



Simulation - eine praktische Einführung
Autor: Dr.-Ing. Alfred Kamusella



- ohne Altersbeschränkung -



Dieses Beispiel wurde ursprünglich aufbereitet, um Gymnasial-Schülern bei einem Besuch an der Universität durch selbstständiges, praktisches Üben das Erlebnis der numerischen Simulation zu vermitteln. Dazu wurde als Objekt ein Antrieb mit einem Gleichstrom-Motor gewählt. Im Rahmen einer studentischen Komplexübung zur Dimensionierung eines Nadelantriebes für einen Brailleschrift-Präger dient dieses Beispiel zur Einarbeitung in das Bedienkonzept von SimulationX.

1. SimulationX - ein Simulationsprogramm

- Oberflächlicher Einblick

2. Modell eines elektrischen Antriebs

- Motor mit Netzteil
- Einschaltstrom - Experiment
- Drehzahl - Experiment
- Regelkreis - Experiment

3. Was ist nun eigentlich Simulation?

- Eine erschöpfende Antwort

Einzusendende Ergebnisse:

- Teilnehmer der Lehrveranstaltung **Optimierung** schicken ihre Lösung per Mail an a.kamusella@tu-dresden.de
- Als Anhang dieser Mail ist mit (**xx**=Teilnehmer-Nummer 01...99) die konfigurierte Modelldatei **Antrieb_xx.isx** in einem Archiv-File (z.B. ZIP) zu senden.

Einsendeschluss:

- Die Nacht vor dem ersten Übungskomplex. Die Nacht endet morgens um 10:00 Uhr!



Abgerufen von „http://index.php?title=Software:_SimX_-_Einfuehrung_-_DC-Motor&oldid=23250“



SimulationX - ein Simulationsprogramm

Das von der **ESI Germany GmbH - Dresden** entwickelte Simulationssystem gehört zu den Modernsten:

- Wir wollen SimulationX in dieser praktischen Einführung nutzen, um zu demonstrieren, wie das mit dem Simulieren auf dem Computer so funktioniert.
- Das Programm steht als freie "Express Edition" z.B. für Lehrzwecke mit eingeschränktem Funktionsumfang zur Verfügung. Interessenten können diese Version aus dem ESI-Kundencenter nach *Registrierung* herunterladen und auf einem privaten Windows-Computer installieren.
- **Wichtig:** Teilnehmer der **Lehrveranstaltung "Optimierung"** erhalten die zu nutzende Programmversion direkt vom Betreuer!

Nun geht es los - wir starten SimulationX!



Die Programmoberfläche ab Version 4.0 hat in der Grundeinstellung folgende wesentlichen Bereiche:

Bibliotheken

- SimulationX 4.0
 - Allgemein
 - Signalglieder
 - Mechanik
 - Translatorische Mechanik (1D)
 - Rotatorische Mechanik (1D)
 - Ebene Mechanik (2D)
 - MKS-Mechanik
 - Hilfsmittel
 - Antriebstechnik (1D)
 - Antriebstechnik (MKS)
 - Elektrotechnik / Elektronik
 - Elektromechanik
 - Magnetik

Modell1

Kommerzielle Anwendung ist nicht erlaubt!

Strukturansicht des Modells

Parameter Aktuelle Werte des

Kommentar	Name	Aktueller Wert	Maßeinheit
Masse	m	1	kg
Anfangsweg	x0		m
Anfangsgeschwindigkeit	v0		m/s

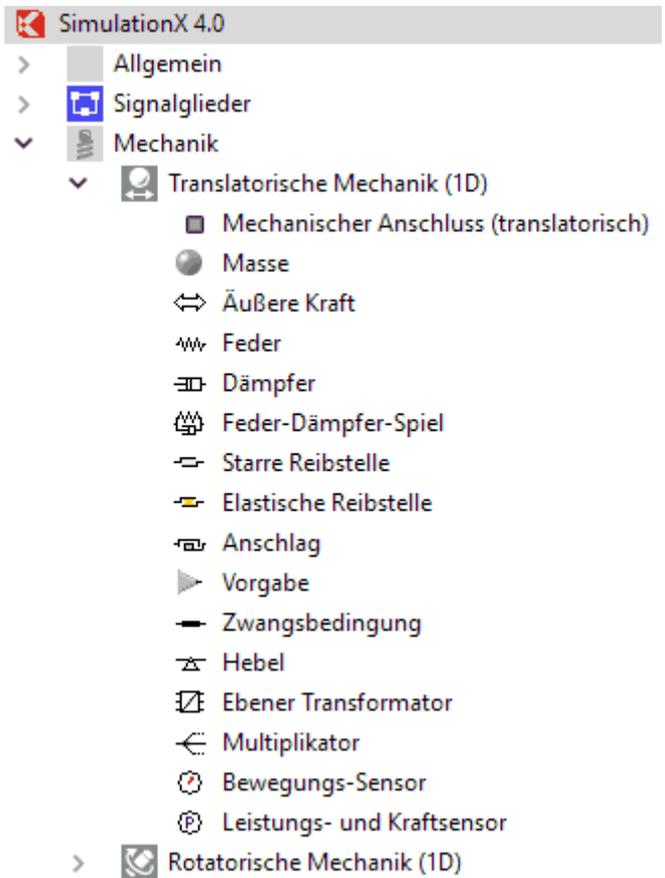
Ergebnisse gewählten Modell-Elements

Kommentar	Name	Aktueller Wert	Maßeinheit	Protokoll
Trägheitskraft	Fa	0	N	<input checked="" type="checkbox"/>
Weg	x	0	m	<input checked="" type="checkbox"/>
Geschwindigkeit	v	0	m/s	<input checked="" type="checkbox"/>
Beschleunigung	a	0	m/s ²	<input checked="" type="checkbox"/>
Änderung der kinetischen Energie	Pk	0	kW	<input checked="" type="checkbox"/>

Ausgabe

Ausgabebereich (Protokoll)

- In der **Leiste der Modellbibliothek** findet man ein reichhaltiges Angebot an Elemente-Typen aus allen möglichen Fachgebieten.
- In den **Bereich für die Strukturansicht des Modells** kann man diese Elemente einfach mit dem Cursor ziehen:
 - Wie in einem "echten" Versuchsstand kann man die Elemente miteinander verbinden.
 - Zum Verbinden nutzt man wieder den Cursor.
 - Das gelingt aber nur, wenn die Element-Anschlüsse zueinander passen (auch in der wirklichen Welt funktioniert eine elektrische Leitung meist nicht besonders gut als Wasserleitung!)
- Im **Modellexplorer** wird das aufgebaute Modell als Baumstruktur dargestellt:
 - Die Form dieser Darstellung entspricht der Ordneransicht im Windowsexplorer.
 - In dieser Baumstruktur werden die Elemente und die Verbindungen abgebildet.
- Die **Werte des aktuell ausgewählten Modell-Elements** erscheinen in einem separaten Bereich (Auswahl in der Strukturansicht oder in der Baumstruktur)
- Der **Ausgabebereich** dient dem Protokoll, so erscheinen dort z.B. Fehlermeldungen bei der Arbeit mit dem Modell (hoffentlich nicht!).



← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Einfuehrung_-_DC-Motor_-_Programm&oldid=26711“



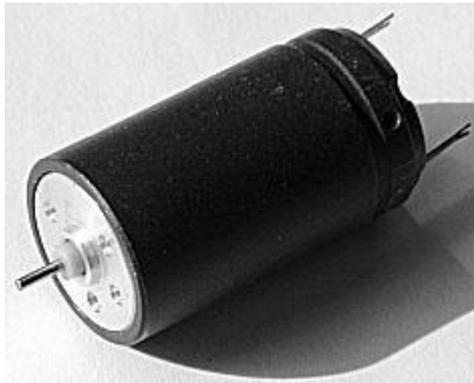
Modell eines elektrischen Antriebs

Es gibt nur wenige elektrische Geräte, die keinen elektrischen Antrieb enthalten (z.B. einfache Wasserkocher oder Leuchten). Ansonsten findet man fast überall mindestens einen kleinen Motor versteckt.

- Oft reicht es nicht aus, dass sich der Motor dreht:
 - der Motor soll z.B. zu jedem Zeitpunkt die richtige Drehzahl besitzen;
 - zu bestimmten Zeitpunkten soll sich der Motor z.B. um einen bestimmten Winkel gedreht haben;
 - Motoren dürfen z.B. nicht zu schnell beschleunigen oder bremsen, damit nichts kaputt geht;
 - es muss manchmal schneller gehen als man gucken kann, z.B. in Bruchteilen von Sekunden und
 - der Antrieb darf meist fast nichts kosten!
- Das Konstruieren eines modernen Antriebssystems ist meist eine anspruchsvolle Aufgabe:
 - In der Werkstatt basteln, bis es funktioniert - das funktioniert oft nicht oder dauert viel zu lange!
 - Außerdem kostet jede Stunde Arbeit in der Werkstatt ungefähr 100 € - das kann keiner bezahlen.
 - Deshalb entwickelt man immer häufiger zuerst numerische Modelle, mit denen man auf dem Computer eine Lösung sucht.
 - Erst wenn man weiß, wie man den Antrieb aufbauen muss, lässt man sich ein Versuchsmuster in der Werkstatt bauen.
- An einem vereinfachten Lehrbeispiel soll das Arbeiten mit solchen Computer-Modellen gezeigt werden.

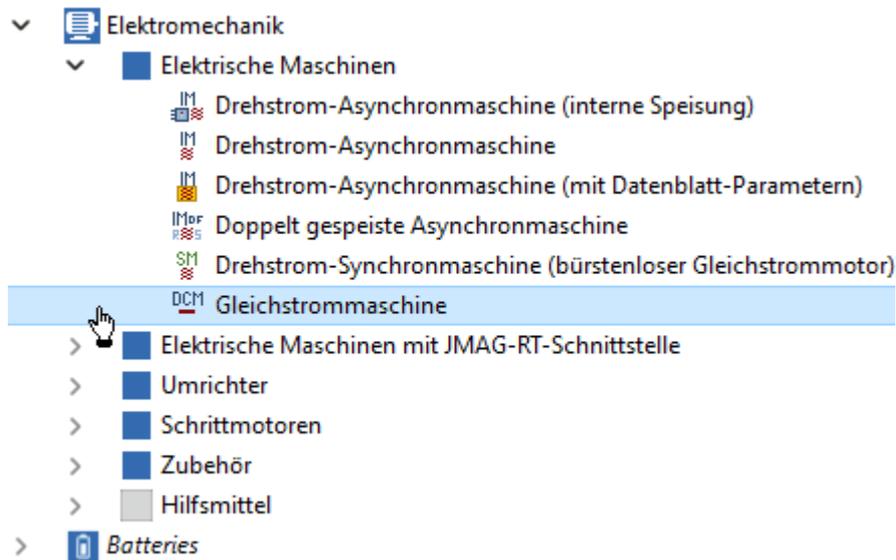
Damit es nun richtig losgehen kann, müssen wir zuvor unseren Modell-Ansichtsbereich entrümpeln, falls wir zum Ausprobieren dort ein paar Modell-Elemente verknüpft hatten:

- Wir schließen das Modell (*Datei > Modell Schließen*) ohne Speichern und beginnen danach mit einem neuen Modell (*Datei > Neu*).
- Den Aufbau unseres Modells wollen wir mit dem Motor beginnen:

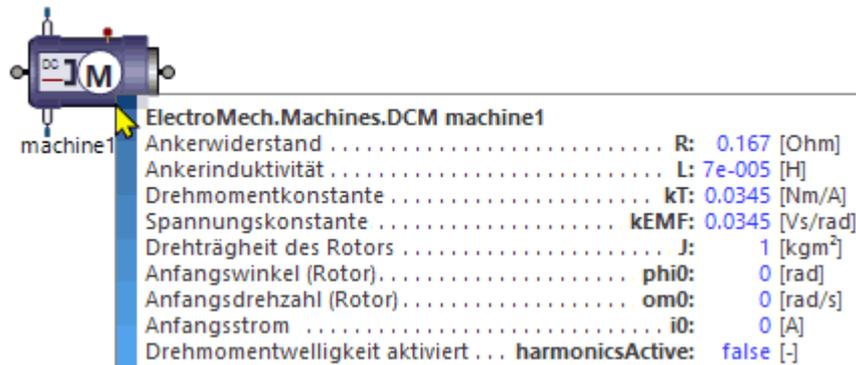


Gleichstrom-Motor:

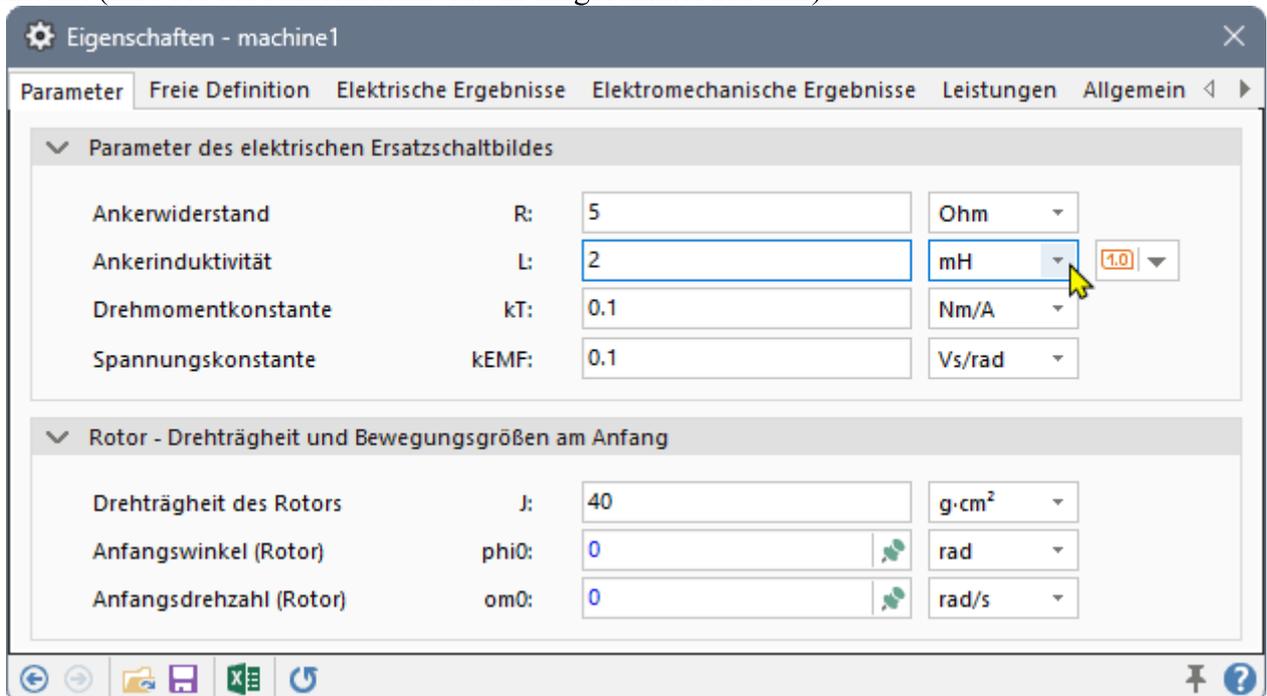
- Dazu zieht man einfach das zugehörige Modell-Element aus der Bibliotheksleiste auf die Arbeitsfläche einer neuen Modell-Datei (Strukturansicht des Modells):



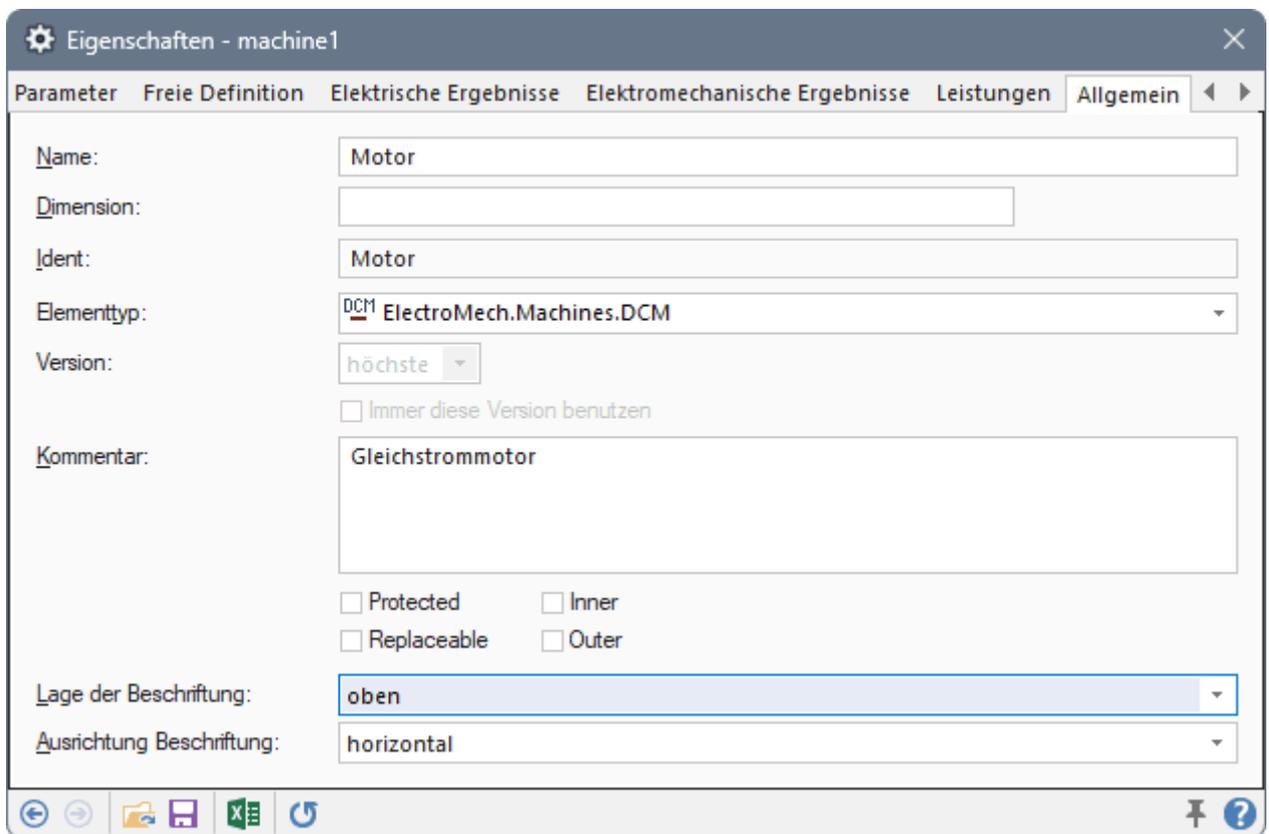
- Bewegt man danach dort den Cursor über dieses Modell-Element, erscheint ein kleines Fenster mit den aktuellen elektrischen und mechanischen Parameterwerten:



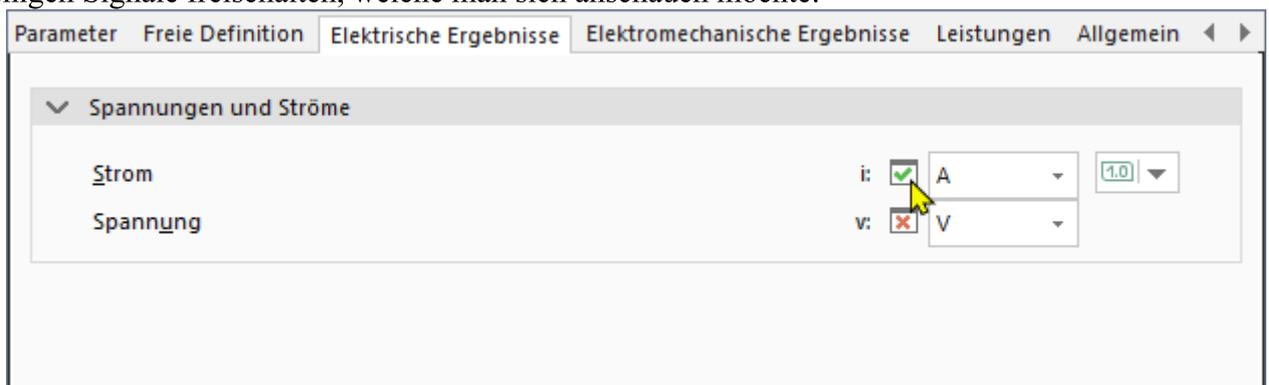
- Unser Motor, den wir f ur den Antrieb benutzen m ochten, besitzt nat urlich ganz andere Parameter. Nach einem Doppelklick auf das Motor-Symbol  ffnet sich ein Eigenschaftsfenster.
- Hier tragen wir die "richtigen" Parameter ein. Dabei muss man beachten, dass man die richtige Ma einheit gew ahlt hat (z.B. bei der Ankerinduktivit at **H** umgeschaltet auf **mH**):



- Unser Motor erhielt automatisch den Namen **machine1**. Direkt in der grafischen Modellansicht oder in der Register-Karte *Allgemein* des Eigenschaftsdialogs kann daf ur eine g unstigere Bezeichnung eintragen und auch Lage/Ausrichtung der Beschriftung  andern:

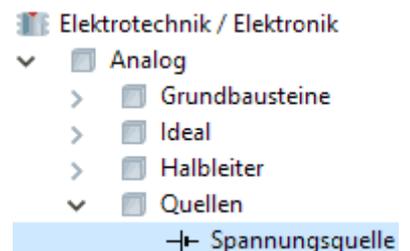
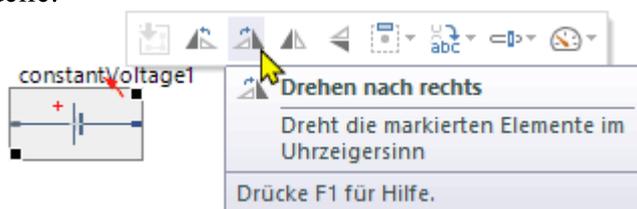


- Nach dem Einschalten des Motors wollen wir uns den Signalverlauf des elektrischen Stromes anschauen, der dann durch die Ankerspule des Motors fließt. In der Register-Karte *Elektrische Ergebnisse* kann man diejenigen Signale freischalten, welche man sich anschauen möchte:

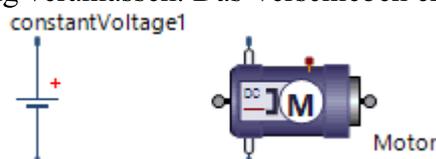


Stromversorgung (Netzteil):

- Zum Einschalten des Motors benötigt man noch ein Netzteil. Der Motor soll mit 24 V betrieben werden. Wir benutzen aus der Bibliothek eine Konstant-Spannungsquelle:



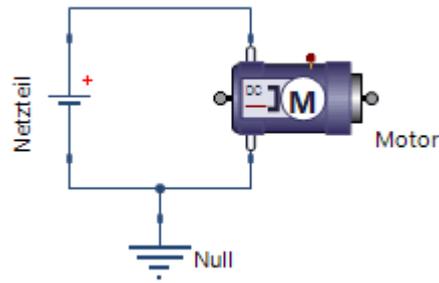
- Die waagerechte Lage des Symbols ist ungünstig für die Verbindung mit dem Motor. Die Elemente kann man jedoch noch beliebig drehen und verschieben. Nach einem Mausklick auf das Element kann man die gewünschte Drehung oder Spiegelung veranlassen. Das Verschieben erfolgt dann einfach mit dem Cursor:



- Der Name der Spannungsquelle soll **Netzteil** heißen. Die Lage der Beschriftung kann man in der Registerkarte *Allgemein* der Element-Eigenschaften ändern (im Beispiel auf "links" und "vertikal"). Auch

die Position der Motor-Beschriftung wurde nachträglich noch geändert.

- Das Netzteil soll eine Spannung von **24 V** liefern.
- Die Anschlüsse des Netzteils müssen mit den Anschlüssen des Motors verbunden werden (einfach mit der linken Maustaste!).
- Um die Schaltung mit einem Null-Potential zu versehen, benötigen wir noch eine elektrische **Masse** (in der Bibliothek unter *Elektrotechnik/Elektronik-Analog-Grundbausteine*):



- Auf der folgenden Seite wird beschrieben, wie wir in einem ersten Experiment den Anstieg des Motorstroms nach dem Anlegen der Betriebsspannung bei "festgehaltenem" Rotor beobachten können.

Datei > Speichern nicht vergessen (mit Teilnehmer-Nr. xx=01..99): **Antrieb_xx.isx**

← →

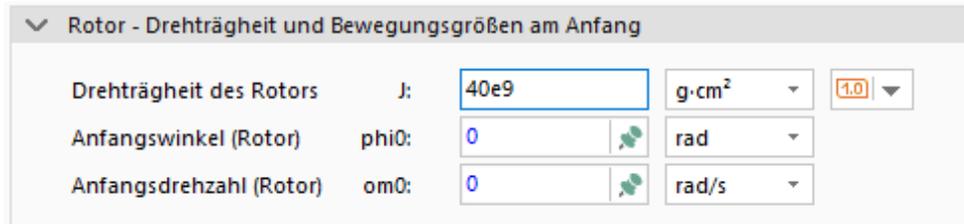
Abgerufen von „http://index.php?title=Software:_SimX_-_Einfuehrung_-_DC-Motor_-_Antriebsmodell&oldid=26751“



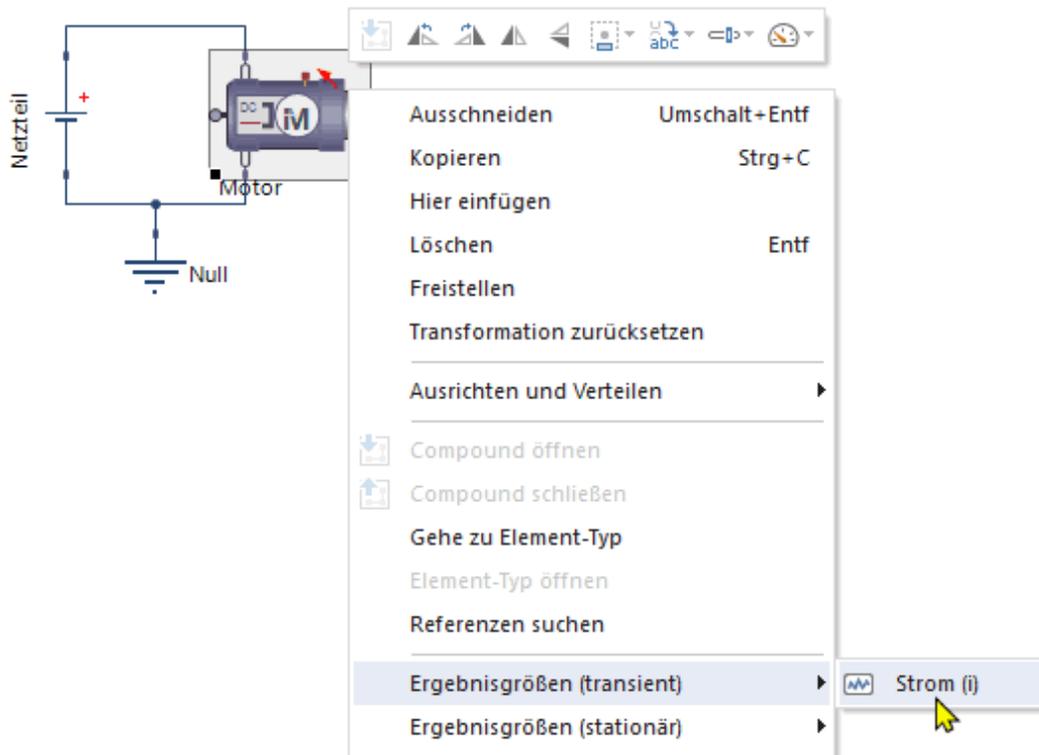
Einschaltstrom-Experiment

In einem ersten Experiment untersuchen wir nur Reaktion der elektrischen Seite des elektro-mechanischen Wandlers auf das Anlegen einer konstanten Spannung:

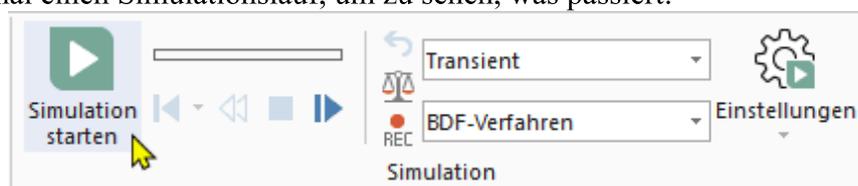
- Dazu ist es erforderlich, eine Bewegung des Rotors zu verhindern, weil diese eine elektrische Spannung in der Ankerspule induziert und damit den elektrischen Strom beeinflusst.
- Man könnte mittels zusätzlicher Modell-Elemente ein mechanisches Festhalten des Rotors realisieren.
- Viel einfacher ist eine temporäre Erhöhung der Rotor-Drehträgheit auf den Wert "**unendlich**", wofür praktisch schon ein Faktor von **1e9** ausreichend ist:



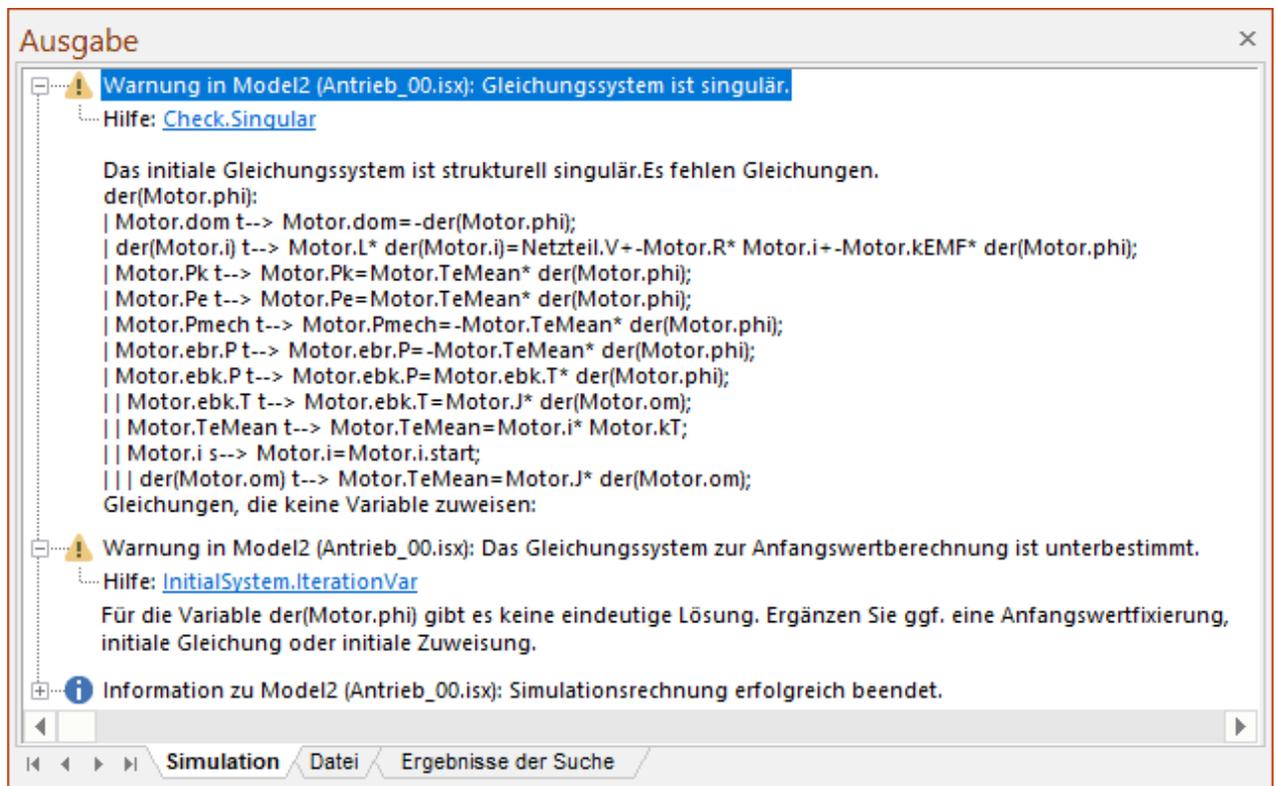
Damit man der Verlauf des Motorstroms nach dem Einschalten betrachten kann, muss man zuvor diese Ergebnisgröße für die Darstellung auswählen (mittels rechten Mausklick auf Motor):



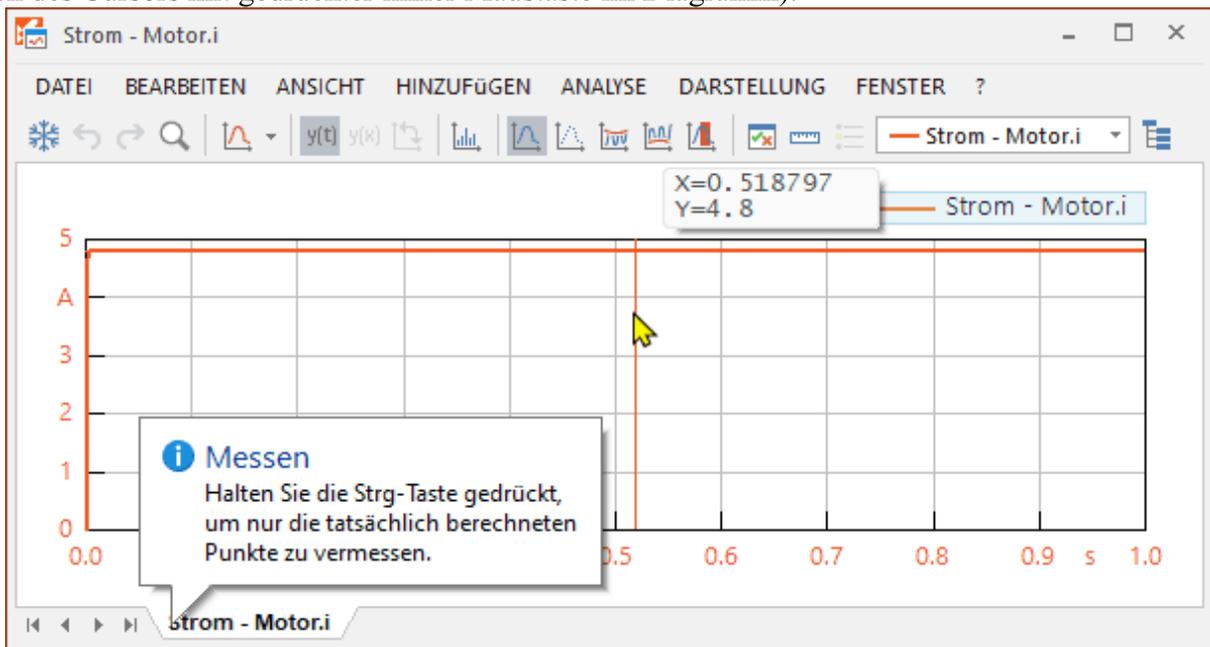
- Es wird ein leeres Ergebnisfenster für den Motor-Strom geöffnet (noch keine Ergebnisse berechnet).
- Wir starten einfach mal einen Simulationslauf, um zu sehen, was passiert:



- Und schon sind wir in der realen Welt der Warnungen und Fehlermeldungen angekommen!
- Gelb hinterlegt springt sofort eine Warnung ins Auge, welche wir näher analysieren müssen, ob sie für unsere Simulationsergebnisse von Bedeutung ist.
- Nach Doppelklick auf die Titelzeile des Ausgabe-Fensters (unten im Programmfenster) wird dieses separiert und wir können es auf eine anschauliche Größe ziehen und die einzelnen Warnungen öffnen:

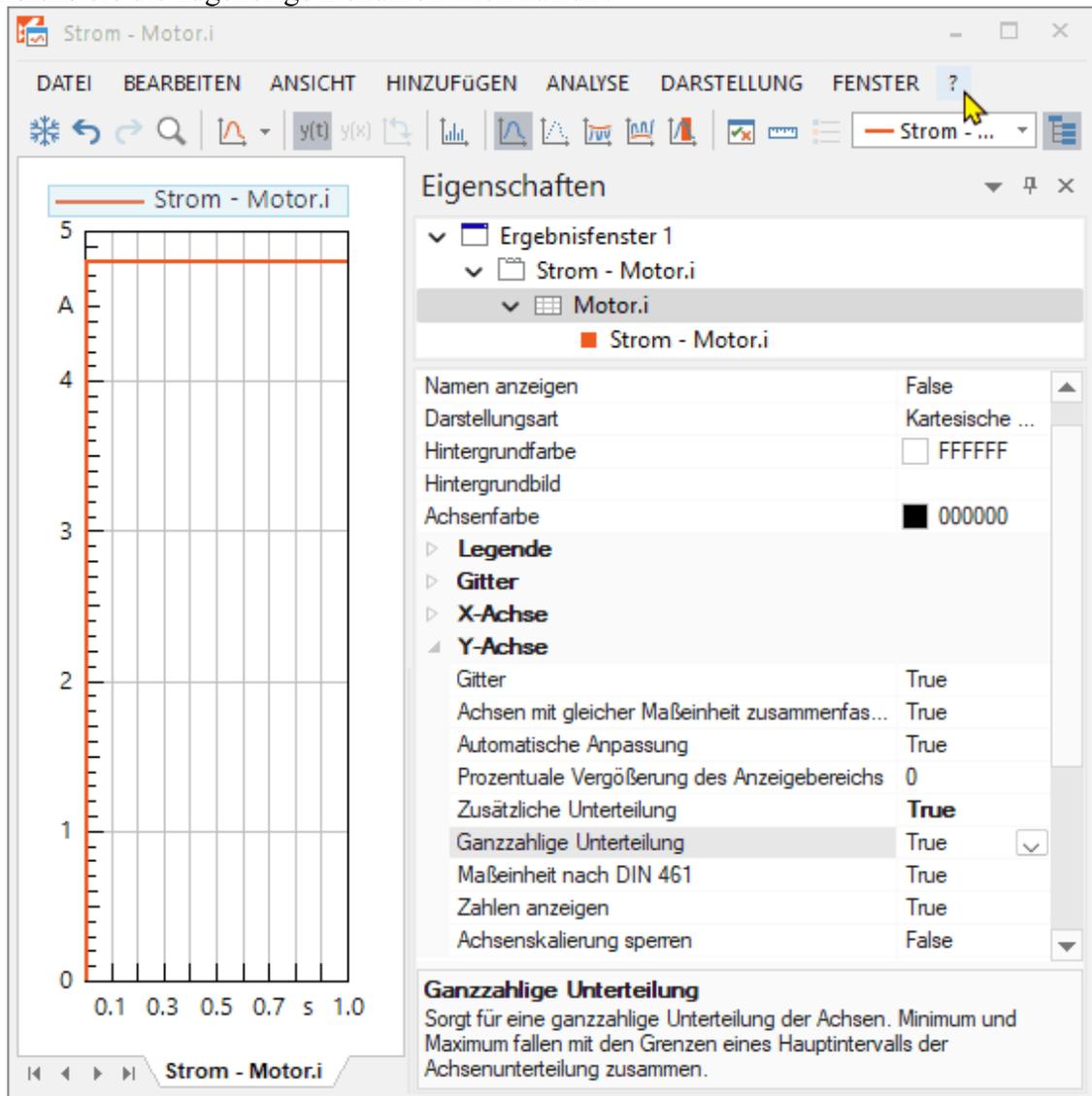


- Für die zeitliche Ableitung des Motor-Drehwinkels "der(Motor.phi)" fehlt angeblich ein eindeutiger Anfangswert. Diese Ableitung ϕ/dt entspricht der Rotor-Drehgeschwindigkeit, für welche wir jedoch einen Anfangswert fest vorgegeben haben.
- Die Programmierer des DC-Motors haben nicht berücksichtigt, dass jemand auf die Idee kommen könnte, diesen Motor ohne Last zu betreiben, denn der Anfangswert für diese Drehgeschwindigkeit wird von der (hier nicht vorhandenen) mechanischen Verbindung zur Last übernommen. Ein erneuter Simulationslauf würde auch mit veränderten Parametern solche Warnungen bei fehlenden Anfangswerten von Zustandsgrößen nicht mehr anzeigen.
- Es ist natürlich für den Laien schwierig, aus den aufgelisteten Modellanweisungen immer die richtigen Schlussfolgerungen zu ziehen! Man kann jedoch davon ausgehen, dass Warnungen meist keinen Einfluss auf die Ergebnissenauigkeit haben, solange anscheinend sinnvolle Signalverläufe berechnet wurden. Diese Ergebnisse sollte man aber nach Warnungen besonders kritisch analysieren!
- Im Ergebnisfenster wird im Beispiel ein konstanter Strom von 4,8 A im Zeitbereich von 0 bis 1 Sekunde angezeigt. Das war laut Ohmschem Gesetz bei $V=24\text{ V}$ und $R=5\text{ Ohm}$ zu erwarten (Anzeige des Wertes durch Bewegungen des Cursors mit gedrückter linker Maustaste im Diagramm):



In den Ergebnisfenstern erfolgt standardmäßig eine automatische Skalierung der Y-Achse zwischen dem minimalen und maximalem Signalwert (im Beispiel 0...4,8 A) mit Auf- bzw. Abrundung auf den nächsten ganzzahligen Wert (im Beispiel 0...5 A). Die entsprechenden Einstellungen kann man über die Eigenschaftsleiste des Ergebnisfensters vornehmen bzw. überprüfen (*Ansicht > Eigenschaftsleiste*):

- Hinter jedem "Ergebnisfenster" verbirgt sich ein komplexes System zur Darstellung, Verarbeitung und Archivierung unterschiedlichster Ergebniswerte.
- Einen Eindruck von den umfangreichen Möglichkeiten erhält man, wenn man über den Hilfe-Menüpunkt (?) des Ergebnisfensters die zugehörige Dokumentation aufruft:



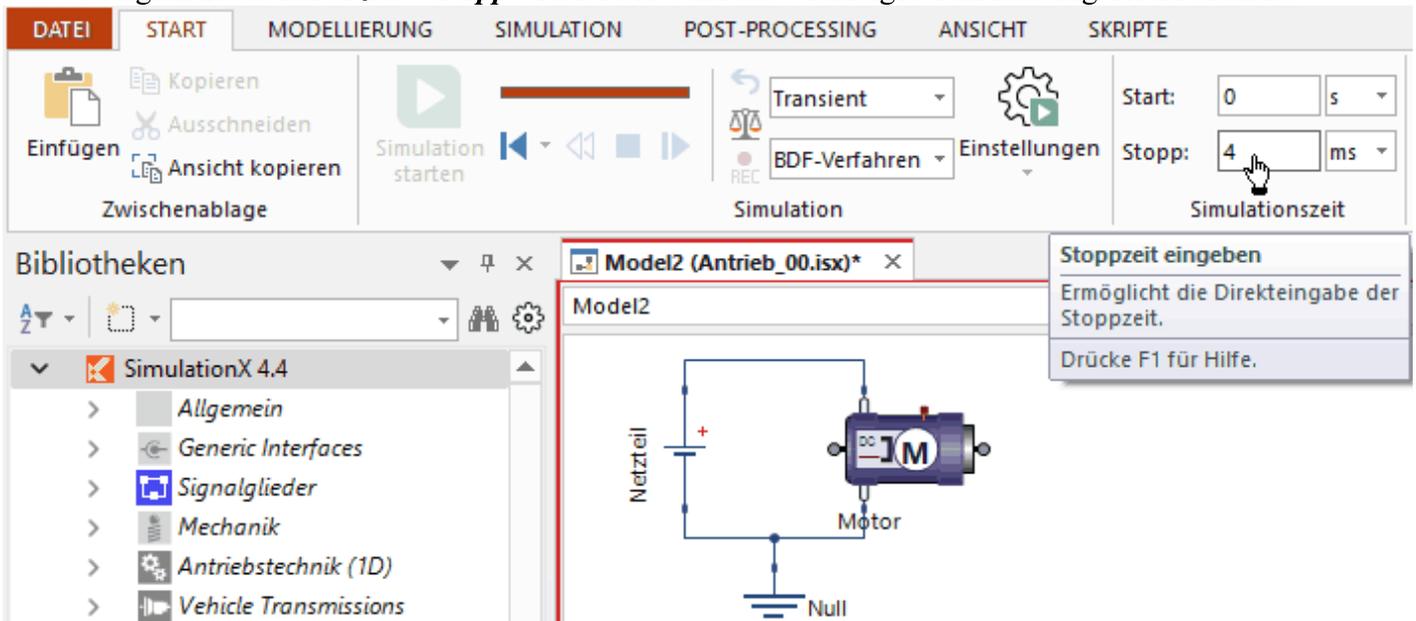
- Die Funktionsvielfalt widerspiegelt sich in der Eigenschaftsleiste in einer verwirrenden Vielfalt von Einstellmöglichkeiten:
 - Die Baumstruktur bildet die hierarchische Struktur innerhalb des Ergebnisfensters ab (**Fenster > Seite > Panel > Ergebnisgröße**).
 - Nach Wahl eines Elements innerhalb dieser hierarchischen Struktur erscheint die zugehörige Liste von Eigenschaften im darunterliegenden Feld.
 - Nach Wahl einer Eigenschaft in dieser Liste wird dazu im untersten Feld der Eigenschaftsliste eine Erläuterung eingeblendet.
 - Der Wert einer gewählten Eigenschaft ist direkt änderbar.
- Unser Ziel einer sinnvollen Skalenteilung der Y-Achse zwischen 0 und 5 A können wir bereits mit den Eigenschaften des Panels erreichen (falls durch Standardvorgabe nicht realisiert, dann ändern):
 - **Automatische Anpassung = True** (*Anpassung an veränderte Min/Max*)
 - **Zusätzliche Unterteilung = True** (*Ablesbarkeit von Kommastellen*)
 - **Ganzzahlige Unterteilung = True** (*Skalengrenzen gerundet auf nächste ganze Zahl*)
- **Hinweis:** Von den Standardvorgaben abweichende Werte werden "fett" dargestellt.

Im Motor befindet sich eine Spule, welche eine bestimmte Induktivität besitzt. Deshalb dürfte der Strom nicht "schlagartig" seinen Endwert erreichen, sondern muss vom Wert=0 A kontinuierlich ansteigen:

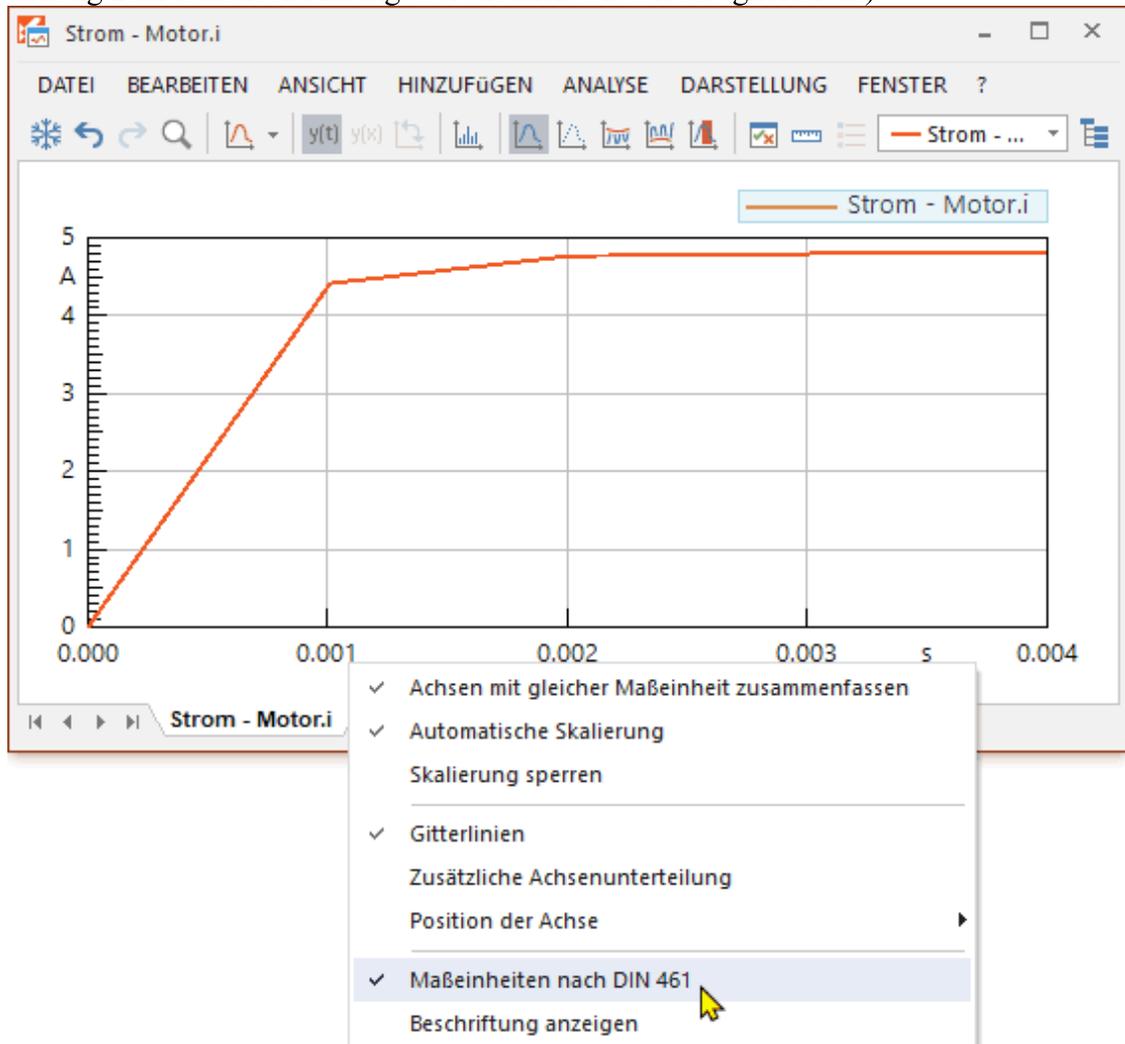
- Der standardmäßig simulierte Zeitbereich von 1 s ist für unsere Spule viel zu groß! Die elektrische Zeitkonstante **T** für den Stromanstieg beträgt nur
$$T=L/R=0.002/5 \text{ s}=0.4 \text{ ms}$$
- Da nach ungefähr der dreifachen Zeitkonstante der Endwert des Stromes fast erreicht ist, genügt eine Simulationszeit von **tStop=4 ms**.

Die Multifunktionsleiste gestattet neben anderen Einstellungen die Festlegung des berechneten Zeitbereiches:

- Wir verringern **Simulationszeit > Stopp** von 1 s auf 4 ms mit vorheriger Umschaltung der Zeiteinheit:



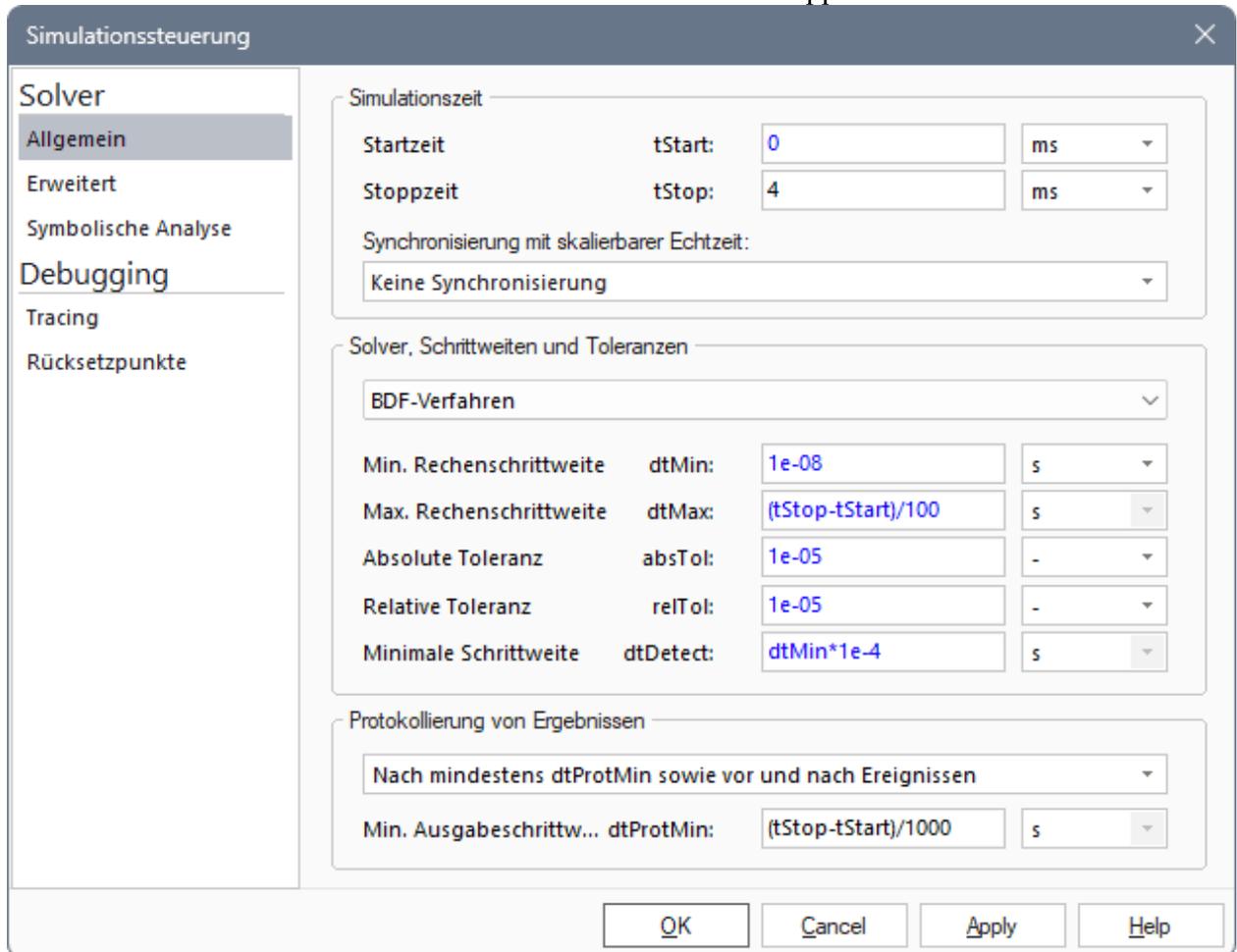
- Nach **Simulation > Rücksetzen** starten wir erneut einen Simulationslauf (die nun angezeigte, bereits bekannte Warnung zum fehlenden Anfangswert können wir wieder ignorieren!):



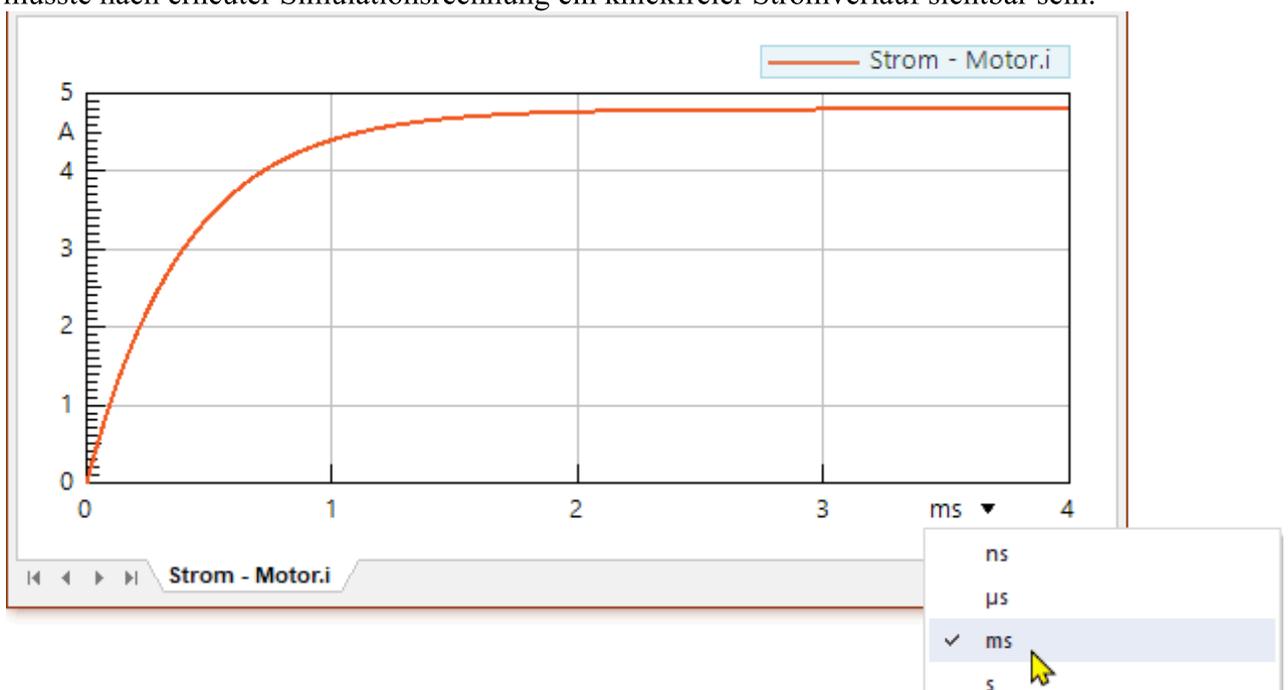
- Hinweis:** Mittels des Kontext-Menüs (der rechten Maustaste) auf den Bereich einer Koordinatenachse kann man diese grundlegend konfigurieren, ohne dafür die komplexe Einstellungsleiste zu öffnen.
- Wir sehen nun den erwarteten Stromanstieg von 0 A auf 4.8 A, allerdings ist dieser Verlauf entgegen unserer Erwartung sehr "eckig".
- Die alleinige Reduktion der Simulationendzeit genügt also noch nicht! Wir müssen über **Simulation > Einstellungen** den Dialog für die komplette "Simulationssteuerung" öffnen. Dabei genügt vorläufig die Registerkarte "Allgemein".
- Ursache für den eckigen Kurvenverlauf ist die Vorgabe für die "minimale Ausgabeschrittweite" **dtProtMin=1 ms**. Die Stromkurve wird demzufolge im Beispiel aus Geradenstücken von 1 ms Länge

approximiert!

- Damit Kurvendarstellungen unabhängig vom Zeitbereich einigermaßen "rund" sind, sollte man die Ausgabeschrittweite abhängig vom simulierten Zeitbereich machen. Für die Max. Rechenschrittweite ist z.B. standardmäßig $dtMax=(tStop-tStart)/100$ eingetragen.
- **Wichtig:** Die Ausgabeschrittweite $dtProtMin$ muss auf einen kleineren Wert als $dtMax$ gesetzt werden, z.B.: $dtProtMin=(tStop-tStart)/1000$ (ansonsten wird 1 ms benutzt!).
- Zusätzlich sollte die Protokollierung von Ergebnissen "*Nach mindestens $dtProtMin$ sowie vor und nach Ereignissen*" erfolgen. Damit werden auch Unstetigkeiten in Signalverläufen "exakt" abgebildet.
- Außerdem sollte man einheitlich die Maßeinheit **ms** für Start- und Stopzeit verwenden:



- Nun müsste nach erneuter Simulationsrechnung ein knickfreier Stromverlauf sichtbar sein:



- Es ist sinnvoll, die Maßeinheit der Zeitachse auf Millisekunden (**ms**) zu ändern.

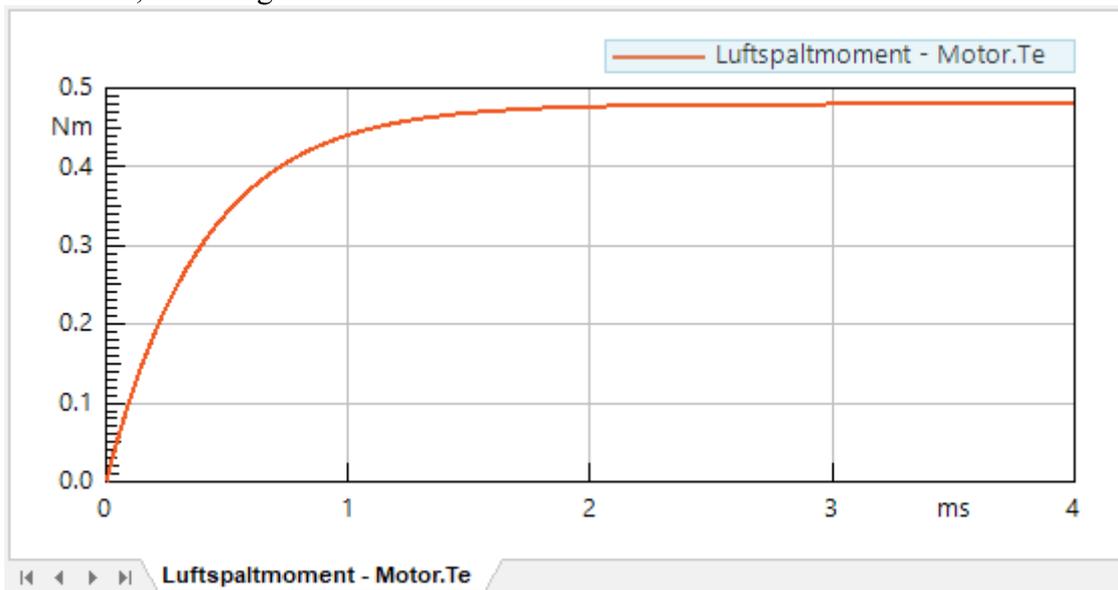
Im folgenden Experiment werden wir die Drehung des Motors und die Rückwirkung der Motor-Drehung auf den zeitlichen Verlauf des Stroms untersuchen.



Drehzahl-Experiment

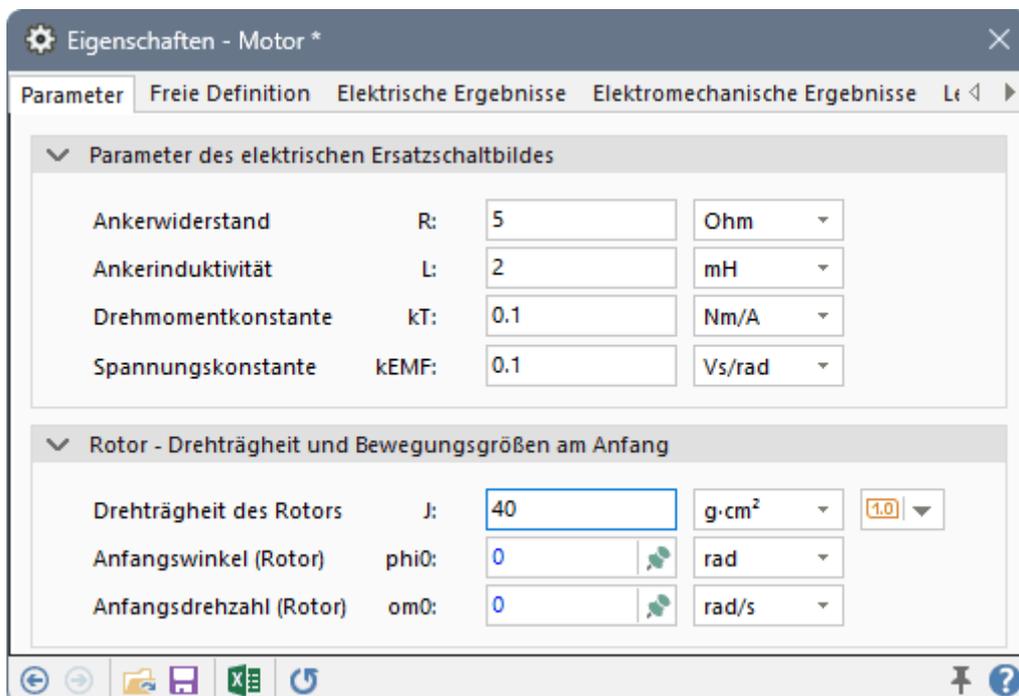
Wenn ein elektrischer Strom durch die Spule des Motors fließt, wird ein Drehmoment erzeugt und der Motor beginnt sich zu drehen. Diesen Vorgang werden wir jetzt mit dem Modell simulieren. Zusätzlich zum Strom wollen wir das Drehmoment des Motors als Signalverlauf darstellen:

- Luftspaltmoment T_e als elektromechanische Ergebnisgröße des Motors für die Ausgabe aktivieren.
- Für den Motor dieses Luftspaltmoment T_e in einem neuen Signalfenster darstellen.
- Simulation zurücksetzen und starten, damit der Signalverlauf berechnet wird.
- Der Verlauf des Drehmoments T_e sieht nach entsprechender Konfiguration des Ergebnisfensters genauso aus wie der Stromverlauf, denn es gilt $T_e = k_T \cdot i$:

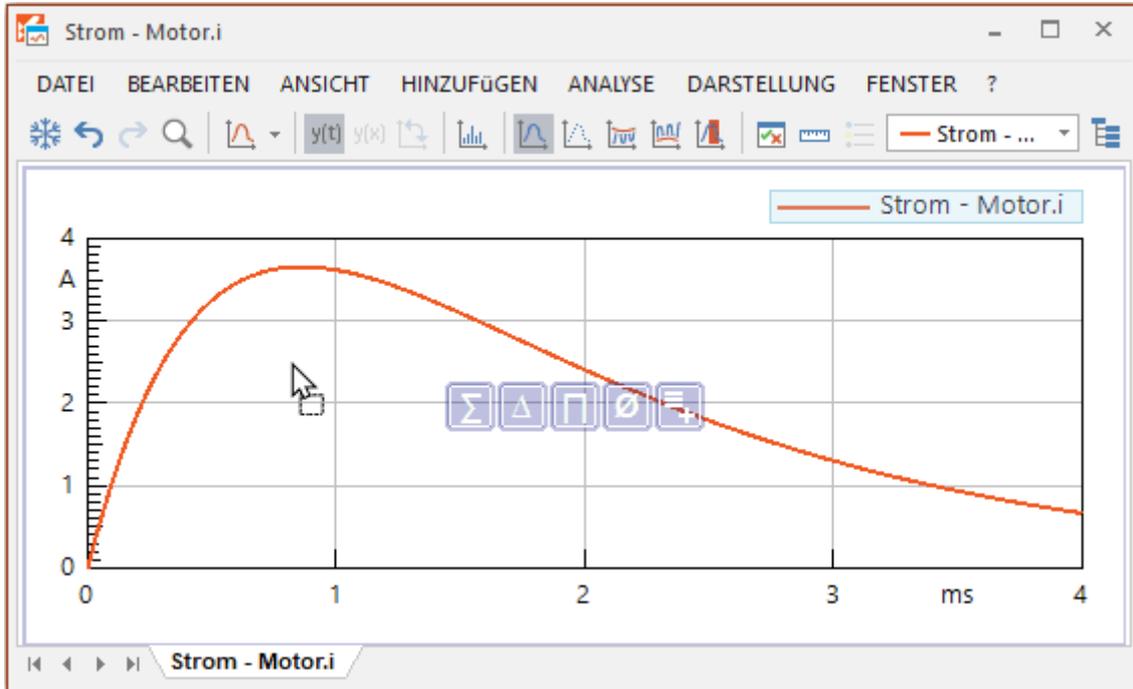


- **Hinweis:** Die Option "Ganzzahlige Unterteilung" wirkt auch bei vorangestellten Nullen entsprechend des Maximalwertes.

Unser Motor-Modell besitzt noch eine "unendlich" große Drehträgheit! Damit sich der Rotor wieder normal drehen kann, müssen wir den Wert der Drehträgheit im Modell-Element auf den richtigen Wert zurücksetzen:

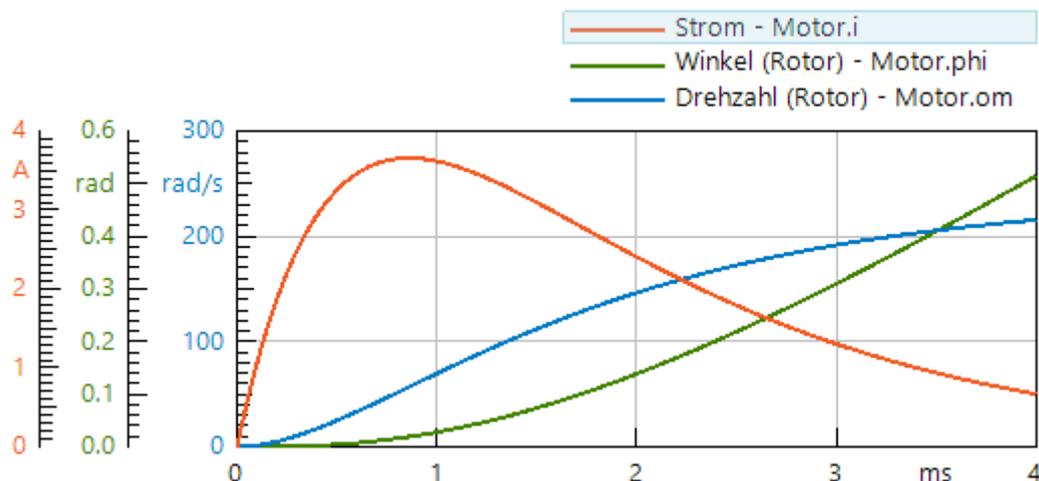


- Der Rotor unseres Motors besitzt eine Drehträgheit $J=40 \text{ g}\cdot\text{cm}^2$.
- Wir wollen uns in Form von Signalen anschauen, um welchen Winkel ϕ sich der Rotor dreht und wie groß die Drehgeschwindigkeit ω wird.
- Damit haben wir inzwischen vier Signalfenster auf dem Bildschirm, so dass es etwas unübersichtlich wird:
 - Das Signal-Fenster **Motor.Te** können wir schließen.
 - Die übrigen Signale ziehen wir mit dem Cursor per **Drag&Drop** einfach in das Signal-Fenster für den Strom (Hinweis: Ziehen der Signal-Legende, z.B. — Winkel (Rotor) - Motor.phi).
 - Im Zielfenster deutet beim **Drag&Drop** ein farbiger Rahmen an, ob das vorhandene Diagramm-Panel auf der Seite genutzt wird, oder ob ein neues Diagramm-Panel angelegt wird:



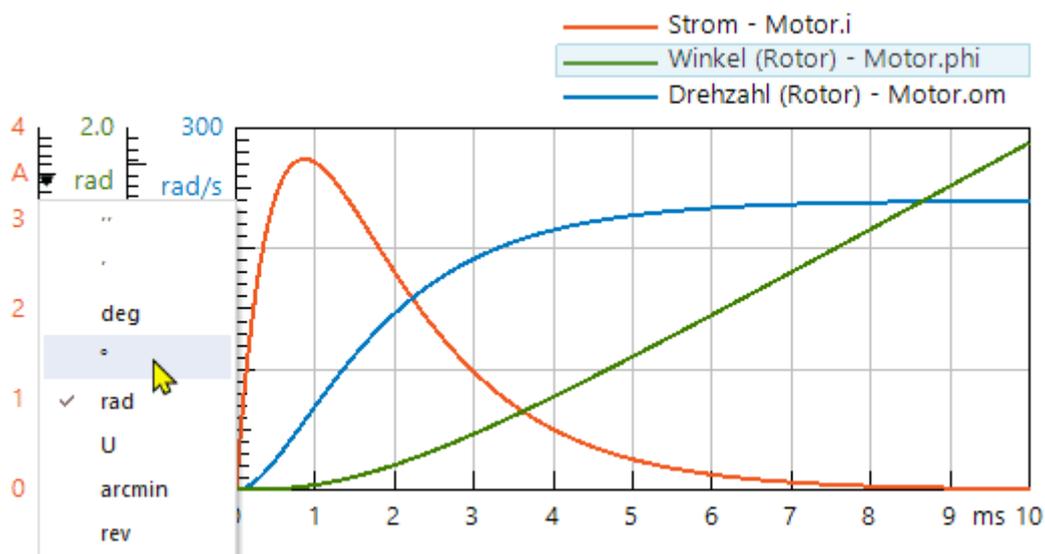
- Es wäre möglich, dass hinzugefügte Signal mit dem vorhandenen Signal zu verknüpfen. Die möglichen Verknüpfungsoperationen werden als farbige Symbole eingeblendet.
 - Wir schieben die Signale ohne Verknüpfung in das vorhandene Diagramm-Panel.
 - Hinweis:** Besitzen die Zeitachsen vor dem Zusammenkopieren der Signale unterschiedliche Einheiten, muss man nachträglich die drei Zeitachsen zu einer gemeinsamen Achse zusammenfassen (Kontext-Menü der Zeitachsen). Als Maßeinheit verwenden wir wieder die Millisekunde.

Nun wollen wir uns die Zeit nehmen und die Signal-Verläufe genauer betrachten:

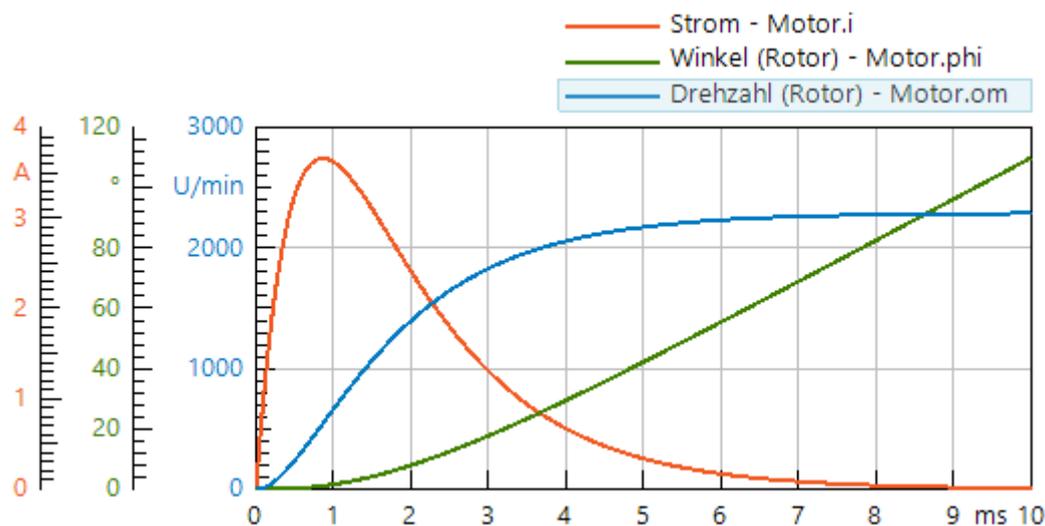


- Der Strom i steigt wieder von Null beginnend an. Aber anstatt einen Endwert von 4,8 A zu erreichen, verringert sich der Motor-Strom danach wieder.
- Ursache für das Absinken des Motorstroms ist die beginnende Drehbewegung des Motors. Die Rotor-Drehzahl ω erhöht sich stetig.
- Der vom Rotor zurückgelegte Winkel ϕ wird dabei immer größer.

- Am Ende der durchgeführten Simulation haben die Signale noch keinen Endwert erreicht. Deshalb wollen wir in der Simulationssteuerung **tStop=10 ms** setzen. Ohne vorheriges *Rücksetzen* kann man die Simulationsrechnung durch *Simulation starten* beginnend von der bisherigen Endzeit 4 ms fortsetzen lassen.
- Anstatt der Maßeinheit **rad** wollen wir die anschaulichere Einheit **°** verwenden (Hinweis: das Grad **°** ist identisch mit **deg**). Das bewerkstelligt man einfach über das Menü an der Maßeinheit:

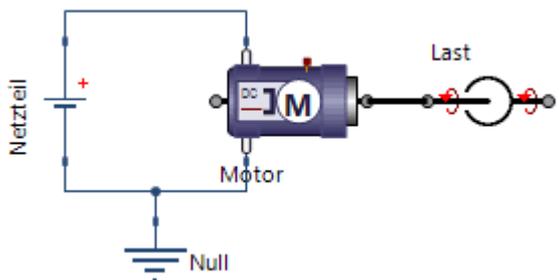


- Genauso lassen wir uns die Drehzahl in **U/min** anzeigen:

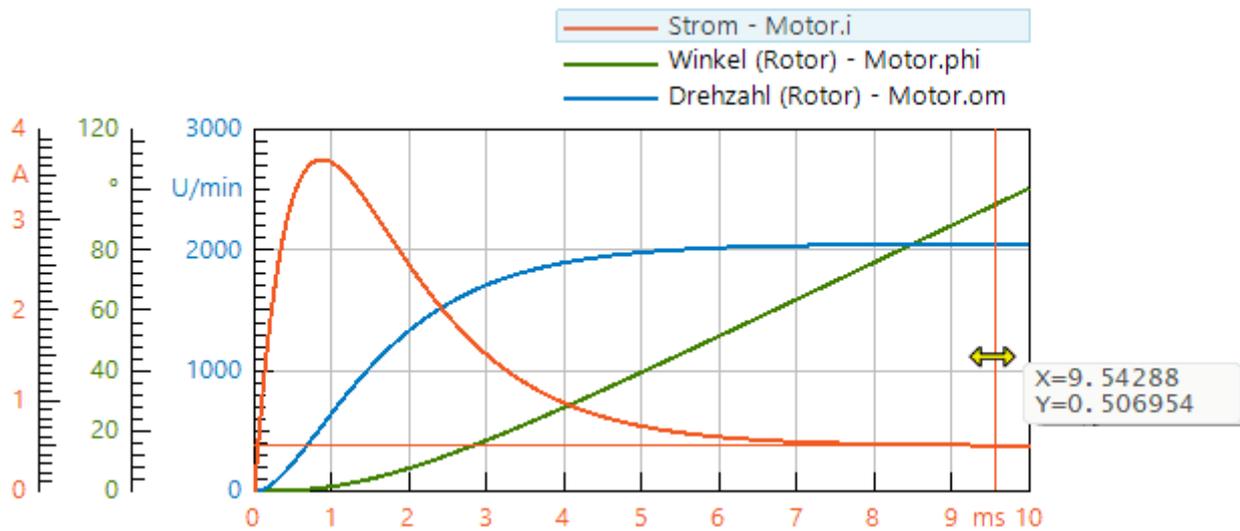


- Der verwendete Motor erreicht ohne Belastung eine Enddrehzahl von ungefähr 2300 U/min.
- Die Enddrehzahl ist erreicht, wenn der Motorstrom kein antreibendes Drehmoment mehr erzeugt. Dann wird der Rotor nicht mehr beschleunigt.
- Ein Motor wirkt gleichzeitig als Generator. Der Motorstrom geht im Leerlauf (ohne Reibung) auf Null, wenn die infolge der Rotor-Drehung induzierte Spulenspannung gleich der Betriebsspannung ist. Dann ist die Spannungsdifferenz über der "Induktivität" der Spule gleich Null.

Abschließend zu diesem Experiment werden wir den Rotor noch mit einer Dämpfung belasten (Rotatorische Mechanik). Ein Dämpfer-Element erzeugt ein Last-Moment, das proportional zur Drehzahl ist:



- Von Null beginnend wollen wir die Last (**Dämpfung b**) vorsichtig erhöhen, bis ein Endstrom von ungefähr $i=0,5\text{ A}$ fließt:



- Die Enddrehzahl verringert sich bei dieser Belastung um ca. 10%.

Hinweis: den Endstrom $i(t_{\text{Stop}})$ kann man auf zwei Arten ermitteln

1. Wahl von **Motor.i** im Ergebnisfenster (z.B. über die Legende) und Wahl des Zeitpunktes **X** mit dem Cursor (Siehe vorheriges Bild).
2. Wahl des Elements Motor in der Modellstruktur-Ansicht oder im Modellexplorer. Der aktuelle End-Wert des Motorstroms erscheint in der zugehörigen Liste der Ergebnisse (unabhängig davon, ob das Ergebnis als Signal aktiviert wurde):

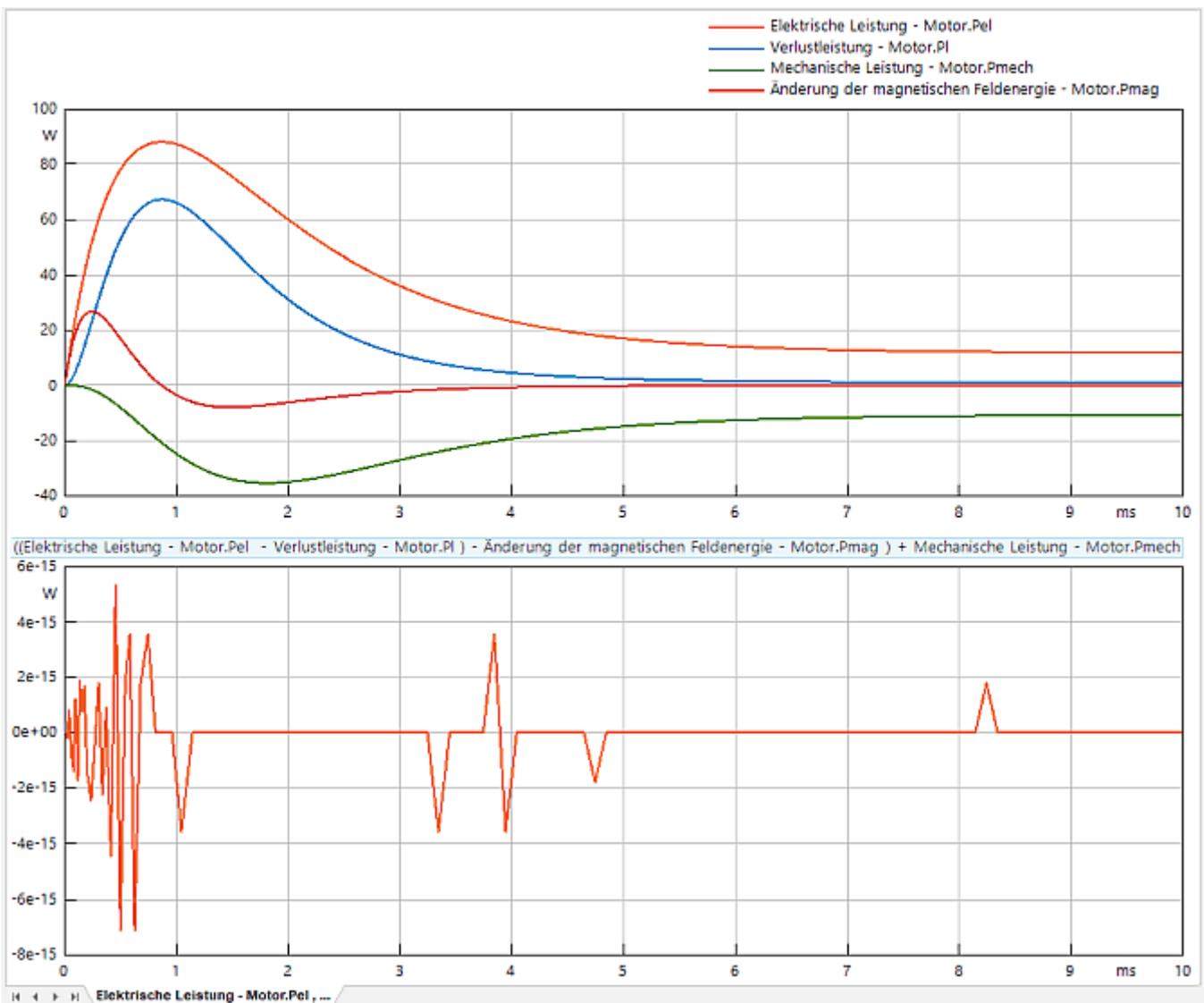
Parameter			
Kommentar	Name	Aktueller Wert	Maßeinheit
Ankerwiderstand	R	5	Ohm
Ankerinduktivität	L	2	mH
Drehmomentkonstante	kT	0.1	Nm/A
Spannungskonstante	kEMF	0.1	Vs/rad
Drehträgheit des Rotors	J	40	g·cm ²

Ergebnisse				
Kommentar	Name	Aktueller Wert	Maßeinheit	Protokoll
► Strom	i	0.505107058455173	A	✓
Spannung	v	24	V	✗
Luftspaltmoment	Te	0.0505107058455173	Nm	✓
Mittelwert des Luftspaltmoments	TeMean	0.0505107058455173	Nm	✗
Inneres Moment	Ti	0.0500511442777947	Nm	✗
Beschleunigungsmoment (Rotor)	Ta	0.000459561567722658	Nm	✗
Winkel (Rotor)	phi	1.75573157438274	rad	✓
Drehzahl (Rotor)	om	214.811778016286	rad/s	✓
Beschleunigung (Rotor)	alp	114.890391930664	rad/s ²	✗
Elektrische Leistung	Pel	0.0121225694029242	kW	✗

Komponenten

- Model2
 - DCM1 Motor
 - Netzteil
 - connection1
 - connection2
 - Null
 - Last
 - connection3

- Im Endzustand wird die elektrische Leistung von $P_{el}=12\text{ W}=24\text{ V}\cdot 0,5\text{ A}$ teilweise in der Motor-Spule verheizt ($P_{I}=1,25\text{ W}=0,5^2\text{ A}^2\cdot 5\text{ Ohm}$).
- Der Rest von $P_{mech}=10,75\text{ W}$ entspricht der im Motor-Luftspalt bereitgestellten mechanischen Leistung.
- Die Änderung der magnetischen Feldenergie P_{mag} ist im eingeschwungenen Zustand praktisch Null.
- Wie sich die Verhältnisse zwischen den einzelnen Leistungsanteilen (elektrisch, thermisch, magnetisch, mechanisch) während des Anfahrvorgangs ändern, kann man sich anhand der berechneten Signalverläufe anzeigen lassen (Maßeinheit von Kilowatt auf Watt ändern):



- Fasst man in einem neuen Panel unter Beachtung der Vorzeichen alle Leistungsanteile als Summe zusammen, so ergibt sich der Wert=0 (mit numerischem Rauschen). Der Energie-Erhaltungssatz wird in unserem Modell für den Motor also nicht verletzt.
- **Hinweis:** Auf Grundlage der bisherigen Kenntnisse sollte es möglich sein, wie im Bild gezeigt, in einem neuen Panel des gleichen Ergebnisfensters durch Kopieren (*Drag&Drop* mit gedrückter <Strg>-Taste) und Benutzung der eingblendeten Operatoren mit den Leistungssignalen die vorzeichenrichtige Summenbildung vorzunehmen. Man muss dabei vor allem beachten, dass **Pmech** bereits mit negativem Vorzeichen berechnet wird!

Speichern des Modells nicht vergessen!

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Einfuehrung_-_DC-Motor_-_Drehzahl&oldid=26781“

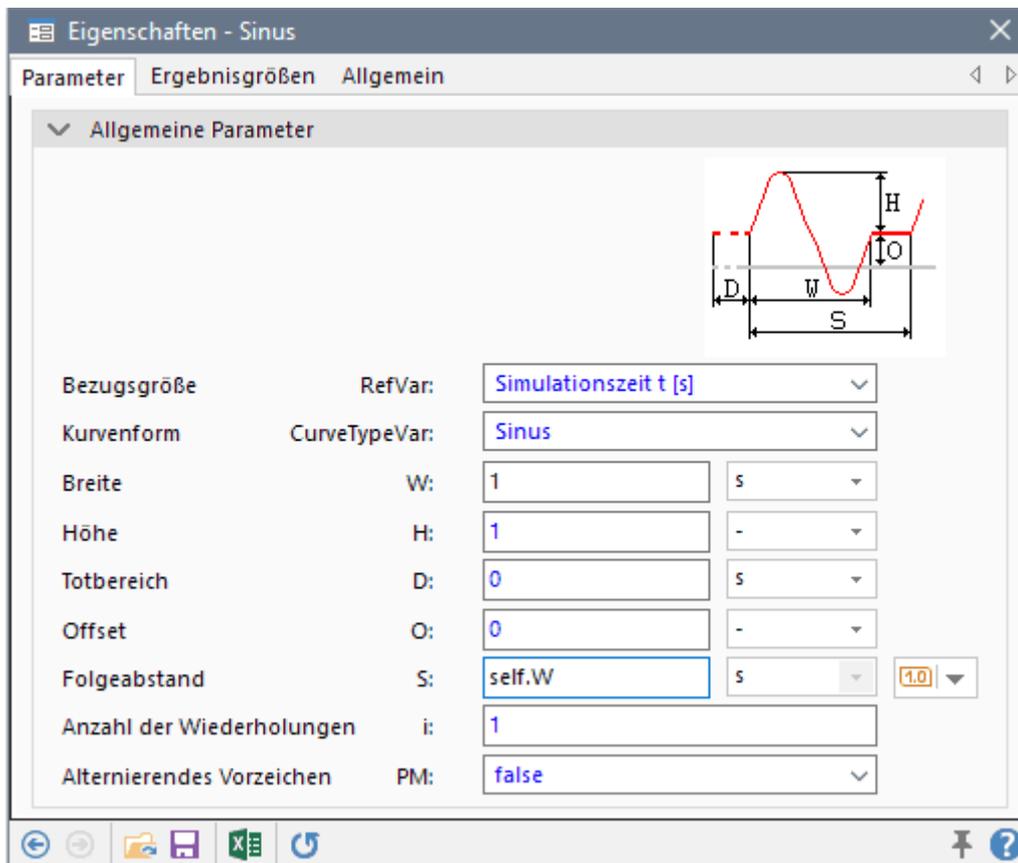
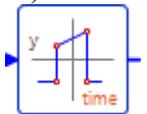


Regelkreis-Experiment

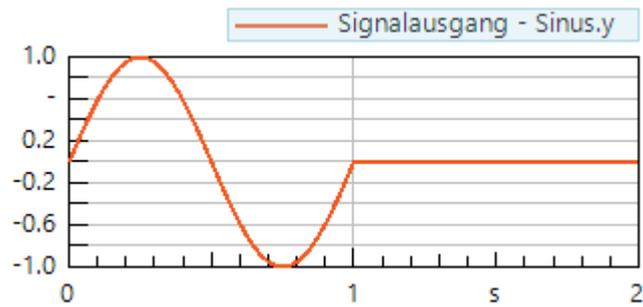
Oft soll ein Antrieb nicht nur ein- und ausgeschaltet werden, sondern sich nach einer vorgegebenen Sollkurve bewegen. Das kann man mit einem Gleichstrom-Motor hinreichend genau nur mit einem Regelkreis realisieren

In unserem Beispiel soll der Motor erst in eine Richtung hochdrehen und dann die Bewegung umkehren. Zum Schluss soll er wieder stehen bleiben. Der gesamte Vorgang soll 1 Sekunde dauern. Bei der Bewegung soll der Motor möglichst ruckfrei beschleunigen, also ganz "zart" die Drehzahl ändern. Dafür wollen wir zuerst einen Sollwert-Generator für die Vorgabe der Winkelgeschwindigkeit aufbauen:

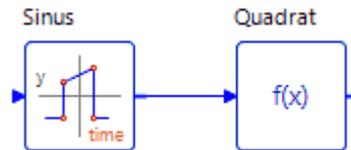
- Es bietet sich an, für diesen Bewegungsvorgang eine komplette Sinus-Schwingung (=1 Sinus-Impuls) als Grundlage zu nehmen.
- Wir finden in der Bibliothek das Element **Impulsgenerator** unter "Signalglieder - Signalquellen".
- Nach dem Einfügen in unser Antriebsmodell mit dem Namen "**Sinus**" konfigurieren wir diesen Generator als Sinusgenerator mit einer Impulslänge von 1 s:



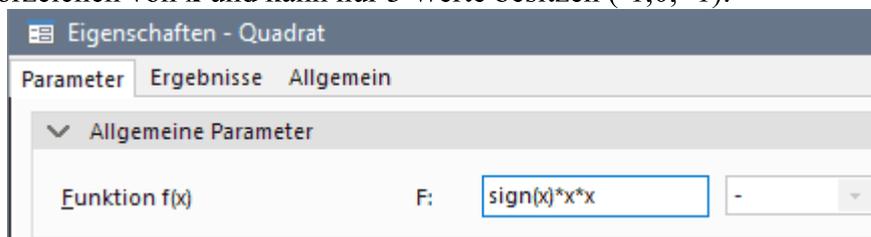
- **Hinweis:** Der Name *self* bezeichnet das eigene Element. SimulationX ergänzt diesen Vorsatz automatisch für alle Bezeichner, welche sich auf das eigene Element beziehen. Im Beispiel ist der Folgeabstand S also immer so groß wie die Impuls-Breite W .
- Wenn wir die Simulationszeit für das Modell auf 2 s erhöhen, erhalten wir folgendes Ausgangssignal vom Sinusgenerators:



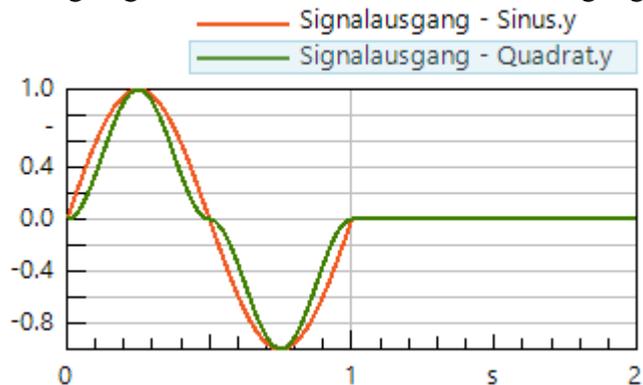
Ein solcher Drehzahl-Verlauf wäre aber noch nicht sanft genug. Am Anfang und Ende gäbe es einen ordentlichen Ruck! Deshalb quadrieren wir den Funktionsverlauf unter Beibehaltung des Vorzeichens:



- Wir nutzen dafür das Element $f(x)$ direkt aus dem Bibliotheksordner "Signalglieder".
- $\text{sign}(x)$ liefert das Vorzeichen von x und kann nur 3 Werte besitzen (-1,0,+1):



- Die Zusammenschaltung beider Signalglieder liefert nun eine sanfte Bewegungssollkurve:

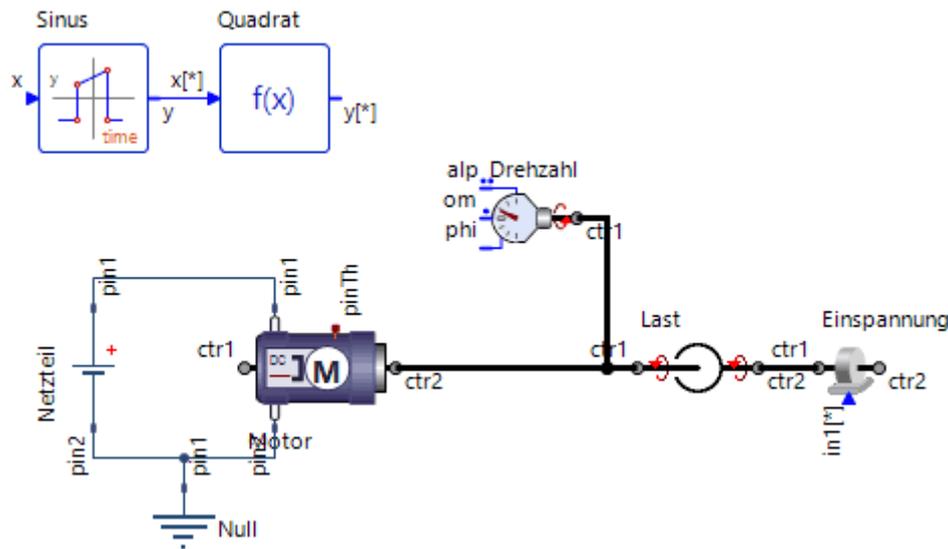


Zusätzlich zum Sollwert-Generator benötigen wir einen Sensor, welcher die aktuelle Drehzahl misst. In der Praxis funktioniert das z.B. mit einer Codescheibe in einem optischen inkrementalen Geber:



Wir wollen das im Beispiel stark vereinfacht mit einem trägheitslosen, analogen Sensor-Element aus der Bibliothek realisieren:

- Den "Bewegungssensor" findet man in der Bibliothek im Ordner "Rotatorische Mechanik (1D)". Dieser Sensor liefert nach seinem Verbinden mit dem Last-Element alle Bewegungsgrößen an seinen Ausgängen (phi, om, alp):



- Das Sensor-Element dreht man in der Modellstruktur in die gewünschte Position.
- Damit man sieht, an welchem Anschluss welches Signal anliegt, kann man die Anschlüsse innerhalb der Modellstruktur einblenden (Ansicht > Beschriftung > Anschlüsse). Diese Beschriftung sollte man dann wieder ausschalten, weil sie meist stört!

Hinweise zu unverbundenen Anschlüssen:

- Freie Anschlüsse eines Kraftelements (z.B. Feder, Dämpfer, Reibung) sind im Modell "automatisch" fest eingespannt. Um Fehler zu vermeiden, sollte man möglichst (wie gezeigt) die in der Realität vorhandene Einspannung als Element ergänzen (Rotatorische Mechanik(1D) - Element "Vorgabe" - konfiguriert als "Einspannung").
- Ein sehr "gemeiner" Modellierungsfehler wäre z.B. die Verbindung des Drehzahlsensors mit dem "freien" Anschluss **ctr2** der Last-Dämpfung. Dann würde die Dämpfung praktisch unwirksam, da nicht mehr die Relativbewegung zwischen dem Rotor und dem Motorgehäuse erfasst würde!
- Der "freie" Anschluss **Motor.ctr1** entspricht dem Motorgehäuse, das praktisch auch fest eingespannt ist. Wegen der Übersichtlichkeit wurde hier auf eine Verbindung mit einem Einspann-Element verzichtet.

Als Amplitude für unseren Sollwert-Verlauf wollen wir **100 rad/s** vorgeben. Dazu müssen wir die Quadrat-Funktion um diesen Faktor ergänzen):

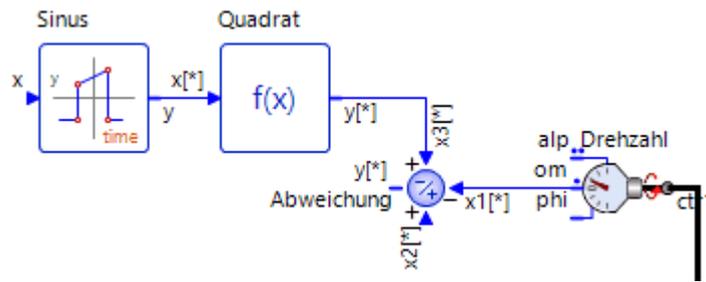
$$F: \text{sign}(\text{self.x}) * \text{self.x} * \text{self.x} * 100$$

Hinweise zu Maßeinheiten:

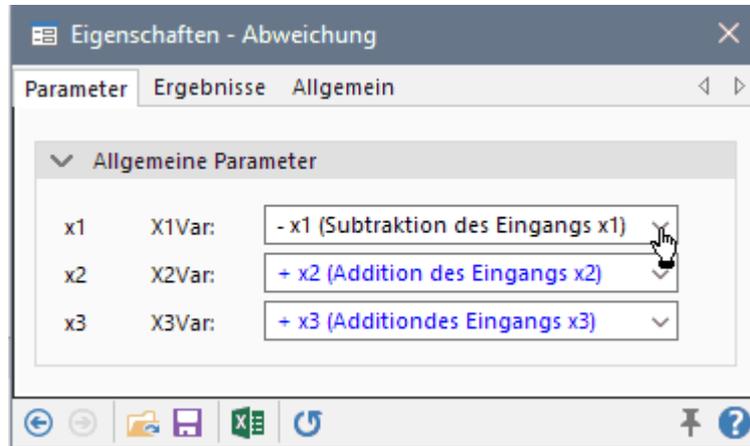
- Die Modellberechnung in SimulationX erfolgt grundsätzlich mit den SI-Einheiten!
- Verwendet man eine Modellgröße in einer Formel, so enthält die Modellgröße immer den aktuellen Zahlenwert für die SI-Einheit.
- Deshalb wurde im Beispiel eine Drehzahlvorgabe in der SI-Einheit rad/s verwendet, ansonsten hätte man die Umrechnung in der Formel entsprechend berücksichtigen müssen!
- Nur für die Eingabe der Parameter und die Darstellung von Ergebnisgrößen können handliche Vorsätze zu den SI-Einheiten gewählt werden. SimulationX nimmt dann automatisch die Umrechnung zur SI-Einheit vor.

Im nächsten Schritt muss man die Regelabweichung als Differenz **Sollwert minus Istwert** bilden:

- Wir nutzen für die Bildung der Abweichung aus der Bibliothek einen Summationspunkt (direkt im Ordner "Signalglieder"):

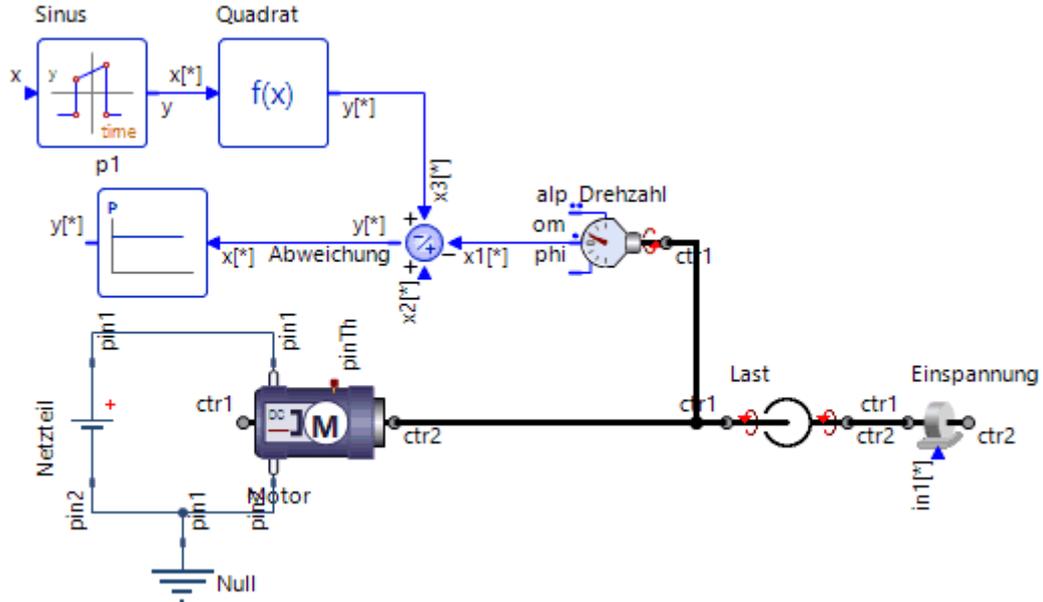


- Im Beispiel wurde dieses Element um 180° gedreht und der Eingang x_1 mit dem Istwert gespeist. Der Sollwert geht in den Eingang x_3 . Um $Y=x_3-x_1$ zu bilden, muss man den Summationspunkt wie folgt konfigurieren:



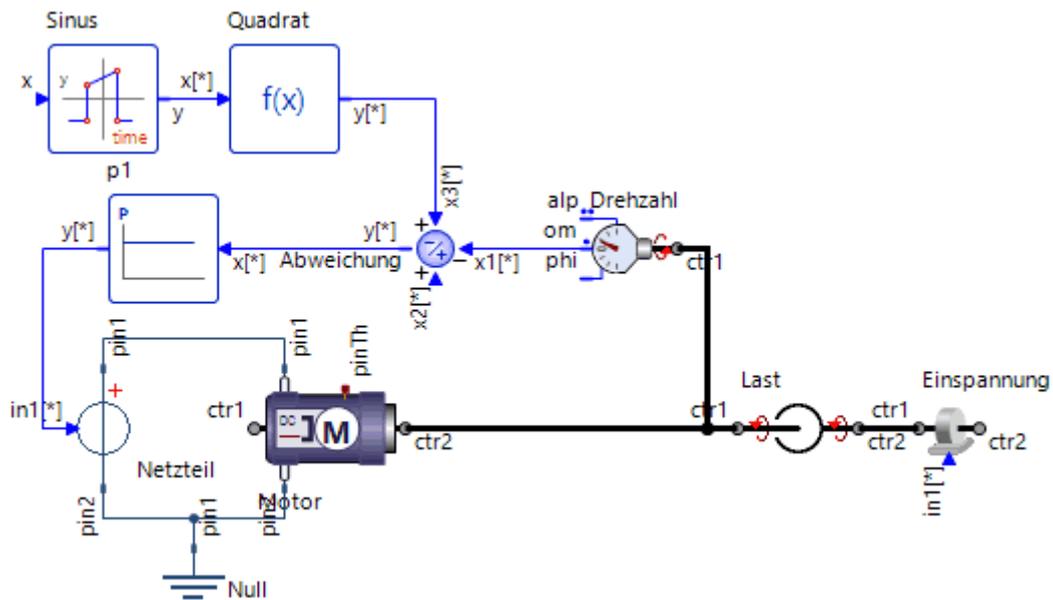
Der Regler wertet die Regel-Abweichung aus und beeinflusst dann die Betriebsspannung für den Motor so, dass diese Abweichung möglichst klein wird:

- Wir nutzen einen einfachen P-Regler (P-Element in der Bibliothek unter "Signalglieder - Lineare Signalglieder") und speisen seinen Eingang mit der Regelabweichung.
- Ein P-Regler verstärkt mit dem Faktor G die eingespeiste Regelabweichung (Proportional-Regler):

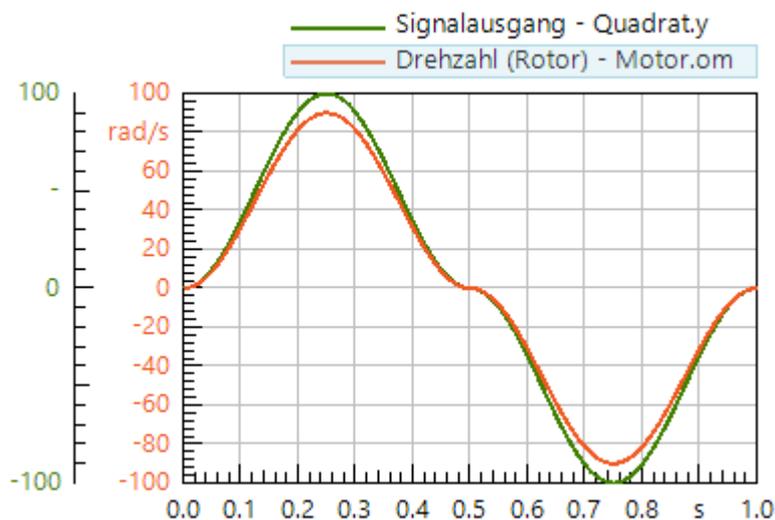


Mit dem resultierenden Ausgangswert des Reglers muss man nun entsprechend die Betriebsspannung für den Motor beeinflussen. Dazu benötigen wir eine gesteuerte Spannungsquelle:

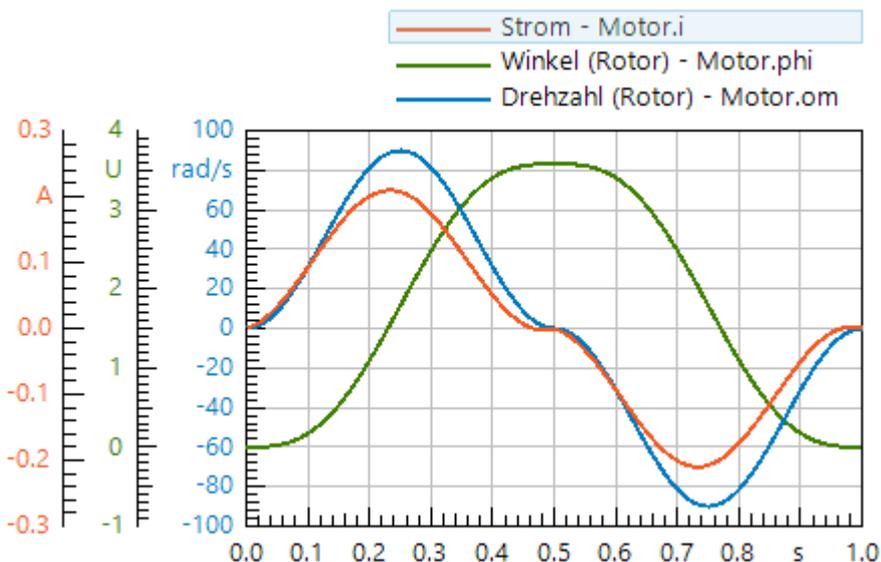
- Wir ersetzen die "Konstantspannungsquelle" (im Modell löschen) durch eine "Spannungsquelle mit Signaleingang" (unter "Elektronik - Analog - Quellen").
- Der Ausgang **Regler.y** wird mit dem Signaleingang **in1** der Spannungsquelle verbunden:



- Als Ausgangsspannung des Netzteils soll direkt der Ausgangswert **Regler.y** benutzt werden (entspricht **in1** des Netzteils).
- Durch Ändern des Verstärkungsfaktors **G** des Reglers kann man nun eine hinreichende Übereinstimmung zwischen Sollwert- und Istwert-Verlauf der Drehzahl erreichen.
- Es ist günstig, beide Signal-Verläufe in einem Fenster darzustellen.
- Die Verstärkung sollte man möglichst niedrig halten, weil hohe Verstärkungen zu instabilen Reglerverhalten führen.
- Der dargestellte Drehzahlverlauf ergibt sich bei $G=1$:



- Der Regler gewährleistet, dass durch die Motorspule zu jedem Zeitpunkt der erforderliche Strom fließt:



Achtung!

Leider wird der Proportional-Regler in der Praxis nie so gut funktionieren wie hier im Modell:

- Im Modell wurden Zeitverzögerungen im Antriebsstrang, z.B. infolge von Nachgiebigkeiten in der Antriebswelle, nicht berücksichtigt.
- In der Praxis führt das zum Schwingen des Antriebssystems schon bei relativ kleinen Verstärkungen.
- Mit den möglichen kleinen Verstärkungen kann man nur eine ziemlich große Regelabweichung realisieren. Deshalb ergänzt man in der Praxis den Regler z.B. um einen Integral-Anteil. Damit kann man im Idealfall die Regelabweichung stark reduzieren.

Speichern des Modells nicht vergessen!

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Einfuehrung_-_DC-Motor_-_Regelkreis&oldid=26789“



Was ist nun eigentlich Simulation?



Umgangssprachlich ist ein **Simulant** jemand, welcher nicht vorhandene Symptome einer Krankheit vortäuscht, um daraus Vorteile zu ziehen. Diese Art Simulation haben wir hier nicht betrachtet! Bei uns ging es darum, durch Experimentieren neue Erkenntnisse zu gewinnen. Im betrachteten Beispiel wäre ein Erkenntnisziel, wie man den Regler dimensionieren sollte, damit der Antrieb sich wie gewünscht verhält.

Um diese Erkenntnis zu gewinnen, führten wir Experimente durch ("mal schauen, was passiert"). Allerdings verwendeten wir dabei nicht den echten Antrieb, sondern benutzten ein Computermodell.

Verallgemeinert kann man sagen:

Simulation ist jegliche Benutzung eines Modells als Ersatzobjekt, um damit Erkenntnisse über das modellierte Objekt zu gewinnen.

Dazu einige Erläuterungen:

- Simulation hat nicht unbedingt etwas mit der Benutzung eines Computers zu tun. Wenn man z.B. zwei Bälle und eine Kerze zur Verdeutlichung der von der Erde aus sichtbaren Mondphasen benutzt, ist das auch eine Simulation.
- Auch wenn man über etwas nachdenkt, "simuliert" man. Man nutzt in den Gedankenexperimenten dann Sprach-Konstruktionen als Ersatz-Objekte für das, was in der Realität stattfinden soll.
- Simulation ist etwas typisch Menschliches. Denn bevor man eine geplante Handlung ausführt, hat man sich meist bereits anhand von Ersatz-Objekten (Modellen) überlegt, was dann passiert.
- Leider ist die "Simulationsleistung" des menschlichen Gehirns ziemlich beschränkt. Durch die Benutzung von Computer-Modellen kann der Mensch auf eine höhere Simulationsleistung zugreifen. Denn viele Operationen des menschlichen Nachdenkens kann man in Form von Computer-Algorithmen formulieren (z.B. im Sinne "if ... then ... else;" oder im Sinne "Künstlicher Intelligenz unter Nutzung *Neuronaler Netze*"). So kann man mit Computer-Simulationen ganz neue Lösungen für komplizierte Problemstellungen finden!

Bei aller Technik-Begeisterung sollte man jedoch beachten:

- **"Nicht alles lässt sich simulieren!"**
- Letztendlich funktioniert nur die reale Welt richtig und darf als Prüfstein für jegliche Simulationsergebnisse nicht vergessen werden!



Abgerufen von „http://index.php?title=Software:_SimX_-_Einfuehrung_-_DC-Motor_-_Simulation&oldid=23326“