benötigt ein Datenfile namens 'schonda', welches gemeinsam mit dem Quellcode des DFEM-Programms kostenlos im Internet von den Seiten des Autors der vorliegenden Publikation heruntergeladen werden kann. Dieses Datenfile 'schonda' hat lediglich den Zweck, CPU-Zeit zu sparen, sofern in der Initialisierungsphase des Hauptprogramms die dort eingegebenen Geometrieparameter seit dem letzten Lauf des Programms nicht verändert wurden. Die Initialisierungsphase dient nämlich der automatischen Generierung der Vernetzung (in finite Elemente), der Entwicklung der Fourierreihe und einiger Vorarbeiten, die die Geometrie des Systems beschreiben. Wurde diese Vernetzung aber seit dem letzten Lauf des Programms nicht geändert, so ist eine erneute Berechnung der Initialisierungsphase nicht nötig, und deshalb können die Ergebnisse der Initialisierungsphase direkt aus dem Datenfile 'schonda' aus dem vorangehenden Lauf des Programms entnommen werden, so dass die CPU-Zeit für den Initialisierungsprozess eingespart werden kann. Dies erleichtert den wiederholten Ablauf des Programms zur Optimierung der Systemparameter wesentlich.

Wir wollen jetzt in Abschnitt 5 die Systemparameter gemäß der Einstellung diskutieren, wie sie im Anhang des vorliegenden Artikels wiedergegeben ist.

Die typische Konstruktion eines EMDR-Konverters beginnt mit der Suche eines geeigneten Magneten. Ist er gefunden, und ist seine Feldstärke vermessen (zum Beispiel mit einer Hall-Sonde), so gibt man seine Abmessungen in die Eingabe-Datenzeilen ein und konstruiert sodann eine Stromverteilung, die anhand geeigneter Dauermagnet-Emulationsspulen das gemessene Feld theoretisch reproduziert.

Danach moduliert man die Turbo-Spule und gibt die restlichen Daten für den DFEM-Algorithmus ein. Zuletzt angepasst wird die Startdrehzahl des rotierenden Magneten, zu allerletzt die Kapazität des Kondensators im Turbo-Schwingkreis. Das Kriterium für die Startdrehzahl des rotierenden Magneten geht auf die zur Verfügung stehende Lagerung zurück. Man darf die vorhandene Lagerung nicht mit der End-Drehzahl im stabilen Betriebszustand der Maschine überlasten. Das Kriterium für die Justierung der Kapazität ist die maximal erreichbare Magnet-Rotationsdrehzahl. Eine große Drehzahl ist für die Wandlung der Raumenergie günstiger als eine kleine Drehzahl, und die maximale Drehzahl, die auf dem Bildschirm ausgegeben wird, ist im Vergleich zur Startdrehzahl ein direktes Maß für die Zunahme der Drehzahl während der berechneten Betriebsdauer. Zunächst einmal wird bei sehr kleinem Lastwiderstand die Kapazität so justiert, dass diese Drehzahl-Zunahme maximal wird.

Dabei stellt man häufig fest (je nach gewählter Geometrie), dass die gewandelte mechanische Leistung wesentlich größer ist, als der elektrische Leistungsgewinn im System. (Darin liegt auch der Grund, warum wir später in Abschnitt 6 die eigentliche Nutzleistung des Raumenergie-Motors mechanisch an der drehenden Welle abgreifen wollen.)

Man variiert nun sukzessive in kleinen Schritten den Lastwiderstand und beobachtet dabei die Veränderung des mechanischen und des elektrischen Leistungsgewinns (wobei man immer wieder den Kondensator nachjustieren muß). Je nach Einstellung des Konverters können beide Leistungen gleichzeitig zunehmen oder abnehmen oder auch gegenläufig. Man darf den Lastwiderstand nicht zu stark erhöhen, sonst dämpft man die Maschine derart massiv, dass sie nicht mehr ordentlich anlaufen kann. Ist die mechanische Leistungsaufnahme der Maschine groß, ggf. sogar deutlich größer als die elektrische Leistungsabgabe am Lastwiderstand, so stören wir uns daran überhaupt nicht, denn eine starke mechanische Leistungszunahme verleiht der Maschine später in Abschnitt 6 eine hohe Nutzleistung. Im übrigen stellt eine hohe mechanische Leistungszunahme sicher, dass die mechanisch gewonnene Leistung mit Sicherheit ausreicht, um die Reibung zu überwinden.

Bei der Justierung erkennt man, dass bereits in der theoretischen Berechnung sowohl der Lastwiderstand als auch die Kapazität mit einer ziemlich hohen Genauigkeit eingestellt werden müssen. Dies macht uns eindrucksvoll klar, dass entsprechende Anforderungen hinsichtlich der feinen Variation und der Einstellung sowohl an den Lastwiderstand als auch an die Kapazität auch im praktischen Aufbau gerichtet werden. Hier sind drei numerisch signifikante Stellen durchaus sinnvoll. Im Zusammenhang mit den nachfolgend dargestellten Strom- und Spannungs- Werten im elektrischen Schwingkreis stellt dies sicherlich erhebliche Anforderungen an die beiden Bauteile *Lastwiderstand* und *Kondensator* dar.

Das hier gezeigte Rechenbeispiel ist nicht im Bezug auf die erzielbare Leistung ausoptimiert, da dies erst in Abschnitt 6 unter Berücksichtigung der zur Verfügung gestellten mechanischen Leistung sinnvoll sein wird.

Trotzdem sei hier angemerkt, dass man die aus der Raumenergie gewandelte Leistung deutlich erhöhen kann, wenn man die Zahl der Windungen der Turbo-Spule ein wenig erhöht (was allerdings die Spannung am Kondensator deutlich erhöht). Ein sehr wirkungsvolles Mittel zur Erhöhung der erzielbaren Leistung ist auch die Drehzahl. Mit ihr steigt die erreichbare Leistung gewaltig an. Die hier verwendeten 30.000 Umdrehungen/min werden als eher moderat angesehen. Diese Begrenzung dient dazu, die Anforderungen an die Lagerung des rotierenden Dauermagneten nicht zu groß werden zu lassen (was für erste Prototypen ganz angenehm sein kann). Denkt man jedoch an typische Drehzahlen von Turbolader-Rotoren (im Automobilbereich) im Bereich von 100.000 Umdrehungen/min oder noch wesentlich höher, so ist klar, dass man eine gewaltige Leistungssteigerung durch eine Erhöhung der Drehzahl auch bei dem hier vorgestellten Raumenergie-Konverter ohne allzu große Probleme erreichen kann. Noch höhere Drehzahlen von einigen 100.000 Umdrehungen/min, wie etwa in der Zahnmedizin üblich, oder bei Turbo-Vakuumpumpen würden mit Sicherheit dazu führen, dass auch ein dicker Kupferdraht als Spule schnell beginnen würde zu erhitzen und ggf. zu schmelzen.

Will man die Berechnung und deren Ergebnisse dokumentieren, so gibt man unmittelbar vor dem endgültigen Verlassen des Programms (vor dem letzten <Wait>) ein „D“ ein, gefolgt von „Return“. Das Programm zeichnet nun die Ergebnisse der Berechnung auf, und zwar in einem File namens „Auswertung“, welches man mit Textverarbeitungsprogramm lesen kann. Die dokumentierten Ergebnisse in dem hier beschriebenen Rechenbeispiel sehen dabei in etwa wie folgt aus:

### DFEM-Simulation eines EMDR-Motors (hier noch ohne mechanische Leistungs-Entnahme)

Parameter zum Lösen der Dgl. und zur Darstellung der Ergebnisse:

AnzP = 10000000 Zum Lösen der Dgl.: Anzahl der tatsächlich berechneten Zeit-Schritte  
 dt = 2.000E-0007 {Sekunden} Dauer der Zeitschritte zur iterativen Lsg. der Dgl.  
 Abstd= 1 {Nur Vorbereitung, nicht zum Lösen der Dgl.: Jeden wievielten Punkt soll ich plotten ins Excel?}  
 PlotAnfang = 0 {Zum Lösen der Dgl.: Erster-Plot-Punkt: Anfang des Daten-Exports nach Excel}  
 PlotEnde = 10000000 {Zum Lösen der Dgl.: Letzter-Plot-Punkt: Ende des Daten-Exports nach Excel}  
 PlotStep = 400 {Zum Lösen der Dgl.: Schrittweite des Daten-Exports nach Excel}

Es folgt die Eingabe der beiden Spulen, vgl. Zeichnung \*2 von S.1 :

Die Spulen werden nach Vorgabe der Geometrieparameter automatisch vernetzt.

Spsw = 0.010000 Angabe in Metern: Die Spulen-Aufgliederung ist in Spsw-Schritten  
 xo = 0, Angaben in Vielfachen von Spsw, Geometrieparameter nach Zeichnung\*2 von S.1  
 yo = 6, Angaben in Vielfachen von Spsw, Geometrieparameter nach Zeichnung\*2 von S.1  
 zo = 5, Angaben in Vielfachen von Spsw, Geometrieparameter nach Zeichnung\*2 von S.1  
 Ninput = 100 Zahl der Wicklungen der Input-Spule  
 Nturbo = 9 Zahl der Wicklungen der Turbo-Spule  
 nebeninput = 10 Windungen nebeneinander in der Input-Spule  
 ueberinput = 10 Windungen uebereinander in der Input-Spule  
 neberturbo = 3 Windungen nebeneinander in der Turbo-Spule  
 ueberturbo = 3 Windungen uebereinander in der Turbo-Spule

Bsw = 1.0E-0002 Magnetfeld-Speicherung nach \*1 von S.2 in Zentimeter-Schritten

Ich emuliere hier das Magnetfeld eines 1T-Magneten durch ein Spulenpaar nach \*1 von S.5:

MEyo = 5.00000E-0002 y-Koordinaten der Magnetfeld-Emulationsspulen nach \*1 von S.5  
 MEro = 1.00000E-0002 Radius der Magnetfeld-Emulationsspulen nach \*1 von S.5  
 MEI = 1.58998E+0004 Strom des Magnetfeld-Emulationsspulenpaares nach \*1 von S.5, Angabe in Ampere

Allgemeine technische Größen:

DD = 0.0100000 {Meter} {Durchmesser des Spulendrahtes zur Angabe der Drahtstärke  
 rho = 1.350000000000000E-0008 {Ohm\*m} {Spez. elektr. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193  
 rhoMag = 7.800000000000000E+0003 {kg/m^3} {Dichte des Magnet-Materials, Eisen, Kohlrausch Bd.3  
 CT = 9.83000E-0005 {Farad} {Kapazität des Kondensators, der mit in der Turbo-Spule (!) in Reihe geschlattet  
 CI = 1.00000E-0004 {Farad} {Kapazität des Kondensators, der mit in der Input-Spule (!) in Reihe geschlattet

Sonstige (zur Eingabe):

Rlast = 6.400000E-0002 {Ohm} Ohm'scher Lastwiderstand im LC-Turbo-Schwingkreis  
 UmAn = 30000.00 {U/min} Anfangsbedingung mechanisch - Rotierender Magnet: Startdrehzahl  
 Uc = 0.00 {Volt} Anfangsbedingung elektrisch - Kondensatorspannung am TURBO-Kondensator  
 Il = 0.00 {Ampere} Anfangsbedingung elektrisch - Spulenstrom im TURBO-Schwingkreis

Abgeleitete Parameter. Die Größen werden aus den obigen Parametern berechnet, es ist keine Eingabe möglich:

DLI:=4\*(yo+zo)\*Spsw\*Ninput = 44.00000 {Meter} Länge des Spulendrahtes, Input-Spule  
 DLT:=4\*(yo+zo)\*Spsw\*Nturbo = 3.96000 {Meter} Länge des Spulendrahtes, Turbo-Spule  
 RI:=rho\*(DLI)/(pi/4\*DD\*DD) = 0.00756 {Ohm} Ohm'scher Widerstand des Spulendrahtes, Input-Spule  
 RT:=rho\*(DLT)/(pi/4\*DD\*DD) = 0.00068 {Ohm} Ohm'scher Widerstand des Spulendrahtes, Turbo-Spule  
 Breitel:=nebeninput\*DD = 0.10000 Breite und Höhe des Input-Spulenlörpers  
 Hoehel:=ueberinput\*DD = 0.10000 Breite und Höhe des Input-Spulenlörpers  
 BreiteT:=neberturbo\*DD = 0.03000 Breite und Höhe des Turbo-Spulenlörpers  
 HoeheT:=ueberturbo\*DD = 0.03000 Breite und Höhe des Turbo-Spulenlörpers  
 fkl:=Sqrt(Hoehel\*Hoehel+4/pi\*2\*yo\*2\*zo)/Hoehel = 123.61179 Korrekturfaktor zur Induktivität der kurzen Input-Spule  
 fkT:=Sqrt(Hoehel\*Hoehel+4/pi\*2\*yo\*2\*zo)/HoeheT = 412.02703 Korrekturfaktor zur Induktivität der kurzen Turbo-Spule  
 LI:=muo\*(2\*yo+Bretel)\*(2\*zo+Bretel)\*Ninput\*Ninput/(Hoehel\*fkl) = 1.24238647244960E-0001 Induktivität Input-Spule

$LT = \mu_0 \cdot (2 \cdot y_0 + \text{Breite}_T) \cdot (2 \cdot z_0 + \text{Breite}_T) \cdot N_{\text{turbo}} \cdot N_{\text{turbo}} / (\text{Höhe}_T \cdot f_k T) = 9.93606632469255E-0004$  Induktivität Turbo-Spule  
 $\omega_{mT} = 1 / \sqrt{LT \cdot CT} = 3.19974964955735E+0003$  Kreis-Eigenfrequenz des Turbo-Spulen-Schwingkreises aus LT & CT  
 $TT = 2 \cdot \pi / \omega_{mT} = 1.96364903362012E-0003$  Schwingungsdauer des Turbo-Spulen-Schwingkreises aus LT & CT.  
 $M_{\text{mag}} = \rho_{\text{mag}} \cdot (\pi \cdot M_{\text{Ero}} \cdot M_{\text{Ero}}) \cdot (2 \cdot M_{\text{Eyo}}) = 0.245$  kg Masse des Magneten (Rotation um Querachse !!)  
 $J = M_{\text{mag}} / 4 \cdot (M_{\text{Ero}} \cdot M_{\text{Ero}} + 4 \cdot M_{\text{Eyo}} \cdot M_{\text{Eyo}} / 3) = 2.10329628157837E-0004$  Trägheitsmoment des Magneten bei Rotation um Querachse

Anzeige einiger auszurechnender Parameter:  
 Magnet: Start-Winkelgeschw.:  $\omega_{mAn} = 3141.592654$  rad/sec  
 Magnet: Startdrehzahl, Umdr./sec.:  $UmSec = 500.0000000000$  Hz  
 Masse des Magnet = 0.245044 kg  
 Traegheitsmoment Magnet bei QUER-Rotation 2.10329628157837E-0004 kg\*m^2  
 Gesamtdauer der Betrachtung: 2.00000000000000E+0000 sec.  
 Excel-Export: 0.00000E+0000... 2.00000E+0000 sec., Step 8.00000E-0005 sec.  
 Das sind 25000 Datenpunkte (also Zeilen).

\*\*\*\*\*

Einige Ergebnisse der Berechnung:  
 Anfangs-Energie im System: 1037.93511187 Joule  
 End-Energie im System: 1171.18167853 Joule  
 Leistungs-Aenderung im System: 66.62328333 Watt  
 Am Lastwiderstand entnommene Gesamtenergie = 13.29139772 Joule  
 entsprechend einer mittleren entnommenen Leistg: 6.64569885828765E+0000 Watt  
 Ueber Input-Spannung zugefuehrte Gesamt-Energie: 0.00000000000000E+0000 Joule  
 entsprechend einer mittleren zugefuehrten Leistg: 0.00000000000000E+0000 Watt  
 bei einer Betrachtungs-Dauer von 2.00000000000000E+0000 sec.

Oben sieht man eine Protokollierung der eingegebenen und der daraus hergeleiteten Größen. Unten findet man den Abschnitt „Einige Ergebnisse der Berechnung“, dem wir jetzt unsere Aufmerksamkeit zuwenden wollen:

Bei einer Betrachtungs-Dauer von 2 Sekunden (die benötigte CPU-Zeit liegt bei den meisten Computern deutlich höher) wird eine mechanische Leistungszunahme von 66.62 Watt im System beobachtet, und dazu eine elektrische Leistungsabgabe an den Lastwiderstand von 6.645 Watt. Beide Leistungen sind ohne klassische Energiequelle erzeugt worden, wurden also der Raumenergie (zumindest aus einer bisher unbekanntem Energiequelle) entnommen und gewandelt.

Vergrößert man die Betrachtungs-Dauer, so kann eine Verringerung der Leistungs-Wandlung beobachtet werden. Dies liegt aber nicht dominant an unvollständiger Konvergenz des Algorithmus (aufgrund der Größe der Zeitschritte), sondern es hat seinen Grund in der Tatsache, dass die Maschine im Anlaufbetrieb ein anderes Verhalten zeigt, als nach dem Einschwingen in ein stabiles Fließgleichgewicht. Wir erkennen dies eindeutig in Abb.12, bei der die Spalte „I“ des Excel-Datenprotokolls angezeigt ist, die die Winkelgeschwindigkeit des rotierenden Magneten enthält (abzurufen in dem vom Programm automatisch erzeugten File „test“.)

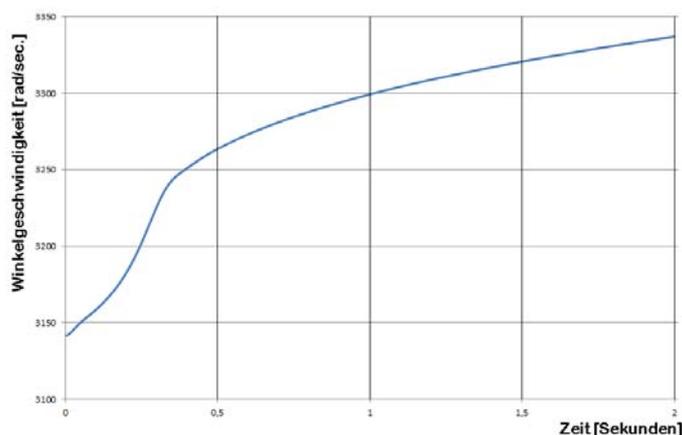


Abb.12:

Winkelgeschwindigkeit des rotierenden Magneten, die sich als Funktion der Zeit und ohne klassische Energiezufuhr erhöht. Zu Beginn des Betriebs sieht man die starke Energie-Aufnahme während des Hochlaufs der Maschine. Gegen Ende der Betrachtungs-Dauer konvergiert der Betriebszustand allmählich gegen die Sättigung, hat diese aber noch nicht erreicht.

Die Kondensator-Spannung entnimmt man aus der Ladung des Kondensators, deren zeitliche Entwicklung wir in Abb.13 sehen. Allerdings erkennt man aufgrund der begrenzten Auflösung der Grafik nur die Einhüllende der tatsächlichen Schwingung, da bei der tatsächlichen Berechnung 10.000.000 Punkte berücksichtigt wurden, die bei einer Computer-Grafik natürlich nicht alle dargestellt werden können.

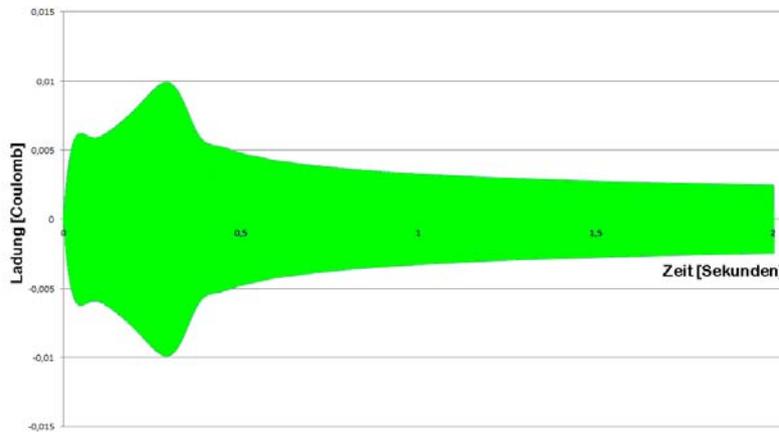


Abb.13:

In grüner Farbe erkennt man die Einhüllende der Ladung als Funktion der Zeit, die in Form einer Schwingung in den Kondensator hinein fließt und diesen wieder verlässt.

Das Maximum der Spannungsamplitude liegt bei  $U_{C,max} = \frac{Q_{max}}{C_{Turbo}} = \frac{0.01C}{98.3\mu F} \approx 102 Volt$ . Es sollte kein

Problem sein, solche Kondensatoren zu bekommen. Erhöht man die Zahl der Windungen, so kann man auch noch deutlich größere Spannungen anfahren und somit die elektrisch entnehmbare Leistung wesentlich erhöhen (und damit übrigens auch die mechanisch zur Verfügung gestellte Leistung).

Der Spulenstrom ist in Abb. 14 mit einem Maximum von etwas über 30 A zu erkennen. Erhöht man die Zahl der Windungen oder senkt man den Lastwiderstand ab, so lässt sich nach geeigneter Justierung der restlichen Systemparameter eine wesentliche Erhöhung des Spulenstroms erwirken. Bei einem Spulendraht wie in Abb.11 mit einem Durchmesser von 10 mm (Querschnittsfläche 78,5 mm<sup>2</sup> ist dies keinerlei Problem.

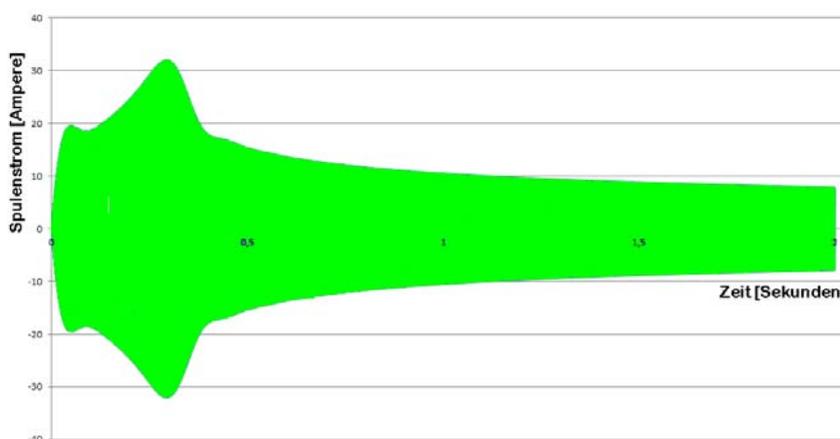


Abb.14:

Einhüllende des Spulenstroms als Funktion der Zeit.

Die Spulenspannung, deren graphische Darstellung eine ähnliche Form der Einhüllenden zeigt, wie Abb.13 und Abb.14, sollte eine vergleichbare Amplitude haben wie die Kondensatorspannung, wenn der Konverter einigermaßen vernünftig eingestellt ist. Mit einem Wert von 99 Volt ist dieses Kriterium überraschend gut erfüllt.

Die Beschleunigung des rotierenden Magneten, deren Einhüllende wir in Abb. 15 sehen, zeigt eine merkliche Oszillation innerhalb einer Periode, woran man erkennt, daß zwischen den Ladungen im LC-Schwingkreis und den Magneten durchaus merkliche Kräfte ausgetauscht werden. (Mit entsprechender mechanischer Festigkeit muss die Anordnung im praktischen Experiment aufgebaut sein.) Das Fließgleichgewicht des eingeschwungenen Zustands ist erreicht, wenn diese Oszillation symmetrisch um die Abszisse stattfindet - was hier noch nicht ganz der Fall ist, wie wir auch schon aus der Diskussion zu Abb.12 erwarten. Da auch am Ende des Berechnungs-Zeitraumes der Mittelwert der Winkelbeschleunigung noch deutlich im positiven liegt, hat der Rotor offensichtlich noch nicht die Maximaldrehzahl erreicht.

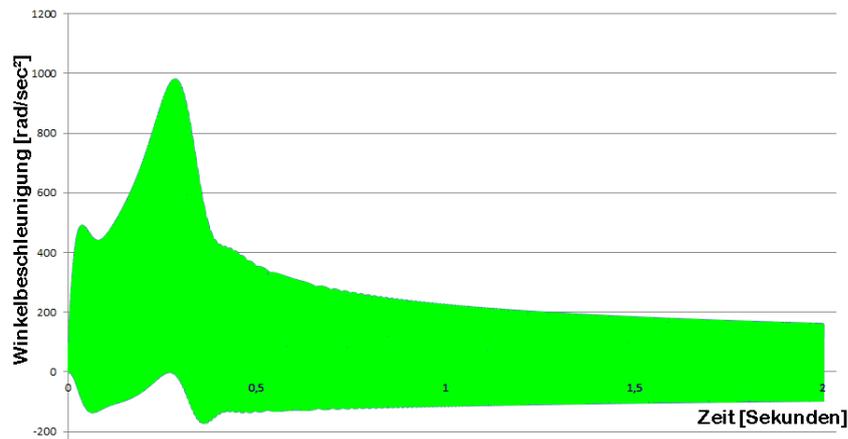


Abb.15:

Einhüllende der Winkelbeschleunigung als Funktion der Zeit.

Die Frage nach der Qualität der numerischen Iteration beantworten wir nach Betrachtung eines hochauflösenden Zeitfensters, dass wir auf eine der bisher diskutierten Größen setzen. Ein Beispiel hierfür ist Abb. 16, in der ein Fenster von ca. 5 ms mit etwa fünf Schwingungsperioden der Winkelbeschleunigung wiedergibt. Sind die Kurven unregelmäßig oder gar gezackt, so liegt der typische Fall eine Unterabtastung vor, der eine Verringerung der diskreten Zeitschritte „dt“ für ratsam erscheinen lässt, wodurch sich natürlich bei gleichbleibender Gesamt-Betrachtungsdauer die Zahl der Rechenschritte und damit die CPU-Zeit erhöht. Daß in unserem Beispiel eine zufriedenstellende Abtastung vorliegt, erkennt man in Abb.16.

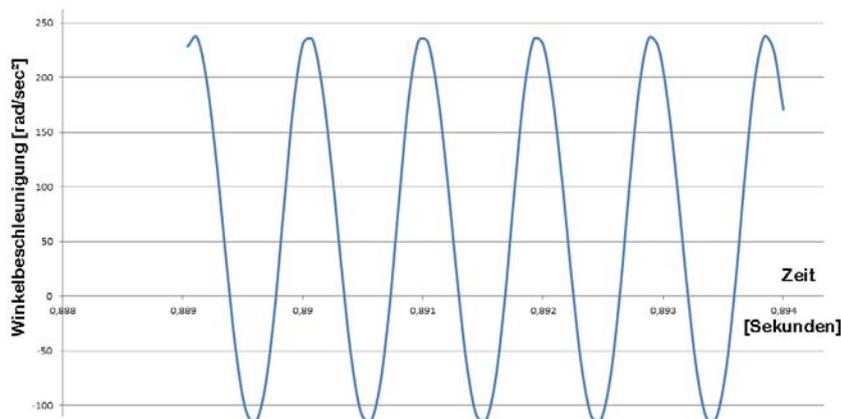


Abb.16:

Betrachtung mit hoher Zeitauflösung zwecks Begutachtung der Qualität der numerischen Iteration.

Obwohl dem System permanent Leistung entnommen wird (siehe Abb. 17), wächst die Gesamtenergie im System während des gesamten Untersuchungszeitraums streng monoton (siehe Abb. 18). Diese Beobachtung lässt sich bis zum Erreichen des stabilen Zustandes Fließgleichgewicht fortsetzen.

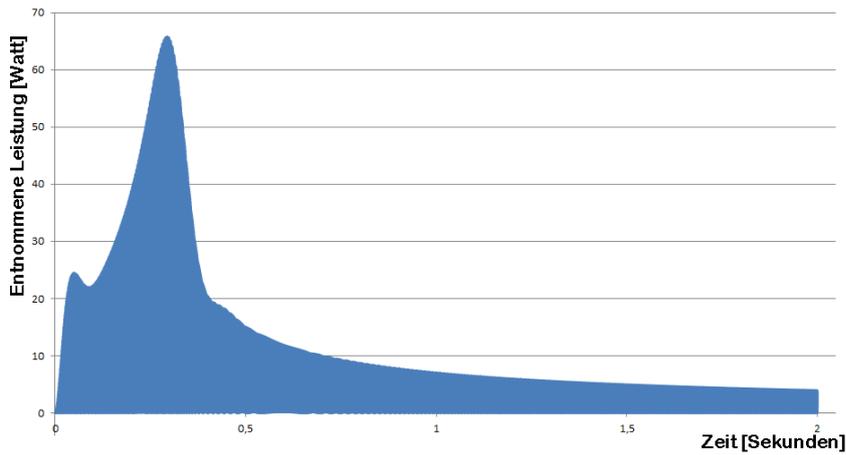


Abb.17:  
Elektrische Leistungs-Entnahme am Lastwiderstand.

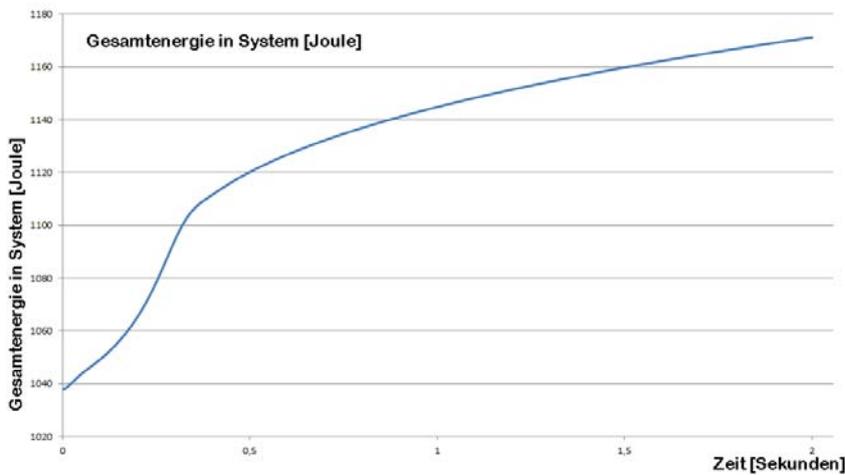


Abb.18:  
Gesamtenergie im System, also in der Spule, im Kondensator und in der Drehbewegung des rotierenden Magneten.

Eine zugeführte Leistung über den Input-Schwingkreis gibt es in unserem Beispiel nicht, da die Maschine als Selbstläufer arbeitet. Aus diesem Grund erübrigt sich eine Darstellung dieser Größe.

Eine Langzeit-Analyse der am Lastwiderstand entnehmbaren elektrischen Leistung findet man in Abb.19. Von den vorangehenden Kurzzeitanalysen unterscheidet sie sich lediglich durch die Ausdehnung des gesamten Beobachtungszeitraumes auf 40 Sekunden. Ganz offensichtlich konvergiert die am Lastwiderstand entnehmbare elektrische Leistung gegen Null, was man auch durch eine weitere Verlängerung des Untersuchungszeitraums deutlich belegen kann.

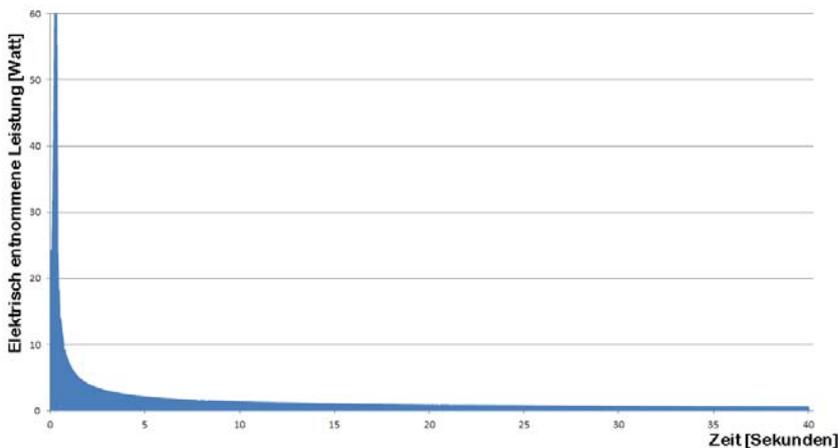


Abb.19:  
Am Lastwiderstand elektrisch entnehmbare Leistung in der Langzeit-Analyse.

Der Grund ist folgender:

Um leistungsstark arbeiten zu können, muss eine endliche (und bleibende) Phasenverschiebung zwischen der elektrischen Schwingung im LC-Schwingkreis und der mechanischen Rotation des Magneten vorhanden sein und aufrecht erhalten werden. Diese Phasenverschiebung wird unbedingt benötigt, damit das von der Spule erzeugte Magnetfeld den Magneten in genau demjenigen Moment beschleunigt, in dem es besonders stark ist, aber hingegen den Magneten genau in demjenigen Moment besonders wenig abbremst, in dem es besonders schwach ist.

Man kann dies durch ein ganz einfaches Modell veranschaulichen. Die Schwingung im LC-Schwingkreis sorgt dafür, dass die in diesem Schwingkreis enthaltene Energie zwischen der Energie des elektrostatischen Feldes im Kondensator und der Energie des magnetischen Feldes in der Spule hin- und her- schwingt. Dies macht sie genau zwei mal in jeder Schwingungsperiode, denn die Kondensatorplatten werden wechselseitig positiv oder negativ aufgeladen, die Spulen werden wechselseitig nach „Nordpol“ und „Südpol“ aufgeladen. In denjenigen Momenten, in denen die Feldenergie gerade in der Spule ist, wird der rotierende Magnet deutlich beschleunigt, wohingegen der rotierende Magnet wenig abgebremst wird, zu denjenigen Zeitpunkten, in denen die Energie gerade im Kondensator ist. Man betrachte hierzu Abb.20, welches den Fall der günstigen Phasenverschiebung für eine starke Rotation beschreibt. Wir sehen dort acht Momentaufnahmen, die in äquidistanten Zeitabständen aufeinander folgen und von 1-8 durchnummeriert sind. Die im violett gezeichneten Kondensator enthaltene Ladung ist mit positivem und negativem Vorzeichen an den Kondensatorplatten gekennzeichnet, und zwar umso mehr, je mehr Ladung die entsprechenden Kondensatorplatten enthalten. Fließt die Ladung durch den Schwingkreis, so gibt ein schwarzer Pfeil die Flußrichtung an, wobei die Ladung natürlich durch die blau gezeichnete Spule fließen muss, und dabei einen Strom in der Spule hervorruft, der das am Spulenrand markierte Magnetfeld verursacht. Auch hier symbolisiert die Anzahl der Symbole („N“ und „S“) die Stärke des Feldes.

Beginnen wir mit unserer Betrachtung bei Bild-Teil Nr.1. Zum Zeitpunkt  $t_1$  ist der Kondensator noch leicht geladen, aber es existiert auch ein mäßiger Stromfluss durch die Spule, der für ein entsprechendes Magnetfeld sorgt. Aufgrund der Polarität der Magnetfelder (von Spule und Dauermagnet relativ zueinander) wird der Dauermagnet in eine Rechts-Drehung versetzt. In Bild-Teil Nr.2 ist der Kondensator vollständig entladen, so dass der Strom durch die Spule maximal ist. Hier liegt die gesamte Feldenergie als Magnetfeld-Energie vor. Dadurch werden die Pole des Dauermagneten besonders stark angezogen, d.h. die Rotation wird besonders deutlich verstärkt. Bei Bild-Teil Nr.3 hat sich das Magnetfeld der Spule inzwischen schon ein Stück weit abgebaut, aber aufgrund der Stellung des Dauermagneten kann das Magnetfeld der Spule den Dauermagneten ohnehin nicht mehr beschleunigen. Würde das Feld der Spule nicht vollständig abgeklungen sein, so würde in Bild-Teil Nr.4 das zuvor vorhandene Magnetfeld die Rotation des Dauermagneten sogar wieder abbremsen. Dass dies nicht passiert, hat seine Ursache darin, dass die gesamte schwingende Energie des LC-Schwingkreises nun in Form eines elektrischen Feldes im Kondensator vorliegt und somit auf den Dauermagneten keine Kräfte ausübt. (Die Spule ist ohne Feld.) Ab nun beginnt aber die Entladung des bei Bild-Teil Nr.4 maximal aufgeladenen Kondensators, die wir in Bild-Teil Nr.5 schon gut beobachten können. Der Entladestrom fließt natürlich in der entgegengesetzten Richtung wie bei Bild-Teil Nr.1, so dass das Magnetfeld der Spule in Bild-Teil Nr.5 die umgekehrte Polarität hat wie in Bild-Teil Nr.1. In Bild-Teil Nr.5 baut sich das Magnetfeld der Spule langsam auf und erreicht in Bild-Teil Nr.6 seine maximale Feldstärke. Entsprechend wird hier der Magnet in Bild-Teil Nr.5 und 6 abermals zur Rechtsdrehung beschleunigt. Bild-Teil Nr.7 entspricht der Umkehrung von Bild-Teil Nr.3. Das noch leicht vorhandene Magnetfeld der Spule gibt kein Drehmoment auf den Dauermagneten aus, weil dieser bereits nach dem Feld ausgerichtet ist. In Analogie zu Bild-Teil Nr.4 sieht man auch in Bild-Teil Nr.8 kein bremsendes Drehmoment auf den Dauermagneten, weil die Ladungen sich gerade im Kondensator befinden und die Spule kein Magnetfeld erzeugt. Abb.20 ist wie eine Animation zu verstehen, die man sich aus acht aufeinanderfolgenden Bildern zusammengesetzt denken sollte, sodass nach Abb.8 der Vorgang wieder zyklisch bei Abb.1 weitergeht.

Die Beschleunigung der Rotation des Magneten wird solange fortgesetzt, wie die Phasendifferenz zwischen der Position des Magneten und den schwingenden Feldern im LC-Schwingkreis aufrecht erhalten bleibt.

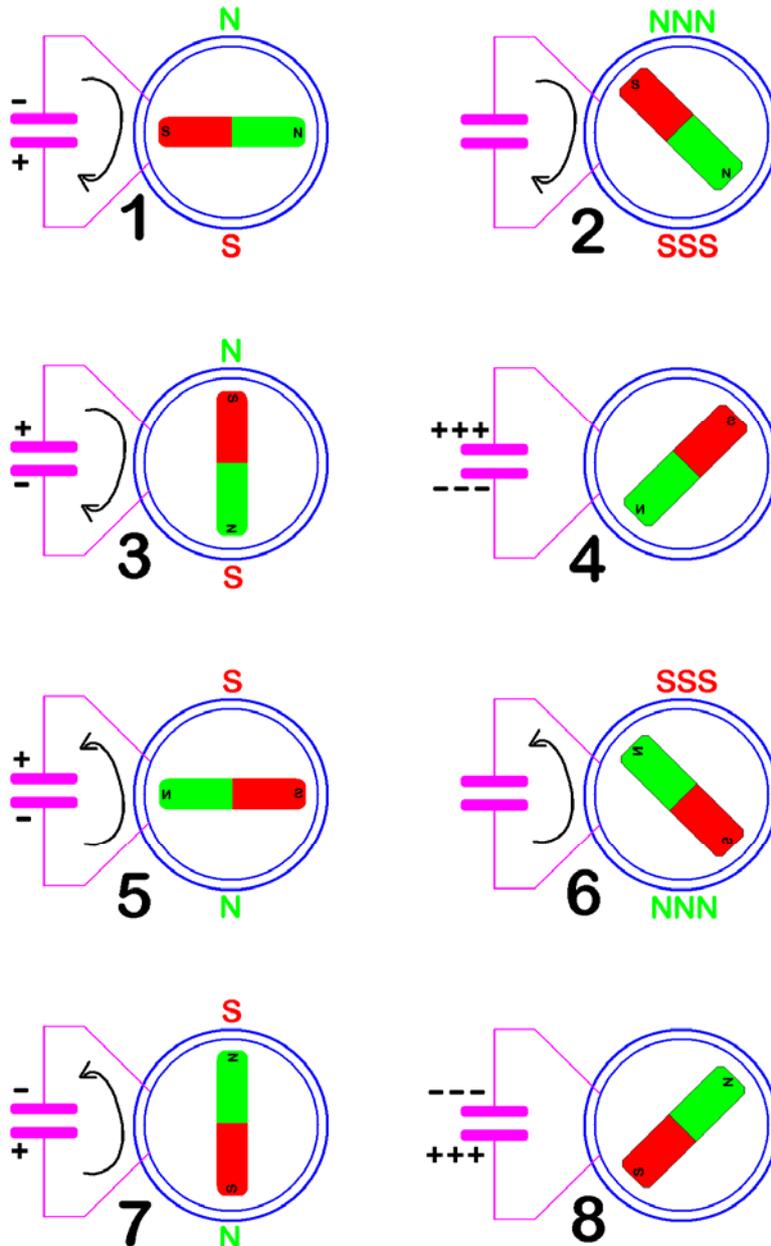


Abb.20:  
Drehung eines Magneten in der Spule eines LC-Schwingkreises mit konstant gehaltener Phasendifferenz zwischen Rotation und elektrischer Schwingung.

Würde der Magnet dauernd beschleunigt werden, aber nicht abgebremst, so würde die Drehzahl der Rotation des Magneten bald die Schwingungsfrequenz des LC-Schwingkreises erreichen. Entsprechend würde die Phasenverschiebung zwischen der elektrischen Schwingung im LC-Schwingkreis und der mechanischen Drehung bald verschwinden, sodass keine Beschleunigung des drehenden Magneten mehr stattfinden kann. Man betrachte hierzu Abb.21.

Wir sehen dort in Bild-Teil Nr.1 eine leichte Beschleunigung des Dauermagneten (durch ein schwaches Magnetfeld in der Spule), die durch das Abbremsen des Dauermagneten in Bild-Teil Nr.3 exakt kompensiert wird. In demjenigen Moment (Bild-Teil Nr.2), in dem das Magnetfeld der Spule seinen Maximalwert erreicht, kann es kein Drehmoment auf den Dauermagneten ausüben, da dieser bereits exakt nach dem Magnetfeld der Spule ausgerichtet ist. Mit anderen Worten heißt das, dass in Summe über die Bild-Teile Nr.1-3 kein Drehmoment auf den Magneten ausgeübt wird. Auch in Bild-Teil Nr.4 beobachten wir kein Drehmoment auf den Dauermagneten, weil hier die gesamte Feldenergie im Kondensator vorliegt und nicht in der Spule. (Hier ist die Spule ohne Magnetfeld.) Die

selbe Argumentation, die wir nun von den Bild-Teilen Nr.1-4 kennen gelernt haben, können wir nochmals bei den Bild-Teilen Nr.5-8 wiederholen. In diesem Betriebszustand des Raumenergie-Konverters hat der rotierende Magnet mit seiner Drehzahl die Schwingungsfrequenz des LC-Schwingkreises eingeholt und kann diese nun nicht überholen, weil kein beschleunigendes Drehmoment mehr vorhanden ist, welches dem Dauermagneten Raumenergie zuführen könnte.

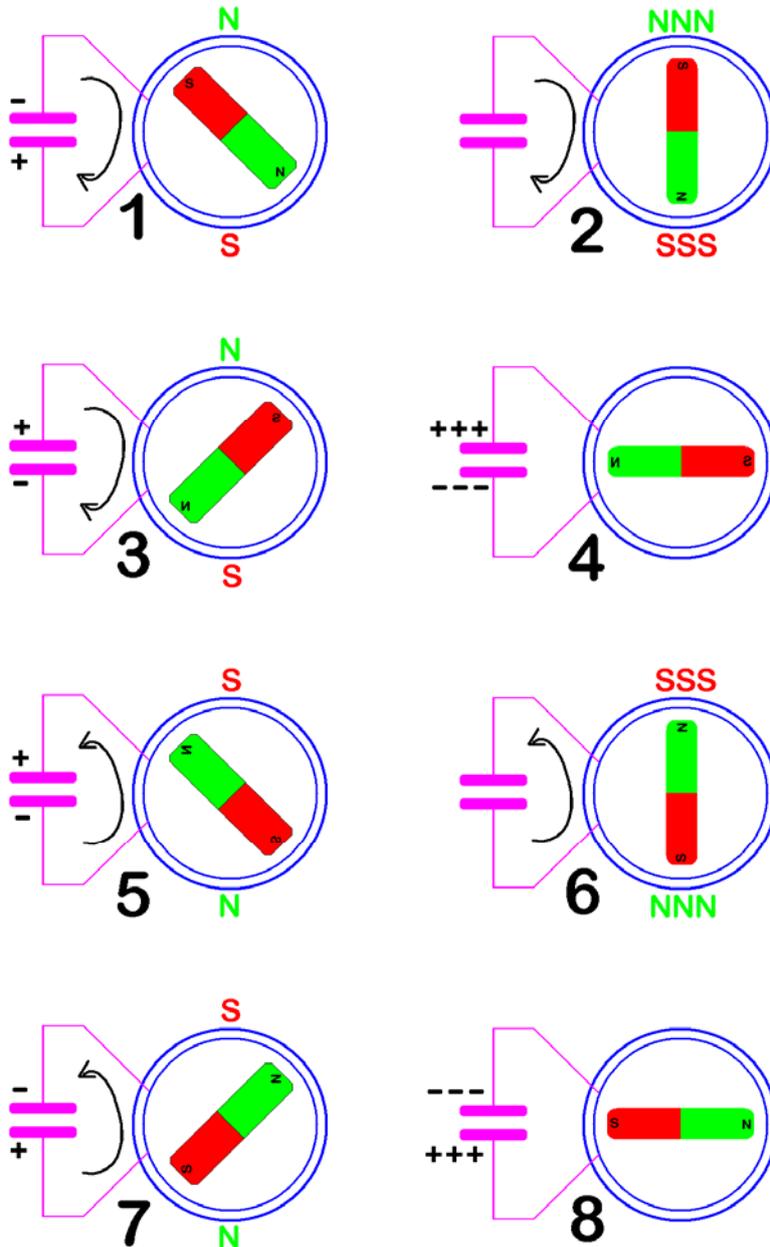


Abb.21:

Drehung eines Magneten in der Spule eines LC-Schwingkreises mit verschwindender Phasendifferenz zwischen Rotation und elektrischer Schwingung.

Aufgrund mangelnder mechanischer Belastung holt die Rotation des Dauermagneten die elektrische Schwingung im LC-Schwingkreis ein, so dass keine Raumenergie mehr gewandelt wird.

Der Zustand führt schließlich zu der Erkenntnis, dass ohne mechanische Leistungsentnahme auch keine elektrische Leistungsentnahme möglich ist.

Die Tatsache dass der drehende Magnet (auch ohne mechanische Belastung) die Schwingung im LC-Schwingkreis niemals überholen kann, verhindert die Unfallgefahr durch überhöhte Drehzahl, weil beim Verschwinden der Phasenverschiebung keine Raumenergie mehr gewandelt werden kann (außer demjenigen Anteil, der benötigt wird, um die Drehzahl gegen die Reibung konstant zu halten). Aber um Raumenergie zu wandeln, muss man eine positive Phasenverschiebung nach Abbild 20 aufrechterhalten. Die optimale Phasenverschiebung entspricht mit 45° genau der Situation in Abb.20. Da dies auf elektrischem Wege (also alleine durch den elektrischen Schwingkreis) nicht erreicht werden kann, ist eine mechanische Belastung der Rotation nötig. Man muss also den rotierenden Dauermagneten mit einem mechanischen Drehmoment belasten, um eine endliche Phasenverschiebung zwischen der Rotation des Magneten und der Schwingung im LC-Schwingkreis aufrecht zu erhalten.

In dieser Hinsicht verhält sich der Raumenergie-Motor grundsätzlich völlig anders als eine klassische elektrische Maschine. Wenn man leicht laufende Lager hat, und die Rotation praktisch ohne mechanische Belastung stattfindet, kann der Motor praktisch keine oder fast keine mechanische Leistung und auch keine oder fast keine elektrische Leistung erbringen. Alleine schwergängige Lager würden schon die Wandlung von Raumenergie begünstigen.

Besser als schwergängige Lager eignet sich natürlich eine mechanische Energieentnahme aus der Magnet-Rotation, die man im Idealfall sogar durch eine Regelung derart beeinflussen kann, dass sich ein langzeitstabiler Betrieb des Raumenergie-Motors erreichen lässt. Die mechanische Leistungsentnahme kann man z.B. an der Drehzahl des Magneten ausrichten, oder an der Phasenverschiebung zwischen der Magnetrotation und der Schwingung im LC-Schwingkreis. Entscheiden wir uns z.B. für die erstgenannte Variante und richten die mechanische Leistungsentnahme aus der drehenden Achse an der Drehzahl aus, so können wir bei Überschreiten eines oberen Schwellenwertes der Drehzahl die Leistungsentnahme erhöhen, um so die Laufgeschwindigkeit des Motors zu dämpfen, und dadurch eine gleichbleibende Phasenverschiebung zwischen der Rotation des Dauermagneten und dem LC-Schwingkreis aufrechterhalten. In ähnlicher Weise definiert man einen unteren Drehzahl-Schwellenwert, bei dessen Unterschreitung die Leistungsentnahme reduziert wird, damit die Rotation des Dauermagneten der Schwingung im LC-Schwingkreis einwandfrei folgen kann. Dies ist die Grundidee, die der mechanischen Leistungsentnahme im nachfolgenden Abschnitt 6 zu Grunde liegt.

Im praktischen Betrieb (den wir im nachfolgenden Abschnitt untersuchen werden), werden wir dann finden, daß immer genau so viel Leistung aus der Raumenergie gewandelt wird, wie der Verbraucher abnimmt – solange wir die Belastung nicht zu groß wählen (denn eine zu große Belastung würde den Motor einfach zum Stillstand bringen, wie man eben jeden Motor durch Überlast abbremst).

## 6. Der EMDR-Konverter mit mechanischer Leistungsentnahme

Wie wir in den Abbildungen 13, 14, 15 und besonders in Abb.19 erkannt haben, läuft die Raumenergie-Wandlung nach Abklingen des Einschwingungsvorgangs bei fehlender Phasenverschiebung zwischen der Dauermagnet-Rotation und dem LC-Schwingkreis asymptotisch gegen die Leistung null. Den Grund haben wir am Ende von Abschnitt 5 erläutert.

Aus diesem Grunde muß für einen nutzbringenden Betrieb des Raumenergie-Konverters das Drehmoment  $M(t)$  in Gleichung (24a) außer dem Beitrag der Magnetkräfte (zwischen Dauermagnet und Spule) auch noch den in Abschnitt 5 begründeten Beitrag der mechanischen Leistungsentnahme enthalten.

Der Letztgenannte wurde in Tab. 1 bereits eingeführt und folgt im Prinzip der einfachen technischen Mechanik, wie in Gleichung (27) wiedergegeben.

$$P_{mech} = M_{mech} \cdot \dot{\varphi} \tag{27}$$

Mechanische Leistungsentnahme, mit dem Drehmoment

$$M_{mech} = c_r \cdot \dot{\varphi} \text{ und } c_r = \text{Reibkoeffizient für}$$

winkelgeschwindigkeitsproportionale Leistungsentnahme.

Damit gelingt es uns nun, den Raumenergie-Motor in einen leistungsstarken Betrieb zu überführen, der langzeitstabil (dauerhaft) aufrechterhalten wird.

Die Dokumentation der Eingabeparameter entnimmt man dem im Anhang befindlichen Quell-Code, der gleichzeitig den letzten Entwicklungsstand des DFEM-Algorithmus zum Zeitpunkt der vorliegenden Publikation wiedergibt.

Die Einführung der geschwindigkeitsproportionalen Leistungsentnahme in die Bewegungs-Differentialgleichungen des Konverter-Systems ist in denjenigen Programmzeilen enthalten, die mit einem vorangestellten Kommentar „{GG}“ beginnen. Die drehzahlabhängige Regelung des für die Leistungsentnahme zuständigen Reibung-Koeffizienten „cr“ ist im Unterprogramm „Reibung\_nachregeln“ zu finden. Dabei wird eine Zielfrequenz „phipZiel“ angesteuert, die aus Gründen der Überzeugungskraft ein wenig höher liegen kann als die Startfrequenz, mit der der Raumenergie-Motor angeworfen werden muss, damit jeder sieht, daß die entnommene Energie nicht aus der Drehzahl des Anlagers stammt. Für den Dauerbetrieb wird dann der Konverter in einem Frequenzband um dieses Ziel Frequenz gehalten, wobei das Frequenzband lediglich den Zweck hat, eine für die Regelung nötige Schalt-Hysterese zu definieren.

Wie man in den Eingabedaten des Quell-Codes entdecken kann, liegt der Lösung der Differentialgleichung eine numerische Iteration über  $10^8$  Schritte der Dauer von je 43nanoSekunden zugrunde. Es wurde untersucht, dass die Zeitauflösung tatsächlich ausreicht, um sicherzustellen, dass der Algorithmus zufriedenstellend auskonvergiert ist.

Die Bildschirmkontrolle der Ausgabedaten direkt im Anschluss an die Berechnungen zeigt folgende Werte:

```
Leistungs-Aenderung im System:      5.04845399 Watt
Am Lastwiderstand entnommene Gesamtenergie = 223.50737922 Joule
entsprechend einer mittleren entnommenen Leistg: 5.19784602848813E+0001 Watt
Ueber Input-Spannung zugefuehrte Gesamt-Energie: 0.00000000000000E+0000 Joule
entsprechend einer mittleren zugefuehrten Leistg: 0.00000000000000E+0000 Watt
Gesamte mechanisch entnommene Energie = 2271.25431928806 Joule
entsprechend einer mittleren Leistung = 528.19867890420 Watt
bei einer Betrachtungs-Dauer von 4.30000000000000E+0000 sec.
```

Wir interpretieren dies wie folgt:

Der Start des Konverters wird mit einer Anfangs-Drehzahl von 30.000 Umdrehungen/min initiiert, während sich im elektrischen LC-Schwingkreis keinerlei Anfangs-Energie befindet. Die Drehung des Magneten induziert eine Spannung in der Spule und bringt somit Energie in den elektrischen LC-Schwingkreis ein. Aus diesem Grunde beginnt der Einschwingzustand mit einer Abnahme der Drehzahl des Dauermagneten. Wir sehen dies in Abb.22. Nach Abklingen des Einschwingzustandes folgt ein minimales Pendeln um die gewünschte Zielfrequenz, wobei die Breite des Pendelns durch die Schalt-Hysterese durch die Regelung der Leistungsentnahme bedingt ist. Wie man sieht, genügt bereits eine sehr schmale Schalt-Hysterese.

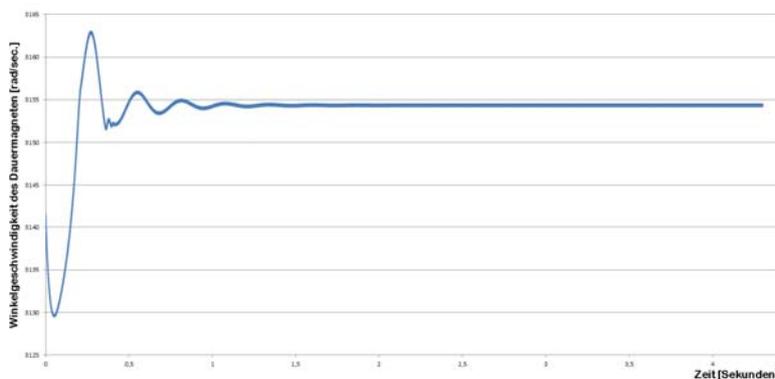


Abb.22:

Winkelgeschwindigkeit des Dauermagneten im leistungsstarken Raumenergie-Konverter.

Die Darstellung ist eine Wiedergabe von spaltet „I“ der Excel-Datenausgabe.

Die Regelung der Leistungsentnahme wird über einen zeitlich variablen Reibungskoeffizienten vorgenommen, dessen dynamische Entwicklung Abb.23 wiedergibt. Offensichtlich ist während des Einschwingvorgangs eine gewisse Regelung nötig, im eingeschwungenen Zustand läuft die Leistungsentnahme jedoch gegen einen sehr stabilen Zustand.

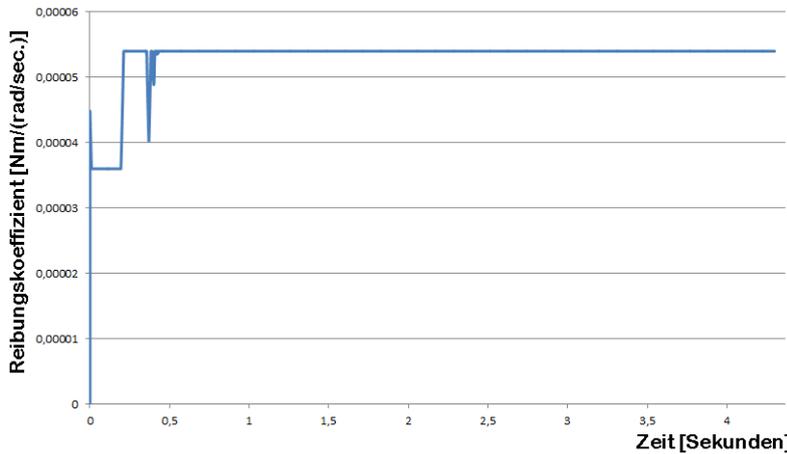


Abb.23:

Dynamische Entwicklung des geregelten Reibungs-Koeffizienten zur Steuerung einer optimalen mechanischen Leistungs-Entnahme im leistungsstarken EMDR-Raumenergie-Konverter.

Die Darstellung ist eine Wiedergabe von spaltet „X“ der Excel-Datenausgabe.

Die tatsächliche entnommene mechanische Leistung sehen wir in Abb.24. In Anbetracht eines Dauermagneten von nur 10 cm Länge ist die freiwerdende Leistung von gut 530 W durchaus erfreulich, zumals noch eine elektrische Leistungsentnahme hinzukommt, und überdies die Auslegung der Maschine noch nicht vollständig leistungsoptimiert wurde.

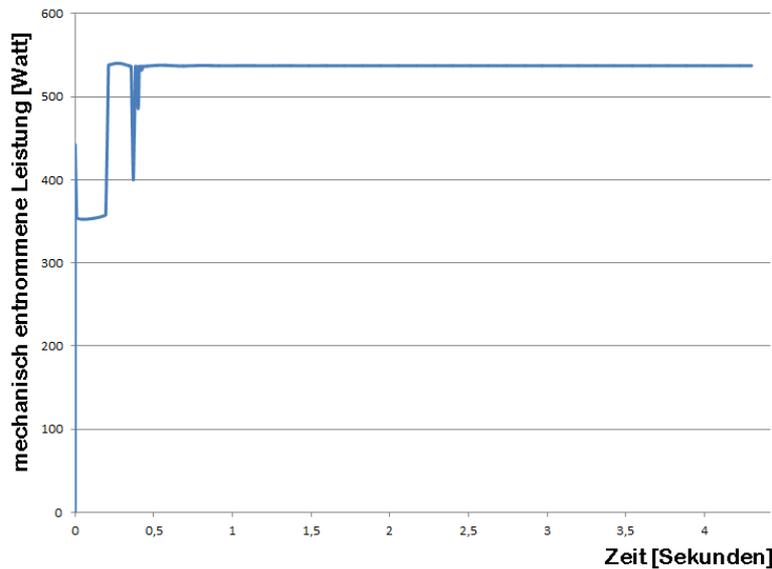


Abb.24:

Mechanische Leistungsentnahme aus dem EMDR-Raumenergie-Konverter.

Die Darstellung ist eine Wiedergabe von spaltet „Y“ der Excel-Datenausgabe.

Um ein Gefühl für die Anforderungen an die bauliche Dimensionierung der Bauteile eines Prototypen zu bekommen, betrachten wir noch die Ströme und Spannungen im LC-Schwingkreis:

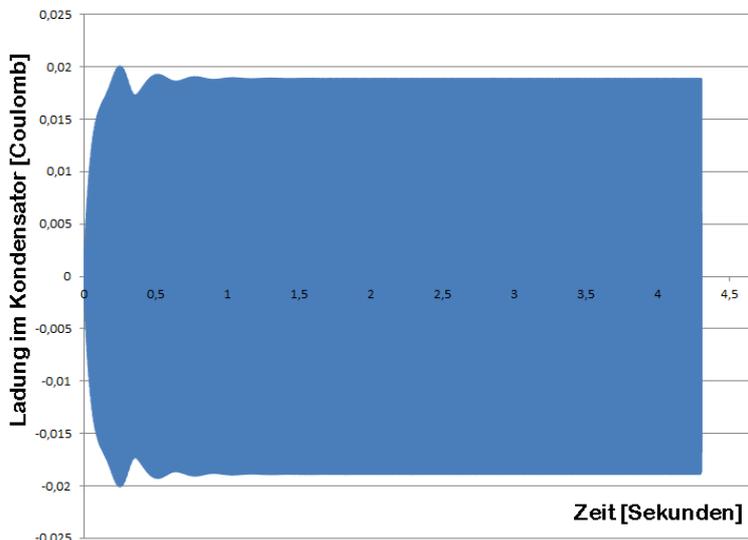


Abb.25:

Kondensator-Ladung  $Q$  im EMDR-Raumenergie-Konverter.

Die Darstellung ist eine Wiedergabe von spaltet „A“ der Excel-Datenausgabe.

Auf die Kondensatorspannung hat der Einschwingvorgang (im Vergleich zum Dauerbetrieb) nur einen eher moderaten Einfluss, da der Kondensator im Anfangszustand ungeladen ist. Abb.25 zeigt die Einhüllende der schwingenden Ladung im Kondensator. Die Kondensatorspannung ergibt sich nach Gleichung (28) mit knapp 200 Volt. Der Wert sollte in der Praxis handhabbar sein. Allerdings muss man bedenken, dass der Kondensator relativ fein schaltbar sein muss, da er der Feinabstimmung des EMDR-Raumenergie-Konverters dient. Eine Abstimmung auf 3-4 signifikante Stellen ist durchaus erforderlich, so dass eine Kondensatorbank nach Abb. 26 sinnvoll erscheint. Eine Hintereinanderschaltung mehrerer Kapazitäten ist ebenso unsinnig, wie die parallel geschaltete Verwendung verschiedener Kondensatoren mit all zu unterschiedlichen Innenwiderständen, da es absolut erforderlich ist, dass alle Kondensatoren innerhalb der Kondensatorbank einigermaßen gleich schnell auf- und entladen werden.

Schließlich muß auch noch betont werden, daß der Kondensator dasjenige Mittel ist, mit dem Unsicherheiten und Näherungen der theoretischen Berechnungen ausgeglichen werden müssen !

$$U_C = \frac{Q}{C} = \frac{0.02C}{101.7 \mu F} = 197V \tag{28}$$

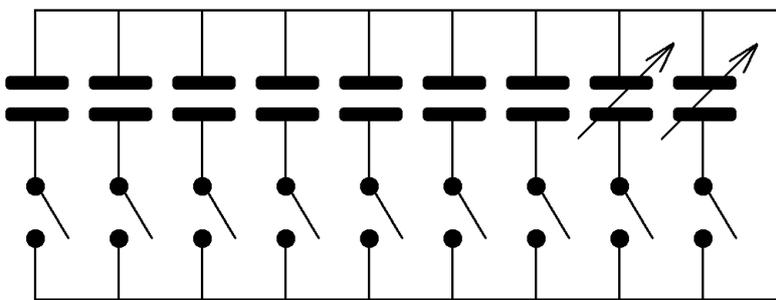


Abb.26: Kondensatorbank zur Feinabstimmung.

Die einzelnen Kondensatoren dürfen durchaus unterschiedliche Kapazitäten haben, so dass ein relativ großer Kapazitätsbereich fein eingestellt werden kann.

Ein Spulenstrom von 60 A (siehe Abb. 27) bereitet uns keinerlei Kopfzerbrechen, denn wir haben in Voraussicht der zu erwarteten Werte zum Wickeln der Spule bereits einen Leiter mit einer Querschnittsfläche von 78,5 mm<sup>2</sup> (Drahtdurchmesser 10 mm) vorgesehen.

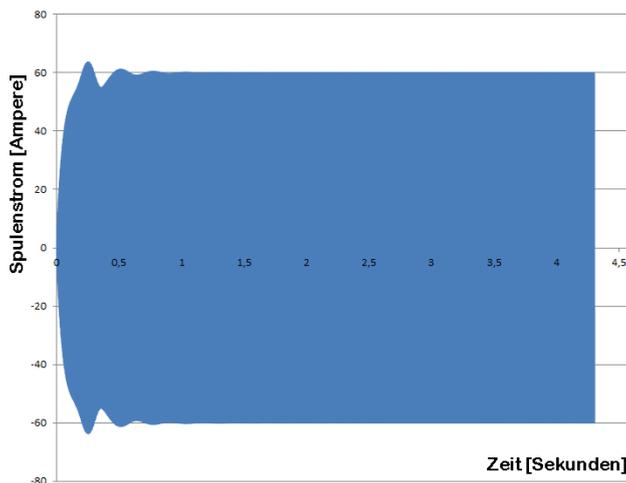


Abb.27: Spulenstrom  $\dot{Q}$  im EMDR-Raumenergie-Konverter.

Die Darstellung ist eine Wiedergabe von spaltet „B“ der Excel-Datenausgabe.

Die Spannung über der Spule ermitteln wir nach Gleichung (29) aus dem Induktionsgesetz. Ihre Amplitude liegt dem Betrage nach bei knapp 200 V (sie paßt gut zur Kondensatorspannung, was für die sinnvolle Abstimmung der Auslegung spricht), was ebenfalls keine Probleme in der Handhabung (speziell im Hinblick auf die nötige Isolation) mit sich bringen dürfte.

$$U_L = -L \cdot \dot{I} = 9.936 \cdot 10^{-4} \text{ Henry} \cdot 200000 \frac{\text{A}}{\text{sec}} = -199 \text{ Volt} \tag{29}$$

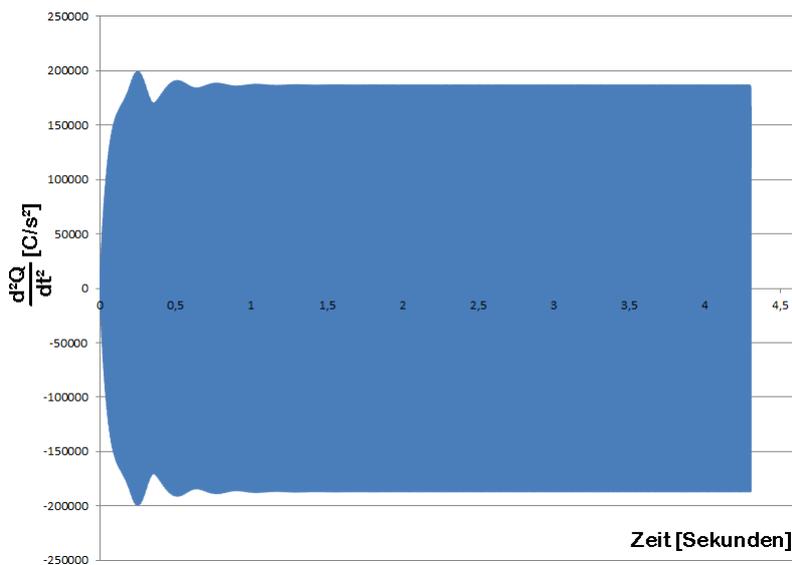


Abb.28:

Ableitung des Spulenstroms nach der Zeit  $\ddot{Q}$  im EMDR-Raumenergie-Konverter.

Die Darstellung ist eine Wiedergabe von spaltet „C“ der Excel-Datenausgabe.

Die Winkelbeschleunigung (Spalte „J“ im Excel-Datenexport) mit einer Amplitude von knapp 1000  $\text{rad/s}^2$  offenbart in Anbetracht der Masse des Magneten (von 245 Gramm) und seines Trägheitsmoments der Rotation (von  $2.1 \cdot 10^{-4} \text{ kg} \cdot \text{m}^2$ ) den Hinweis, dass der Raumenergie Motor montageseitig eine ordentliche Stabilität aufweisen muss, um einem zentrifugalen Wegfliegen der Teile im Betrieb vorzubeugen.

Die Gesamtenergie im System ist nach dem Einschwingvorgang in engen Grenzen (aufgrund der Schalt-Hysterese der mechanischen Leistungsentnahme) gleichbleibend, wie wir in Abb. 29 erkennen. Entsprechendes Verhalten lässt sich in allen Kanälen der Energie-Analyse aller Komponenten des EMDR-Systems erkennen, da im eingeschwungenen Dauerbetrieb lediglich Energie zwischen den unterschiedlichen Komponenten des Systems hin und her schwingt, und die aus der Raumenergie gewonnene Leistung sofort vom Lastwiderstand (elektrisch) bzw. von der mechanischen Entnahme abgezogen wird.

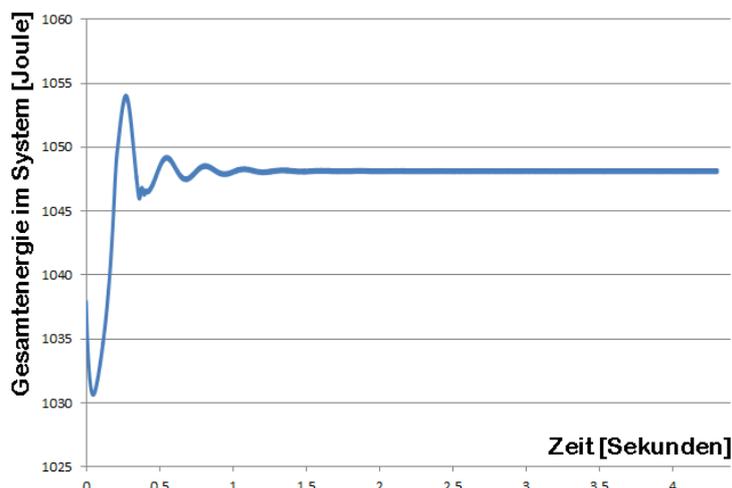


Abb.29:

Gesamtenergie im System.

Die Darstellung ist eine Wiedergabe von spaltet „T“ der Excel-Datenausgabe.

#### Zusatzfrage:

Darf die mechanische Belastung auch ohne Regelung betrieben werden ?  
Dies würde die Herstellung eines Prototypen wesentlich erleichtern.

#### Antwort:

Ja, auch bei konstantem Belastungs-Koeffizienten (const: „cr = crAnfang“) stellt sich ein dauerhaft stabiler eingeschwungener Zustand ein, sofern der Belastungs-Koeffizient innerhalb gewisser nicht all

zu enger Grenzen vorgegeben wird. Betreibt man zum Beispiel den Algorithmus mit den im Anhang gezeigten Parametern und legt dabei ohne Regelung des Belastungs-Koeffizienten „crAnfang:=37E-6“ fest, so bildet sich auch rasch ein ein geschwungener Zustand aus, wie wir ihn in Abb.30 sehen. Auch die entnommene Leistung stellt sich dabei rasch auf einen konstanten Wert ein, siehe Abb.31.

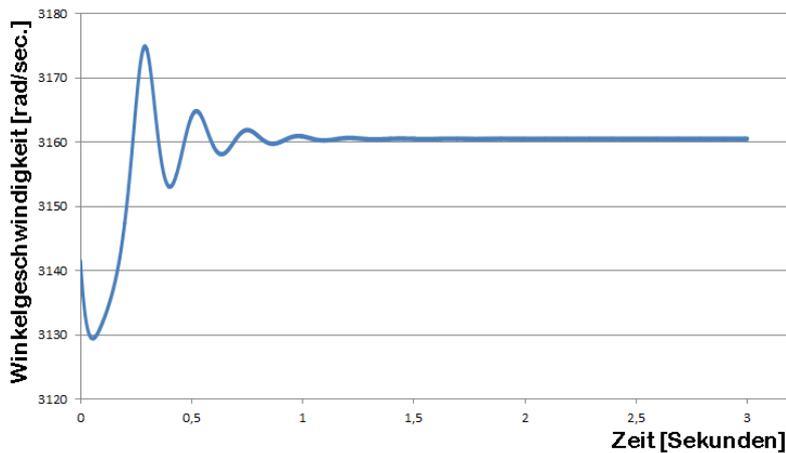


Abb.30:  
Winkelgeschwindigkeit des rotierenden Dauermagneten.  
Die Darstellung ist eine Wiedergabe von spaltet „I“ der Excel-Datenausgabe.

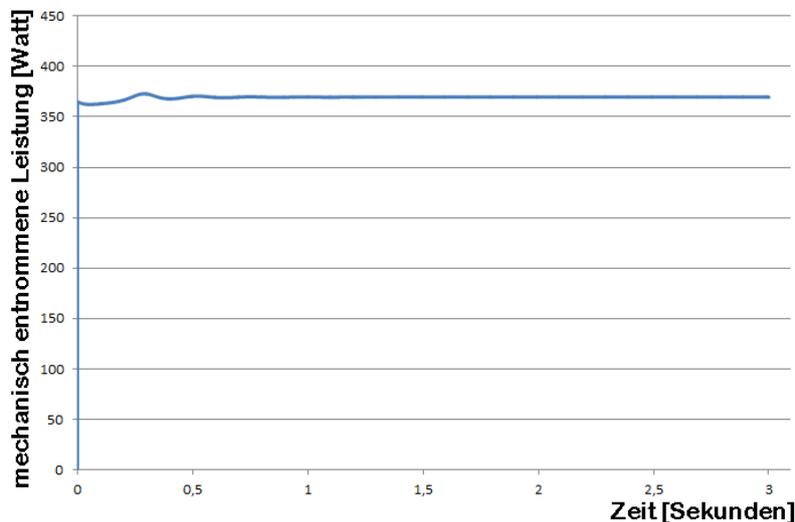


Abb.31:  
Mechanisch an der Drehachse des rotierenden Dauermagneten entnommene Leistung bei geschwindigkeitsproportionaler Belastung.  
Die Darstellung ist eine Wiedergabe von spaltet „Y“ der Excel-Datenausgabe.

Man muss dabei nur darauf achten, dass der Belastungs-Koeffizient (Reibung-Koeffizient  $c_r$ ) nicht zu groß eingestellt wird, weil eine zu hohe mechanische Belastung in der Einschwingphase dem System zu viel Energie entzieht, so dass es nicht einen günstigen stabilen Zustand erreichen kann, wie er für den Dauerlauf geeignet ist. Würde man das System zum Beispiel mit dem Mittelwert des Belastungs-Koeffizienten nach Abb.23 starten ( $c_r:=c_{rAnfang}:=54E-6$ ), so würde der Konverter bereits in der Anlaufphase so stark abgebremst werden, dass er niemals den Betriebszustand zu Abb.23 erreichen könnte. Beispiele für mögliche Variationen des Belastungs-Koeffizienten zeigt die nachfolgende Aufstellung:

- $c_r:=c_{rAnfang}:=2.0 \cdot 10^{-5} \Rightarrow P_{mech} = 201 \text{ Watt}$  (mündet in einen stabilen Betriebszustand)
- $c_r:=c_{rAnfang}:=2.5 \cdot 10^{-5} \Rightarrow P_{mech} = 251 \text{ Watt}$  (mündet in einen stabilen Betriebszustand)
- $c_r:=c_{rAnfang}:=3.0 \cdot 10^{-5} \Rightarrow P_{mech} = 300 \text{ Watt}$  (mündet in einen stabilen Betriebszustand)
- $c_r:=c_{rAnfang}:=3.5 \cdot 10^{-5} \Rightarrow P_{mech} = 349 \text{ Watt}$  (mündet in einen stabilen Betriebszustand)
- $c_r:=c_{rAnfang}:=3.7 \cdot 10^{-5} \Rightarrow P_{mech} = 369 \text{ Watt}$  (mündet in einen stabilen Betriebszustand)
- $c_r:=c_{rAnfang}:=4.0 \cdot 10^{-5} \Rightarrow P_{mech} = 247 \text{ Watt}$  (Drehzahl sinkt kontinuierlich, kein stabiler Betrieb)
- $c_r:=c_{rAnfang}:=5.3 \cdot 10^{-5} \Rightarrow P_{mech} = 117 \text{ Watt}$  (Mittelwert von Abb.23)

Offensichtlich steigt die mechanisch entnehmbare Leistung praktisch linear mit wachsender Belastung an, solange die Belastung nicht zu groß wird, um das Einschwingen in einen stabilen Betriebs-

zustand bereits in der Anlaufphase zu unterdrücken (stabil langzeit-tauglicher Bereich:  $cr:=2.0 \cdot 10^{-5}$  ...  $3.7 \cdot 10^{-5}$ ). Dieser stabile Betrieb gelingt bereits bei  $cr:=4.0 \cdot 10^{-5}$  nicht mehr, erst recht nicht, wenn man den unter der Regelung zu Abb.23 eingestellten Wert von  $cr:=5.3 \cdot 10^{-5}$  ungeregelt als konstante Belastung vorgibt.

Letztlich lässt sich also feststellen, dass die Regelung des Belastungs-Koeffizienten (also der Belastung) für den Betrieb des Konverters im Prinzip erlässlich ist, und somit für einen Prototypen nicht unbedingt benötigt wird, dass diese Regelung aber andererseits den Vorteil hat, die entnehmbare Leistung zu maximieren. Mit Regelung wurde nämlich erreicht:

$cr_{\text{Anfang}}:=4.5 \cdot 10^{-5}$ , nach Regelung  $cr:=5.4 \cdot 10^{-5}$ , =>  $P_{\text{mech}} = 537$  Watt (Dauerbetrieb lt. Abb.23)

Der Grund liegt darin, dass die Belastung in denjenigen Momenten, in denen das Drehmoment während der Anlaufphase schwach ist, heruntergeregelt wird, aber in denjenigen Momenten, in denen die Leistung zur Verfügung steht, nach oben geregelt wird.

## 7. Praktische Hinweise, Anleitung zum Aufbau eines EMDR-Konverters

Mit Abschluss von Abschnitt 6 ist die Theorie des EMDR-Systems besprochen. Wir wollen aber jetzt noch einige praktische Hinweise betrachten, die für den Aufbau eines Prototyps interessant sein könnten. [PC 11]

Herzstück des EMDR-Konverters ist der Magnet. Seine Beschaffung kann einen gewissen Aufwand bedeuten. In [Tur 11] wurde ein Scheibenmagnet mit einer „in-plane“ Magnetisierung vorgeschlagen, und die Berechnung des Konverters war auf ein homogenes Magnetfeld beschränkt. Diese Berechnung ist eine relativ grobe Näherung. Die in der jetzt vorgelegten neuen Publikation dargestellte Rechenmethode ist in der Lage, beliebige Magnetformen einer Berechnung zuzuführen. Aus Gründen der einfachen Beschaffbarkeit der Magneten, wurde für die neue genaue Rechnung ein simpler zylindrischer Stabmagnet vorausgesetzt, was zwar einen erhöhten Rechenaufwand bedeutet, aber die Beschaffung erleichtert.

Bei Verwendung eines Stabmagneten muss man aber an den Luftwiderstand denken, der aufgrund der hohen Drehzahl als ernst zu nehmende mechanische Belastung angesehen werden kann. Wünscht man diese Belastung, auch im Hinblick auf die Aufrechterhaltung einer Phasendifferenz nach Abb. 20, so ist nichts dagegen einzuwenden. Möchte man jedoch die Leistung nicht verlieren, die ein sich wie ein Propeller drehender Stabmagnet aufgrund des Luftwiderstands verzehrt, so könnte man den Magneten in eine runde Scheibe einbetten, die man zum Beispiel aus Kunststoff oder einem anderen nicht ferromagnetischen und nicht leitenden Material fertigen kann (siehe Abb.32).

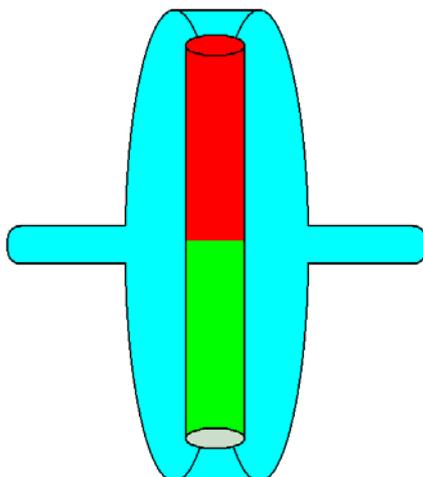


Abb.32:

Mögliche Kapselung (Einbettung) eines rotierenden Stabmagneten zur Minimierung des Luftwiderstandes bei der Rotation.

Dass eine ferromagnetische Kapselung auf jeden Fall vermieden werden muss, ist sofort klar, weil ein Ferromagnetikum die Feldlinien des Stabmagneten führen (umlenken) würde, so dass diese die Spule nicht mehr erreichen könnten.

Vermutlich könnte es sinnvoll sein, eine nichtleitende Kapsel zu verwenden, um Wirbelstromverluste zu vermeiden. Ob Wirbelströme auftreten, und inwieweit diese die Funktionsweise des Raumentriebs-Konverters beeinträchtigen würden, wurde bisher weder theoretisch noch experimentell untersucht. Störende Einflüsse können nicht ausgeschlossen werden.

Gelegentlich wurden auch nicht ferromagnetische Kapseln diskutiert, weil die mechanische Stabilität eines Neodym-Magneten wesentlich geringer ist, als die mechanische Stabilität einer Eisen-Kobalt-Nickel-Legierung. In diesem Zusammenhang dient die Kapsel um dazu, ein Zerschlagen des Magneten aufgrund der hohen Zentrifugalkräfte zu verhindern. Wer das Risiko von Wirbelstromverlusten als gering einschätzt, mag vielleicht einen nicht ferromagnetischen Edelstahl oder vielleicht auch Kupfer einsetzen. Wer das Risiko von Wirbelstromverlusten als hoch einschätzt, ist vermutlich besser beraten, wenn er eine Eisen-Kobalt-Nickellegierung als Dauermagneten (und dann eben ohne Kapselung) verwendet, zumal es derartige Legierungen gibt, die bei einer Aufmagnetisierungs-Feldstärke von knapp zwei Tesla bereits eines remanentes Feld von knapp über einem Tesla am Stabende des Dauermagneten erbringen können.

Weniger große Sorgen bereiten Wirbelstromverluste in der Drehachse, die die Rotation des Magneten in den Lagern fixiert. Wer sehr große Bedenken in Bezug auf Wirbelstromverluste hat, mag an dieser Stelle mit einem Glasfaser-verstärkten Kunststoff arbeiten und die beiden Lager (links und rechts des Magneten) sehr dicht neben den Magneten setzen. Allerdings ergeben sich daraus ernsthafte technische Durchführungsschwierigkeiten im Bezug auf die Zufuhr der Anfangsdrehzahl zur Initialisierung der Bewegung und ebenso im Bezug auf die Abfuhr der erzeugten mechanischen Leistung. Wer solchermaßen sicher sein will, dass Wirbelströme garantiert vermieden werden, ist auf die Verwendung von Keramik-Lagern angewiesen, da im Inneren der Spule kein leitfähiges Material verwendet werden darf. Leider sind Keramiklager für sehr hohe Drehzahlen ungünstiger geeignet als Lager aus Stahl.

Vermutlich sollte es daher an dieser Stelle effizienter sein, eine etwas längere metallische Achse zu verwenden, die man links und rechts durch die Spule hindurch führen kann, und diese Achse dann mit Stahl-Kugellagern zu halten, die hinreichend weit von der Spule entfernt sind. Dies erleichtert auch wesentlich das Einleiten der Initialisierungsdrehzahl und das Ausleiten der mechanisch erzeugten Leistung.

Eine zweite Spule, die in dieser Publikation als Input-Spule bezeichnet wurde, kann auch als Output-Spule benutzt werden, um dem System auf elektrischem Wege erhöhte Leistung zu entziehen. Über die Windungszahl dieser Output-Spule ließe sich dann eine Anpassung der Strom-Spannung-Charakteristik an die Bedürfnisse eines Verbrauchers erwirken, oder ebenso auch an die Möglichkeiten des Raumentriebs-Konverters.

Ein Vorschlag zur Verringerung der Drehzahl der Magnetrotation ist die Verwendung eines Multipol-Magneten, wie er zum Beispiel in Abb. 33 angedeutet ist. Verteilt man zum Beispiel 16 Stabmagneten (also Dipole) auf dem Umfang eines Rades, so ist die Zahl der Polwechsel pro Umdrehung 16 anstatt von 2 wie beim einfachen Dipol-Stabmagneten. Eine Erhöhung der Zahl der Polwechsel pro Umdrehung um einen Faktor 8 bedeutet eine Absenkung der Drehzahl um einen Faktor 8. Auf diese Weise könnte man zum Beispiel den elektrischen Schwingkreis mit einer Frequenz von 32.000 Schwingungen/min schwingen lassen und dabei die mechanische Rotation eben nicht mit 32.000 Umdrehungen/min, sondern nur mit 4000 Umdrehungen/min laufen lassen. Beliebige andere Zahlenkombinationen sind problemlos nach Bedarf einstellbar.

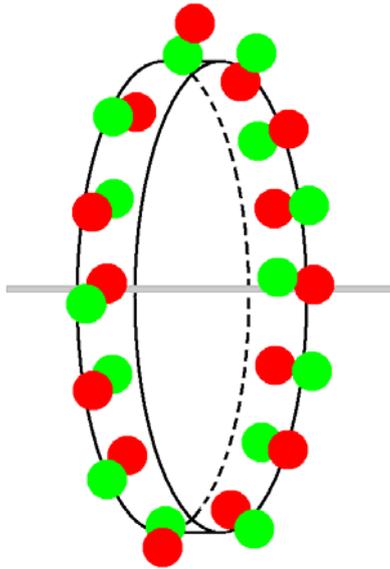


Abb.33:

Multipol-Magnet, der dadurch hergestellt werden kann, dass man eine Reihe von Stabmagneten auf dem Umfang eines Rades verteilt.

Auf die Notwendigkeit einer stabilen Lagerung wird ausdrücklich hingewiesen, da der umlaufende Magnet während jeder Umdrehung ständig gebremst und beschleunigt wird. Die Situation ist bei einem Multipol-Magneten höherer Ordnung wesentlich weniger kritisch als bei einem Dipol, wie z.B. bei einem einfachen Stabmagneten.

## 8. Resumée

Als Ergebnis dieser Arbeit wird ein realitätsnahes Berechnungsverfahren für beliebige Raumenergie-Motoren mit elektrischem und/oder magnetischem Funktionsprinzip vorgestellt, mit dem jeder versierte Fachmann seinen eigenen Raumenergie-Konverter auslegen kann, aber auch auf vorhandene in der Literatur dargestellte Konverter nachrechnen kann.

Wichtiger Bestandteil der vorgestellten Arbeit ist aber auch die Auslegung eines konkreten Beispiels für einen EMDR-Raumenergie-Konverter, den jeder geschickte Experimentator nachbauen kann. Die Beschreibungen sind praxisnah genug, daß man sie auch ohne weitere Nachrechnungen in ein Experiment umsetzen können sollte. Bei dem hier vorgestellten Beispiel wurden praktische Belange wie die Verfügbarkeit von Bauteilen und deren Dimensionierung ebenso berücksichtigt, wie technische Aspekte, die man bei der Inbetriebnahme eines Prototypen berücksichtigen sollte.

Trotzdem darf man nicht vergessen, dass die vorliegende Publikation auf einer rein theoretischen Ausarbeitung basiert, dass sich also beim experimentellen Aufbau eines Prototypen mit Sicherheit noch zahlreiche bisher nicht bekannte offene Fragen ergeben, die gelöst werden müssen, bevor der Prototyp in der Praxis tatsächlich funktioniert. Aber dies sollte keine existentiellen Probleme aufwerfen.

Der Autor des vorliegenden Artikels würde ausgesprochen gerne selbst einen Prototypen aufbauen (insbesondere auf dem Hintergrund der Tatsache, daß er selbst als Experimentalphysiker ausgebildet ist), aber leider hat er keinen Zugang zu einem Labor und auch nicht zu einer Werkstatt.

## 9. Literatur-Hinweise

- [Ans 08] Finite Elemente Programm ANSYS, John Swanson (1970-2008)  
ANSYS, Inc. Software Products, <http://www.ansys.com>
- [Bea 02] Motionless Electromagnetic Generator, Tom Bearden et. al.  
US Patent, 6,362,718 vom 26. März 2002  
Erfinder: Patrick L. Stephen, Thomas E. Bearden, James C. Hayes, Kenneth D. Moore,  
James L. Kenny  
Auch nachzulesen unter <http://www.cheniere.org>
- [Bor 99] Borland Pascal (Delphi 5 aus 1999 oder neuere Version)
- [Bro 08] Taschenbuch der Mathematik  
Ilja N Bronstein, Konstantin A Semendjajew, Gerhard Musiol, Heiner Mühlig,  
Verlag Harri Deutsch, 7.Auflage, 2008, ISBN 978-3-8171-2017-8
- [Dub 90] Dubbel - Taschenbuch für den Maschinenbau , 17.Auflage  
W. Beitz, K.-H. Küttner et. al., Springer-Verlag. 1990. ISBN 3-540-52381-2
- [Ger 95] Gerthsen Physik, H. Vogel  
Springer Verlag. 1995. ISBN 3-540-59278-4
- [Hoh 11] <http://www.magnetmotor.at/> von Dietmar Hohl  
Im Internet finden sich zahlreiche selbstlaufende Magnet-Konverter. An dieser Stelle sei ein  
Beispiel exemplarisch erwähnt.
- [Hur 40] The Invention of Hans Coler, Relating To An Alleged New Source Of Power.  
R. Hurst, B.I.O.S. Final Report No. 1043, B.I.O.S.Trip No. 2394  
B.I.O.S. Target Number: C31/4799, British Intelligence Objectives Sub-Committee
- [Jac 81] Klassische Elektrodynamik, John David Jackson  
Walter de Gruyter Verlag. 1981. ISBN 3-11-007415-X
- [Jeb 06] Die Urkraft aus dem Universum, Klaus Jebens  
Jupiter-Verlag. 2006. ISBN 3-906571-23-8
- [Kep 10] Keppe Motor Manual 1.0, 2008, Norberto Keppe and Cláudia B. S. Pacheco.  
<http://www.keppemotor.com/> and <http://www.keppemotor.com/manual1.php>
- [Koh 96] Praktische Physik, 3 Bde., Friedrich Kohlrausch, et. al.  
Verlag B.G.Teubner Stuttgart, 1996, ISBN-13: 978-3519230014
- [Mar 88-98]
- Der Kugellager-Motor und der Huber-Effekt,  
Stefan Marinov, raum&zeit 32 (1988) 81-84
  - Repetition of Silvertooth's experiment for measuring the aether drift",  
Stefan Marinov, Speculations in Science and Technology 12 No.3 (1989) 187-179
  - Propulsive and Rotating Ampère Bridges and the Principle of Relativity  
Stefan Marinov, Physics Essays 4 No.1 (1991) 30-36
  - The Missing Magnetic Force Law, Galilean Electrodynamics 9 No.2  
Stefan Marinov, (March/April 1998) 35-37

- [Mie 84] Kompendium Hypertechnik. Tachyonenenergie, Hyperenergie, Antigravitation.  
Sven Mielordt, Berlin, 1984  
Nachdruck der 4. Auflage vom raum&zeit Verlag, ISBN 3-89005-005-0
- [Nie 83] Konversion von Schwerkraft-Feld-Energie. Revolution in Technik, Medizin, Gesellschaft.  
Von Hans A. Nieper  
MIT-Verlag, Oldenburg, 1983, 4. erw. Auflage, ISBN 3-925188-00-2
- [PC 11] private communication, Gespräche mit zahlreichen Kollegen, die aufgrund ihrer Vielzahl an dieser Stelle nicht alle namentlich genannt werden können.
- [Stö 07] Taschenbuch der Physik, Horst Stöcker  
Verlag Harri Deutsch. 2007. ISBN-13 987-3-8171-1720-8
- [Tur 10a] Example of a simple Algorithm for the Construction of Zero-point-energy Converters  
Claus W. Turtur, PHILICA.COM, ISSN 1751-3030, Article number 207, (9. Okt. 2010)
- [Tur 10b] DFEM-Computation of a Zero-point-energy Converter with realistic Parameters for a practical setup  
Claus W. Turtur, PHILICA.COM, ISSN 1751-3030, Article number 213, (7. Dez. 2010)
- [Tur 11] DFEM-Simulation of a Zero-point-energy Converter with realisable Dimensions and a Power-output in the Kilowatt-range.  
Claus W. Turtur, PHILICA.COM, ISSN 1751-3030, Article number 219, (7. Feb. 2011)

## 10. Anhang: Quell-Code des DFEM-Algorithmus (EMDR\_009i.dpr)

[Bor 99]

Hinweis: Pascal-Programmierzeilen, deren Länge eine Text-Zeile in der Publikation übersteigt, wurden rechtsbündig fortgesetzt. Diese Ausrichtung muß wieder entfernt werden, wenn man das Programm im Pascal-Compiler laufen lassen will. Bequemer ist es, den Original-Quellcode des Programm von meiner Internet-Seite herunter zu laden, selbstverständlich kostenlos (zusammen mit dem File „schonda“, das man zwecks Lauffähigkeit des Programms benötigt).

```

Program KM_009i;
{$APPTYPE CONSOLE}
uses
  Windows,
  Messages,
  SysUtils,
  Classes,
  Graphics,
  Controls,
  Forms,
  Dialogs;

Const Bn=7;           {Schritt-Anzahl der Magnetfeld-Speicherung nach *2 von S.2}
Const SpNmax=200;    {Maximal mögliche Anzahl der Stützpunkte der Spulen (Inupt und Turbo)}
Const FlNmax=2000;   {Maximal mögliche Anzahl der Flächenelemente der Spulen (Inupt und Turbo)}
Const MESEanz=200;   {Tatsächliche Anzahl der Magnetfeld-Emulations-Spulen-Elemente, gerade Anzahl wählen}
Const AnzPmax=35000; {Dimensionierung der Arrays für den Plot (für den Datenexport nach Excel)}

Var epo,muo : Double; {Naturkonstanten}
    Bsw      : Double; {Schritt-Weite der Magnetfeld-Speicherung nach *1 von S.2}
    Spsw     : Double; {Schritt-Weite der Spulen-Aufgliederung nach *2 von S.1}
    SpN      : Integer; {Anzahl der Stützpunkte der Spulen}
    FlN      : Integer; {Anzahl der Flächenelemente der Spulen}
    LiGe     : Double; {Lichtgeschwindigkeit}
    xo,yo,zo : Integer; {Geometrieparameter nach Zeichnung *2 von S.1}
    Ninput   : Integer; {Zahl der Wicklungen der Input-Spule}
    Nturbo   : Integer; {Zahl der Wicklungen der Turbo-Spule}
    PsiSFE   : Double; {magnetischer Fluß durch ein Spulen-Flächenelement}
    PsiGES   : Double; {magnetischer Fluß durch die gesamte Spule}
    B1,B2,B3,B4,B5 : Double; {Fourier-Koeffizienten, allgemein}
    B1T,B2T,B3T,B4T,B5T : Double; {Fourier-Koeffizienten, Turbo-Spule}
    B1I,B2I,B3I,B4I,B5I : Double; {Fourier-Koeffizienten, Input-Spule}
    Bldreh,phase : Double; {Koeffizienten zur Drehmoments-Schnellberechnung}
    MEyo, MEro, MEI : Double; {Abmessungen und Strom des Magnetfeld-Emulationsspulenpaares nach *1 von S.5}
    Bx,By,Bz : Array [-Bn..Bn,-Bn..Bn,-Bn..Bn] of Double; {Kartes. Komp. der magn. Induktion, Dauermagnet}
    MESEx,MESEy,MESEz : Array [1..MESEanz] of Double; {Orte der Magnetfeld-Emulations-Spulen-Elemente}
    MESEdx,MESEdy,MESEdz : Array [1..MESEanz] of Double; {Laufrichtungen der Magnetfeld-Emulations-Spulen-Elemente}
    OrtBx,OrtBy,OrtBz : Array [-Bn..Bn,-Bn..Bn,-Bn..Bn] of Double; {Kartes. Komp. der Orte, an denen das Feld Bx, By, Bz ist.}
    SpIx,SpIy,SpIz : Array [1..SpNmax] of Double; {Stützpunkte der Polygonzüge der Input-Spule, kartesische Koordinaten}
    SpTx,SpTy,SpTz : Array [1..SpNmax] of Double; {Stützpunkte der Polygonzüge der Turbo-Spule, kartesische Koordinaten}
    SIx,SIy,SIz : Array [1..SpNmax] of Double; {Ort als Mittelpunkt der Leiterschleifen-Elemente}
    STx,STy,STz : Array [1..SpNmax] of Double; {Ort als Mittelpunkt der Leiterschleifen-Elemente}
    dSIx,dSIy,dSIz : Array [1..SpNmax] of Double; {Richtungsvektoren der Leiterschleifen-Elemente der Input-Spule}
    dSTx,dSTy,dSTz : Array [1..SpNmax] of Double; {Richtungsvektoren der Leiterschleifen-Elemente der Turbo-Spule}
    FlIx,FlIy,FlIz : Array [1..FlNmax] of Double; {Flächenelemente der Input-Spule, kartesische Koordinaten}
    FlTx,FlTy,FlTz : Array [1..FlNmax] of Double; {Flächenelemente der Turbo-Spule, kartesische Koordinaten}
    BxDR,ByDR,BzDR : Array [-Bn..Bn,-Bn..Bn,-Bn..Bn] of Double; {gedrehtes Magnetfeld}
    OrtBxDR,OrtByDR,OrtBzDR : Array [-Bn..Bn,-Bn..Bn,-Bn..Bn] of Double; {gedrehte Ortsvektoren}

    {Zum Lösen der Bewegungs-Differentialgleichung:}
    phi,phip,phipp : Array[0..AnzPmax] of Double; {Drehwinkel und dessen Ableitungen}
    Q,Qp,Qpp : Array[0..AnzPmax] of Double; {Ladung und deren Ableitungen in der Turbo-Spule}
    QI,QpI,QppI : Array[0..AnzPmax] of Double; {Ladung und deren Ableitungen in der Input-Spule}
    phio,phipo,phippo,phim,phipm,phippm : Double; {Winkel und dessen Ableitungen Index "io" und "io-1"}
    qoT,qpoT,qppoT,qmT,qpmT,qppmT : Double; {Ladung und deren Ableitungen Index "io" und "io-1" in Turbo-Spule}
    qoI,qpoI,qppoI,qmI,qpmI,qppmI : Double;

```



```

Writeln(fout,'Es folgt die Eingabe der beiden Spulen, vgl. Zeichnung *2 von S.1 :');
Writeln(fout,'Die Spulen werden nach Vorgabe der Geometrieparameter automatisch vernetzt. ');
Writeln(fout,'Spsw = ',Spsw:12:6,' Angabe in Metern: Die Spulen-Aufgliederung ist in Spsw-Schritten');
Writeln(fout,'xo = ',xo,', Angaben in Vielfachen von Spsw, Geometrieparameter nach *2 von S.1');
Writeln(fout,'yo = ',yo,', Angaben in Vielfachen von Spsw, Geometrieparameter nach *2 von S.1');
Writeln(fout,'zo = ',zo,', Angaben in Vielfachen von Spsw, Geometrieparameter nach *2 von S.1');
Writeln(fout,'Ninput = ',Ninput:9,' Zahl der Wicklungen der Input-Spule');
Writeln(fout,'Nturbo = ',Nturbo:9,' Zahl der Wicklungen der Turbo-Spule');
Writeln(fout,'nebeninput = ',Round(nebeninput):9,' Windungen nebeneinander in der Input-Spule');
Writeln(fout,'ueberinput = ',Round(ueberinput):9,' Windungen uebereinander in der Input-Spule');
Writeln(fout,'nebenturbo = ',Round(nebenturbo):9,' Windungen nebeneinander in der Turbo-Spule');
Writeln(fout,'ueberturbo = ',Round(ueberturbo):9,' Windungen uebereinander in der Turbo-Spule');
Writeln(fout,' ');
Writeln(fout,'Bsw = ',Bsw:9,' Magnetfeld-Speicherung nach *1 von S.2 in Zentimeter-Schritten');
Writeln(fout,'Ich emuliere hier das Magnetfeld eines 1T-Magneten durch ein Spulenpaar *1 von S.5:');
Writeln(fout,'MEyo = ',MEyo:14,' y-Koordinaten der Magnetfeld-Emulationsspulen nach *1 von S.5');
Writeln(fout,'MEro = ',MEro:14,' Radius der Magnetfeld-Emulationsspulen nach *1 von S.5');
Writeln(fout,'MEI = ',MEI:14,' Strom des Magnetfeld-Emulationsspulenpaares nach *1 von S.5, Ampere');
Writeln(fout,' ');
Writeln(fout,'Allgemeine technische Größen:');
Writeln(fout,'DD = ',DD:12:7,' {Meter} {Durchmesser des Spulendrahtes zur Angabe der Drahtstärke}');
Writeln(fout,'rho = ',rho,', {Ohm*m} {Spez. elektr. Widerstand von Kupfer, je nach Temperatur,
Kohlrausch,T193}');
Writeln(fout,'rhoMag = ',rhoMag,', {kg/m^3} {Dichte des Magnet-Materials, Eisen, Kohlrausch Bd.3}');
Writeln(fout,'CT = ',CT:14,' {Farad} {Kapazität des Kondensators, der mit in der Turbo-Spule (!) in
Reihe geschaltet}');
Writeln(fout,'CI = ',CI:14,' {Farad} {Kapazität des Kondensators, der mit in der Input-Spule (!) in
Reihe geschaltet}');
Writeln(fout,' ');
Writeln(fout,'Sonstige (zur Eingabe):');
Writeln(fout,'Rlast = ',Rlast:15,' {Ohm} Ohm'scher Lastwiderstand im LC-Turbo-Schwingkreis');
Writeln(fout,'UmAn = ',UmAn:10:2,' {U/min} Anfangsbedingung mechanisch - Rotierender Magnet:
Startdrehzahl');
Writeln(fout,'Uc = ',Uc:10:2,' {Volt} Anfangsbedingung elektrisch - Kondensatorspannung am TURBO-
Kondensator');
Writeln(fout,'Il = ',Il:10:2,' {Ampere} Anfangsbedingung elektrisch - Spulenstrom im TURBO-
Schwingkreis');
Writeln(fout,' ');
Writeln(fout,'Mechanische Leistungs-Entnahme, proportional zur Geschwindigkeit, aber mit Nachregelung zur
konst. Drehzahl:');
Writeln(fout,'Koeffizient einer geschw-prop. mechan. Leistungs-Entnahme: ',crAnfang:17:12,'
Nm/(rad/s)');
Writeln(fout,'Ziel-Drehzahl, für mechanische Reibungs-Nachregelung: ',phipZiel:17:12,' U/min');
Writeln(fout,' ');
Writeln(fout,'Abgeleitete Parameter. Die Größen werden aus den obigen Parametern berechnet, es ist keine
Eingabe möglich:');
Writeln(fout,'DLI:=4*(yo+zo)*Spsw*Ninput = ',DLI:10:5,' {Meter} Länge des Spulendrahtes, Input-Spule');
Writeln(fout,'DLT:=4*(yo+zo)*Spsw*Nturbo = ',DLT:10:5,' {Meter} Länge des Spulendrahtes, Turbo-Spule');
Writeln(fout,'RI:=rho*(DLI)/(pi/4*DD*DD) = ',RI:10:5,' {Ohm} Ohm'scher Widerstand des Spulendrahtes,
Input-Spule');
Writeln(fout,'RT:=rho*(DLT)/(pi/4*DD*DD) = ',RT:10:5,' {Ohm} Ohm'scher Widerstand des Spulendrahtes,
Turbo-Spule');
Writeln(fout,'BreiteI:=nebeninput*DD = ',BreiteI:10:5,' Breite und Höhe des Input-Spulenlörpers');
Writeln(fout,'HoeheI:=ueberinput*DD = ',HoeheI:10:5,' Breite und Höhe des Input-Spulenlörpers');
Writeln(fout,'BreiteT:=nebenturbo*DD = ',BreiteT:10:5,' Breite und Höhe des Turbo-Spulenlörpers');
Writeln(fout,'HoeheT:=ueberturbo*DD = ',HoeheT:10:5,' Breite und Höhe des Turbo-Spulenlörpers');
Writeln(fout,'fkI:=Sqrt(HoeheI*HoeheI+4/pi*2*yo*2*zo)/HoeheI = ',fkI:10:5,' Korrekturfaktor zur
Induktivität der kurzen Input-Spule');
Writeln(fout,'fkT:=Sqrt(HoeheT*HoeheT+4/pi*2*yo*2*zo)/HoeheT = ',fkT:10:5,' Korrekturfaktor zur
Induktivität der kurzen Turbo-Spule');
Writeln(fout,'LI:=muo*(2*yo+BreiteI)*(2*zo+BreiteI)*Ninput*Ninput/(HoeheI*fkI) = ',LI,' Induktivität
Input-Spule');
Writeln(fout,'LT:=muo*(2*yo+BreiteT)*(2*zo+BreiteT)*Nturbo*Nturbo/(HoeheT*fkT) = ',LT,' Induktivität
Turbo-Spule');
Writeln(fout,'omT:=1/Sqrt(LT*CT) = ',omT,' Kreis-Eigenfrequenz des Turbo-Spulen-Schwingkreises aus LT &
CT');
Writeln(fout,'TT:=2*pi/omT = ',TT,' Schwingungsdauer des Turbo-Spulen-Schwingkreises aus LT & CT. ');
Writeln(fout,'Mmag:=rhoMag*(pi*MEro*MEro)*(2*MEyo) = ',Mmag:8:3,' kg Masse des Magneten {Rotation um
Querachse !!}');
Writeln(fout,'J:=Mmag/4*(MEro*MEro+4*MEyo*MEyo/3) = ',J,' Trägheitsmoment des Magneten bei Rotation um
Querachse');
Writeln(fout,' ');
Writeln(fout,'Anzeige einiger auszurechnender Parameter:');
Writeln(fout,'Magnet: Start-Winkelgeschw.: omAn = ',omAn:15:6,' rad/sec');
Writeln(fout,'Magnet: Startdrehzahl, Umdr./sec.: UmSec = ',UmSec:15:10,' Hz');
Writeln(fout,'Masse des Magnet = ',Mmag:10:6,' kg');
Writeln(fout,'Traegheitsmoment Magnet bei QUER-Rotation',J,' kg*m^2');
Writeln(fout,'Gesamtdauer der Betrachtung: ',AnzP*dt,' sec. ');

```

```

Writeln(fout,'Excel-Export: ',PlotAnfang*dt:14,'...',PlotEnde*dt:14,' sec., Step ',PlotStep*dt:14,'
sec. ');
Writeln(fout,'Das sind ',(PlotEnde-PlotAnfang)/PlotStep:8:0,' Datenpunkte (also Zeilen).');
Writeln(fout,' ');
Writeln(fout,'*****');
Writeln(fout,' ');
Writeln(fout,'Einige Ergebnisse der Berechnung:');
Writeln(fout,'Anfangs-Energie im System:      ',AnfEnergie:14:8,' Joule');
Writeln(fout,'End-Energie im System:                ',EndEnergie:14:8,' Joule');
Writeln(fout,'Leistungs-Aenderung im System:',(EndEnergie-AnfEnergie)/(AnzP*dt):14:8,' Watt');
Writeln(fout,'Am Lastwiderstand entnommene Gesamtenergie = ',Wentnommen:14:8,' Joule');
Writeln(fout,'entsprechend einer mittleren entnommenen Leistg:',Wentnommen/(AnzP*dt),' Watt');
Writeln(fout,'Ueber Input-Spannung zugefuehrte Gesamt-Energie: ',Ezuf,' Joule');
Writeln(fout,'entsprechend einer mittleren zugefuehrten Leistg:',Ezuf/(AnzP*dt),' Watt');
Writeln(fout,'Gesamte mechanisch entnommene Energie = ',Ereib:18:11,' Joule');
Writeln(fout,'entsprechend einer mittleren Leistung = ',Ereib/(AnzP*dt):18:11,' Watt');
Writeln(fout,'bei einer Betrachtungs-Dauer von',(AnzP*dt),' sec. ');
Close(fout);
end;

Procedure Wait;
Var Ki : Char;
begin
  Write('<W>'); Read(Ki); Write(Ki);
  If Ki='e' then Halt;
  If Ki='E' then Halt;
  If Ki='d' then Dokumentation_des_Ergebnisses;
  If Ki='D' then Dokumentation_des_Ergebnisses;
end;

Procedure ExcelAusgabe(Name:String;Spalten:Integer);
Var fout : Text;      {Bis zu 14 Spalten zum Aufschreiben der Ergebnisse}
    lv,j,k : Integer; {Laufvariablen}
    Zahl : String;    {Die ins Excel zu druckenden Zahlen}
begin
  Assign(fout,Name); Rewrite(fout); {File öffnen}
  For lv:=0 to AnzP do {von "plotanf" bis "plotend"}
  begin
    If (lv mod Abstd)=0 then
    begin
      For j:=1 to Spalten do
      begin {Kolumnen drucken, zuerst 3*Ladung, dann 3*Winkel, dann 8 freie Felder}
        If j=1 then Str(Q[lv]:19:14,Zahl);
        If j=2 then Str(Qp[lv]:19:14,Zahl);
        If j=3 then Str(Qpp[lv]:19:14,Zahl);
        If j=4 then Str(phi[lv]:19:14,Zahl);
        If j=5 then Str(phip[lv]:19:14,Zahl);
        If j=6 then Str(phipp[lv]:19:14,Zahl);
        If j=7 then Str(KG[lv]:19:14,Zahl);
        If j=8 then Str(KH[lv]:19:14,Zahl);
        If j=9 then Str(KI[lv]:19:14,Zahl);
        If j=10 then Str(KJ[lv]:19:14,Zahl);
        If j=11 then Str(KK[lv]:19:14,Zahl);
        If j=12 then Str(KL[lv]:19:14,Zahl);
        If j=13 then Str(KM[lv]:19:14,Zahl);
        If j=14 then Str(KN[lv]:19:14,Zahl);
        For k:=1 to Length(Zahl) do
        begin {Keine Dezimalpunkte verwenden, sondern Kommata}
          If Zahl[k]<>'.' then write(fout,Zahl[k]);
          If Zahl[k]='.' then write(fout,',');
        end;
        Write(fout,chr(9)); {Daten-Trennung, Tabulator}
      end;
      Writeln(fout,''); {Zeilen-Trennung}
    end;
  end;
  Close(fout);
end;

Procedure ExcelLangAusgabe(Name:String;Spalten:Integer);
Var fout : Text;      {Zeit-Skala und bis zu 25 Daten-Spalten zum Aufschreiben der Ergebnisse}
    lv,j,k : Integer; {Laufvariablen}
    Zahl : String;    {Die ins Excel zu druckenden Zahlen}
begin
  If (Spalten>25) then
  begin
    Writeln('FEHLER: Zu viele Spalten. Soviele Daten-Arrays sind nicht vorhanden. ');
    Writeln(' => PROGRAMM WURDE ANGEHALTEN : STOP !');
  end;
end;

```

```

Wait; Wait; Halt;
end;
Assign(fout,Name); Rewrite(fout); {File Öffnen}
For lv:=0 to LPP do {von "plotanf" bis "plotend"}
begin
  If (lv mod Abstd)=0 then
  begin
    For j:=0 to Spalten do
    begin {Kolumnen drucken, zuerst 3*Ladung, dann 3*Winkel, dann 8 freie Felder}
      If j=0 then Str(Zeit[lv]:19:14,Zahl); {Markieren der Zeit-Skala}
      If j=1 then Str(Q[lv]:19:14,Zahl); {Turbo-Spule}
      If j=2 then Str(Qp[lv]:19:14,Zahl); {Turbo-Spule}
      If j=3 then Str(Qpp[lv]:19:14,Zahl); {Turbo-Spule}
      If j=4 then Str(QI[lv]:19:14,Zahl); {Input-Spule}
      If j=5 then Str(QpI[lv]:19:14,Zahl); {Input-Spule}
      If j=6 then Str(QppI[lv]:19:14,Zahl); {Input-Spule}
      If j=7 then Str(phi[lv]:19:14,Zahl); {Magnet}
      If j=8 then Str(phip[lv]:19:14,Zahl); {Magnet}
      If j=9 then Str(phipp[lv]:19:14,Zahl); {Magnet}
      If j=10 then Str(KK[lv]:19:14,Zahl); {Auxiliary}
      If j=11 then Str(KL[lv]:19:14,Zahl); {Auxiliary}
      If j=12 then Str(KM[lv]:19:14,Zahl); {Auxiliary}
      If j=13 then Str(KN[lv]:19:14,Zahl); {Auxiliary}
      If j=14 then Str(KO[lv]:19:14,Zahl); {Auxiliary}
      If j=15 then Str(KP[lv]:19:14,Zahl); {Auxiliary}
      If j=16 then Str(KQ[lv]:19:14,Zahl); {Auxiliary}
      If j=17 then Str(KR[lv]:19:14,Zahl); {Auxiliary}
      If j=18 then Str(KS[lv]:19:14,Zahl); {Auxiliary}
      If j=19 then Str(KT[lv]:19:14,Zahl); {Auxiliary}
      If j=20 then Str(KU[lv]:19:14,Zahl); {Auxiliary}
      If j=21 then Str(KV[lv]:19:14,Zahl); {Auxiliary}
      If j=22 then Str(KW[lv]:19:14,Zahl); {Auxiliary}
      If j=23 then Str(KX[lv]:19:14,Zahl); {Auxiliary}
      If j=24 then Str(KY[lv]:19:14,Zahl); {Auxiliary}
      If j=25 then Str(KZ[lv]:19:14,Zahl); {Auxiliary}
      For k:=1 to Length(Zahl) do
      begin {Keine Dezimalpunkte verwenden, sondern Kommata}
        If Zahl[k]<>'.' then write(fout,Zahl[k]);
        If Zahl[k]='.' then write(fout,',');
      end;
      Write(fout,chr(9)); {Daten-Trennung, Tabulator}
    end;
    Writeln(fout,''); {Zeilen-Trennung}
  end;
end;
Close(fout);
end;

Function Sgn(Zahl:Integer):Double;
Var merk : Double;
begin
  merk:=0;
  If Zahl>0 then merk:=+1;
  If Zahl<0 then merk:=-1;
  Sgn:=merk;
end;

Procedure Magnetfeld_zuweisen_01; {homogenes Magnetfeld}
Var i,j,k : Integer;
begin
  For i:=-Bn to Bn do {in x-Richtung}
  begin
    For j:=-Bn to Bn do {in y-Richtung}
    begin
      For k:=-Bn to Bn do {in z-Richtung}
      begin
        Bx[i,j,k]:=0.0; {Telsa}
        By[i,j,k]:=1.0; {Telsa}
        Bz[i,j,k]:=0.0; {Telsa}
        OrtBx[i,j,k]:=i*Bsw;
        OrtBy[i,j,k]:=j*Bsw;
        OrtBz[i,j,k]:=k*Bsw;
      end;
    end;
  end;
end;

Procedure Magnetfeld_zuweisen_02; {willkürlicher Versuch eines inhomogenen Magnetfeldes}

```

```

Var i,j,k : Integer;
begin
  For i:=-Bn to Bn do {in x-Richtung}
  begin
    For j:=-Bn to Bn do {in y-Richtung}
    begin
      For k:=-Bn to Bn do {in z-Richtung}
      begin
        Bx[i,j,k]:=-Sgn(i)/(i*i+j*j+k*k+1); If i=0 then Bx[i,j,k]:=0; {Telsa}
        By[i,j,k]:= 10/(i*i+j*j+k*k+1); {Telsa}
        Bz[i,j,k]:=-Sgn(k)/(i*i+j*j+k*k+1); If k=0 then Bz[i,j,k]:=0; {Telsa}
        OrtBx[i,j,k]:=i*Bsw;
        OrtBy[i,j,k]:=j*Bsw;
        OrtBz[i,j,k]:=k*Bsw;
      { Writeln('Ort:',OrtBx[i,j,k]:12:8,', ',OrtBy[i,j,k]:12:8,', ',OrtBz[i,j,k]:12:8); Wait; }
      end;
    end;
  end;
end;

Procedure Magnetfeld_zuweisen_03;
Var KRPx,KRPy,KRPz : Double; {kartesische Komponenten des Kreuzprodukts im Zähler}
  lmsbetrag : Double; {Betrag im Nenner}
  lmsbetraghoch3 : Double; {Hilfsvariable}
  qwill : Double; {Ladung willkürlich nach S.7}
  om : Double; {Frequenz zum Anpassen von qwill an I}
  t : Double; {Zeit als Laufvariable von 0 ... 2*pi/om}
  sx,sy,sz : Double; {Aufpunkt, an dem das Feld bestimmt werden soll}
  dHx,dHy,dHz : Double; {Infinitesimales Feldelement für Biot-Savert}
  Hx,Hy,Hz : Double; {Gesamtfeld am Ort des Aufpunkts}
  dphi : Double; {Aufteilung der Spulenrings}
  Hxkl,Hykl,Hzkl : Double; {klassische Berechnung im Vgl.}
  Nenner : Double; {Hilfsgröße für klassische Feldberechnung}
  i2,j2,k2 : Integer;{Laufvariable zum Durchgehen des felderfüllten Raumes}
  Bxmax,Bymax,Bzmax : Double; {maximaler Feldwert auf der y-Achse}
Procedure Berechne_dH;
begin
  KRPx:=-om*MEro*cos(om*t)*(MEyo-sy);
  KRPy:+=om*MEro*cos(om*t)*(MEro*cos(om*t)-sx)+om*MEro*sin(om*t)*(MEro*sin(om*t)-sz);
  KRPz:=-om*MEro*sin(om*t)*(MEyo-sy);
  lmsbetrag:=Sqr(MEro*cos(om*t)-sx)+Sqr(MEyo-sy)+Sqr(MEro*sin(om*t)-sz);
  lmsbetrag:=Sqrt(lmsbetrag);
  lmsbetraghoch3:=lmsbetrag*lmsbetrag*lmsbetrag;
  If lmsbetraghoch3<=1E-50 then begin dHx:=0; dHy:=0; dHz:=0; end;
  If lmsbetraghoch3>=1E-50 then
  begin
    dHx:=qwill*KRPx/4/pi/lmsbetraghoch3*dphi/2/pi;
    dHy:=qwill*KRPy/4/pi/lmsbetraghoch3*dphi/2/pi;
    dHz:=qwill*KRPz/4/pi/lmsbetraghoch3*dphi/2/pi;
  end;
  { Writeln('Infinitesimales Feldelement: ',dHx:12:7,', ',dHy:12:7,', ',dHz:12:7,' A/m'); }
end;
Procedure Berechne_Hges;
Var ilok : Integer;{Laufvariable für Zählschleife zur Zerteilung der Spule}
begin
  Hx:=0; Hy:=0; Hz:=0; {Initialisierung des Gesamtfelds für die Addition der Feldelemente}
  qwill:=1; om:=2*pi*MEI/qwill; {Ladung und der Kreisfrequenz in der Magnetfeld-Emulationsspule}
  dphi:=2*pi/1000; {Radianten bei der Aufteilung der Spule in 1000 Abschnitte}
  For ilok:=0 to 999 do {1000 Zählschritte}
  begin
    t:=ilok*dphi/om; {Laufvariable (Zeit), zur Umrundung der Spule}
  { Writeln('ilok = ',ilok:4,' => ',om*t:12:6); Wait; }
    Berechne_dH; {Infinitesimales Feldelement nach Biot-Savart berechnen}
    Hx:=Hx+dHx;
    Hy:=Hy+dHy;
    Hz:=Hz+dHz;
  end;
  { Writeln('Gesamtes Feld am Aufpunkt. : ',Hx:12:7,', ',Hy:12:7,', ',Hz:12:7,' A/m'); }
  Hxkl:=0; Hzkl:=0; {klassische Berechnung im Vgl.}
  Nenner:=Sqrt(MEro*MEro+(MEyo-sy)*(MEyo-sy)); Nenner:=2*Nenner*Nenner*Nenner;
  Hykl:=MEI*MEro*MEro/Nenner; {Der klassische Vergleich geht nur entlang der y-Achse.}
  { Writeln('Vgl. klass. entlang y-Achse: ',Hxkl:12:7,', ',Hykl:12:7,', ',Hzkl:12:7,' A/m'); }
end;
begin
  Writeln; Writeln('Magnetfeld Emulations-Spulenpaar nach *1 von S.5');
  Writeln('y-Koordinaten der Magnetfeld-Emulationsspulen nach *1 von S.5: ',MEyo:8:5,' m');
  Writeln('Radius der Magnetfeld-Emulationsspulen nach *1 von S.5: ',MEro:8:5,' m');
  Writeln('Strom der Magnetfeld-Emulationsspulen nach *1 von S.5: ',MEI:8:5,' Ampere');

```

```

Writeln('Anzahl der Schritte: ',Bn,' hoch 3 => ', 2*Bn+1,' Bildschirm-Aktionspunkte je Spule.');
```

```
{ Zuerst die obere Spule durchlaufen lassen: }
```

```
For i2:=-Bn to Bn do {in x-Richtung}
```

```
begin
```

```
  For j2:=-Bn to Bn do {in y-Richtung}
```

```
  begin
```

```
    For k2:=-Bn to Bn do {in z-Richtung}
```

```
    begin
```

```
      OrtBx[i2,j2,k2]:=i2*Bsw;  sx:=OrtBx[i2,j2,k2];
```

```
      OrtBy[i2,j2,k2]:=j2*Bsw;  sy:=OrtBy[i2,j2,k2];
```

```
      OrtBz[i2,j2,k2]:=k2*Bsw;  sz:=OrtBz[i2,j2,k2];
```

```
      Berechne_Hges;
```

```
      Bx[i2,j2,k2]:=muo*Hx; {Telsa}
```

```
      By[i2,j2,k2]:=muo*Hy; {Telsa}
```

```
      Bz[i2,j2,k2]:=muo*Hz; {Telsa}
```

```
{      Write(OrtBx[i2,j2,k2]:10:6,' ',OrtBy[i2,j2,k2]:10:6,' ',OrtBz[i2,j2,k2]:10:6);
```

```
      Writeln(' =>',Bx[i2,j2,k2]*1E8:7:4,'E-8, ',By[i2,j2,k2]*1E8:7:4,'E-8, ',Bz[i2,j2,k2]*1E8:7:4,'E-8
```

```
                                                    Tesla');
```

```
      Wait; }
```

```
    end;
```

```
  end;
```

```
  Write('.');
```

```
end; Writeln(' Obere Spule ist durchgerechnet.');
```

```
{ Writeln('Obere Spule, Feld am Ursprung: ');
```

```
Writeln(Bx[0,0,0],', ',By[0,0,0],', ',Bz[0,0,0]*1E8:7:4,' T'); }
```

```
{ Dann die untere Spule dazu addieren: }
```

```
MEyo:=-MEyo; {Position der unteren Spule}
```

```
For i2:=-Bn to Bn do {in x-Richtung}
```

```
begin
```

```
  For j2:=-Bn to Bn do {in y-Richtung}
```

```
  begin
```

```
    For k2:=-Bn to Bn do {in z-Richtung}
```

```
    begin
```

```
      OrtBx[i2,j2,k2]:=i2*Bsw;  sx:=OrtBx[i2,j2,k2];
```

```
      OrtBy[i2,j2,k2]:=j2*Bsw;  sy:=OrtBy[i2,j2,k2];
```

```
      OrtBz[i2,j2,k2]:=k2*Bsw;  sz:=OrtBz[i2,j2,k2];
```

```
      Berechne_Hges;
```

```
      Bx[i2,j2,k2]:=Bx[i2,j2,k2]+muo*Hx; {Telsa}
```

```
      By[i2,j2,k2]:=By[i2,j2,k2]+muo*Hy; {Telsa}
```

```
      Bz[i2,j2,k2]:=Bz[i2,j2,k2]+muo*Hz; {Telsa}
```

```
{      Write(OrtBx[i2,j2,k2]:10:6,' ',OrtBy[i2,j2,k2]:10:6,' ',OrtBz[i2,j2,k2]:10:6);
```

```
      Writeln(' =>',Bx[i2,j2,k2]*1E8:7:4,'E-8, ',By[i2,j2,k2]*1E8:7:4,'E-8, ',Bz[i2,j2,k2]*1E8:7:4,'E-8
```

```
                                                    Tesla');
```

```
      Wait; }
```

```
    end;
```

```
  end;
```

```
  Write('.');
```

```
end; Writeln(' Untere Spule ist durchgerechnet.');
```

```
MEyo:=-MEyo; {MEyo zurücksetzen.}
```

```
Writeln('Gesamtes Feld am Koordinaten-Ursprung: ');
```

```
Writeln(Bx[0,0,0],', ',By[0,0,0],', ',Bz[0,0,0], ' T');
```

```
Writeln; Writeln('Gesamtes Feld im Zentrum der oberen Spule:');
```

```
{oberen Spulenmittelpunkt suchen:} sx:=0; sy:=MEyo; sz:=0;
```

```
Berechne_Hges; BXmax:=muo*Hx; BYmax:=muo*Hy; BZmax:=muo*Hz;
```

```
{unteren Spulenmittelpunkt suchen:} sx:=0; sy:=-MEyo; sz:=0;
```

```
Berechne_Hges; BXmax:=BXmax+muo*Hx; BYmax:=BYmax+muo*Hy; BZmax:=BZmax+muo*Hz;
```

```
Writeln(BXmax,', ',BYmax,', ',BZmax,' T');
```

```
Writeln('Ist dieses Feld gewünscht ? ? ! ? ? ! ? ?');
```

```
Wait; Wait;
```

```
end;
```

```
Procedure Magnetfeld_anzeigen;
```

```
Var i,j,k : Integer;
```

```
begin
```

```
  Writeln('Feld "Magnetische Induktion" des Dauermagneten:');
```

```
  For i:=-Bn to Bn do {in x-Richtung}
```

```
  begin
```

```
    For j:=-Bn to Bn do {in y-Richtung}
```

```
    begin
```

```
      For k:=-Bn to Bn do {in z-Richtung}
```

```
      begin
```

```
        Write('x,y,z=',OrtBx[i,j,k]*100:5:2,' ',OrtBy[i,j,k]*100:5:2,' ',OrtBz[i,j,k]*100:5:2,'cm =>
```

```
                                                    B=');
```

```
        Write(Bx[i,j,k]:8:4,' ');
```

```
        Write(By[i,j,k]:8:4,' ');
```

```
        Write(Bz[i,j,k]:8:4,' T ');
```

```
        Wait;
```

```
      end;
```

```
    end;
```

```
  end;
```



```

SIx[2*zo+2*yo+i+1] :=-xo*Spsw;  SIy[2*zo+2*yo+i+1] :=yo*Spsw;  SIz[2*zo+2*yo+i+1] :=(zo-i-0.5)*Spsw;
                                     {Ort,Mittelpunkt}
STx[2*zo+2*yo+i+1] :=+xo*Spsw;  STy[2*zo+2*yo+i+1] :=yo*Spsw;  STz[2*zo+2*yo+i+1] :=(zo-i-0.5)*Spsw;
                                     {Ort,Mittelpunkt}
dSIx[2*zo+2*yo+i+1]:=0;          dSIy[2*zo+2*yo+i+1]:=0;          dSIz[2*zo+2*yo+i+1]:=-Spsw;
                                     {Richtungsvektor}
dSTx[2*zo+2*yo+i+1]:=0;          dSTy[2*zo+2*yo+i+1]:=0;          dSTz[2*zo+2*yo+i+1]:=-Spsw;
                                     {Richtungsvektor}
end;
For i:=0 to 2*yo do
begin  {und zu guter Letzt in y-Richtung wieder runter}
  SpIx[4*zo+2*yo+i+1]:=-xo*Spsw;  SpIy[4*zo+2*yo+i+1]:=(yo-i)*Spsw;          SpIz[4*zo+2*yo+i+1]:=-zo*Spsw;
                                     {Stützpunkt}
  SpTx[4*zo+2*yo+i+1]:=+xo*Spsw;  SpTy[4*zo+2*yo+i+1]:=(yo-i)*Spsw;          SpTz[4*zo+2*yo+i+1]:=-zo*Spsw;
                                     {Stützpunkt}
  SIx[4*zo+2*yo+i+1] :=-xo*Spsw;  SIy[4*zo+2*yo+i+1] :=(yo-i-0.5)*Spsw;  SIz[4*zo+2*yo+i+1] :=-zo*Spsw;
                                     {Ort,Mittelpunkt}
  STx[4*zo+2*yo+i+1] :=+xo*Spsw;  STy[4*zo+2*yo+i+1] :=(yo-i-0.5)*Spsw;  STz[4*zo+2*yo+i+1] :=-zo*Spsw;
                                     {Ort,Mittelpunkt}
  dSIx[4*zo+2*yo+i+1]:=0;          dSIy[4*zo+2*yo+i+1]:=-Spsw;          dSIz[4*zo+2*yo+i+1]:=0;
                                     {Richtungsvektor}
  dSTx[4*zo+2*yo+i+1]:=0;          dSTy[4*zo+2*yo+i+1]:=-Spsw;          dSTz[4*zo+2*yo+i+1]:=0;
                                     {Richtungsvektor}
end;  {Der letzte Punkt ist dem ersten identisch}
SpN:=4*zo+4*yo+1;
Writeln('Anzahl der Punkte der Spulen-Linienaufteilung: von 1 - ',SpN);
If SpN>SpNmax then
begin
  Writeln('--- ERROR --- zu viele Spulen-Linienelemente');
  Writeln('--- ABHILFE -> Array groesser dimensionieren');
  Wait; Wait; Halt;
end;
{Dann die Flächenelemente:}
For j:=1 to 2*yo do
begin
  For i:=1 to 2*zo do
  begin
    FlIx[i+(j-1)*2*zo]:=-xo*Spsw;  FlIy[i+(j-1)*2*zo]:=(j-0.5-yo)*Spsw;  FlIz[i+(j-1)*2*zo]:=(i-0.5-
    zo)*Spsw;
    FlTx[i+(j-1)*2*zo]:=+xo*Spsw;  FlTy[i+(j-1)*2*zo]:=(j-0.5-yo)*Spsw;  FlTz[i+(j-1)*2*zo]:=(i-0.5-
    zo)*Spsw;
  end;
end;
FlN:=4*zo*yo;
Writeln('Anzahl der Flaechelemente jeder Spulen: von 1 - ',FlN);
If FlN>FlNmax then
begin
  Writeln('--- ERROR --- zu viele Spulen-Flaechelemente');
  Writeln('--- ABHILFE -> Array groesser dimensionieren');
  Wait; Wait; Halt;
end;
end;

Procedure Spulen_anzeigen;  {Spule für den Input der Steuer-Energie}
Var i : Integer;
begin
  Writeln('Input-Sp.-> Stuetzpunkte des Polygons, Orte der FE, Richtungsvektoren der FE:');
  For i:=1 to SpN do
  begin
    Writeln('SP [' ,i:5,']= ',SpIx[i]*100:10:6,', ',SpIy[i]*100:10:6,', ',SpIz[i]*100:10:6,' cm ');
    Writeln('ORT[' ,i:5,']= ', SIx[i]*100:10:6,', ', SIy[i]*100:10:6,', ', SIz[i]*100:10:6,' cm ');
    Writeln('RV [' ,i:5,']= ',dSIx[i]*100:10:6,', ',dSIy[i]*100:10:6,', ',dSIz[i]*100:10:6,' cm ');
    Wait;
  end;
  Writeln('Turbo-Sp.-> Stuetzpunkte des Polygons, Orte der FE, Richtungsvektoren der FE:');
  For i:=1 to SpN do
  begin
    Writeln('SP [' ,i:5,']= ',SpTx[i]*100:10:6,', ',SpTy[i]*100:10:6,', ',SpTz[i]*100:10:6,' cm ');
    Writeln('ORT[' ,i:5,']= ', STx[i]*100:10:6,', ', STy[i]*100:10:6,', ', STz[i]*100:10:6,' cm ');
    Writeln('RV [' ,i:5,']= ',dSTx[i]*100:10:6,', ',dSTy[i]*100:10:6,', ',dSTz[i]*100:10:6,' cm ');
    Wait;
  end;
  Writeln('Input-Spule -> Flaechelemente, deren Mittelpunktspositionen:');
  For i:=1 to FlN do
  begin
    Write('x,y,z[' ,i:5,']= ',FlIx[i]*100:10:6,', ',FlIy[i]*100:10:6,', ',FlIz[i]*100:10:6,' cm ');
    Wait;
  end;
end;

```

```

Writeln('Turbo-Spule -> Flaechenelemente, deren Mittelpunktspitionen:');
For i:=1 to FlN do
begin
  Write('x,y,z[' ,i:5,']= ',FlTx[i]*100:10:6,', ',FlTy[i]*100:10:6,', ',FlTz[i]*100:10:6,' cm ');
  Wait;
end;
Writeln('-----');
end;

Procedure Magnet_drehen(fi:Double); {Drehen um Drehwinkel "fi":}
Var i,j,k : LongInt; {Laufvariablen}
begin
  fi:=fi/180*pi; {Umrechnen des Winkels in Radianten}
  For i:=-Bn to Bn do {x-Anteile}
  begin
    For j:=-Bn to Bn do {y-Anteile}
    begin
      For k:=-Bn to Bn do {z-Anteile}
      begin
        {Drehung der Ortsvektoren:}
        OrtBxDR[i,j,k]:=+OrtBx[i,j,k]*cos(-fi)+OrtBy[i,j,k]*sin(-fi);
        OrtByDR[i,j,k]:=-OrtBx[i,j,k]*sin(-fi)+OrtBy[i,j,k]*cos(-fi);
        OrtBzDR[i,j,k]:=+OrtBz[i,j,k];
        {Drehung der Feldstärke-Vektoren:}
        BxDR[i,j,k]:=+Bx[i,j,k]*cos(-fi)+By[i,j,k]*sin(-fi);
        ByDR[i,j,k]:=-Bx[i,j,k]*sin(-fi)+By[i,j,k]*cos(-fi);
        BzDR[i,j,k]:=+Bz[i,j,k];
        {Magnetfeld zeilenweise erst ungedreht und dann gedreht anzeigen:}
        {
          Write('x,y,z=',OrtBx[i,j,k]:5:2,', ',OrtBy[i,j,k]:5:2,', ',OrtBz[i,j,k]:5:2,'mm => B=');
          Write(Bx[i,j,k]:8:4,', ');
          Write(By[i,j,k]:8:4,', ');
          Write(Bz[i,j,k]:8:4,' T '); Writeln;
          Write('x,y,z=',OrtBxDR[i,j,k]:5:2,', ',OrtByDR[i,j,k]:5:2,', ',OrtBzDR[i,j,k]:5:2,'mm => B=');
          Write(BxDR[i,j,k]:8:4,', ');
          Write(ByDR[i,j,k]:8:4,', ');
          Write(BzDR[i,j,k]:8:4,' T ');
          Wait; Writeln; }
        end;
      end;
    end;
  end;
end;

Procedure Feldstaerke_am_Ort_suchen(xpos,ypos,zpos:Double); {an dem Ort suche ich die Feldstärke}
Var ixo,iyo,izo : Integer;
    ix,iy,iz : Integer;
    dist,disto : Double;
begin
  {Zuerst suche ich, welcher Feldort zum xpos,ypos,zpos den kürzesten Abstand hat.}
  ixo:=0; iyo:=0; izo:=0;
  disto:=Sqrt(Sqr(xpos-OrtBxDR[ixo,iyo,izo])+Sqr(ypos-OrtByDR[ixo,iyo,izo])+Sqr(zpos-
OrtBzDR[ixo,iyo,izo]));
  { Writeln('Anfangs-Abstand vom Nullpunkt: ',disto*100:1:15,' cm'); }
  For ix:=-Bn to Bn do {x-Suche}
  begin
    For iy:=-Bn to Bn do {y-Suche}
    begin
      For iz:=-Bn to Bn do {z-Suche}
      begin
        dist:=Sqrt(Sqr(xpos-OrtBxDR[ix,iy,iz])+Sqr(ypos-OrtByDR[ix,iy,iz])+Sqr(zpos-OrtBzDR[ix,iy,iz]));
        If dist<=disto then
        begin
          ixo:=ix; iyo:=iy; izo:=iz;
          disto:=dist;
          {
            Write('Position: ',OrtBxDR[ix,iy,iz]*100:8:5,', ',OrtByDR[ix,iy,iz]*100:8:5,',
              ',OrtBzDR[ix,iy,iz]*100:8:5,' cm'); }
          {
            Writeln(disto); } {Wait;}
          end;
        end;
      end;
    end;
  end;
  { Writeln('Punkt Nummer (ixo,iyo,izo): ',ixo,', ',iyo,', ',izo); }
  {Dann gebe ich das Magnetfeld ebendort an:}
  { Writeln('Magnetfeld dort: ',BxDR[ixo,iyo,izo]:8:4,', ',ByDR[ixo,iyo,izo]:8:4,',
    ',BzDR[ixo,iyo,izo]:8:4,' T '); }
  {Jetzt brauche ich noch den magnetischen Fluß durch das dortige Spulen-Flächenelement:}
  PsiSFE:=BxDR[ixo,iyo,izo]*Spsw*Spsw; {nach *1 von S.3}
  { Writeln('Magn. Fluss durch Spulen-Flaechenelement: ',PsiSFE,' T*m^2'); }
end;

```

```

Procedure Gesamtfluss_durch_Input_Spule; {gemäß *2 von S.3}
Var i : Integer;
begin
  PsiGES:=0;
  For i:=1 to FlN do
    begin
      Feldstaerke_am_Ort_suchen(FlIx[i],FlIy[i],FlIz[i]);
      PsiGES:=PsiGES+PsiSFE;
    end;
end;

Procedure Gesamtfluss_durch_Turbo_Spule; {gemäß *2 von S.3}
Var i : Integer;
begin
  PsiGES:=0;
  For i:=1 to FlN do
    begin
      Feldstaerke_am_Ort_suchen(FlTx[i],FlTy[i],FlTz[i]);
      PsiGES:=PsiGES+PsiSFE;
    end;
end;

Procedure FourierDatenspeicherung(PSIF : Array of Double); {Magnetischer Fluß für Fourier-Entwicklung}
Var i : Integer;
    fout : Text;
begin
  Assign(fout,'PSIF.DAT'); Rewrite(fout); {File öffnen}
  Writeln('FOURIER - HIER:');
  For i:=0 to 360 do Writeln(fout,PSIF[i]);
  Close(fout);
end;

Procedure FourierEntwicklung;
Var i : Integer;
    PSIF : Array [0..360] of Double;
    fin : Text;
    Qsplus,Qsmitte,Qsminus : Double;
    Qanf,Q1p,Q1m,Q2p,Q2m,Q3p,Q3m : Double; {für B1,2,3 - Iteration}
    Q4p,Q4m,Q5p,Q5m : Double; {für B4,5 - Iteration}
    Qsminimum : Double; {zur Minimums-Suche}
    weiter : Boolean;
Function QuadSum1:Double;
Var merk : Double;
    i : Integer;
begin
  merk:=0; {'i' ist Laufvariable für den Winkel, anzugeben in Grad}
  For i:=0 to 360 do merk:=merk+Sqr(PSIF[i]-B1*sin(i/360*2*pi));
  QuadSum1:=merk;
end;
Function Fourier(t,Ko1,Ko2,Ko3,Ko4,Ko5:Double):Double;
Var merk : Double;
begin {'t' ist Variable für den Winkel, anzugeben in Grad}
  merk:=Ko1*sin(t/360*2*pi);
  merk:=merk+Ko2*sin(2*t/360*2*pi);
  merk:=merk+Ko3*sin(3*t/360*2*pi);
  merk:=merk+Ko4*sin(4*t/360*2*pi);
  merk:=merk+Ko5*sin(5*t/360*2*pi);
  Fourier:=merk;
end;
Function QuadSum3(Koeff1,Koeff2,Koeff3:Double):Double;
Var merk : Double;
    i : Integer;
begin
  merk:=0; {'i' ist Laufvariable für den Winkel, anzugeben in Grad}
  For i:=0 to 360 do merk:=merk+Sqr(PSIF[i]-Koeff1*sin(i/360*2*pi)-Koeff2*sin(2*i/360*2*pi)-
    Koeff3*sin(3*i/360*2*pi));
  QuadSum3:=merk;
end;
Function QuadSum5(Koeff1,Koeff2,Koeff3,Koeff4,Koeff5:Double):Double;
Var merk : Double;
    i : Integer;
begin
  merk:=0;
  For i:=0 to 360 do {'i' ist Laufvariable für den Winkel, anzugeben in Grad}
    begin
      If PSIF[i]>>0 then merk:=merk+Sqr(PSIF[i]-Fourier(i,Koeff1,Koeff2,Koeff3,Koeff4,Koeff5));
    end;
end;

```

```

QuadSum5:=merk;
end;
begin
Assign(fin,'PSIF.DAT'); Reset(fin); {File öffnen}
Writeln('FOURIER - ENTWICKLUNG:');
For i:=0 to 360 do Readln(fin,PSIF[i]);
Close(fin);
B1:=0; {Mittelwert über erste Periode als Startwert für Grundschiwingung}
For i:=0 to 180 do B1:=B1+PSIF[i];
{Zuerst die Größenordnung von B1 abschätzen;}
B1:=B1/90; {writeln('B1 : ',B1); Wait;}
{Jetzt B1 anpassen über die Minimierung der Abweichungsquadrate;}
Repeat
  B1:=0.99*B1; QSminus:=QuadSum1;
  B1:=B1/0.99; QSmitte:=QuadSum1;
  B1:=1.01*B1; QSplus:=QuadSum1; B1:=B1/1.01;
  weiter:=false;
  If QSminus<QSmitte then begin B1:=0.99*B1; weiter:=true; end;
  If QSplus<QSmitte then begin B1:=1.01*B1; weiter:=true; end;
{ Writeln('QS: ',QSminus,', ',QSmitte,', ',QSplus); }
Until Not(weiter);
writeln('B1-vorab : ',B1,', QS = ',QSmitte);
{Die Werte zur Kontrolle herauschreiben;}
AnzP:=360; Abstd:=1;
For i:=0 to 360 do {'i' ist Laufvariable für den Winkel, anzugeben in Grad}
begin
  Q[i]:=PSIF[i]; Qp[i]:=B1*sin(i/360*2*pi);
end;
{Dann B1 & B2 & B3 anpassen über die Minimierung der Abweichungsquadrate;}
{Startwerte für B2 suchen;}
B2:=0;
B2:=+B1/10; QSplus:=QuadSum3(B1,B2,0);
B2:=-B1/10; QSminus:=QuadSum3(B1,B2,0);
If QSplus<QSminus then B2:=+B1/10;
If QSminus<QSplus then B2:=-B1/10;
{Startwerte für B3 suchen;}
B3:=0;
B3:=+B1/10; QSplus:=QuadSum3(B1,B2,B3);
B3:=-B1/10; QSminus:=QuadSum3(B1,B2,B3);
If QSplus<QSminus then B3:=+B1/10;
If QSminus<QSplus then B3:=-B1/10;
Writeln('AnfB1,2,3: ',B1:20,' ',B2:20,' ',B3:20);
{Jetzt über Iteration die B1, B2, B3 fitten;}
Repeat
  {QuadSummen berechnen;}
  Qanf:=QuadSum3(B1,B2,B3);
  Q1p:=QuadSum3(B1*1.01,B2,B3); Q1m:=QuadSum3(B1*0.99,B2,B3);
  Q2p:=QuadSum3(B1,B2*1.01,B3); Q2m:=QuadSum3(B1,B2*0.99,B3);
  Q3p:=QuadSum3(B1,B2,B3*1.01); Q3m:=QuadSum3(B1,B2,B3*0.99);
  {Kleinste QuadSumme suchen;}
  QSminimum:=Qanf;
  If Q1p<QSminimum then QSminimum:=Q1p; If Q1m<QSminimum then QSminimum:=Q1m;
  If Q2p<QSminimum then QSminimum:=Q2p; If Q2m<QSminimum then QSminimum:=Q2m;
  If Q3p<QSminimum then QSminimum:=Q3p; If Q3m<QSminimum then QSminimum:=Q3m;
  {Koeffizienten zur kleinsten QuadSumme einstellen;}
  weiter:=false;
  If Q1p=QSminimum then begin B1:=B1*1.01; weiter:=true; end;
  If Q1m=QSminimum then begin B1:=B1*0.99; weiter:=true; end;
  If Q2p=QSminimum then begin B2:=B2*1.01; weiter:=true; end;
  If Q2m=QSminimum then begin B2:=B2*0.99; weiter:=true; end;
  If Q3p=QSminimum then begin B3:=B3*1.01; weiter:=true; end;
  If Q3m=QSminimum then begin B3:=B3*0.99; weiter:=true; end;
{ Writeln('QS: ',QSminimum); }
Until Not(weiter);
Writeln('Nun B1 = ',B1:17,', B2 = ',B2:17,' B3 = ',B3:17);
Writeln('Zugehoerige Quadsum: ',Quadsum3(B1,B2,B3));
{Die Werte zur Kontrolle herauschreiben;}
For i:=0 to 360 do
begin
  Qpp[i]:=B1*sin(i/360*2*pi)+B2*sin(2*i/360*2*pi)+B3*sin(3*i/360*2*pi);
end;
{Nun will ich alle Ausreißer mit mehr als 75% Abweichung löschen;}
For i:=0 to 360 do
begin
  If Abs(PSIF[i]-(B1*sin(i/360*2*pi)-B2*sin(2*i/360*2*pi)-B3*sin(3*i/360*2*pi)))>Abs(0.75*B1) then
    PSIF[i]:=0;
end;
{Dazu will nun eine Fourier-Reihe mit 5 Koeffizienten fitten;}

```

```

{Startwerte für B4 suchen;}
B4:=0;
B4:=+B1/40;  QSpplus:=QuadSum5(B1,B2,B3,B4,0);
B4:=-B1/40;  QSminus:=QuadSum5(B1,B2,B3,B4,0);
If QSpplus<QSminus then B4:=+B1/40;
If QSminus<QSpplus then B4:=-B1/40;
{Startwerte für B5 suchen;}
B5:=0;
B5:=+B1/40;  QSpplus:=QuadSum5(B1,B2,B3,B4,B5);
B5:=-B1/40;  QSminus:=QuadSum5(B1,B2,B3,B4,B5);
If QSpplus<QSminus then B5:=+B1/10;
If QSminus<QSpplus then B5:=-B1/10;
WriteLn('Und B4,5: ',B4:20,' ',B5:20);
WriteLn('Anf Quadsum: ',QuadSum5(B1,B2,B3,B4,B5));
{Jetzt über Iteration die B1, B2, B3, B4, B5 fitten;}
Repeat
  {QuadSummen berechnen;}
  Qanf:=QuadSum5(B1,B2,B3,B4,B5);
  Q1p:=QuadSum5(B1*1.01,B2,B3,B4,B5);      Q1m:=QuadSum5(B1*0.99,B2,B3,B4,B5);
  Q2p:=QuadSum5(B1,B2*1.01,B3,B4,B5);      Q2m:=QuadSum5(B1,B2*0.99,B3,B4,B5);
  Q3p:=QuadSum5(B1,B2,B3*1.01,B4,B5);      Q3m:=QuadSum5(B1,B2,B3*0.99,B4,B5);
  Q4p:=QuadSum5(B1,B2,B3,B4*1.01,B5);      Q4m:=QuadSum5(B1,B2,B3,B4*0.99,B5);
  Q5p:=QuadSum5(B1,B2,B3,B4,B5*1.01);      Q5m:=QuadSum5(B1,B2,B3,B4,B5*0.99);
  {Kleinste QuadSumme suchen;}
  QSminimum:=Qanf;
  If Q1p<QSminimum then QSminimum:=Q1p;    If Q1m<QSminimum then QSminimum:=Q1m;
  If Q2p<QSminimum then QSminimum:=Q2p;    If Q2m<QSminimum then QSminimum:=Q2m;
  If Q3p<QSminimum then QSminimum:=Q3p;    If Q3m<QSminimum then QSminimum:=Q3m;
  If Q4p<QSminimum then QSminimum:=Q4p;    If Q4m<QSminimum then QSminimum:=Q4m;
  If Q5p<QSminimum then QSminimum:=Q5p;    If Q5m<QSminimum then QSminimum:=Q5m;
  {Koeffizienten zur kleinsten QuadSumme einstellen;}
  weiter:=false;
  If Q1p=QSminimum then begin B1:=B1*1.01; weiter:=true; end;
  If Q1m=QSminimum then begin B1:=B1*0.99; weiter:=true; end;
  If Q2p=QSminimum then begin B2:=B2*1.01; weiter:=true; end;
  If Q2m=QSminimum then begin B2:=B2*0.99; weiter:=true; end;
  If Q3p=QSminimum then begin B3:=B3*1.01; weiter:=true; end;
  If Q3m=QSminimum then begin B3:=B3*0.99; weiter:=true; end;
  If Q4p=QSminimum then begin B4:=B4*1.01; weiter:=true; end;
  If Q4m=QSminimum then begin B4:=B4*0.99; weiter:=true; end;
  If Q5p=QSminimum then begin B5:=B5*1.01; weiter:=true; end;
  If Q5m=QSminimum then begin B5:=B5*0.99; weiter:=true; end;
{ WriteLn('QS: ',QSminimum); }
Until Not(weiter);
WriteLn('Ergebnis: B1 = ',B1:17,' B2 = ',B2:17,' B3 = ',B3:17);
WriteLn('          B4 = ',B4:17,' B5 = ',B5:17);
WriteLn('Endliche Quadsum: ',Quadsum5(B1,B2,B3,B4,B5));
{Die Werte zur Kontrolle herausschreiben;}
For i:=0 to 360 do
begin
  phipp[i]:=Fourier(i,B1,B2,B3,B4,B5)
end;
ExcelAusgabe('fourier.dat',6);
end;

Function FlussI(alpha:Double):Double;
Var merk : Double; {Hier ist alpha in 'radianten' anzugeben.}
begin
  merk:=B1I*sin(alpha);
  merk:=merk+B2I*sin(2*alpha);
  merk:=merk+B3I*sin(3*alpha);
  merk:=merk+B4I*sin(4*alpha);
  merk:=merk+B5I*sin(5*alpha);
  FlussI:=merk;
end;

Function FlussT(alpha:Double):Double;
Var merk : Double; {Hier ist alpha in 'radianten' anzugeben.}
begin
  merk:=B1T*sin(alpha);
  merk:=merk+B2T*sin(2*alpha);
  merk:=merk+B3T*sin(3*alpha);
  merk:=merk+B4T*sin(4*alpha);
  merk:=merk+B5T*sin(5*alpha);
  FlusT:=merk;
end;

Procedure SinusEntwicklung_fuer_Drehmoment;

```

```

Var i,j,jmerk : Integer;
PSIF : Array [0..360] of Double;
fin : Text;
QSalt,QSneu : Double;
weiter : Boolean;
Qanf,QB1plus,QB1minus,Qphaseplus,Qphaseminus : Double; {für numerische Iteration}
QSminimum : Double; {Zur Suche des kleinsten Abweichungsquadrates.}
Function QuadSum2(B1lok,phaselok:Double):Double;
Var merk : Double;
i : Integer;
begin
merk:=0; {'i' ist Laufvariable für den Winkel, anzugeben in Grad}
For i:=0 to 360 do merk:=merk+Sqr(PSIF[i]-B1lok*sin((i+phaselok)/360*2*pi));
QuadSum2:=merk;
end;
begin
Assign(fin,'PSIF.DAT'); Reset(fin); {File öffnen}
Writeln('FOURIER-ENTWICKLUNG FUER DIE SCHNELLE DREHMOMENTS-BERECHNUNG:');
For i:=0 to 360 do Readln(fin,PSIF[i]);
Close(fin);
B1:=0; {Startwert für Grundschiwingung "B1" suchen}
For i:=0 to 360 do
begin
If PSIF[i]>B1 then B1:=PSIF[i];
end;
Writeln('Startwert von B1: ',B1); Wait;
phase:=0; QSalt:=QuadSum2(B1,phase); jmerk:=Round(phase); {Startwert für Grundschiwingung "phase" suchen}
For j:=1 to 360 do
begin
phase:=j; QSneu:=QuadSum2(B1,phase);
If QSneu<QSalt then
begin
QSalt:=QSneu;
jmerk:=j;
{ Writeln(phase,' => ',QSalt); Wait; }
end;
phase:=jmerk;
end;
Writeln('Startwert von phase: ',phase); Wait;
{Jetzt folgt noch eine genaue Iteration der Parameter:}
Repeat
{QuadSummen berechnen:}
Qanf:=QuadSum2(B1,phase);
QB1plus:=QuadSum2(B1*1.0001,phase);
QB1minus:=QuadSum2(B1*0.9999,phase);
Qphaseplus:=QuadSum2(B1,phase*1.0001);
Qphaseminus:=QuadSum2(B1,phase*0.9999);
{Kleinste QuadSumme suchen:}
QSminimum:=Qanf;
If QB1plus<QSminimum then QSminimum:=QB1plus;
If QB1minus<QSminimum then QSminimum:=QB1minus;
If Qphaseplus<QSminimum then QSminimum:=Qphaseplus;
If Qphaseminus<QSminimum then QSminimum:=Qphaseminus;
{Koeffizienten zur kleinsten QuadSumme einstellen:}
weiter:=false;
If QB1plus=QSminimum then begin B1:=B1*1.0001; weiter:=true; end;
If QB1minus=QSminimum then begin B1:=B1*0.9999; weiter:=true; end;
If Qphaseplus=QSminimum then begin phase:=phase*1.0001; weiter:=true; end;
If Qphaseminus=QSminimum then begin phase:=phase*0.9999; weiter:=true; end;
Writeln('QS: ',QSminimum);
Until Not(weiter);
phase:=phase/360*2*pi; {Phase auf Radianten einstellen}
Bldreh:=B1; {Drehmoment-Amplitude weitergeben.}
end;

Function Schnell_Drehmoment(winkel:Double):Double;
begin
Schnell_Drehmoment:=Bldreh*sin(winkel+phase);
end;

Procedure Magfeld_Turbo_Berechnen(rx,ry,rz,Strom:Double);
Var i : Integer;
sx,sy,sz : Double; {Orte der Leiterschleifen-Elemente}
dsx,dsy,dsz : Double; {Laufrichtungen der Leiterschleifen-Elemente}
AnzLSE : Integer; {Anzahl der Leiterschleifen-Elemente}
smrx,smry,smrz : Double; {Differenzen für das nachfolgende Kreuzprodukt}
krpx,krpy,krpz : Double; {Kreuzprodukt in Biot-Savart}
smrbetrhoch3 : Double; {Betragsbildung für Nenner}

```

```

    dHx,dHy,dHz      : Double; {Infinitesimales Magnetfeld}
    Hgesx,Hgesy,Hgesz:Double;{Gesamt-Magnetfeld der Input-Spule am Aufpunkt}
begin
{ Spulen_anzeigen; } {Optional aufrufbares Unterprogramm.}
AnzLSE:=SpN-1;
If AnzLSE<>4*yo+4*zo then
begin
    Writeln('Da timmt wat nich: Vernetzung der felderzeugenden Spule ist falsch. ');
    Writeln('Problem bei der TURBO-Spule: ');
    Writeln('Anzahl der Stuetzpunkte der Spulen, AnzLSE = ',AnzLSE);
    Writeln('Hingegen: 4*yo+4*zo = ',4*yo+4*zo);
    Wait; Wait; Halt;
end;
{Ort und Richtungsvektoren der Leiterschleifen-Elemente feststellen, Feld am Aufpunkt ausrechnen nach
                                                    Biot-Savart:}
Hgesx:=0; Hgesy:=0; Hgesz:=0;
For i:=1 to AnzLSE do
begin
    sx:=SpTx[i]; sy:=SpTy[i]; sz:=SpTz[i];      {Orte der Leiterschleifen-Elemente}
    dsx:=dSTx[i]; dsy:=dSTy[i]; dsz:=dSTz[i];  {Richtungsvektoren der Leiterschleifen-Elemente}
    smrx:=sx-rx; smry:=sy-ry; smrz:=sz-rz;     {Differenzen für das nachfolgende Kreuzprodukt}
    krpx:=dsy*smrz-dsz*smry; krpy:=dsz*smrx-dsx*smrz; krpz:=dsx*smry-dsy*smrx; {Kreuzprodukt}
    smrbetrhoch3:=Sqrt(Sqr(smrx)+Sqr(smry)+Sqr(smrz));
    If smrbetrhoch3<Spsw/1000 then
    begin
        Writeln('Mechanische Kollision -> Magnet beruehrt Turbo-Spule. STOP. ');
        Writeln('Spulen-Element bei: ',sx:18,', ',sy:18,', ',sz:18,'m. ');
        Writeln('Magnet-Ort bei: ',rx:18,', ',ry:18,', ',rz:18,'m. ');
        Wait; Wait; Halt;
    end;
    smrbetrhoch3:=smrbetrhoch3*smrbetrhoch3*smrbetrhoch3; {Betragsbildung für Nenner in Biot-Savart}
    dHx:=Strom*krpx/4/pi/smrbetrhoch3;          {Finites Magnetfeld des Leiterschleifen-Elements}
    dHy:=Strom*krpy/4/pi/smrbetrhoch3;
    dHz:=Strom*krpz/4/pi/smrbetrhoch3;
    Hgesx:=Hgesx+dHx; Hgesy:=Hgesy+dHy; Hgesz:=Hgesz+dHz; {Summation der Feldelemente}
end;
    {Vorzeichen in der nachfolgenden Zeile gemäß technischer Stromrichtung.}
    BTx:=-muo*Hgesx*Nturbo; BTy:=-muo*Hgesy*Nturbo; BTz:=-muo*Hgesz*Nturbo;
end;

Procedure Magfeld_Input_Berechnen(rx,ry,rz,Strom:Double);
Var i : Integer;
    sx,sy,sz      : Double;      {Orte der Leiterschleifen-Elemente}
    dsx,dsy,dsz  : Double;      {Laufrichtungen der Leiterschleifen-Elemente}
    AnzLSE       : Integer;      {Anzahl der Leiterschleifen-Elemente}
    smrx,smry,smrz : Double;     {Differenzen für das nachfolgende Kreuzprodukt}
    krpx,krpy,krpz : Double;     {Kreuzprodukt in Biot-Savart}
    smrbetrhoch3 : Double;       {Betragsbildung für Nenner}
    dHx,dHy,dHz  : Double;       {Infinitesimales Magnetfeld}
    Hgesx,Hgesy,Hgesz:Double;    {Gesamt-Magnetfeld der Input-Spule am Aufpunkt}
begin
{ Spulen_anzeigen; } {Optional aufrufbares Unterprogramm.}
AnzLSE:=SpN-1;
If AnzLSE<>4*yo+4*zo then
begin
    Writeln('Da timmt wat nich: Vernetzung der felderzeugenden Spule ist falsch. ');
    Writeln('Problem bei der INPUT-Spule: ');
    Writeln('Anzahl der Stuetzpunkte der Spulen, AnzLSE = ',AnzLSE);
    Writeln('Hingegen: 4*yo+4*zo = ',4*yo+4*zo);
    Wait; Wait; Halt;
end;
{Ort und Richtungsvektoren der Leiterschleifen-Elemente feststellen, Feld am Aufpunkt ausrechnen nach
                                                    Biot-Savart:}
Hgesx:=0; Hgesy:=0; Hgesz:=0;
For i:=1 to AnzLSE do
begin
    sx:=SpIx[i]; sy:=SpIy[i]; sz:=SpIz[i];      {Orte der Leiterschleifen-Elemente}
    dsx:=dSIx[i]; dsy:=dSIy[i]; dsz:=dSIz[i];  {Richtungsvektoren der Leiterschleifen-Elemente}
    smrx:=sx-rx; smry:=sy-ry; smrz:=sz-rz;     {Differenzen für das nachfolgende Kreuzprodukt}
    krpx:=dsy*smrz-dsz*smry; krpy:=dsz*smrx-dsx*smrz; krpz:=dsx*smry-dsy*smrx; {Kreuzprodukt}
    smrbetrhoch3:=Sqrt(Sqr(smrx)+Sqr(smry)+Sqr(smrz));
    If smrbetrhoch3<Spsw/1000 then
    begin
        Writeln('Mechanische Kollision -> Magnet beruehrt Input-Spule. STOP. ');
        Writeln('Spulen-Element bei: ',sx:18,', ',sy:18,', ',sz:18,'m. ');
        Writeln('Magnet-Ort bei: ',rx:18,', ',ry:18,', ',rz:18,'m. ');
        Wait; Wait; Halt;
    end;
    smrbetrhoch3:=smrbetrhoch3*smrbetrhoch3*smrbetrhoch3; {Betragsbildung für Nenner in Biot-Savart}

```

```

dHx:=Strom*krpx/4/pi/smrbehrhoch3;           {Finites Magnetfeld des Leiterschleifen-Elements}
dHy:=Strom*krpy/4/pi/smrbehrhoch3;
dHz:=Strom*krpz/4/pi/smrbehrhoch3;
Hgesx:=Hgesx+dHx;  Hgesy:=Hgesy+dHy;  Hgesz:=Hgesz+dHz; {Summation der Feldelemente}
end;           {Vorzeichen in der nachfolgenden Zeile gemäß technischer Stromrichtung.}
BIx:=-muo*Hgesx*Ninput;  BIy:=-muo*Hgesy*Ninput;  BIZ:=-muo*Hgesz;
end;

Function Drehmoment(alpha:Double):Double; {Das Argument ist der Winkel der Magnetstellung "alpha"}
Var i : Integer; {Laufvariable}
    Idlx,Idly,Idlz : Double; {Kartesische Komponenten von dl-Vektor nach (*1 von S.11)}
    Bxlok,Bylok,Bzlok : Double; {lokale Magnetfeld-Werte}
    FLx,FLy,FLz : Double; {Lorentz-Kraft als Kreuzprodukt}
    dMx,dMy,dMz : Double; {Drehmoment, das jedes Leiterschleifen-Element auf seinen gesamten Magneten
                                ausübt.}
    MgesX,MgesY,MgesZ : Double; {Gesamt-Drehmoment der Turbo- & Input- Spule auf den Magneten (aus
                                Emulations-Spulen).}
    rx,ry,rz : Double; {Ortsangabe der Magnetfeld-Emulationsspulen-Elemente nach Dreh-Transformation}
begin
    MgesX:=0; MgesY:=0; MgesZ:=0;
    For i:=1 to MESEanz do
        begin
            {Wir beginnen mit der Berechnung der Lorentz-Kraft auf jedes einzelne Element der Magnetfeld-
                                Emulations-Spule}
            Idlx:=MEI*MESEdx[i]*4*pi*MEro/MESEanz; {Magnetfeld-Emulations-Leiterschleifen-Element}
            Idly:=MEI*MESEdy[i]*4*pi*MEro/MESEanz; {Magnetfeld-Emulations-Leiterschleifen-Element}
            Idlz:=MEI*MESEdz[i]*4*pi*MEro/MESEanz; {Magnetfeld-Emulations-Leiterschleifen-Element}
            {Es folgt die Berechnung der Magnetfeld-Stärke der Input- und Turbo- Spule am Ort des Leiterschleifen-
                                Elements}
            Magfeld_Input_Berechnen(MESEx[i],MESEy[i],MESEz[i],qpoI); {Strom durch die Input-Spule einstellen !!}
            Magfeld_Turbo_Berechnen(MESEx[i],MESEy[i],MESEz[i],qpoT); {Strom "QP" durch die Input-Spule einstellen
                                !!}
            Bxlok:=BIx+BTx;  Bylok:=BIy+BTy;  Bzlok:=BIZ+BTz;  {lokales B-feld der Input- und der Turbo- Spule am
                                Ort des Leiterschleifen-Elements}

            {Kreuzprodukt bilden zur Berechnung der Lorentz-Kraft:}
            FLx:=Idly*Bzlok-Idlz*Bylok;
            FLy:=Idlz*Bxlok-Idlx*Bzlok;
            FLz:=Idlx*Bylok-Idly*Bxlok;
            {Kontrolle der Lorentz-Kraft:}
        { Writeln('Ort: ',MESEx[i],', ',MESEy[i],', ',MESEz[i]);
          Writeln(' dl: ',MESEdx[i],', ',MESEdy[i],', ',MESEdz[i]);
          Writeln('FLo: ',FLx,', ',FLy,', ',FLz);
          }
            {Den wirkenden Ort, an dem das Drehmoment angreift, stellen wir durch eine Drehtransformation gemäß *1
                                von S.12 fest:}
            rx:=+MESEx[i]*cos(-alpha)+MESEy[i]*sin(-alpha);
            ry:=-MESEx[i]*sin(-alpha)+MESEy[i]*cos(-alpha);
            rz:=MESEz[i];
            {Daraus berechnen wir nun das zugehörige Drehmoment-Element, das dieses Lorenzt-Kraft-Element auf dem
                                Magneten ausübt:}

            dMx:=ry*FLz-rz*FLy;           {Drehmoment als Kreuzprodukt M = r x F }
            dMy:=rz*FLx-rx*FLz;
            dMz:=rx*FLy-ry*FLx;
            {Kontrolle des Drehmoments:}
        { Writeln('Dreh:',dMx,', ',dMy,', ',dMz); Wait;
          }
            MgesX:=MgesX+dMx; {Summation aller einzelnen Drehmoment-Elemente zum Gesamt-Drehmoment.}
            MgesY:=MgesY+dMy; {in drei kartesischen Komponenten}
            MgesZ:=MgesZ+dMz; {Wegen der Lagerung des Magneten spielt nur die z-Komponente des Drehmoments mit.}
        end;           {Der Magnet hat nämlich eine starre Achse und rotiert nur um die z-Achse.}
    { Writeln('Drehmoment:',MgesX:20,', ',Mgesy:20,', ',Mgesz:20);
      Drehmoment:=MgesZ;
    }
end;

Procedure Daten_Speichern;
Var fout : Text;
    i,j,k : Integer;
begin
    Assign(fout,'schonda'); Rewrite(fout); {File öffnen}
    {Zuerst die Parameter:}
    Writeln(fout,Spsw);
    Writeln(fout,xo);
    Writeln(fout,yo);
    Writeln(fout,zo);
    Writeln(fout,Ninput);
    Writeln(fout,Nturbo);
    Writeln(fout,Bsw);
    Writeln(fout,MEyo);
    Writeln(fout,MEro);
    Writeln(fout,MEI);

```

```

{Dann das Magnetfeld:} {Die Schritt-Anzahl Bn bleibt "Const."}
For i:=-Bn to Bn do {in x-Richtung}
begin
  For j:=-Bn to Bn do {in y-Richtung}
  begin
    For k:=-Bn to Bn do {in z-Richtung}
    begin
      Writeln(fout,OrtBx[i,j,k]);
      Writeln(fout,OrtBy[i,j,k]);
      Writeln(fout,OrtBz[i,j,k]);
      Writeln(fout,Bx[i,j,k]);
      Writeln(fout,By[i,j,k]);
      Writeln(fout,Bz[i,j,k]);
    end;
  end;
end;
{Spulenzuweisung und Stromverteilung brauche ich nicht speichern, die kann man rechnen lassen.}
{Die Drehmoments-Parameter muß ich auch abspeichern:}
Writeln(fout,B1T);
Writeln(fout,B2T);
Writeln(fout,B3T);
Writeln(fout,B4T);
Writeln(fout,B5T);
Writeln(fout,B1I);
Writeln(fout,B2I);
Writeln(fout,B3I);
Writeln(fout,B4I);
Writeln(fout,B5I);
Writeln(fout,B1dreh);
Writeln(fout,phase);
Writeln(fout,'Die Daten sind alleherausgeschrieben.');
```

```

Close(fout);
end;

Procedure Alte_Parameter_vergleichen;
Var fin : Text;
    x : Double; {Parameter zum Einlesen}
    n : Integer; {Parameter zum Einlesen}
    i,j,k : Integer;
begin
  Assign(fin,'schonda'); Reset(fin); {File öffnen}
  {Zuerst die Parameter:}
  Readln(fin,x); If x<>Spsw then begin schonda:=false; Writeln(' Spsw geändert'); end;
  Readln(fin,n); If n<>xo then begin schonda:=false; Writeln(' xo geändert'); end;
  Readln(fin,n); If n<>yo then begin schonda:=false; Writeln(' yo geändert'); end;
  Readln(fin,n); If n<>zo then begin schonda:=false; Writeln(' zo geändert'); end;
  Readln(fin,n); If n<>Ninput then begin schonda:=false; Writeln('Ninput geändert'); end;
  Readln(fin,n); If n<>Nturbo then begin schonda:=false; Writeln('Nturbo geändert'); end;
  Readln(fin,x); If x<>Bsw then begin schonda:=false; Writeln(' Bsw geändert'); end;
  Readln(fin,x); If x<>MEyo then begin schonda:=false; Writeln(' MEyo geändert'); end;
  Readln(fin,x); If x<>MEro then begin schonda:=false; Writeln(' MEro geändert'); end;
  Readln(fin,x); If x<>MEI then begin schonda:=false; Writeln(' MEI geändert'); end;
  If schonda then Writeln('Die Parameter sind bereits bekannt.');
```

```

  If Not(schonda) then
  begin
    Writeln('Die Parameter sind neu. Es beginnt eine neue Vernetzung.');
```

```

    Wait; Wait;
  end;
end;
{Dann das Magnetfeld:} {Die Schritt-Anzahl Bn bleibt "Const."}
For i:=-Bn to Bn do {in x-Richtung}
begin
  For j:=-Bn to Bn do {in y-Richtung}
  begin
    For k:=-Bn to Bn do {in z-Richtung}
    begin
      Readln(fin,OrtBx[i,j,k]);
      Readln(fin,OrtBy[i,j,k]);
      Readln(fin,OrtBz[i,j,k]);
      Readln(fin,Bx[i,j,k]);
      Readln(fin,By[i,j,k]);
      Readln(fin,Bz[i,j,k]);
    end;
  end;
end;
Writeln('Das Magnetfeld ist gelesen.');
```

```

{Spulenzuweisung und Stromverteilung brauche ich nicht speichern, die kann man rechnen lassen.}
{Die Drehmoments-Parameter muß ich auch abspeichern:}
Readln(fin,B1T);
```

```

Readln(fin,B2T);
Readln(fin,B3T);
Readln(fin,B4T);
Readln(fin,B5T);
Readln(fin,B1I);
Readln(fin,B2I);
Readln(fin,B3I);
Readln(fin,B4I);
Readln(fin,B5I);
Writeln('Die Parameter zur Berechnung des magnetischen Flusses sind gelesen.');
```

```

Readln(fin,Bldreh);
Readln(fin,phase);
Writeln('Die Parameter zur schnellen Berechnung des Drehmoments sind gelesen.');
```

```

Writeln('Damit steht alles fuer den DFEM-Algorithmus bereit.');
```

```

Close(fin);
end;

Function U7:Double;           {Input-Spannung für den Input-Schwingkreis}
Var UAmpl : Double;          {Spannungs-Amplitude}
    Pulsdauer : LongInt;     {Pulsdauer in Zeitschritten von "dt"}
    Phasenshift : Double;    {Phasendifferenz zwischen oberem Umkehrpunkt und Spannungs-Impuls}
    Umerk : Double;          {Merk-Wert für die Ausgabe der Spannung}
begin
    Umerk:=0;                {Initialisierung des Merk-Werts für die Spannungs-Ausgabe}
    UAmpl:=6E-6;             {Volt, Spannungs-Amplitude}
    Pulsdauer:=20;           {Pulsdauer, Anzahl der Zeitschritte von "dt"}
    Phasenshift:=000;        {Phasendifferenz zwischen oberem Umkehrpunkt und Spannungs-Impuls in Zeitschritten
                                von "dt"}
    { If i<=Pulsdauer then Umerk:=UAmpl;           {falls gewünscht: Start-Impuls geben}
      If i>=Pulsdauer then           {Getriggerte Pulse im Betrieb geben}
      begin
        If (i>=iumk+Phasenshift)and(i<=iumk+Pulsdauer+Phasenshift) then {Hier wird das Trigger-Signal am
                                                oberen Umkehrpunkt festgemacht.}
          begin Umerk:=UAmpl; end; {Spannung anlegen}           {Alternativ könnte man ihn z.B. auch
                                                                am Nulldurchgang festmachen.}
        end;
        U7:=Umerk*0; {Wir wollen jetzt keine Energie-Zufuhr. Die Maschine soll ein Selbstläufer werden.}
      end;
end;

Function Reibung_nachregeln:Double;
Var merk:Double;
begin {Eine kleine Schalt-Hysterese muß ich einbauen;}
    merk:=cr; {Falls ich nicht außerhalb der Schalt-Hysterese liege.}
    If (phipo/2/pi*60)>1.000001*phipZiel then merk:=cr*1.000001; {Wenn's zu schnell läuft => abbremsen}
    If (phipo/2/pi*60)<0.999999*phipZiel then merk:=cr*0.999999; {Wenn's zu langsam läuft => weniger
                                                                    Reibung}
    If (merk<0.8*crAnfang) then merk:=0.8*crAnfang; {Regelung nicht zu arg schwingen lassen, vor Allen nicht
                                                                    Aufschwingen.}
    If (merk>1.2*crAnfang) then merk:=1.2*crAnfang; {Regelung nicht zu arg schwingen lassen, vor Allen nicht
                                                                    Aufschwingen.}
    Reibung_nachregeln:=merk;
end;

Begin {Hauptprogramm}
{ Eingabe-Daten-Anmerkung: Die Input-Spannung für die Input-Spule steht als letztes Unterprogramm vor dem
                                                                    Beginn des Hauptprogramms.}

{ Initialisierung - Vorgabe der Werte: }           {Wir arbeiten in SI-Einheiten}
Writeln('DFEM-Simulation des EMDR-Motors.');
```

```

{ Naturkonstanten:}
epo:=8.854187817E-12{As/Vm}; {Magnetische Feldkonstante}
muo:=4*pi*1E-7{Vs/Am};      {Elektrische Feldkonstante}
LiGe:=Sqrt(1/muo/epo){m/s};  Writeln('Lichtgeschwindigkeit c = ',LiGe, ' m/s');
```

```

{ Zum Lösen der Dgl. und zur Darstellung der Ergebnisse:}
AnzP:=100000000; {Zum Lösen der Dgl.: Anzahl der tatsächlich berechneten Zeit-Schritte}
dt:=43E-9;      {Sekunden} {Zum Lösen der Dgl.: Dauer der Zeitschritte zur iterativen Lsg. der Dgl.}
Abstd:=1;       {Nur für die Vorbereitung, nicht zum Lösen der Dgl.: Jeden wievielten Punkt soll ich
                                                                    plotten ins Excel}
PlotAnfang:=0000; {Zum Lösen der Dgl.: Erster-Plot-Punkt: Anfang des Daten-Exports nach Excel}
PlotEnde:=100000000; {Zum Lösen der Dgl.: Letzter-Plot-Punkt: Ende des Daten-Exports nach Excel}
PlotStep:=4000;    {Zum Lösen der Dgl.: Schrittweite des Daten-Exports nach Excel}
{ Die beiden Spulen, vgl. Zeichnung *2 von S.1 :} {Die Spulen werden nach Vorgabe der Geometrieparameter
                                                                    automatisch vernetzt}
Spsw:=0.01; {Angabe in Metern: Die Spulen-Aufgliederung ist in 0.01-Meter-Schritten}
xo:=0; yo:=6; zo:=5; {Angaben in Vielfachen von Spsw} {Geometrieparameter nach Zeichnung*2 von S.1}
Spulen_zuweisen;     {Spule für den Input der Steuer-Energie}
Ninput:=100;         {Zahl der Wicklungen der Input-Spule}
Nturbo:=9;           {Zahl der Wicklungen der Turbo-Spule}
nebeninput:=10;      {Windungen nebeneinander in der Input-Spule}

```

```

ueberinput:=10;      {Windungen uebereinander in der Input-Spule}
neberturbo:=3;      {Windungen nebeneinander in der Turbo-Spule}
ueberturbo:=3;      {Windungen uebereinander in der Turbo-Spule}
If nebeninput*ueberinput<>Ninput then
begin
  Writeln; Writeln('Windungszahl falsch: So kann man die Input-Spule nicht anordnen !');
  Wait; Wait; Halt;
end;
If neberturbo*ueberturbo<>Nturbo then
begin
  Writeln; Writeln('Windungszahl falsch: So kann man die Turbo-Spule nicht anordnen !');
  Wait; Wait; Halt;
end;
{ Spulen_anzeigen;      {Optionales Unterprogramm zur Kontrolle der Positionen.}
{ Dauermagnet-Emulation;} Writeln; {Magnetfeld muß nach Messung mit Hall-Sonde eingegeben werden.}
Bsw:=1E-2; {Magnetfeld-Speicherung nach *1 von S.2 in Zentimeter-Schritten}
{Ich emuliere hier das Magnetfeld eines 1T-Magneten durch ein Spulenpaar nach *1 von S.5}
MEyo:=0.05; {y-Koordinaten der Magnetfeld-Emulationsspulen nach *1 von S.5}
MEro:=0.01; {Radius der Magnetfeld-Emulationsspulen nach *1 von S.5}
MEI:=15899.87553474; {Strom des Magnetfeld-Emulationsspulenpaares nach *1 von S.5, Angabe in Ampere}
schonda:=true; Alte_Parameter_vergleichen;
If Not(schonda) then Magnetfeld_zuweisen_03; {Magnetfeld zuweisen und anzeigen}
Stromverteilung_zuweisen_03; {Stromverteilung in den Magnetfeld-Emulations-Spulen zuweisen.}
{ Allgemeine technische Größen:}
DD:=0.010; {Meter} {Durchmesser des Spulendrahtes zur Angabe der Drahtstärke}
rho:=1.35E-8; {Ohm*m} {Spez. elektr. Widerstand von Kupfer, je nach Temperatur, Kohlrausch,T193}
rhoMag:=7.8E3; {kg/m^3} {Dichte des Magnet-Materials, Eisen, Kohlrausch Bd.3}
CT:=101.7E-6; {150E-6;} {Farad} {Kapazität des Kondensators, der mit in der Turbo-Spule (!) in Reihe
                                                    geschaltet}
CI:=100E-6; {Farad} {Kapazität des Kondensators, der mit in der Input-Spule (!) in Reihe
                                                    geschaltet}
{ Sonstige (zur Eingabe):}
Rlast:=0.030; {Ohm} {Ohm'scher Lastwiderstand im LC-Turbo-Schwingkreis}
UmAn:=30000; {U/min} {Anfangsbedingung mechanisch - Rotierender Magnet: Startdrehzahl}
Uc:=0;{Volt} Il:=0; {Ampere} {Anfangsbedingung elektrisch - Kondensatorspannung = 0, Kein Spulenstrom}
{ Mechanische Leistungs-Entnahme (geschwindigkeits-proportional, aber nicht nur Reibung:}
crAnfang:=45E-6; {Koeffizient einer geschwindigkeits-proportionalen Reibung zwecks mechanischer
                                                    Leistungs-Entnahme}
phpZiel:=30100; {Ziel-Drehzahl, an der die Reibungs-Nachregelung ausgerichtet wird.}
{ Abgeleitete Parameter. Die Größen werden aus den obigen Parametern berechnet, es ist keine Eingabe
                                                    möglich:}
DLI:=4*(yo+zo)*Spw*Ninput; {Meter} {Länge des Spulendrahtes, Input-Spule}
DLT:=4*(yo+zo)*Spw*Nturbo; {Meter} {Länge des Spulendrahtes, Turbo-Spule}
RI:=rho*(DLI)/(pi/4*DD*DD); {Ohm} {Ohm'scher Widerstand des Spulendrahtes, Input-Spule}
RT:=rho*(DLT)/(pi/4*DD*DD); {Ohm} {Ohm'scher Widerstand des Spulendrahtes, Turbo-Spule}
BreiteI:=nebeninput*DD; HoeheI:=ueberinput*DD; {Breite und Höhe des Input-Spulenlörpers}
BreiteT:=neberturbo*DD; HoeheT:=ueberturbo*DD; {Breite und Höhe des Turbo-Spulenlörpers}
fkI:=Sqrt(HoeheI*HoeheI+4/pi*2*yo*2*zo)/HoeheI; {Korrekturfaktor zur Induktivität der kurzen Input-Spule}
fkT:=Sqrt(HoeheT*HoeheT+4/pi*2*yo*2*zo)/HoeheT; {Korrekturfaktor zur Induktivität der kurzen Turbo-Spule}
Writeln('Induktivitaets-Korrektur: fkI = ',fkI:12:5,', fkT = ',fkT:12:5);
LI:=muo*(2*yo+BreiteI)*(2*zo+BreiteI)*Ninput*Ninput/(HoeheI*fkI); {Geometrische Mittelung =>
                                                    Induktivität der Input-Spule}
LT:=muo*(2*yo+BreiteT)*(2*zo+BreiteT)*Nturbo*Nturbo/(HoeheT*fkT); {Geometrische Mittelung =>
                                                    Induktivität der Turbo-Spule}
omT:=1/Sqrt(LT*CT); {Kreis-Eigenfrequenz des Turbo-Spulen-Schwingkreises aus LT & CT}
TT:=2*pi/omT; {Schwingungsdauer des Turbo-Spulen-Schwingkreises aus LT & CT.}
Mmag:=rhoMag*(pi*MEro*MEro)*(2*MEyo);{Masse des Magneten} {Rotation des Magneten um Querachse !!}
J:=Mmag/4*(MEro*MEro+4*MEyo*MEyo/3); {Trägheitsmoment des Magneten bei Rotation, siehe *2 von S.13 und
                                                    Dubbel S.B-32}
{ Sonstige, auch abgeleitete (aus den obigen Parametern berechnete) Größen:}
omAn:=UmAn/60*2*pi; {Rotierender Magnet: Winkelgeschwindigkeit (rad/sec.), Startdrehzahl}
UmSec:=UmAn/60; {Rotierender Magnet: Umdrehungen pro Sekunde, Startdrehzahl}
{ Anzeige der Werte:}
Writeln('*****');
Writeln('Anzeige einiger auszurechnender Parameter:');
Writeln('Laenge des Spulendrahtes, Input-Spule: ',DLI,' m');
Writeln('Laenge des Spulendrahtes, Turbo-Spule: ',DLT,' m');
Writeln('Ohm'scher Widerstand des Input-Spulendrahts: RI = ',RI:8:2,' Ohm');
Writeln('Ohm'scher Widerstand des Turbo-Spulendrahts: RT = ',RT:8:2,' Ohm');
Writeln('Induktivitaet der Input-Spule, ca.: LI = ',LI,' Henry');
Writeln('Induktivitaet der Turbo-Spule, ca.: LT = ',LT,' Henry');
Writeln('Eigen-Kreisfreq des Turbo LT-CT-Schwinkreises: omT = ',omT:8:4,' Hz (omega)');
Writeln('=> Schwingungsdauer TT = 2*pi/omT = ',TT:15,'sec. ');
Writeln('Magnet: Start-Winkelgeschw.: omAn = ',omAn,' rad/sec');
Writeln('Magnet: Startdrehzahl, Umdr./sec.: UmSec = ',UmSec:15:10,' Hz');

```

```

Writeln('Masse des Magnet = ',Mmag:10:6,' kg');
Writeln('Traegheitsmoment Magnet bei QUER-Rotation',J,' kg*m^2');
Writeln('Gesamtdauer der Betrachtung: ',AnzP*dt,' sec.');
```

Writeln('Excel-Export: ',PlotAnfang\*dt:14,'...',PlotEnde\*dt:14,' sec., Step ',PlotStep\*dt:14,' sec.');

Writeln('Das sind ',(PlotEnde-PlotAnfang)/PlotStep:8:0,' Datenpunkte (also Zeilen).');

If ((PlotEnde-PlotAnfang)/PlotStep)>AnzPmax then

```

begin
  Writeln; Writeln('FEHLER: Zu viele Datenpunkte.');
```

Writeln('So viele Datenpunkte koennen in Excel nicht dargestellt werden.');

```

  Writeln('=> Berechnung wird hier GESTOPPT.');
```

Wait; Wait; Halt;

```

end;
{ Wait; }
{ Hilfsarbeiten: Für die Vorbereitungen brauche ich AnzP=360, danach brauche ich wieder den eingegebenen Wert.}

AnzPmerk:=AnzP; {Merken des Wertes für später}
AnzP:=360;      {Eine Umdrehung in Winkel-Grad-Schritten}

{ Ein Test der Daten-Transport-Routine ins Excel:}
For i:= 1 to AnzP do
begin
  Q[i]:=i*dt; Qp[i]:=2*i*dt; Qpp[i]:=3*i*dt;      phi[i]:=4*i*dt; phip[i]:=5*i*dt; phipp[i]:=6*i*dt;
  KG[i]:=7*i; KH[i]:=8*i; KI[i]:=9*i; KJ[i]:=10*i; KK[i]:=11*i; KL[i]:=12*i; KM[i]:=13*i; KN[i]:=14*i;
end;
{ExcelAusgabe('test.dat',14);} {Optionales Upgm. zur Datenausgabe nach Excel.}
{Alle Felder zurücksetzen, um möglicher Verwirrung für später vorzubeugen}
For i:= 1 to AnzP do
begin
  Q[i]:=0; Qp[i]:=0; Qpp[i]:=0;      phi[i]:=0; phip[i]:=0; phipp[i]:=0;
  KG[i]:=0; KH[i]:=0; KI[i]:=0; KJ[i]:=0; KK[i]:=0; KL[i]:=0; KM[i]:=0; KN[i]:=0;
end;

{ Hier beginnt das Rechenprogramm.}
{Teil 1: Eine Test-Berechnung der Drehmoment-Wirkung der beiden Spulen auf den Magneten:}
Writeln; {Wir beginnen mit der Bestimmung des Magnetfeld des beiden Spulen an einem beliebigen Aufpunkt}

Writeln('Hier steht vorerst zu Testzwecken die Feldberechnung der Input- und Turbo-Spule');
Magfeld_Input_Berechnen(-0.00,0.01,0.01,1.0); {drei kartesische Komponenten für Aufpunkt und Strom = 1.0 Ampere}

Writeln('B_Input_x,y,z:',BIx:19,' ',BIy:19,' ',BIz:19,' T');
Magfeld_Turbo_Berechnen(+0.00,0.01,0.01,1.0); {drei kartesische Komponenten für Aufpunkt und Strom = 1.0 Ampere}

Writeln('B_Turbo_x,y,z:',BTx:19,' ',BTy:19,' ',BTz:19,' T');
merk:=Sqrt((2*yo*Spsw*2*zo*Spsw)+Sqr(xo*Spsw)); merk:=merk*merk*merk;
Writeln('Vgl->Input: Runde Leiterschleife, Feld im Ursprung:
      ',muo*Ninput*1.0*2*yo*Spsw*2*zo*Spsw/2/merk,' T');
```

Writeln('Vgl->Turbo: Runde Leiterschleife, Feld im Ursprung:
 ',muo\*Nturbo\*1.0\*2\*yo\*Spsw\*2\*zo\*Spsw/2/merk,' T');

{Die Berechnung des Magnetfeldes der beiden Spulen (Input & Turbo) ist jetzt getestet und funktioniert.}

```

If Not(schonda) then
begin
  {Das ist nur eine Kontrolle.}
  For i:=0 to 360 do
  begin
    KN[i]:=Drehmoment(i/180*pi);
    Writeln(i:4,'Grad => Drehmoment-Komponente: Mz = ',KN[i]); {Das Argument ist der Winkel der Magnetstellung "alpha"}
```

end;

```

  ExcelAusgabe('drehmom.dat',14); {Optionales Upgm. zur Datenausgabe nach Excel.}
  Writeln('Damit ist die Drehmoments-Berechnung des Magneten geschafft.');
```

end;

{Teil 2: Ausprobieren der Flußberechnung durch die beiden Spulen unter Magnet-Drehung (führt später zur induz. Spannung):}

```

If Not(schonda) then
begin
  Writeln('Es folgt die Berechnung des magnetischen Flusses fuer Geometrie "03"');
  Magnet_drehen(00); {Drehwinkel in Grad angeben, 0...360}
  Gesamtfluss_durch_Input_Spule; Writeln('Gesamtfluss durch Input-Spule: ',PsiGES,' T*m^2');
  Magnet_drehen(01); {Drehwinkel in Grad angeben, 0...360}
  Gesamtfluss_durch_Input_Spule; Writeln('Gesamtfluss durch Input-Spule: ',PsiGES,' T*m^2');
  Writeln('-----');
  Magnet_drehen(00); {Drehwinkel in Grad angeben, 0...360}
  Gesamtfluss_durch_Turbo_Spule; Writeln('Gesamtfluss durch Turbo-Spule: ',PsiGES,' T*m^2');
  Magnet_drehen(01); {Drehwinkel in Grad angeben, 0...360}
  Gesamtfluss_durch_Turbo_Spule; Writeln('Gesamtfluss durch Turbo-Spule: ',PsiGES,' T*m^2');
  Writeln('-----');
```

end;

{Ergebnis bis hier: Die Differenz zwischen beiden erlaubt die Berechnung der induzierten Spannung}

```

{ Test: Einmal den Magneten drehen und den magnetischen Fluss / die induzierte Spannung messen:}
{Zum Test stehen 360 Zeiteinheiten = 360*dt = 36 Millisekunden für eine Umdrehung, entsprechend 1666.666
                                                                                               U/min}

If Not(schonda) then
begin
  Writeln('Zuerst die Input-Spule:');
  For i:= 0 to 360 do {Zuerst probiere ich's mit der Input-Spule}
  begin
    phi[i]:=i; {Angabe in Grad}
    Magnet_drehen(phi[i]); Gesamtfluss_durch_Input_Spule; {setzt auf "PsiGES" den Ergebnis-Wert ab.}
    PSIinput[i]:=PsiGES; {Dies ist der magnetische Fluß durch die Input-Spule}
    Writeln('phi = ',phi[i]:5:1,' grad => magn. ges. Fluss = ',PSIinput[i],' T*m^2');
    If i=0 then UindInput[i]:=0;
    If i>0 then UindInput[i]:=-Ninput*(PSIinput[i]-PSIinput[i-1])/dt;
    KG[i]:=0; KH[i]:=PSIinput[i]; KI[i]:=UindInput[i]; {Zur Excel-Ausgabe weiterleiten}
  end; Writeln('-----');
  Writeln('Danach die Turbo-Spule:');
  For i:= 0 to 360 do {Danach probiere ich's auch noch mit der Turbo-Spule}
  begin
    phi[i]:=i; {Angabe in Grad}
    Magnet_drehen(phi[i]); Gesamtfluss_durch_Turbo_Spule; {setzt auf "PsiGES" den Ergebnis-Wert ab.}
    PSIturbo[i]:=PsiGES; {Dies ist der magnetische Fluß durch die Turbo-Spule}
    Writeln('phi = ',phi[i]:5:1,' grad => magn. ges. Fluss = ',PSIturbo[i],' T*m^2');
    If i=0 then Uindturbo[i]:=0;
    If i>0 then Uindturbo[i]:=-Nturbo*(PSIturbo[i]-PSIturbo[i-1])/dt;
    KJ[i]:=0; KK[i]:=PSIturbo[i]; KL[i]:=Uindturbo[i]; {Zur Excel-Ausgabe weiterleiten}
    KM[i]:=0; KN[i]:=KN[i]; {Am Ende noch zwei Leerspalten}
  end;
  {Jetzt muß man das Signal noch gegen Rauschen glätten:}
  FourierDatenspeicherung(PSIturbo); FourierEntwicklung;
  B1T:=B1; B2T:=B2; B3T:=B3; B4T:=B4; B5T:=B5;
  (**)Writeln('Aktuelle Kontrolle der Fourier-Koeffizienten für den Turbo-Fluß:');
  (**)writeln(B1T:13,' ',B2T:13,' ',B3T:13,' ',B4T:13,' ',B5T:13); Wait;
  FourierDatenspeicherung(PSIinput); FourierEntwicklung;
  B1I:=B1; B2I:=B2; B3I:=B3; B4I:=B4; B5I:=B5;
  {Kontroll-Output der geglätteten Fluß-Werte in den ersten beiden Spalten des Excel-Datensatzes:}
  For i:=0 to 360 do
  begin
    {FlussI und FlussT gibt den geglätteten magnetischen Fluß durch die Spulen an.}
    KJ[i]:=FlussI(i/360*2*pi); {Der Lagewinkel des Magneten wird in Radianten angegeben.}
    KM[i]:=FlussT(i/360*2*pi); {Der Lagewinkel des Magneten wird in Radianten angegeben.}
  end;
end;

{Die Drehmoments-Berechnung absorbiert noch zu viel CPU-Zeit, um mit sehr feiner Zeit-Schrittweite
                                                                                               rechnen zu können.}
{Daher entwickle ich jetzt auch eine Fourier-Reihe zur Beschleunigung der Drehmoments-Berechnung:}
If Not(schonda) then
begin
  qpoT:=1; qpoI:=0; {Schnell-Berechnungs-Kalibrierung nur für Turbo-Spule, 1A, aber ohne Input-Spule}
  Writeln('Drehmoment in einen Sinus-Term umrechnen, zur spaeteren CPU-Zeit Ersparnis:');
  For i:=0 to 360 do
  begin
    {Das gesamt-Drehmoment, das der Magnet im Feld der beiden Spulen (Input&Turbo) aufnimmt.}
    KN[i]:=Drehmoment(i*2*pi/360); {Der Lagewinkel des Magneten wird in Radianten angegeben.}
    Write('.'); {Writeln(KN[i]);}
  end;
  FourierDatenspeicherung(KN); SinusEntwicklung_fuer_Drehmoment;
  Writeln('Drehmom-Ampl: ',B1dreh,' und Phase: ',phase);
  {Kontrolle, ob die Schnell-Drehmoments-Berechnung richtige Ergebnisse liefert:}
  For i:=0 to 360 do
  begin
    KG[i]:=Schnell_Drehmoment(i*2*pi/360); {Der Lagewinkel des Magneten wird in Radianten angegeben.}
  end;
end;
{Daten abspeichern, falls eine Parameter-Konfiguration vorliegt:}
{If Not(schonda) then} Daten_Speichern;
{Damit ist die Vorbereitung beendet.}

{Ich kontrolliere jetzt, ob alle Parameter und Daten mit und ohne "schonda" angekommen sind:}
For i:=0 to 360 do
begin
  {FlussI und FlussT gibt den geglätteten magnetischen Fluß durch die Spulen an.}
  KJ[i]:=FlussI(i*2*pi/360); {Magnetischer Fluss durch Input-Spule, Winkel des Magneten in Radianten}
  KM[i]:=FlussT(i*2*pi/360); {Magnetischer Fluss durch Turbo-Spule, Winkel des Magneten in Radianten}
end;
For i:=0 to 360 do
begin
  KG[i]:=Schnell_Drehmoment(i*2*pi/360); {Drehmoment auf den Magneten, Winkel des Magneten in Radianten}
end;
ExcelAusgabe('kontroll.dat',14); {Optionales Upgm. zur Datenausgabe nach Excel.}

```

```

{Hilfsarbeiten: Zum Lösen der Dgl. brauche ich wieder vorgegebene Anzahl von Iterationsschritten:}
AnzP:=AnzPmerk;
{Damit stehen jetzt alle Daten zur DFEM-Berechnung bereit.}
Writeln('*****');

{Noch eine Daten-Initialisierung: Zurücksetzen aller Felder für die "ExcelLangAusgabe":}
For i:=0 to AnzPmax do
begin
  Zeit[i]:=0; Q[i]:=0; Qp[i]:=0; Qpp[i]:=0; QI[i]:=0; QpI[i]:=0; QppI[i]:=0;
  phi[i]:=0; phip[i]:=0; phipp[i]:=0; KJ[i]:=0; KK[i]:=0; KL[i]:=0; KM[i]:=0;
  KN[i]:=0; KO[i]:=0; KP[i]:=0; KQ[i]:=0; KR[i]:=0; KS[i]:=0; KT[i]:=0;
  KU[i]:=0; KV[i]:=0; KW[i]:=0; KX[i]:=0; KY[i]:=0;
end;

{Initialisierung für die Suche der Maximalwerte zur Strom-, Drehzahl- und Spannungsangabe bei der
Auslegung:}

QTmax:=0; QImax:=0; QpTmax:=0; QpImax:=0; QppTmax:=0; QppImax:=0; phipomax:=0;
Wentnommen:=0; {Initialisierung der entnommenen Energie am Lastwiderstand}
Ereib:=0; {Initialisierung der entnommenen mechanischen Energie über Reibung}

{Initialisierung der Referenz für das Input-Spannungs-Signal:}
steigtM:=false; steigtO:=false;
Ezuf:=0; {Initialisierung der zugeführten Energie über die Input-Spannung}
LPP:=0; {Initialisierung der Anzahl der Datenpunkt für den Excel-Plot}

{ Damit ist alles vorbereitet, und ich kann jetzt anfangen, in die Lösung der System-Differentialgleichung
zu gehen:}

{ Also kommt jetzt der Rechen-Kern:}
{ Zuerst: Anfangsbedingungen einsetzen:}
phio:=0; phipo:=omAn; {phippo:=0;} {Anfangsbedingungen der Magnetrotation (mechanisch)}
{Starten mit gegebener Anfangs-Winkelgeschwindigkeit ist vorgesehen.}
qoT:=CT*Uc; qpoT:=Il; {qppoT:=0;} {Anfangsbedingungen Turbo-Spule (elektrisch)}
{Ein geladener Kondensator im Schwingkreis der Turbo-Spule ist vorgesehen.}
qoI:=0; qpoI:=0; qppoI:=0; {Anfangsbedingungen Input-Spule (elektrisch), zunächst inaktiv}
{ Beim "nullten" Schritt ist der alten Schritt nicht der "minus erste", sondern auch der nullte Schritt:}
{phim:=phio;} {phipm:=phipo;} {phippm:=phippo;}
{qpmT:=qoT;} {qpmI:=qoI;} {qppmT:=qppoT;}
{qpmI:=qoI;} {qpmI:=qoI;} {qppmI:=qppoI;}

{ Jetzt das eigentliche Lösen der Differential-Gleichung:} {Zunächst noch ohne Input-Spule !!}
For i:=0 to AnzP do
begin
  {Initialisierung der Referenz für das Input-Spannungs-Signal:}
  If i=0 then iumk:=0;
  If i>=1 then {Input-Spannungs-Referenz hier an der Turbo-Spule festmachen.}
  begin
    steigtM:=steigtO; {alten Flanken-Steigungs-Zustand merken}
    If qoT>qmT then steigtO:=true;
    If qoT<qmT then steigtO:=false;
    If (steigtM)and(Not(steigtO)) then iumk:=i;
  end;
  {Aktueller Moment der Analyse, laufende Zeit in Sekunden, "Jetzt-Schritt":}
  Tjetzt:=i*dt;
  {Für den neuen Schritt, wird der alte Vorgänger-Schritt zum vorletzten Schritt heruntergezählt:}
  phim:=phio; phipm:=phipo; {phippm:=phippo;} {Drehbewegung}
  qmT:=qoT; qpmT:=qpoT; {qppmT:=qppoT;} {Turbo-Spule}
  qmI:=qoI; qpmI:=qpoI; qppmI:=qppoI; {Input-Spule}
  {Und jetzt rechne ich den neuen Schritt aus:}
  {Zuerst die Drehung des Magneten, Drehmoment kommt aus Spulenströmen. Vorhandene Drehmoments-
  Berechnung benutzen:}
  {KK}phippo:=Schnell_Drehmoment(phim)*qpoT/J; {Upgm "Schnell_Drehmoment" ist skaliert ITurbo=1A &
  IInput=0A, geht linear mit dem Turbo-Strom.}
  {!! Alle Zeilen mit doppelten Ausrufezeichen dienen der zusätzlichen Aufnahme der Input-Spule.}
  {!! phippo:=Drehmoment(phim)/J; {Vollwertige Drehmoment-Berechnung mit Turbo-Spule und Input-Spule,
  geht langsam.}
  {Für phippo muß ich eine der beiden vorangehenden Alternativen verwenden, je nach dem ob nur die Turbo-
  Spule aktiv ist, oder auch die Input-Spule.}
  {Falls die Input-Spule auch aktiv ist, dann soll ich "schonda" von Anfang an auf "false" setzen und immer
  die komplette Vorbereitung durchrechnen.}
  {Alle mit "GG" kommentierten Zeilen dienen einer geschwindigkeits-proportionalen mechanischen
  Leistungsentnahme:}
  {GG}If i=1 then cr:=crAnfang; {Geschwindigkeits-proportionaler Reibungs-Koeffizient}
  {GG}If i>1 then cr:=Reibung_nachregeln; {Dieser wird nachgeregelt, um eine konstante Drehzahl
  einzustellen für stabilen Betrieb der Maschine.}
  {GG}If phipo>0 then phippo:=phippo-cr*phipm/J; {Die neg. Beschleunigung wirkt immer der Geschwindigkeit
  entgegen.}
  {GG}If phipo=0 then phippo:=phippo;
  {GG}If phipo<0 then phippo:=phippo+cr*phipm/J; {Die neg. Beschleunigung wirkt immer der Geschwindigkeit

```

```

entgegen.}
{GG}{Jetzt ist die geschwindigkeits-proportionale Reibung berechnet.}
  If (i mod 100000)=0 then write('.');
  phipo:=phipm+phippo*dt;           {1. Integrationsschritt, ohne mechanische Reibung}
  phio:=phim+phipo*dt;             {2. Integrationsschritt}
{GG}Preib:=cr*phipm*phipo;         {Leistungsentnahme zur geschwindigkeits-proportionalen Reibung}
{GG}Ereib:=Ereib+Preib*dt;         {Gesamte über Reibung entnommene Leistung}
  {Dann die Turbo-Spule. Gedämpfte elektrische Schwingung, dazu induzierte Spannung aufgrund Magnet-
  Drehung:}
{FF}qppoT:=-1/(LT*CT)*qmT-(RT+Rlast)/LT*qpmT; {Dgl. der gedämpfte Schwingung.}
  UinduzT:=-Nturbo*(FlussT(phio)-FlussT(phim))/dt; {Wirkung durch die induzierte Spannung (aufgrund der
  Magnet-Drehung) hinzunehmen}
  qppoT:=qppoT-UinduzT/LT;         {Wirkung der induzierte Spannung auf die zweite Ableitung
  von q, also "qppoT")}
{??}qpoT:=qpmT+qppoT*dt; {-Rlast/(2*LT)*qpmT*dt;} {1. Integrationsschritt, nach *5 von S.6 im alten Skript
  bzw *1 von S.14 im neuen Skript}
  qoT:=qpmT+qpoT*dt;             {2. Integrationsschritt, nach *3 & *4 von S.6 im alten
  Skript bzw *1 von S.14 im neuen Skript}
  {Dann die Input-Spule:} UinduzI:=0;
  qoI:=qmI; qpoI:=qpmI; qppoI:=qppmI; {Die Input-Spule macht noch gar nichts, sie spielt jetzt
  noch nicht mit.}
{Und wenn die Input-Spule doch mitspielt, muß ich die nachfolgenden fünf Zeilen zur Input-Spule
aktivieren:}
{!! qppoI:=-1/(LI*CI)*qmI-RI/LI*qpmI+U7/LI; {Dgl. der gedämpfte Schwingung, dazu Störfunktion für
  Input-Spannung in den Input-Schwingkreis}
{!! UinduzI:=-Ninput*(FlussI(phio)-FlussI(phim))/dt; {Wirkung durch die induzierte Spannung (aufgrund der
  Magnet-Drehung) hinzunehmen}
{!! qppoI:=qppoI-UinduzI/LI; {Wirkung der induzierte Spannung auf die zweite Ableitung
  von q, also "qppoT")}
{!! qpoI:=qpmI+qppoI*dt; {1. Integrationsschritt, nach *5 von S.6 im alten Skript
  bzw *1 von S.14 im neuen Skript}
{!! qoI:=qmI+qpoI*dt; {2. Integrationsschritt, nach *3 & *4 von S.6 im alten
  Skript bzw *1 von S.14 im neuen Skript}
  Pzuf:=U7;{*qpoI} {Über die Input-Spannung zugeführte Leistung}
  Ezuf:=Ezuf+Pzuf*dt; {Zugeführte Energie über die Input-Spannung}
{Achtung: Die Drehmoments-Schnell-Berechnung "phippo" geht so noch nicht für Turbo-Input-Spule. Dazu muß
ich noch die Ströme in die einzelnen Upgme durchreichen.}

{Jetzt muß ich noch die Maximalwerte für Strom, Spannung und Drehzahl der Auslegung bestimmen:}
If Abs(qoT)>QTmax then QTmax:=Abs(qoT); {Maximum der Ladung im Turbo-Kondensator festhalten}
If Abs(qoI)>QImax then QImax:=Abs(qoI); {Maximum der Ladung im Input-Kondensator festhalten}
If Abs(qpoT)>QpTmax then QpTmax:=Abs(qpoT); {Maximum des Stroms in der Turbo-Spule festhalten}
If Abs(qpoI)>QpImax then QpImax:=Abs(qpoI); {Maximum des Stroms in der Input-Spule festhalten}
If Abs(qppoT)>QppTmax then QppTmax:=Abs(qppoT); {Maximum des Ipunkt in der Turbo-Spule festhalten}
If Abs(qppoI)>QppImax then QppImax:=Abs(qppoI); {Maximum des Ipunkt in der Input-Spule festhalten}
If Abs(phipo)>phipomax then phipomax:=Abs(phipo); {Maximum der Winkelgeschwindigkeit des Magneten
festhalten}
Wentnommen:=Wentnommen+Rlast*qpoT*qpoT*dt; {Summierung der am Lastwiderstand im Turbo-Schwingkreis
entnommenen Gesamtenergie}

{Ggf. muß ich jetzt einen Plot-Punkt für's Excel ausgeben:}
If (i>=PlotAnfang)and(i<=PlotEnde) then {diese Punkte ins Excel plotten}
begin
  If ((i-PlotAnfang)mod(PlotStep))=0 then
  begin
    znr:=Round((i-PlotAnfang)/PlotStep);
    Zeit[znr]:=Tjetzt; {Zeitpunkt für Excel abspeichern.}
    Q[znr]:=qoT; Qp[znr]:=qpoT; Qpp[znr]:=qppoT; {Turbo-Spule, im Array (und nur dort) ohne
    Index "T".}
    QI[znr]:=qoI; QpI[znr]:=qpoI; QppI[znr]:=qppoI; {Input-Spule}
    phi[znr]:=phio; phip[znr]:=phipo; phipp[znr]:=phippo; {Drehung des Magneten}
    KK[znr]:=FlussT(phio); KL[znr]:=FlussI(phio); {Magnetischer Fluß durch die Spulen}
    KM[znr]:=UinduzT; KN[znr]:=UinduzI; {In den Spulen induzierte Spannung}
    KO[znr]:=1/2*LT*qpoT*qpoT; {Energie in der Input-Spule}
    KP[znr]:=1/2*LI*qpoI*qpoI; {Energie in der Turbo-Spule}
    KQ[znr]:=1/2*qoT*qoT/CT; {Energie im Input-Kondensator}
    KR[znr]:=1/2*qoI*qoI/CI; {Energie im Turbo-Kondensator}
    KS[znr]:=1/2*J*phipo*phipo; {Energie der Magnet-Rotation}
    KT[znr]:=KO[znr]+KP[znr]+KQ[znr]+KR[znr]+KS[znr]; {Gesamt-Energie im System}
    KU[znr]:=Rlast*qpoT*qpoT; {Am Lastwiderstand entnommene Leistung, nur Turbo-seitig}
    KV[znr]:=U7; {Kontrolle der Input-Spannung im Input-Schwingkreis}
    KW[znr]:=Pzuf; {Zugeführte Leistung über die Input-Spannung}
    KX[znr]:=cr; {geregelter Reibungskoeffizient zur mechanischen Leistungsentnahme}
    KY[znr]:=Preib; {Entnommene mechanische Leistung, emuliert durch geschwindigkeits-proportionale
    Reibung}
    KZ[znr]:=0; {Noch eine Spalte in Reserve, für optionale Daten, die ins Excel sollen.}
    LPP:=znr; {Letzter Plot-Punkt; Wert wird für Datenausgabe benutzt -> ExcelLangAusgabe}
  end;
end;

```

```

end;
AnfEnergie:=KU[0];           {Anfangs-Gesamt-Energie im System}
EndEnergie:=KU[LPP];        {End-Gesamt-Energie im System}
end;
Writeln; Writeln('Anzahl Datenpunkte fuer Excel-Plot: LPP = ',LPP);
Writeln; Writeln('Anzeigen der Amplituden der Auslegung: (nicht Effektivwerte, sondern Spitze)');
Writeln('Input-Kondensator, Spannung, UmaxI =',QImax/CI,' Volt');   {Maximum der Ladung im Input-
                                                                    Kondensator}
Writeln('Turbo-Kondensator, Spannung, UmaxT =',QTmax/CT,' Volt');   {Maximum der Ladung im Turbo-
                                                                    Kondensator}
Writeln('Input-Schwingkreis, Strom, ImaxI =',QpImax,' Ampere');   {Maximum des Stroms in der Input-
                                                                    Spule}
Writeln('Turbo-Schwingkreis, Strom, ImaxT =',QpTmax,' Ampere');   {Maximum des Stroms in der Turbo-
                                                                    Spule}
Writeln('Input-Spule, Spannung, UmaxSI=',LI*QppImax,' Volt'); {Maximum des Ipunkt in der Input-
                                                                    Spule}
Writeln('Turbo-Spule, Spannung, UmaxST=',LT*QppTmax,' Volt'); {Maximum des Ipunkt in der Turbo-
                                                                    Spule}
Writeln('Maximale Magnet-Rotationsdrehzahl =',phipomax,' rad/sec');{Maximum der Winkelgeschwindigkeit
                                                                    des Magneten}
Writeln('Maximale Magnet-Rotationsdrehzahl=',phipomax/2/pi*60:15:6,' U/min. '); {Maximum der
                                                                    Winkelgeschwindigkeit des Magneten}
Writeln('Am Ende erreichte End-Drehzahl = ',phip[LPP]/2/pi*60:15:6,' U/min. ');
Writeln;
Writeln('Anfangs-Energie im System: ',AnfEnergie:18:11,' Joule');
Writeln('End-Energie im System: ',EndEnergie:18:11,' Joule');
Writeln('Energie-Zunahme im System: ',(EndEnergie-AnfEnergie):18:11,' Joule');
Writeln('Leistungs-Aenderung im System:',(EndEnergie-AnfEnergie)/(AnzP*dt):18:11,' Watt');
Writeln('Am Lastwiderstand entnommene Gesamtenergie = ',Wentnommen:18:11,' Joule');
Writeln('entsprechend einer mittleren entnommenen Leistg:',Wentnommen/(AnzP*dt):18:11,' Watt');
Writeln('Ueber Input-Spannung zugefuehrte Gesamt-Energie: ',Ezuf,' Joule');
Writeln('entsprechend einer mittleren zugefuehrten Leistg:',Ezuf/(AnzP*dt),' Watt');
Writeln('Gesamte mechanisch entnommene Energie = ',Ereib:18:11,' Joule');
Writeln('entsprechend einer mittleren Leistung = ',Ereib/(AnzP*dt):18:11,' Watt');
Writeln('bei einer Betrachtungs-Dauer von',(AnzP*dt):18:11,' sec. ');
ExcelLangAusgabe('test.dat',25);
Writeln; Writeln('Fertig gerechnet -> Adele. ');
Wait; Wait;
End.

```