

Software: SimX - Nadelantrieb - Wirkprinzip

Aus OptiYummy

↑

← →

1. Etappe im Übungskomplex "Nadelantrieb" Wirkprinzip-Entscheidung (E-Magnet) Autor: Dr.-Ing. Alfred Kamusella

*Fehler vermeidet man, indem man Erfahrung sammelt.
Erfahrung sammelt man, indem man Fehler macht.
- Laurence J. Peter -*

Modell des Magnetantriebs

- Antriebsmechanik (Präge-Nadel mit Rückholfeder)
- Antriebsmechanik (Papier und Anschläge)
- Elektromagnet
- Bewertungsgrößen 🟢 **Grundlagen-Optimierungsprozess**: Entwurfsproblem → Optimierungsaufgabe
- Signalprozessor-Teilmodell (Compound-Definition)

Numerischer Versuchsstand

- Experiment
- Simulationsmodell
- Entwurfparameter (Nennwerte)
- Forderungen (Restriktionen)
- Wünsche (Gütekriterien)
- Optimierungsverfahren
- Visualisierung

Experimente

- Lokale Suche (Hooke-Jeeves)
- Auswertung (Lokale Suche)
- Gütefunktion (Rastersuche)
- Zielfunktion (Hierarchische Optimierung)
- Ergebnisse der 1. Etappe

Selbststudium:

🟢 ist ein Verweis auf das zugehörige Kapitel im "Vorlesungsinhalt", welches außerhalb der Übungszeit ergänzend zum markierten Übungsabschnitt zu lesen ist.

Einzusendende Ergebnisse:

- Teilnehmer der Lehrveranstaltung **Optimierung** laden ihre Ergebnisse im zugehörigen Opal-Kurs hoch.
- Mit (xx=Teilnehmer-Nummer 01...99) sind dazu die folgenden konfigurierten Modelldateien in einem Archiv-File (z.B. ZIP) zu senden: **Etappe1_xx.isx**, **Etappe1_xx.opy** und **Text-Datei** für Bestwert.

Einsendeschluss:

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

Software: SimX - Nadelantrieb - Wirkprinzip - Aufgabe

Aus OptiYummy

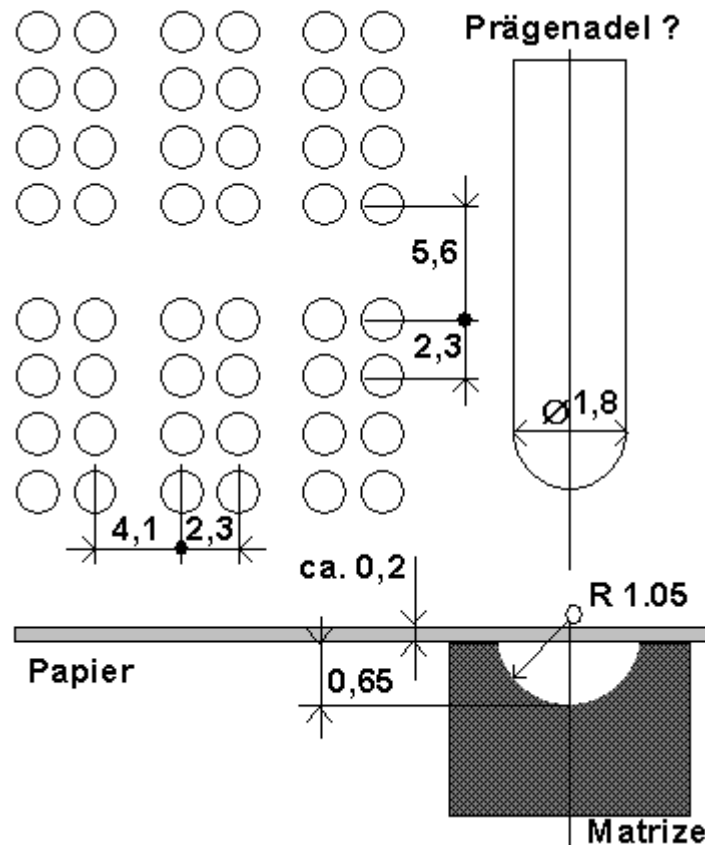
↑



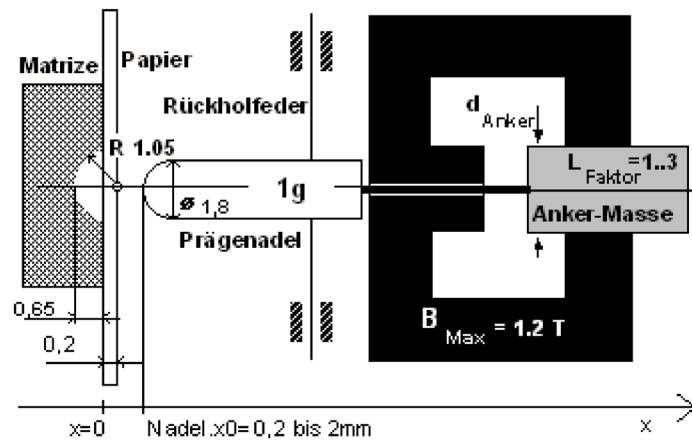
Prägenadel-Antrieb



Als Komponente eines *Druckers* für *Braille- (Blinden)schrift* ist eine Baugruppe zu entwickeln, welche die notwendigen Grübchen in das Papier prägt. Jedes Zeichen besteht aus einer 2x4-Punktmatrix, um auch Sonderzeichen codieren zu können. Bei der klassischen manuellen Prägung haben das Papier, die verwendete Matrize und die Prägenadel-Kuppe folgende Abmessungen:



Es soll eine Prägeschwindigkeit von mindestens **130 Zeichen/s** erreicht werden. Mit einer einzelnen Nadel müsste man ca. **1000 Punkte/s** prägen, was als kaum realisierbar angesehen wird. In der zu untersuchenden Antriebsvariante ist deshalb ein Prägekopf mit einer Spalte aus 4 Präge-Nadeln vorgesehen. Jede Nadel der Baugruppe erhält einen eigenen elektromagnetischen Antrieb in Form eines Topfmagneten. Die Prägenadel soll starr mit dem Anker des Topfmagneten verbunden sein:



Es wird von folgenden Annahmen ausgegangen:

- Die Dicke des zu prägenden Papiers beträgt maximal **0,2 mm**. Nach dem Prägen in der **0,65 mm** tiefen Matrize besitzt das Papier an dieser Stelle nur noch eine Dicke von **0,10 mm**.
- Der erforderliche Hub für die Präge-Nadel beträgt also minimal ca. **0,75 mm**. Falls man eine "Freiflug"-Phase der Nadelspitze über dem Papier vorsieht, vergrößert sich der erforderliche Hub entsprechend.
- Es müssen mindestens **260 Spalten/s** geprägt werden. Für einen Prägezyklus stehen nur $1/260 \text{ s} = \text{ca. } 3,8 \text{ ms}$ zur Verfügung. Mit einem Sicherheitszuschlag von **0,2 ms** sollte ein Nadel-Antrieb einen kompletten **Prägezyklus in 3,6 ms** vollenden können.
- Es wird geschätzt, dass jede Präge-Nadel eine bewegte Masse von insgesamt ca. **0.5 g bis 2 g** besitzt.



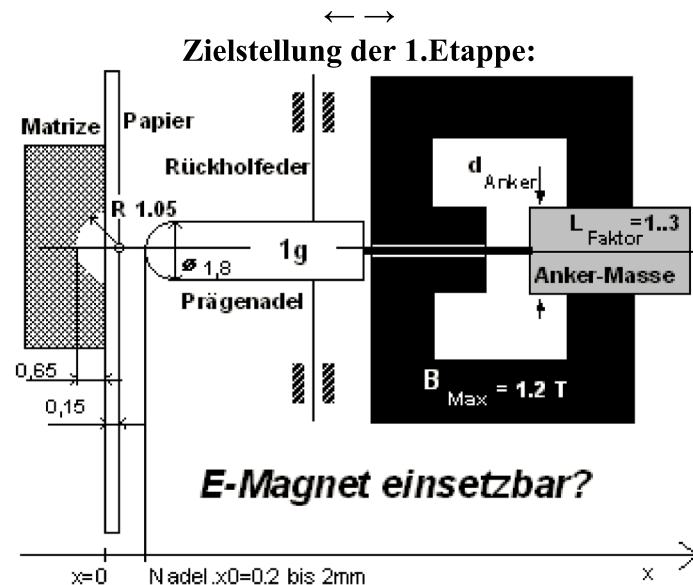
← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Aufgabe&oldid=27738“

Software: SimX - Nadelantrieb - Wirkprinzip - Zielstellung

Aus OptiYummy

↑



Die Idee, als elektro-mechanischen Wandler einen Elektromagneten einzusetzen, ist sicher nahe liegend. Allerdings sind die Anforderungen an die Arbeitsgeschwindigkeit recht hoch! Deshalb ist es fraglich, ob ein E-Magnet überhaupt in der Lage ist, einen Prägezyklus innerhalb von **3,6 ms** auszuführen:

- Die Querschnittsfläche des Ankers im Luftspalt ist zusammen mit der magnetischen Sättigung des Eisens bei hoher Flussdichte verantwortlich für die maximal mögliche Magnetkraft. Bei Vernachlässigung von Streuflüssen gilt näherungsweise:

$$F_{\text{max}} = \frac{B_{\text{max}}^2 \cdot A_{\text{Anker}}}{2 \cdot \mu_0}$$

- Normales Weicheisen erreicht den Sättigungsbereich für die magnetische Flussdichte ungefähr bei **B_{max}=1,2 T**. Viel höher sollte man den Magnetkreis nicht aussteuern!
- Man kann zwar durch Vergrößern des Ankerdurchmessers **d** "beliebig" große Kräfte (**F~d²**) erzeugen, aber gleichzeitig erhöht sich auch die zu beschleunigende Ankermasse (**m~d³**), wenn man von einer Konstanz der Ankerproportionen ausgeht (**Länge~d**).
- Durch Miniaturisierung wird der Anker eines E-Magnet zwar schneller beschleunigt, allerdings reduzieren sich damit auch die möglichen statischen Kräfte an der Wirkstelle. Um das Papier zu prägen, ist jedoch eine Mindestkraft (bzw. minimale Umform-Energie) erforderlich.
- Wahrscheinlich existiert für eine vorgegebene Wirkstelle jeweils eine Grenze für die minimal mögliche Zykluszeit, die prinzipiell mit einem E-Magneten nicht unterschritten werden kann!

Um eine Entscheidung zu treffen, ob ein E-Magnet überhaupt als Aktor für den zu konstruierenden Antrieb geeignet ist, sollte ermittelt werden, wie schnell ein E-Magnet unter diesen Bedingungen prinzipbedingt arbeiten kann. Dafür kann man die numerische Optimierung nutzen, wenn man zuvor geeignete numerische Modelle des Antriebs entwickelt.

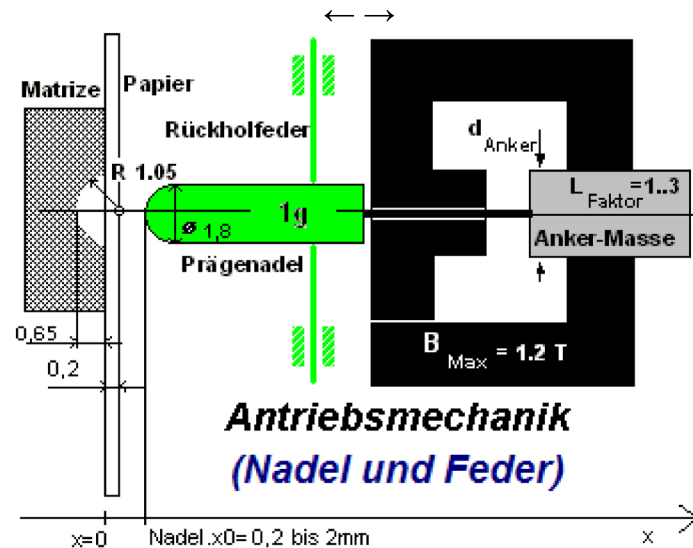
Hinweis:

Man kann die Geometrie des E-Magneten nicht unabhängig von den anderen Komponenten des Antriebs optimieren. Jede Veränderung am E-Magneten erfordert zumindest eine Anpassung der Rückholfeder und der Einschaltzeit des Magneten!

Software: SimX - Nadelantrieb - Wirkprinzip - Nadel

Aus OptiYummy

↑

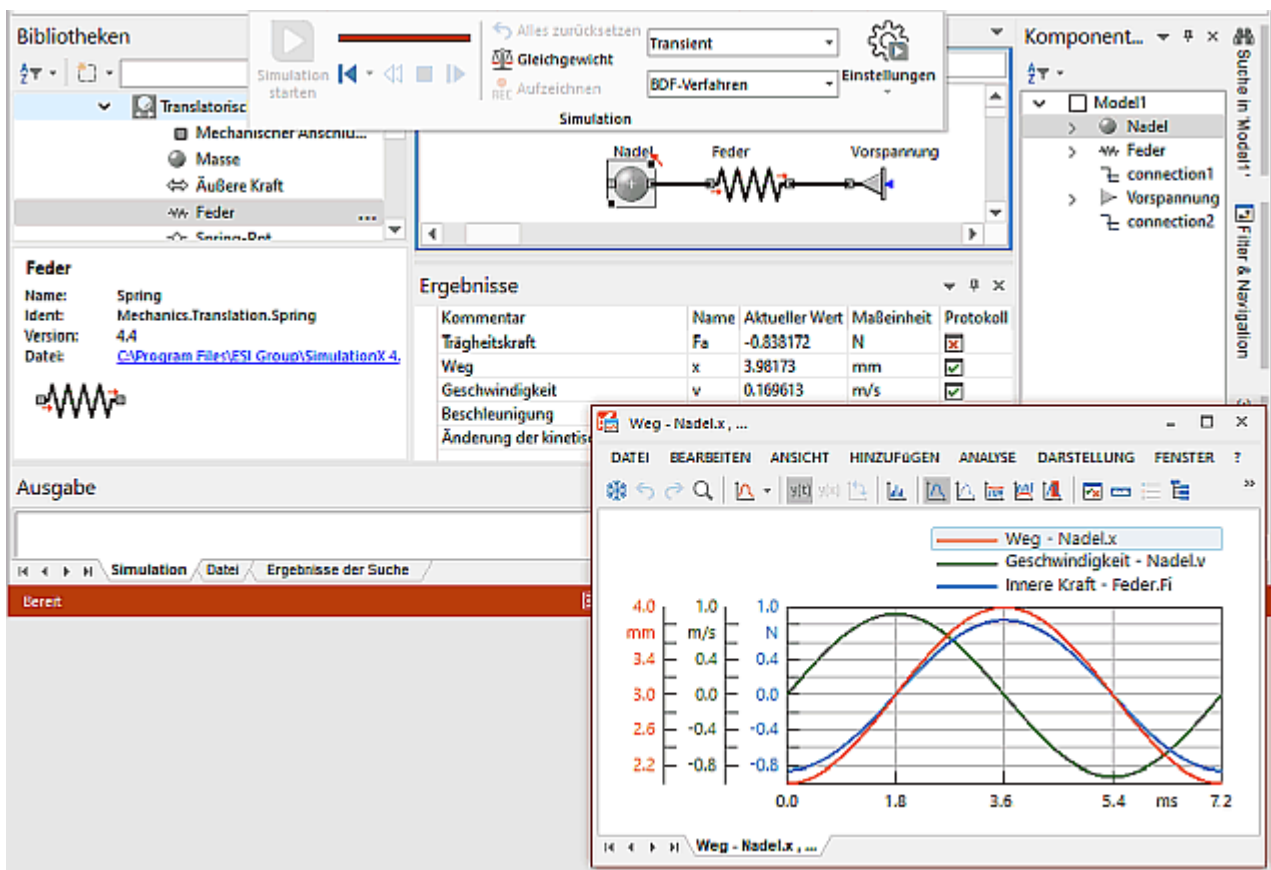


Um ohne Verzögerung die betreute Übungszeit zur ersten Etappe nutzen zu können, sollte ein separates Skript zur Vorbereitung abgearbeitet werden:

- In diesem Skript wurde erläutert, in welchen Abschnitt des konstruktiven Entwicklungsprozesses für einen Blindenschriftträger sich unsere Übungsetappen einordnen.
- Es wurde beschrieben, wie die erforderliche Software auf dem privaten Computer installiert und in Betrieb genommen werden kann.
- Am Beispiel der Antriebsmechanik (Wechselwirkung von Nadelmasse und Rückholfeder) erfolgten erste Schritte zur Einarbeitung in das Programm *SimulationX*.
- Anhand der Parametrisierung des Feder-Masse-Modells entwickelten wir ein Gefühl für die Zeitabläufe und die Größenordnungen von beschleunigter Prägenadel-Masse und wirksamer Rückholfeder innerhalb des Blindenschriftträgers.

Datei > Öffnen > Etappe1_xx.isx → die während der Vorbereitung entstandene individuelle Modelldatei dient nun als Grundlage für die weitere Modellentwicklung:

- Das konfigurierte Ergebnisfenster sollte für die Nadel-Bewegung und die Feder-Kraft die Schwingungsperiode von **7,2 ms** zeigen.
- Es sollte überprüft werden, ob die Simulation noch funktioniert (Zurücksetzen und erneut Starten):



Beachte:

- Voraussetzung für die nachfolgenden Schritte ist die vollständige Abarbeitung der Anleitung zur **"Vorbereitung der Übung"**
- Bei fehlender Vorbereitung (unabhängig von den Gründen) sollte der Einstieg in diese Anleitung zur Vermeidung von Zeitverzug sofort bei der Modellbildung erfolgen:
 1. **Wichtig:** Nutzung der teilnehmer-spezifischen Modelldatei!
 2. Antriebsmechanik (Präge-Nadel mit Rückholfeder) zur Einarbeitung in *SimulationX*
- Die einleitenden Abschnitte der **Übungsvorbereitung** sind später außerhalb der offiziellen Übungszeit zu studieren!

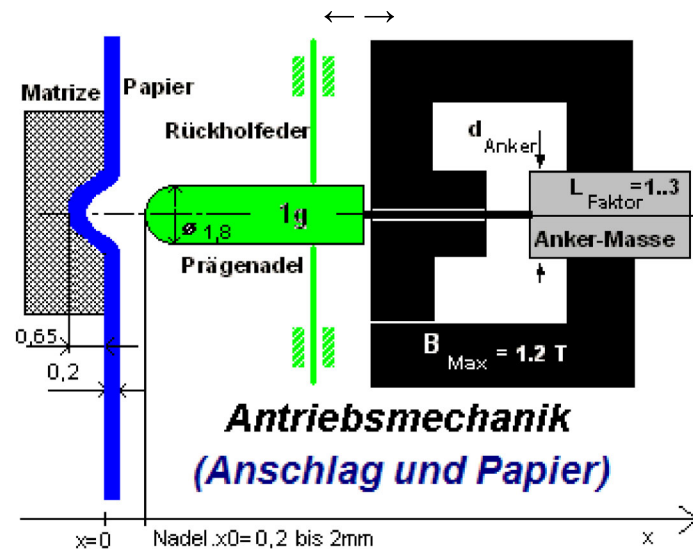
← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Nadel&oldid=27739“

Software: SimX - Nadelantrieb - Wirkprinzip - Papier

Aus OptiYummy

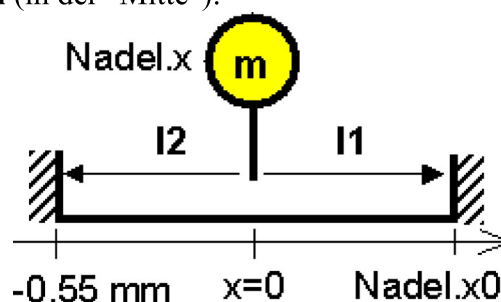
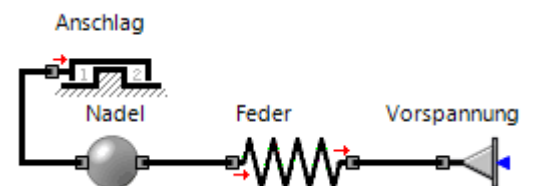
↑



Nadel-Anschlag

Die Bewegung der Nadel wird durch die Anschläge am Matrizenboden und in der Ruhelage begrenzt:

- Das Gestell der Antriebsbaugruppe bildet hierbei die neutrale Bezugsposition für beide Anschlag-Seiten ($x=0$ mm).
- Vereinfacht kann man die Stoßvorgänge an beiden Seiten als ideal plastisch an starren Anschlägen betrachten (nach dem Stoß \rightarrow $Nadel.v=0$ m/s).
- In der Modell-Bibliothek stehen für die "Translatorische Mechanik (1D)" zwei Elementtypen für "Anschläge" zur Verfügung:
 1. Der allgemeine "Anschlag", welcher auch eine veränderliche neutrale Position ermöglicht (relativer Anschlag zwischen zwei beweglichen Elementen),
 2. Der vereinfachte "Anschlag gg. Absolut", welche nur für eine fixe neutrale Position genutzt werden kann (absoluter Anschlag eines beweglichen Elementes an ein feststehendes Element).
- Wir wählen das vereinfachte Anschlagselement, da es numerisch robuster und schneller arbeitet als der relative Anschlag.
- Wir konfigurieren das Anschlagselement als "Starren Anschlag" mit "Plastischem Stoß" ohne "Reibung":
 - Nach dem Prägen des Papiers befindet sich zwischen Nadelspitze und Matrizenboden eine Papierschicht von ca. **0,1 mm** Dicke. Dies entspricht einem Anschlag bei **-0,55 mm** auf dieser Seite.
 - Die Parameter **I1** und **I2** beschreiben die **Abstände** der beiden Anschlagpositionen in Bezug auf die neutrale Position $x=0$ mm (in der "Mitte"):

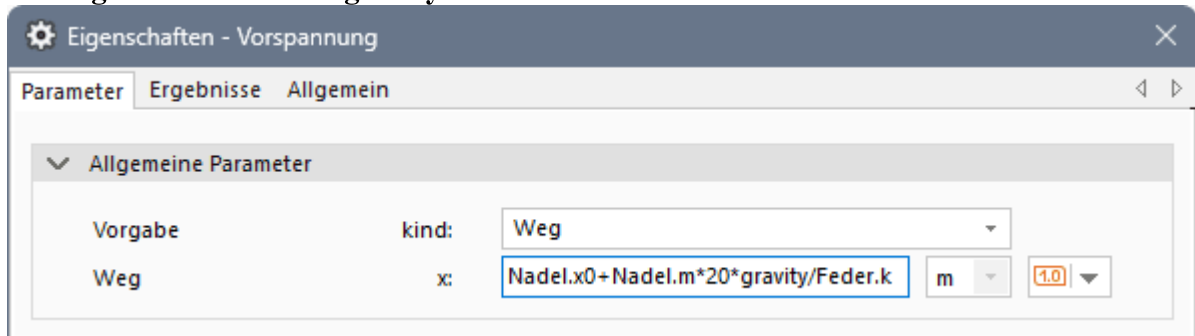


- Die Zeit (**time**) für das Erreichen des Minimum ist abhängig von der konkreten Federsteife (**Feder.k** resultierte im Beispiel noch aus einem tZyklus=3,4 ms).

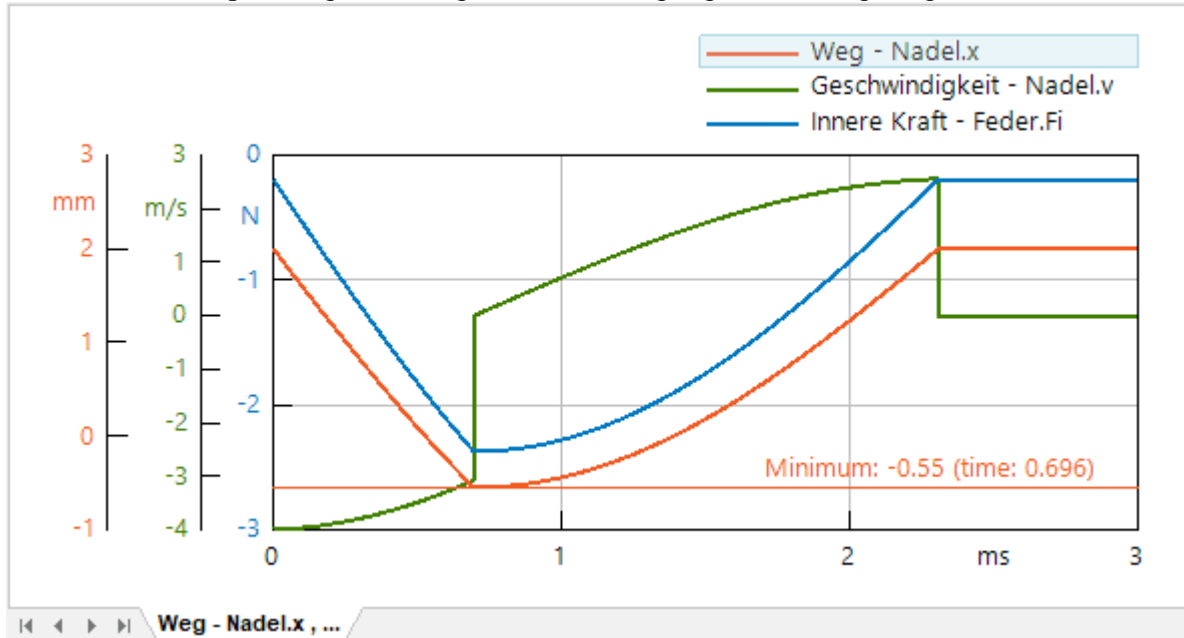
Vorspannung der Feder in der Ruhelage der Nadel

Bisher wurde zum Testen des Bewegungsablaufes der Prägenadel die Rückholfeder um einen konstanten Wert von **1 mm** in der Ruhelage der Nadel vorgespannt. Soll unabhängig von der bewegten Masse und der Steife der Rückholfeder eine Stoßfestigkeit von **20g** (20-fache Erdbeschleunigung) für das Verbleiben der Nadel in der Ruheposition erreicht werden, ist eine Anpassung an die aktuelle Konfiguration des Feder-Masse-Systems erforderlich:

- In Ruhe muss die Haltekraft der vorgespannten Feder dem 20-fachen Gewicht der bewegten Masse entsprechen, wenn ein Betrieb in horizontaler Lage angestrebt wird.
- Aus $20 \cdot g \cdot \text{Nadel.m} = dx \cdot \text{Feder.k}$ (mit $dx = \text{Vorspannung.x} - \text{Nadel.x0}$ und $1g = \text{gravity}$) → **Vorspannung.x = Nadel.m * 20 * gravity / Feder.k**:



- Vor der Simulation mit der aktuellen Vorspannung muss man für die Feder die Protokollierung der Wegdifferenz **Feder.dx** aktivieren. Man kann dann nach der Simulation für die wieder eingenommene Ruhelage der Nadel den berechneten Vorspannweg der Feder als Wert von **Feder.dx** ablesen (im Beispiel ca. **0.2 mm**).
- Der Einfluss dieser Vorspannungsänderung auf den Bewegungsablauf ist gering:



Papier

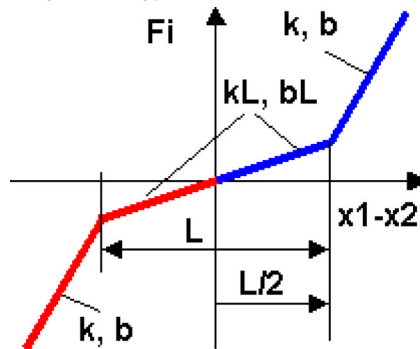
Das zu prägende Papier als Wirkstelle muss im Modell berücksichtigt werden, um Aussagen zur Eignung eines Wirkprinzips für den Aktor machen zu können. Im Beispiel wurden aus Messungen beim Prägevorgang folgende vereinfachte Zusammenhänge abgeleitet:

- Die Kraftwirkung des Papiers auf die Nadel ist Null, solange die Nadel die Papieroberfläche noch nicht berührt ($\text{Nadel.x} \geq 0.2 \text{ mm}$).
- Während des Eindringens verhält sich die Papieroberfläche wie ein Feder-Dämpfer-Element:

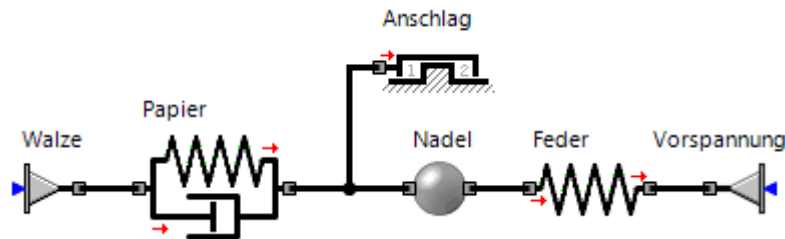
- $k = 36500 \text{ N/m}$
- $b = 5.5 \text{ Ns/m}$
- Nach Erreichen der Rissposition $x_{\text{Riss}} = -0.39 \text{ mm}$ entfällt der elastische Anteil der Kraft (Prägung durch Zerfasern des Papiers). Beim weiteren Eindringen wirkt nur noch der Dämpfungsanteil mit $b=5.5 \text{ Ns/m}$.
- Bei der Rückbewegung der Nadel existiert kein Dämpfungsanteil der Kraft.
- Wir benutzen dafür ein **Feder-Dämpfer-Spiel-Element** aus der Bibliothek.

Diese ziemlich komplizierte Abhängigkeit der Kraft von der Nadelposition, der Bewegungsrichtung der Nadel und vom Zustand des Papiers wollen wir schrittweise nachbilden:

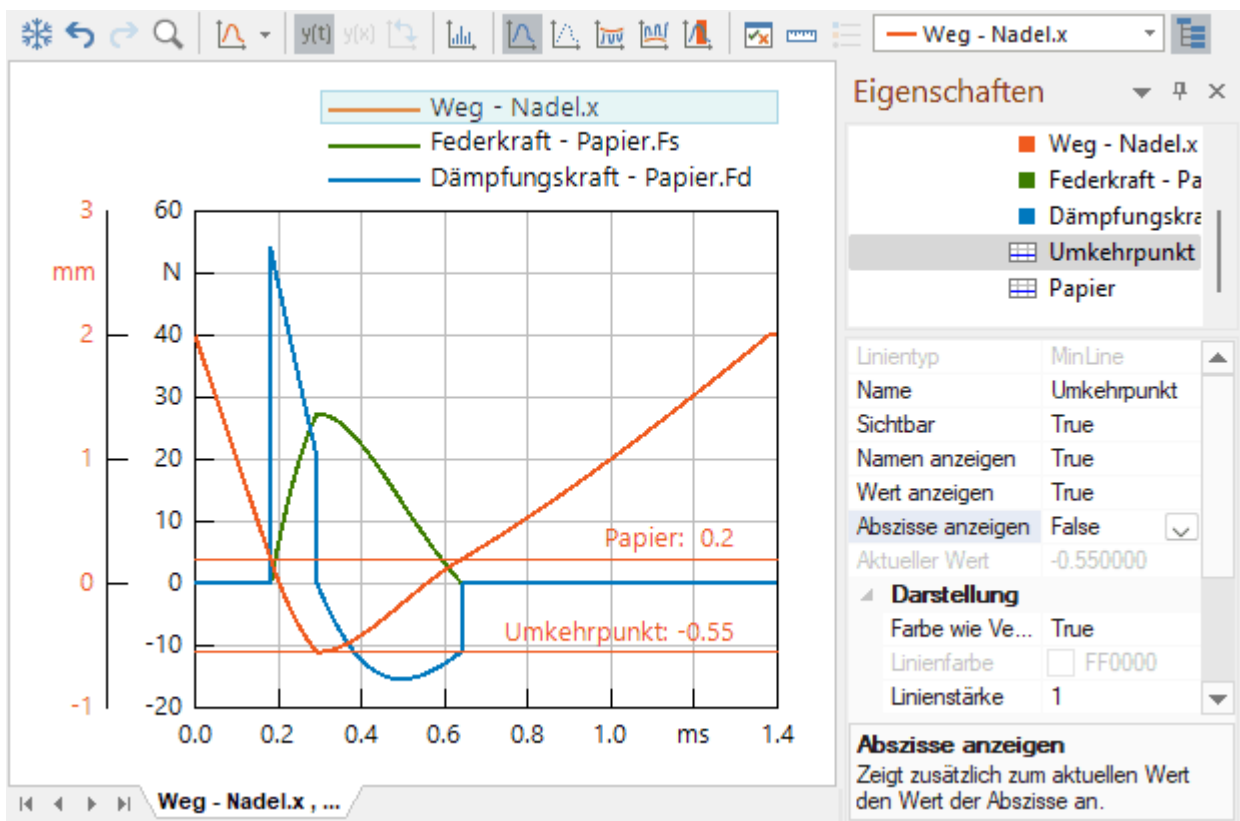
- Der Elementtyp "Feder-Dämpfer-Spiel" gestattet ähnlich wie der "Anschlag" die Modellierung eines Kraftelements mit einem Spielanteil (kräftefreie Nadelbewegung bis zur Papieroberfläche).
- Wir verwenden die Kraft-Kennlinie dieses Elementtyps als "Papier" und nutzen davon aber nur den blauen Bereich rechts von der Kennlinien-Mitte:



- Damit die Anfangsposition **Nadel.x0** sich auf der Kennlinie bei $(x1-x2)=0$ befindet, muss man die Walzen-Seite vom "Papier" ebenfalls mit der Position *Nadel.x0* belegen. Für die "Walze" auf der Seite von $x1$ des Papiers verwenden wir wieder ein Vorgabe-Element aus der Mechanik-Bibliothek mit der Funktion "Weg"-Vorgabe x : **Nadel.x0**:



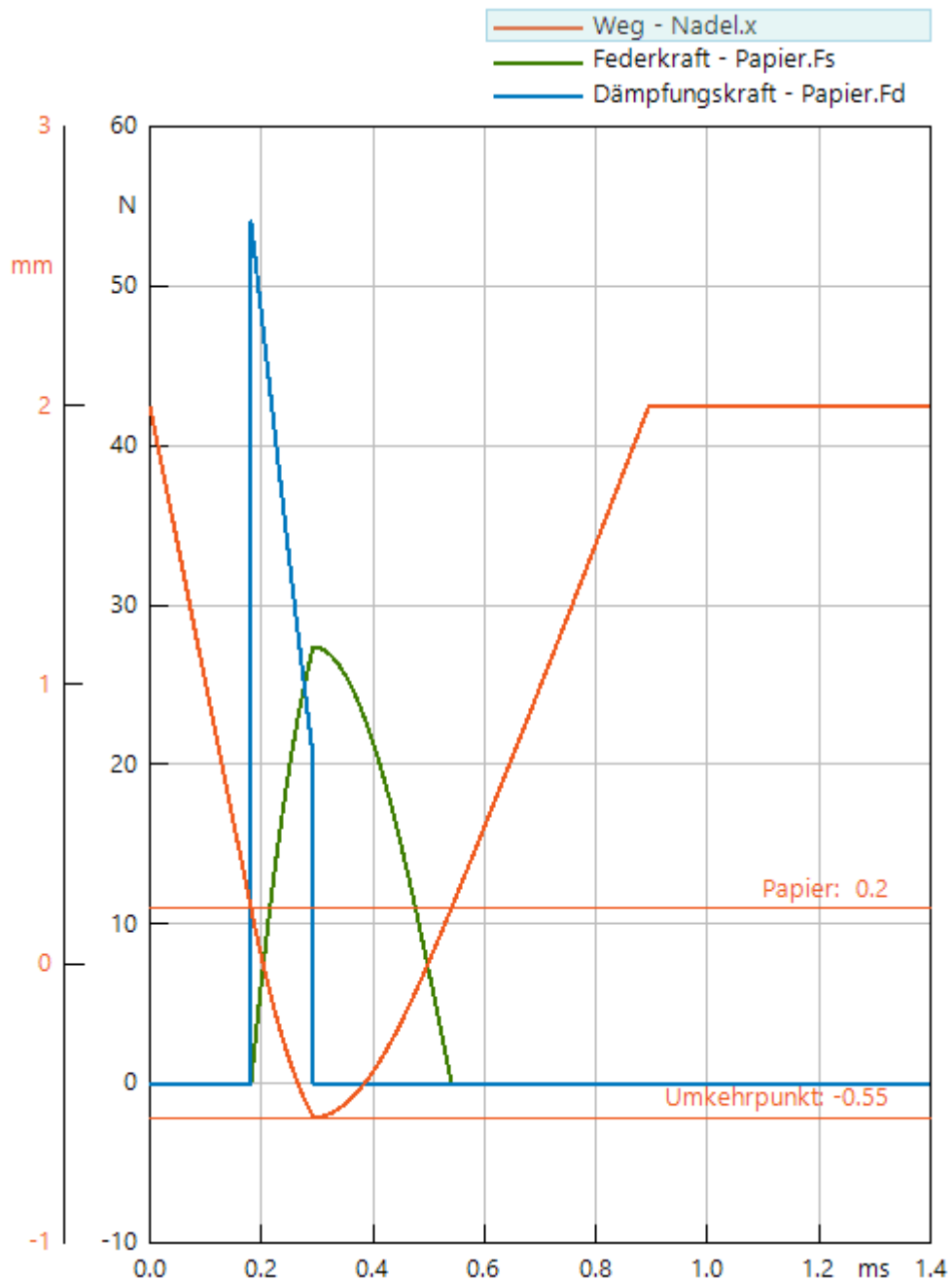
- Wir konfigurieren das Papier zuerst ohne Berücksichtigung der Bewegungsrichtung der Nadel und des Papierzustands:
 - $k = 36500 \text{ N/m}$ (Steifigkeit bei Anlage);
 - $b = 5.5 \text{ Ns/m}$ (Dämpfung bei Anlage);
 - $kL = 0 \text{ N/m}$ (Steifigkeit im Spiel);
 - $bL = 0 \text{ Ns/m}$ (Dämpfung im Spiel);
 - $L = 2 * (\text{Nadel.x0} - 0.0002)$ (Spiel); ← mit *Papierdicke=0.2 mm*
 - **Anfangswerte** für Feder und Dämpfer **ohne Vorspannung**;
- Mit der Simulation sollte man dann sofort den richtigen Verlauf der vom Papier auf die Nadel ausgeübten Kraft überprüfen. Dazu "schleudert" man die Nadel wieder hinreichend kräftig in Richtung Papier (im Beispiel **Nadel.v0=-10 m/s**).
- Wir kontrollieren die relevanten Signalverläufe in einem neuen Ergebnisfenster, welches wir günstig konfigurieren:



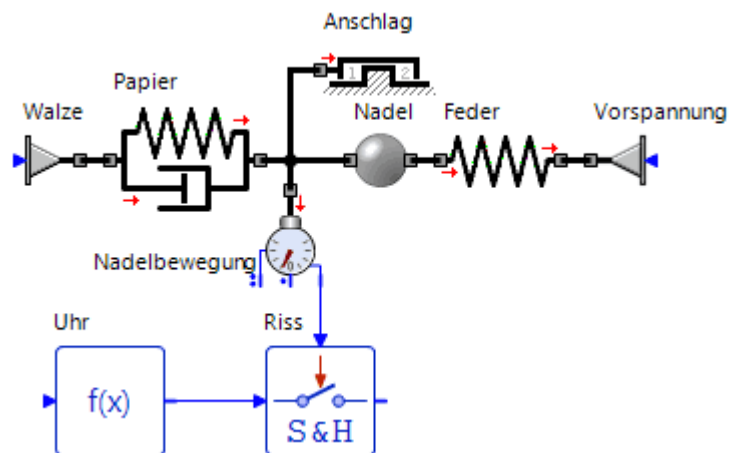
- Die Dämpfungskraft $F_d = b \cdot v$ ist eine Funktion der Eindring-Geschwindigkeit **Nadel.v**.
 - Die Federkraft $F_s = k \cdot x$ ist eine Funktion der Eindringtiefe **Nadel.x**.
 - **Hinzufügen horizontaler Linien** für markante x-Positionen erleichtert die Auswertung der berechneten Kraftverläufe:
 1. "**Minimum**" mit anschließendem Umbenennen des Namens und Ausblenden des Abszissenwertes zeigt die Position des Nadel-Umkehrpunktes (vollständiges Prägen bei **-0,55** mm).
 2. "**Benutzerdefiniert**" mit dem Namen **Papier** und dem Wert **0.2** zeigt die Position der Papieroberfläche.
 - Befindet sich die Nadel oberhalb der Papieroberfläche, darf das Papier keine Kraft auf die Nadel ausüben!
-
- Wenn das Modellverhalten den Erwartungen entspricht, fügen wir nun die Richtungsabhängigkeit der Dämpfungskraft hinzu:

$$b = 5.5 \cdot 0.5 \cdot (1 - \text{sign}(\text{Nadel.v}))$$

Die **sign**-Funktion kann nur die Werte **(-1,0,+1)** annehmen. Die Dämpfung soll Null sein, wenn sich die Nadel in Richtung der Ruhelage bewegt:

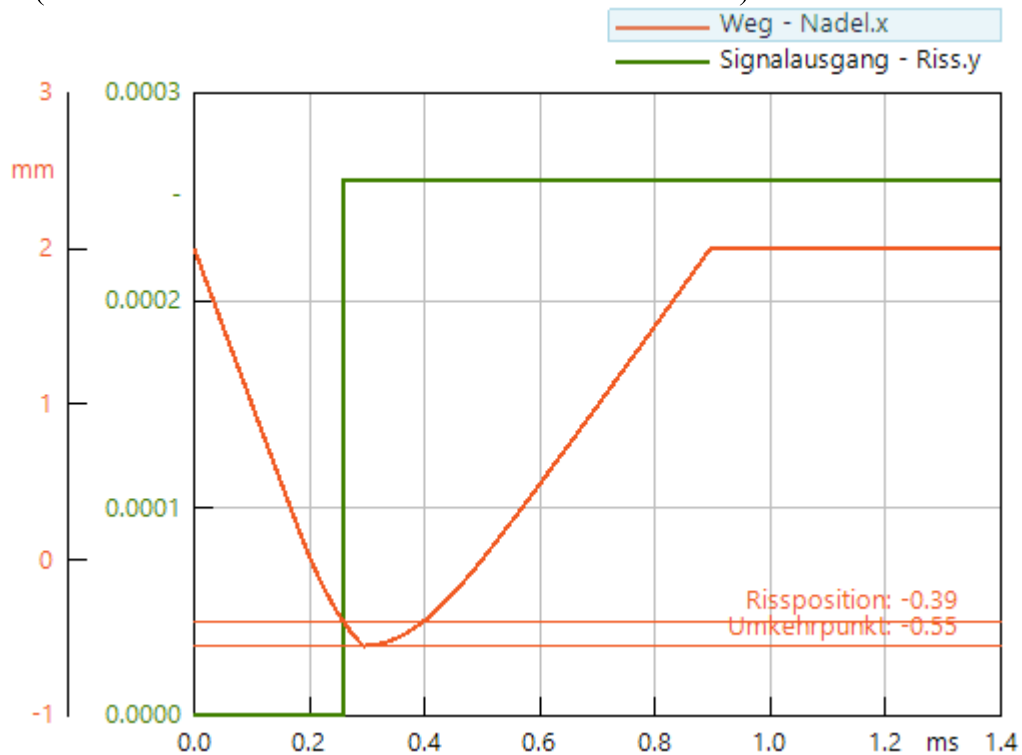


- Schwieriger ist die Berücksichtigung des Papierzustandes (glatt bzw. geprägt). Um diesen zu erfassen, muss man einen sogenannten "Ereignisbeobachter" installieren. Dieser überwacht mit Hilfe eines mechanischen "*Bewegungssensors*" die "Nadelbewegung". Erreicht die Nadel die Rissposition von **-0.39 mm**, so wird dies als Ereignis "Riss" registriert. Es hat sich als günstig erwiesen, dabei gleich den Ereignis-Zeitpunkt zu erfassen:



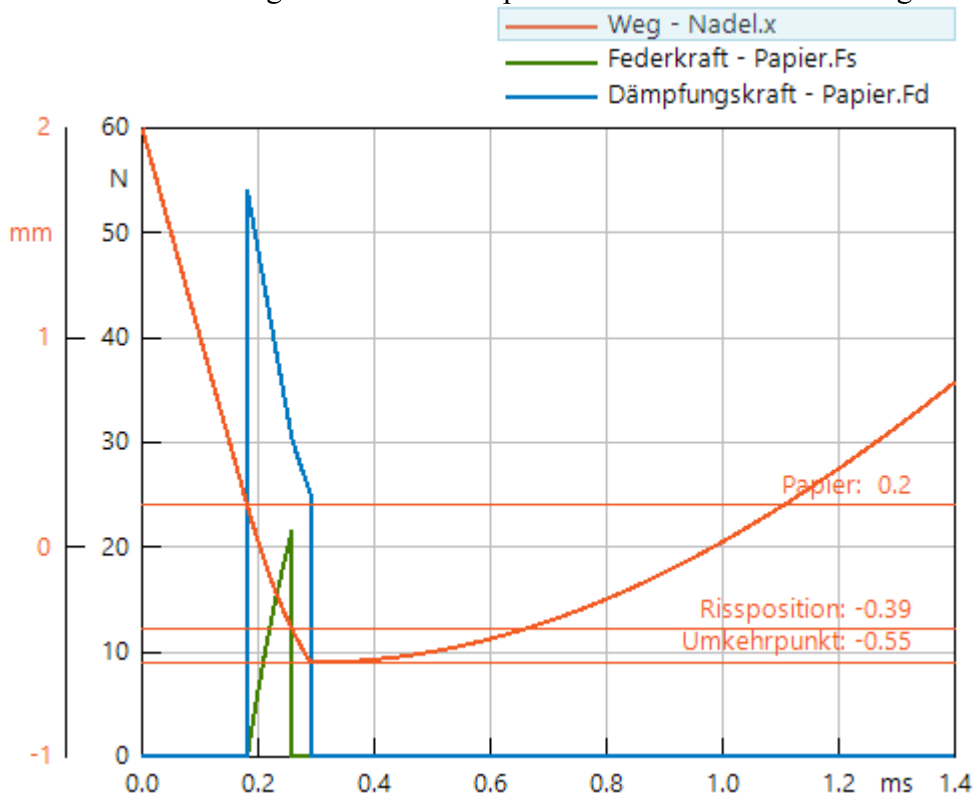
- Die "Riss"-Registrierung realisiert das *spezielle Signalglied* "*Ereignisgesteuertes Abtastglied*".
- Als "Uhr" kann das Signalglied $f(x)$ verwendet werden (Funktionswert gleich aktuelle Simulationszeit **time**).
- Die Ereigniserkennung ist numerisch recht anspruchsvoll. Es darf nur das 1. Ereignis "Nadel erreicht Rissposition" registriert werden (danach ist das Papier "zerrissen").

- Als Ereignis soll nur die Bewegungsrichtung "von oben nach unten" registriert werden (von größeren x-Werten kommend).
- Der **Grenzwert $a=-0.00039$ [m]** definiert die Position der Nadelspitze, wo das Ereignis stattfindet.
- Der Anfangswert für die Ausgangsbelegung ist **$y_0=0$** .
- Wir überprüfen mit der Simulation in einem neuen Ergebnisfenster, ob das Ereignis ordnungsgemäß registriert wird (zusätzliche benutzerdefinierte horizontale Linie= -0.39):



- Danach beziehen wir den Zustand des Papiers in die Kraftwirkung des Papier-Elements mit ein:

$$k=36500*(1-\text{sign}(\text{Riss.y}))$$
- Da die Zeit **time** nicht negativ wird, kann die **sign**-Funktion hier nur die Werte 0 oder 1 annehmen!
- Wir überprüfen dann den vollständigen Verlauf der Papierkraft auf seine Glaubwürdigkeit:



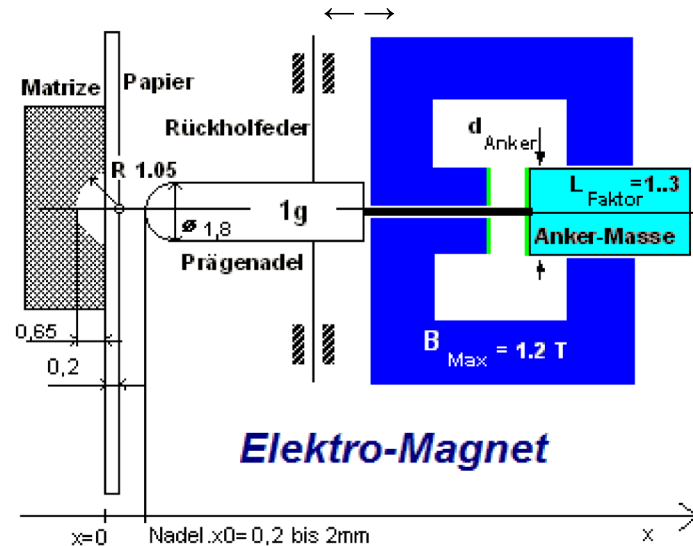
- **Hinweis:** Falls sich das Modell der Antriebsmechanik einschließlich der Wirkstelle glaubwürdig und hinreichend richtig verhält, haben wir die erste Hürde überwunden! Nun können wir uns der eigentlichen Fragestellung dieser Entwurfsetappe widmen: "Ist ein E-Magnet als elektro-mechanischer Wandler geeignet?"

← →

Software: SimX - Nadelantrieb - Wirkprinzip - Elektromagnet

Aus OptiYummy

↑

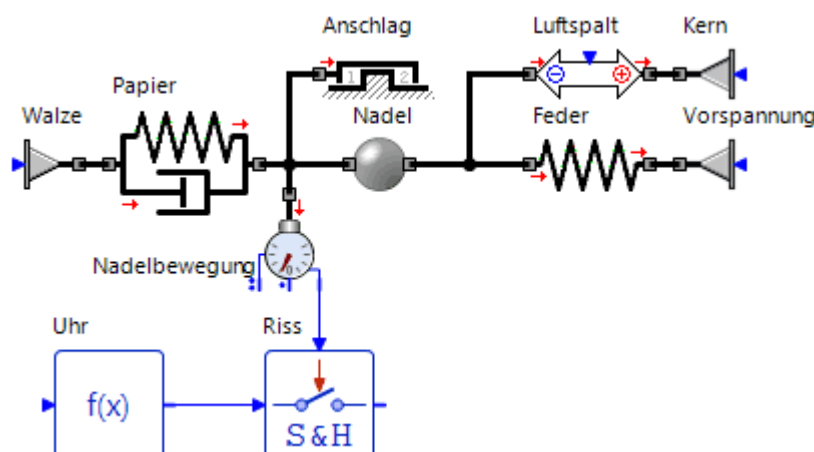


Bisher haben wir die Prägenadel in Ermangelung eines Magnetmodells "persönlich" gegen das Papier geschleudert. Das wird sich nun ändern:

- In der ersten Etappe soll nur ein stark vereinfachtes Verhaltensmodell des Elektro-Magneten erstellt werden, für das kein vertieftes Spezialwissen erforderlich ist.
- Bereitgestellt werden soll für die Antriebsmechanik ein qualitativ ausreichender Magnetkraft-Verlauf.
- Die mögliche Amplitude F_{max} der Magnetkraft im Arbeitsluftspalt wird aus dem Ankerdurchmesser d_{Anker} und der magnetischen Sättigungsflussdichte B_{max} des Eisenmaterials berechnet:

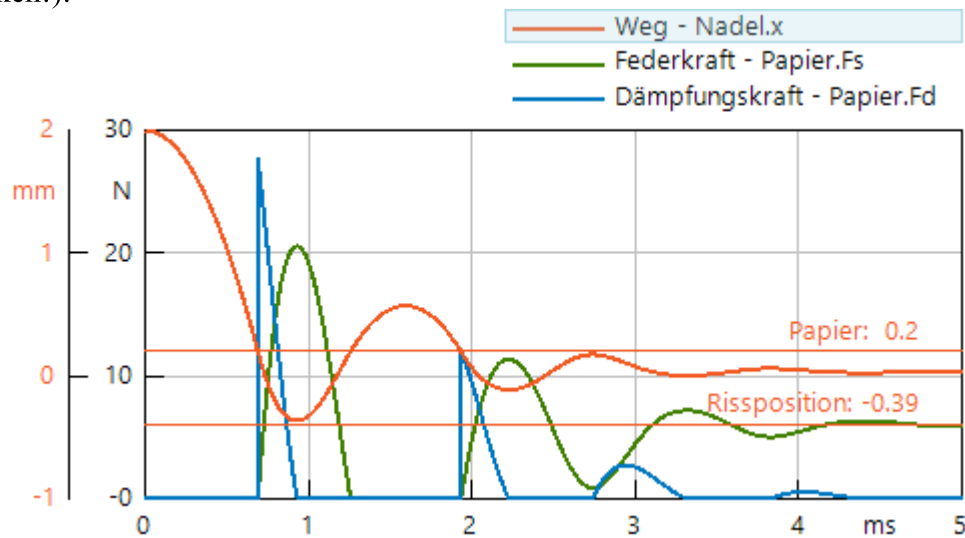
$$F_{max} = \frac{B_{max}^2 \cdot A_{Anker}}{2 \cdot \mu_0}$$

Luftspalt zwischen Anker+Kern

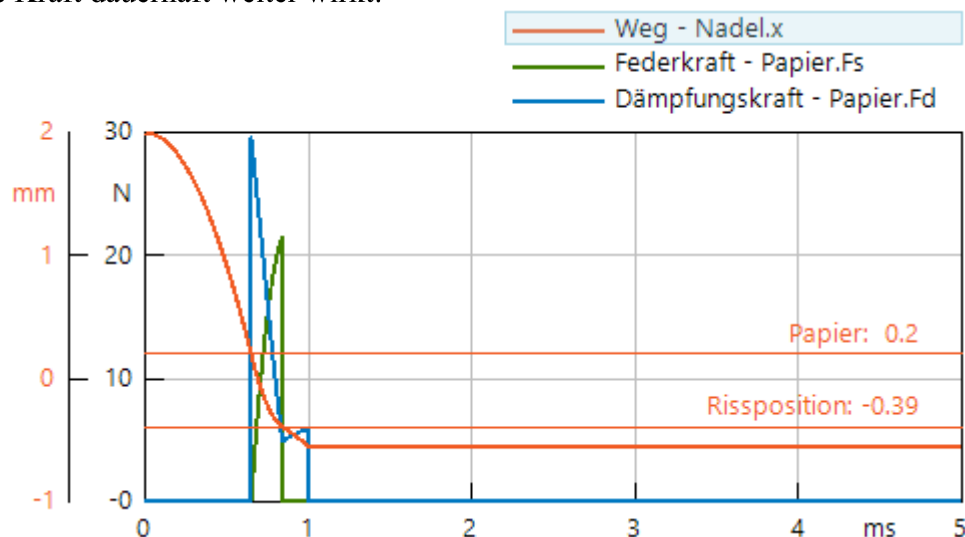


- Die Prägenadel ist starr mit dem beweglichen Anker des Magneten verbunden.
- Der Magnetanker wird durch die Kraft auf die Trennflächen des Arbeitsluftspaltes angetrieben.
- Zur Nachbildung des Luftspalts nutzen wir den Element-Typ "Äußere Kraft".

- Der Kern wird repräsentiert durch eine Einspannung mit dem Typ "Wegvorgabe". Damit der Luftspalt $dx=0$ mm erreicht, wenn die Nadelspitze das Papier komplett geprägt hat, muss man $x=-0,55$ mm für den Kern vorgeben:
- Zum Test der Funktionsfähigkeit des Luftspalts verwenden wir darin als Parameter eine konstante Kraft auf die am Anfang ruhende Nadel (**Nadel.v0=0 m/s!**). Diese Kraft muss größer sein, als die Haltekraft infolge der vorgespannten Feder, damit sich die Nadel bewegt!
- Ist die Kraft zu klein, um das Papier zu zerreißen, so wird dieses einige Male als "Trampolin" benutzt, bevor die Nadel darauf zur Ruhe kommt (falls die Kraft überhaupt ausreichend war, die Nadel bis zum Papier zu drücken!):

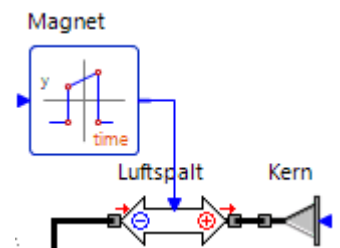


- Erst bei hinreichend großer Kraft wird das Papier zerrissen und die Nadel bleibt am Boden der Matrize liegen, weil die Kraft dauerhaft weiter wirkt:



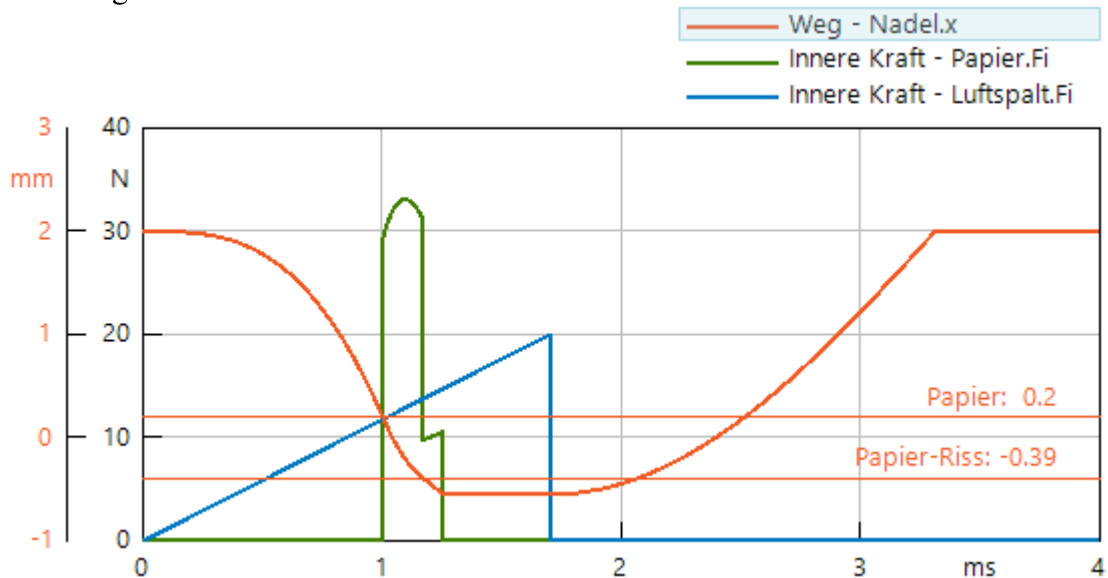
Magnetkraft-Verlauf

- Auch mit wenigen Spezialkenntnissen zum Elektromagneten weiß man, dass die Magnetkraft sich nach dem Einschalten von Null beginnend stetig aufbaut. Ursache ist die Induktivität der Spule, welche den Stromanstieg verzögert.
- Das Ansteigen der Magnetkraft wird durch die magnetische Sättigung des Eisenkreises begrenzt. Der mögliche Maximalwert ergibt sich entsprechend obiger Formel.
- Nachdem die Nadel ein Grübchen in das Papier geprägt hat, muss der Magnet wieder abgeschaltet werden, damit die Feder die Nadel in die Ruhelage zurückholt.
- Als sinnvolle Näherung für die Magnetkraft kann man einen sägezahnförmigen Kraftimpuls verwenden. Dieser lässt sich mittels einer *Signalquelle - Impulsgenerator* konfigurieren:



SignalBlocks.Sources.ImpulsGenerating Magnet		
Bezugsgröße	RefVar:	Simulationszeit t [s] [-]
Kurvenform	CurveTypeVar:	Dreieck [-]
Anstiegsbreite	R:	0.0018 [s]
Breite	W:	self.R [s]
Höhe	H:	20 [-]
Totbereich	D:	0 [s]
Offset	O:	0 [-]
Folgeabstand	S:	0.1 [s]
Anzahl der Wiederholungen	i:	1 [-]
Alternierendes Vorzeichen	PM:	false [-]

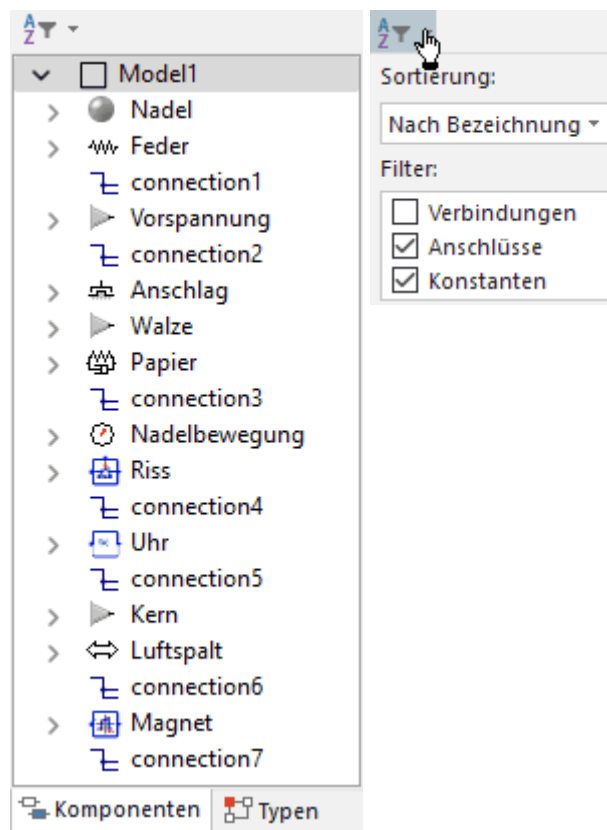
- **Hinweis:** Der konstante Parameterwert im Luftspalt-Element muss durch das Eingangssignal **self.in1** ersetzt werden, damit das Dreieckssignal als Kraftverlauf verwendet wird!
- Zur Validierung des Modellverhaltens wählen wir eine Kraft-Amplitude (im Beispiel: **20 N**), welche einen sicheren Prägezyklus gewährleistet und analysieren die Kraftverläufe in Luftspalt und Papier in einem weiteren Ergebnisfenster:



CAD-Daten

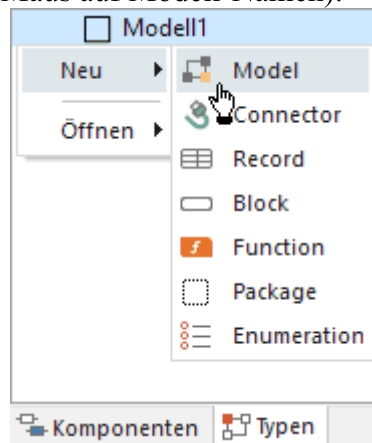
Die Zusammenhänge zwischen den Parametern der konzentrierten Elemente des Antriebmodells und den CAD-Daten der real verwendeten Bauteile bzw. Baugruppen des materiellen Versuchsmusters sollen als neuer Element-Typ "CAD_Data" lokal im Modell definiert werden. Dies erfolgt über den *Modell-Explorer*, welcher unten über zwei Registerkarten verfügt:

1. **Komponenten** = alle Bestandteile der Modellstruktur (Elemente und Connection)
 - Als Bestandteil der Modellstruktur werden standardmäßig auch alle Verbindungen angezeigt.
 - Hinweis: Im Sinne der Übersichtlichkeit sollte man mittels der Filterfunktion z.B. die Verbindungen ausblenden und die Komponenten alphabetisch sortieren.

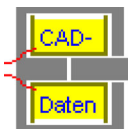
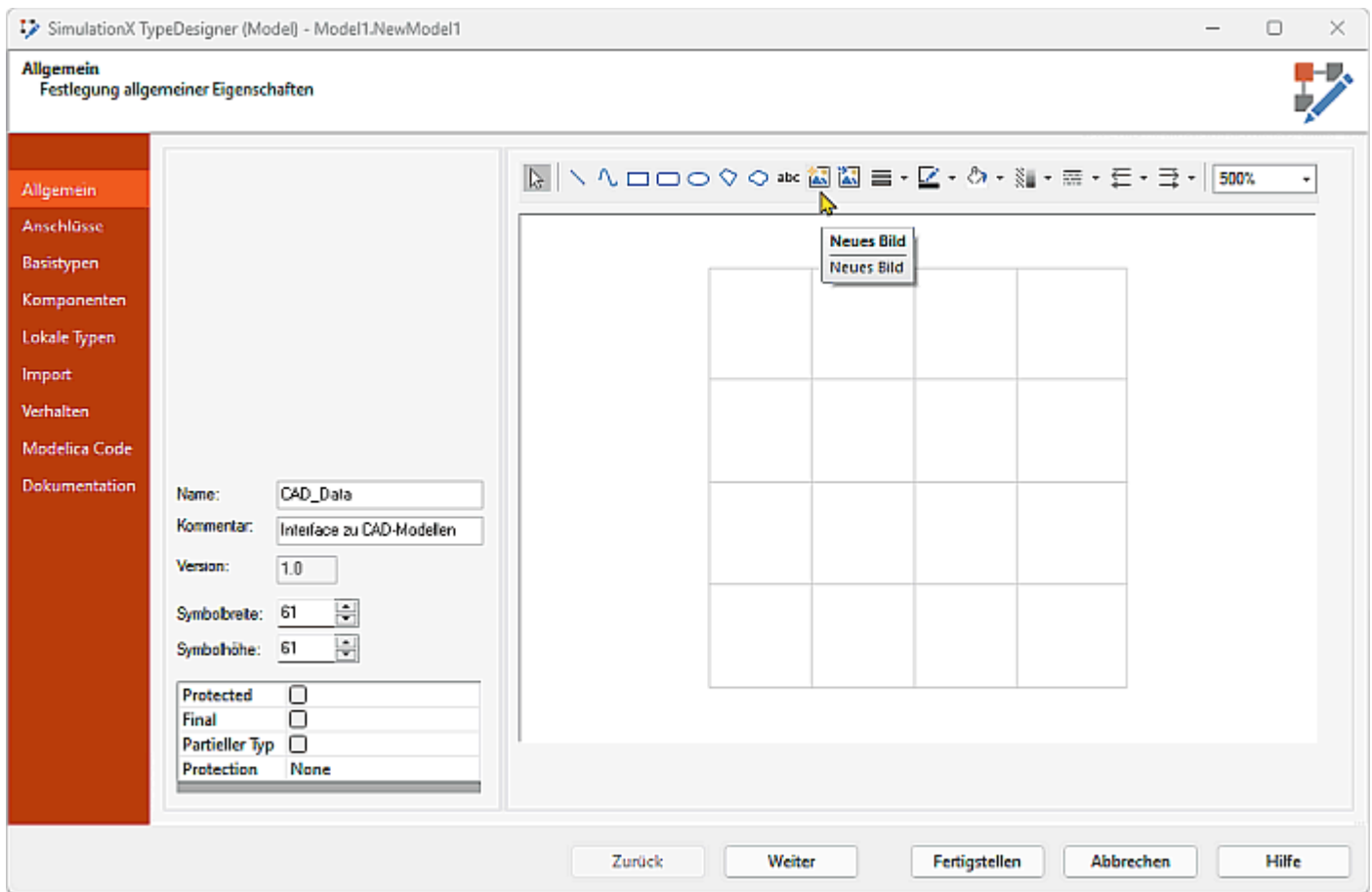


2. **Typen** = lokale Elementtypen analog zu den externen Typen der Modell-Bibliothek

- Der Aufruf des Typ-Designers erfolgt im Modell-Explorer in der Registerkarte "Typen" über das Kontextmenü des Modells (rechte Maus auf Modell-Namen):



- Wir tragen den Namen *CAD_Data* und einen sinnvollen Kommentar ein (z.B. *Interface zu CAD-Modellen*):



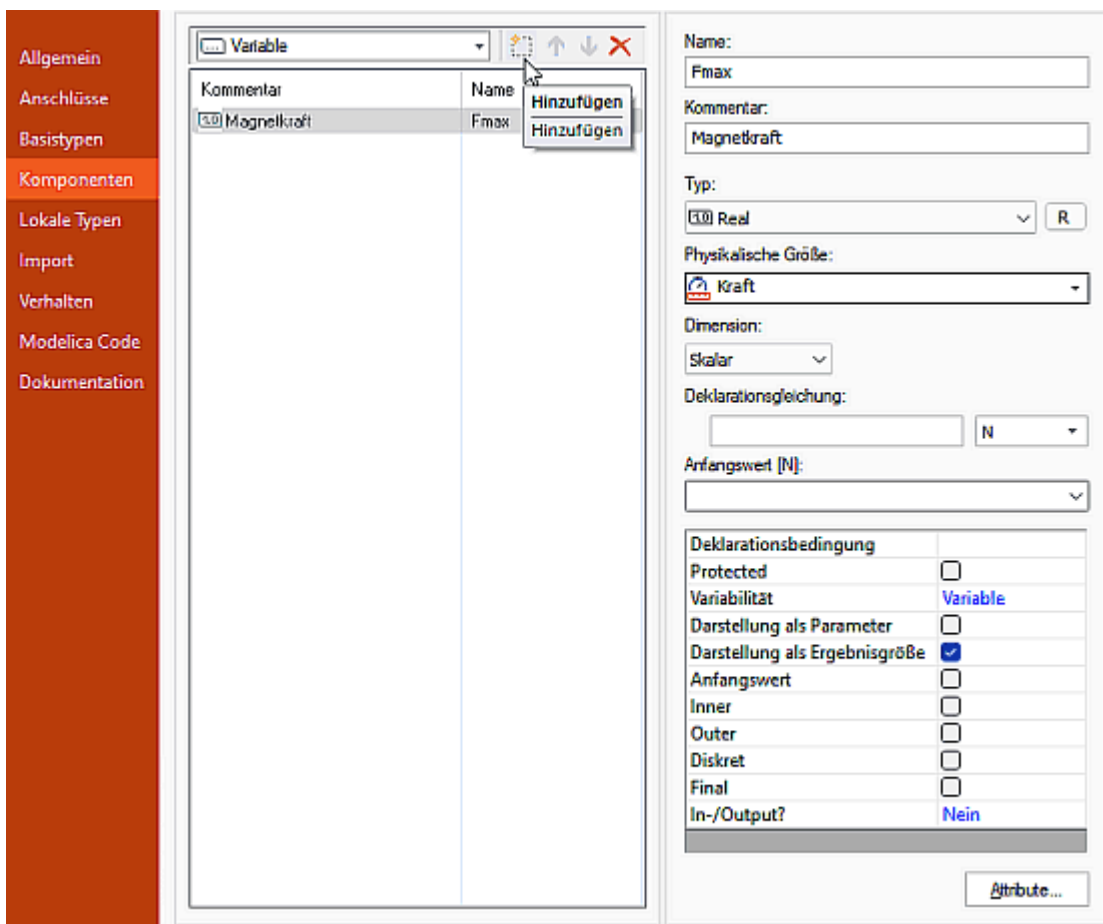
Es soll ein Symbol der Größe **61x61** Pixel definiert werden (z.B. stilisierter Schnitt durch obigen Topfmagnet):

- Nach Einstellung der gewünschten Symbolgröße könnte man im Type-Designer ein **Neues Bild** direkt bearbeiten. Die Farbe "Magenta" wird dabei vom Type-Designer als transparenter Hintergrund interpretiert.
- Komfortabler geht es, wenn man ein anderweitig erstelltes Bild in dieses neue **Bild importiert**, z.B. indem man sich dafür das nebenstehende Bild aus dieser Anleitung zuvor herunterlädt.



Unter **Komponenten** ist die zu berechnende maximal mögliche Kraft **Fmax** als **Variable** zu definieren.

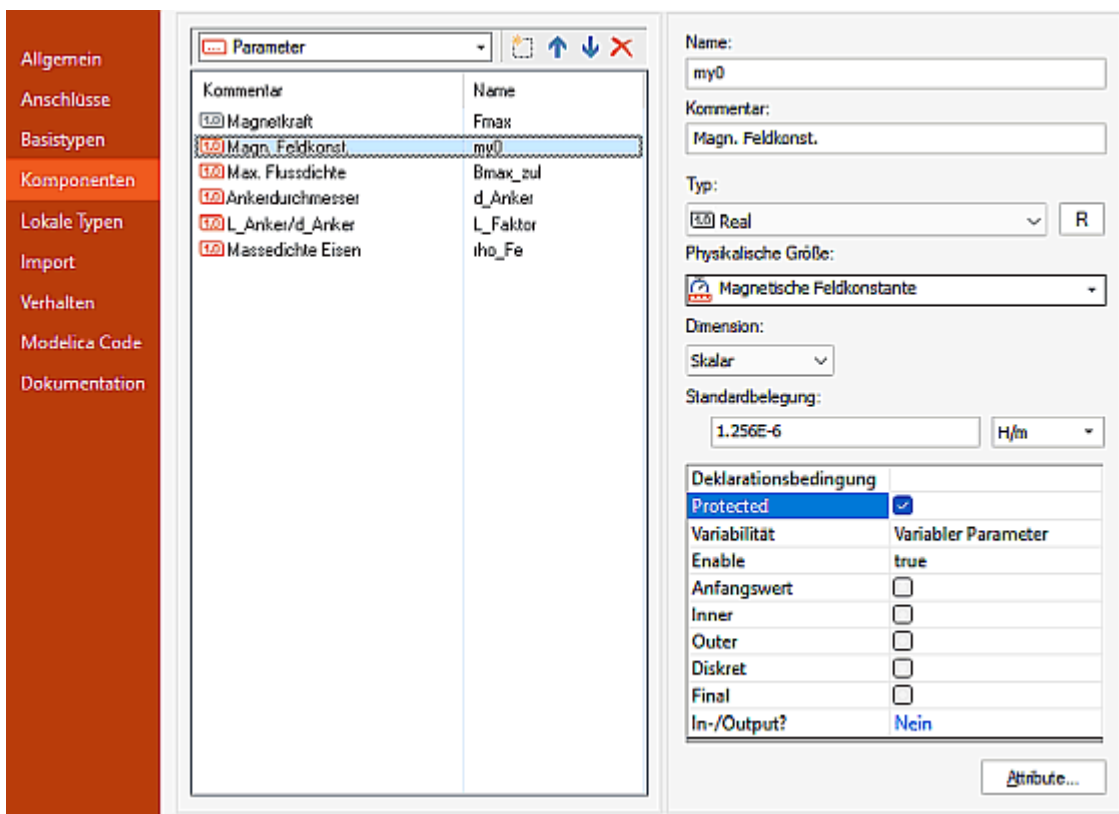
- Als physikalische Größe ist **Fmax** eine mechanische **Kraft**:



Dann definiert man alle erforderlichen **Parameter** mit Name, Kommentar, Maßeinheit und sinnvollem Standardwert:

Name	Kommentar	Wert	Einheit
my0	Magn. Feldkonst.	1.256E-6	H/m
Bmax_zul	Max. Flussdichte	1.2	T
d_Anker	Ankerdurchmesser	10	mm
L_Faktor	L_Anker/d_Anker	1.xx	-
rho_Fe	Massedichte Eisen	7.8	g/cm ³

- Der Längenfaktor wird mit **1.xx** so gewählt, dass ein kurzer, aber technisch noch sinnvoller Anker entsteht (Teilnehmernummer **xx=01..99**).
- Der Wert der Magnetischen Feldkonstante sollte im Eigenschaftsdialog des Elements ausgeblendet und schreibgeschützt werden (entspricht "**Protected**" aktiviert), um einer versehentlichen Veränderung vorzubeugen:

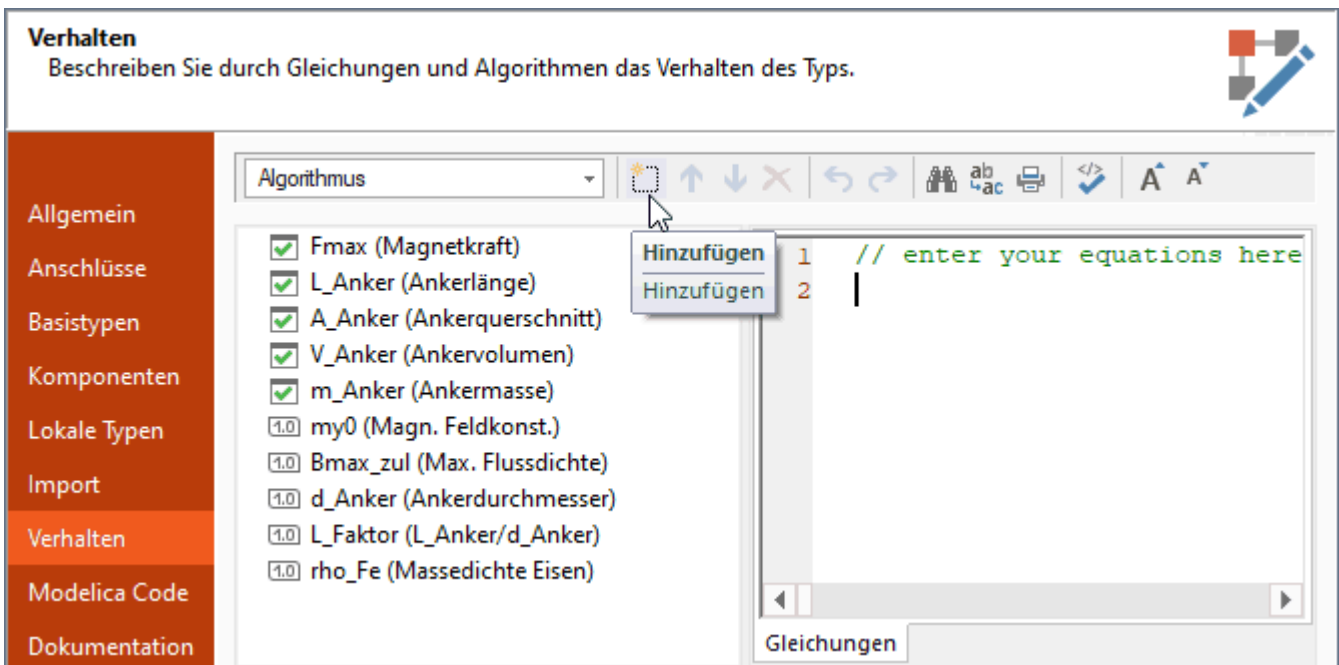


- Außerdem muss man alle zu berechnenden Ergebnis-Größen als **Variable** definieren (für **Fmax** bereits erfolgt!):

Name	Kommentar	phys. Größe	Einheit
L_Anker	Ankerlänge	Länge	mm
A_Anker	Ankerquerschnitt	Fläche	cm ²
V_Anker	Ankervolumen	Volumen	cm ³
m_Anker	Ankermasse	Masse	g
Fmax	Magnetkraft	Kraft	N

Wichtig:

- Falls die SI-Grundeinheiten für den Wert der jeweiligen physikalischen Größe zu unanschaulichen Zahlenwerten führen, sollte man einen geeigneten Präfix wählen (z.B. **cm³** anstatt **m³** für das kleine Ankervolumen). Bei der Belegung der zugehörigen Ergebnisgröße erfolgt dann eine Umrechnung auf die gewählte Größenordnung. Die Maßeinheit der Ergebnisse ist unabhängig davon in Signalfenstern änderbar.
- In unserem Beispiel dürfen obige Variablenwerte nicht als Anfangswert aktiviert werden!
- Für die Variablen darf im Beispiel auf keinen Fall eine Berechnungsformel in das Feld der Deklarationsgleichung eingetragen werden. Dies führt fast immer zu Konflikten mit den Berechnungen im Verhalten-Abschnitt des Modellelements!



Verhalten eines Elementtyps kann durch Algorithmen oder Gleichungen beschrieben werden:

- **Algorithmen** definieren eine Folge von Anweisungen, die innerhalb eines Algorithmus-Abschnittes exakt in der vorgegebenen Reihenfolge abgearbeitet werden. Man ist selbst dafür verantwortlich, dass die verwendeten Operanden zum Zeitpunkt der Benutzung sinnvolle Werte enthalten.

*Syntax: **Variable** := **Ausdruck**;*

- **Gleichungen** (engl. Equation) besitzen eine linke und eine rechte Seite, die durch ein "=" miteinander verknüpft sind. Die Gleichungen werden vor dem Beginn der Simulation automatisch analysiert und die Reihenfolge der Abarbeitung wird innerhalb eines Gleichungsabschnitts intern festgelegt.

*Syntax: **Komponente** = **Ausdruck**;*

Wenn möglich, sollte die Beschreibung des Verhaltens durch Gleichungen erfolgen. Das erhöht die Wahrscheinlichkeit von Optimierungen durch die symbolische Analyse.

- **Syntax** ist hier nur angedeutet. Die Details findet man im Hilfesystem von *SimulationX* und den *Modelica Language Documents*.

Algorithmus als Spezialfall:

- Wir berechnen in unserem CAD-Element ausgehend von den gegebenen Abmessungen und Stoffkonstanten Schritt für Schritt die Parameter der idealisierten Elemente (Punktmasse, Magnetkraft).
- Für diesen Spezialfall sollte im Sinne einer vereinfachten Fehlersuche ein Algorithmenabschnitt benutzt werden.
- Da hierbei die Abarbeitungsreihenfolge der Anweisungen nicht verändert wird, kann man bei falschen Zwischenergebnissen sehr einfach nachvollziehen, bis zu welcher Anweisung noch alles richtig berechnet wurde.
- Algorithmen sind typisch für klassische Programmiersprachen.

Gleichungen als Normalfall:

- Die einzelnen Gleichungen beschreiben Zusammenhänge zwischen physikalischen Größen (Effekte).
- Diese physikalischen Wechselwirkungen sind durch ihre Gleichzeitigkeit gekennzeichnet.
- Die einzelnen Gleichungen wirken deshalb aus Sicht des Nutzers "praktisch" gleichzeitig, es existiert keine Abarbeitungsreihenfolge.
- Gleichungen sind typisch für Modellbeschreibungssprachen.

CAD_Data-Verhalten:

- Da die Modellierung im Normalfall mittels Gleichungen erfolgen sollte, existiert für einen neuen Element-Typ nur ein leerer Abschnitt für "Gleichungen".
- Wir benötigen jedoch einen Algorithmen-Abschnitt und müssen uns einen solchen erst hinzufügen (Siehe obiges Bild).

- In Form eines Algorithmus beschreiben wir die folgenden Zusammenhänge in der richtigen Reihenfolge:

$$F_{max} = \frac{B_{max_zul}^2 \cdot A_{Anker}}{2 \cdot \mu_0}$$

$$m_{Anker} = V_{Anker} \cdot \rho_{Fe}$$

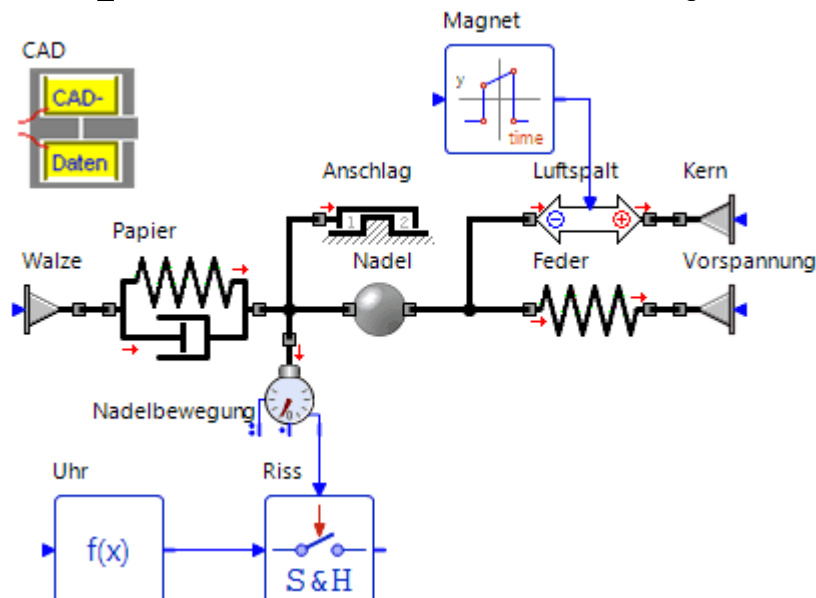
$$L_{Anker} = L_{Faktor} \cdot d_{Anker}$$

$$V_{Anker} = L_{Anker} \cdot A_{Anker}$$

$$A_{Anker} = \frac{\pi}{4} \cdot d_{Anker}^2$$

- Der Algorithmus-Abschnitt wird bei der Simulation immer von oben nach unten durchgerechnet. Es muss gewährleistet sein, dass dabei jede Komponente der rechten Seite einer Anweisung immer einen aktuellen Wert besitzt:

- Die Bezeichner der Komponenten können per *Drag&Drop* aus der Komponenten-Auflistung in die Anweisungen gezogen werden. Alternativ platziert ein Doppelklick einen Komponentenbezeichner auf die aktuelle Cursorposition im Algorithmus-Abschnitt. Jede Anweisung muss mit Semikolon enden.
- Den nicht benötigten Gleichungsabschnitt sollte man löschen. Die Möglichkeit, mehrere Algorithmus- bzw. Gleichungs-Abschnitte zu definieren, soll die Strukturierung des Modell-Elements unterstützen (z.B. getrennte Beschreibung der einzelnen physikalischen Partialsysteme).
- Nach Fertigstellen von *CAD_Data* sichern wir den Arbeitszustand durch Speichern des Modells.

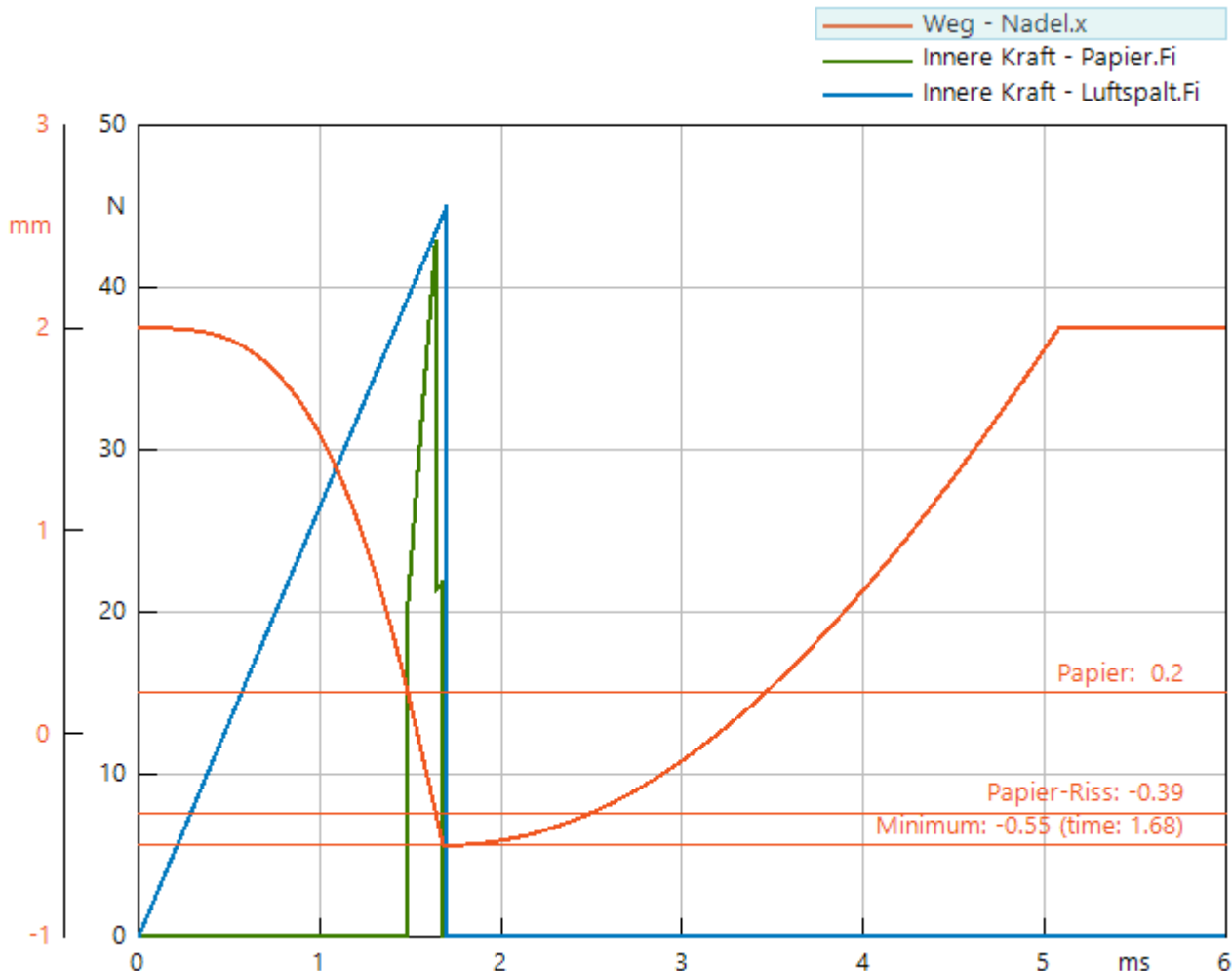


- Den lokalen Typ *CAD_Data* verwenden wir als Element mit der Bezeichnung **CAD** in der Modellstruktur (*Drag&Drop*).

Die Ergebnisse der CAD-Berechnungen müssen im Modell berücksichtigt werden:

- Damit der Wert der max. möglichen Magnetkraft **F_{max}** als Amplitude des Kraft-Impulses verwendet wird, muss man für die Höhe **H** im Impuls-Generator den Namen **CAD.F_{max}** eintragen.
- Die eigentliche Nadelmasse soll **1 g** betragen. Zusätzlich ist mit der Nadel die berechnete Ankermasse **CAD.m_Anker** zu beschleunigen (Summe beider Teilmassen im Parameter **Nadel.m**).

Mit den obigen Anfangswerten müsste der Antrieb sich zumindest qualitativ richtig verhalten. Simulation und Ergebnisfenster sind so zu konfigurieren, dass man einen kompletten Prägezyklus betrachten kann:



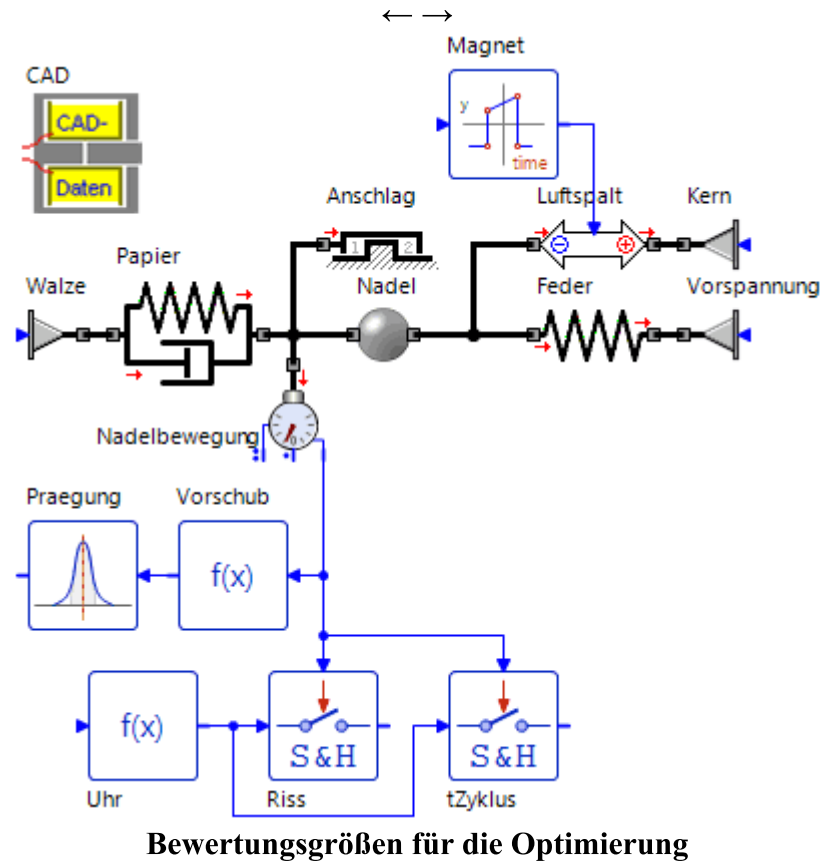
Hinweis zu Lizenzfehler nach Beenden des Typ-Designers (*Express Edition* in Modellen mit 16 Elementen):

- Nach der Verwendung des CAD-Elements im Modell bzw. später auch nach einem Aktualisieren dieses Elements kam es zu einer Fehlermeldung im *SimulationX* ("**Fehler: Die maximale Anzahl der Komponenten wurde erreicht.**").
- Im Ausgabe-Fenster wird dann ein Teil der verwendeten Element-Typen aufgelistet, welche davon angeblich betroffen sind.
- Anscheinend kommt hier das Zählen der in der *Express Edition* zulässigen Elemente durcheinander, weil nicht berücksichtigt wird, dass nur ein Element ersetzt wird.
- Es genügt meist, zuerst ein Element der Modellstruktur anzuklicken und danach die farblich (gelb) hervorgehobene Fehler-Leiste zu schließen.
- Gibt es trotzdem weiterhin diese "Lizenz"-Probleme, muss man das Modell speichern und *SimulationX* beenden. Nach dem erneuten Starten des Programms und Öffnen des Modells funktioniert es wieder bis zum nächsten Beenden des Typ-Designers.

Software: SimX - Nadelantrieb - Wirkprinzip - Bewertung

Aus OptiYummy

↑



Inhaltsverzeichnis

- 1 Vorbetrachtung zur Loesungssuche
- 2 CAD-Daten der Papier- und Matrix-Parameter
- 3 Zeit eines Prägezyklus (tZyklus)
- 4 Max. Nadelvorschub (Praegung)
- 5 Funktionelle Ausgangslösung (kompletter Bewegungszyklus der Nadel)

Vorbetrachtung zur Loesungssuche

Wenn das Modell des Nadel-Antriebs glaubwürdig funktioniert, könnte man nun durch systematisches Verändern der relevanten Modell-Parameter versuchen, eine möglichst günstige Lösung zu finden:

1. Wunsch:

- Realisierung eines möglichst schnellen Prägezyklusses.

2. Veränderbare Parameter:

- CAD.d_Anker (Ankerdurchmesser)
- Feder.k (Federsteife)
- Magnet.R (Einschaltzeit)
- Nadel.x0 (Ruhelage)

3. Einzuhaltende Forderungen:

- Die Nadelspitze muss das Papier komplett prägen, d.h. die Anschlagposition in der Matrize (im Beispiel -0.55 mm) muss erreicht werden.
- Die Zeit für einen Prägezyklus darf maximal 3.6 ms betragen.
- In der Ruhelage darf die Nadelspitze das Papier noch nicht eindrücken.

Mit Ausdauer und etwas Glück findet man in unserem Fall wahrscheinlich sogar die "bestmögliche" Lösung:

- Allerdings muss man dabei beachten, dass man die einzelnen Parameter nicht unabhängig voneinander verstellen sollte.
- Oft findet man nur eine Verbesserung, wenn man mehrere Parameter gleichzeitig im richtigen Verhältnis zueinander ändert.
- Man muss immer darauf achten, dass man alle Forderungen erfüllt.

Für das Finden bestmöglicher Parameter für vorgegebene Systemstrukturen gibt es seit Jahrzehnten ausgereifte Optimierungstools. Leider hat sich dies noch nicht bis an den letzten Ingenieurarbeitsplatz herumgesprochen!

Solche Optimierungstools stellen zielgerichtet an den veränderbaren Parametern. Damit das Optimierungstool "weiß", wie gut das daraus resultierende Verhalten des Antriebs ist, muss man ihm entsprechenden Bewertungsgrößen bereitstellen. Im Beispiel sind dies folgende Kennwerte der Nadelbewegung:

- **tZyklus** (Zeit für einen kompletten Bewegungszyklus der Nadel)
- **Praegung** (Maß [0...1] für den maximalen Vorschub der Nadel innerhalb tZyklus)

Die Erfassung dieser beiden Kennwerte erfolgt in Analogie zum bereits abgeleiteten Riss-Zeitpunkt.

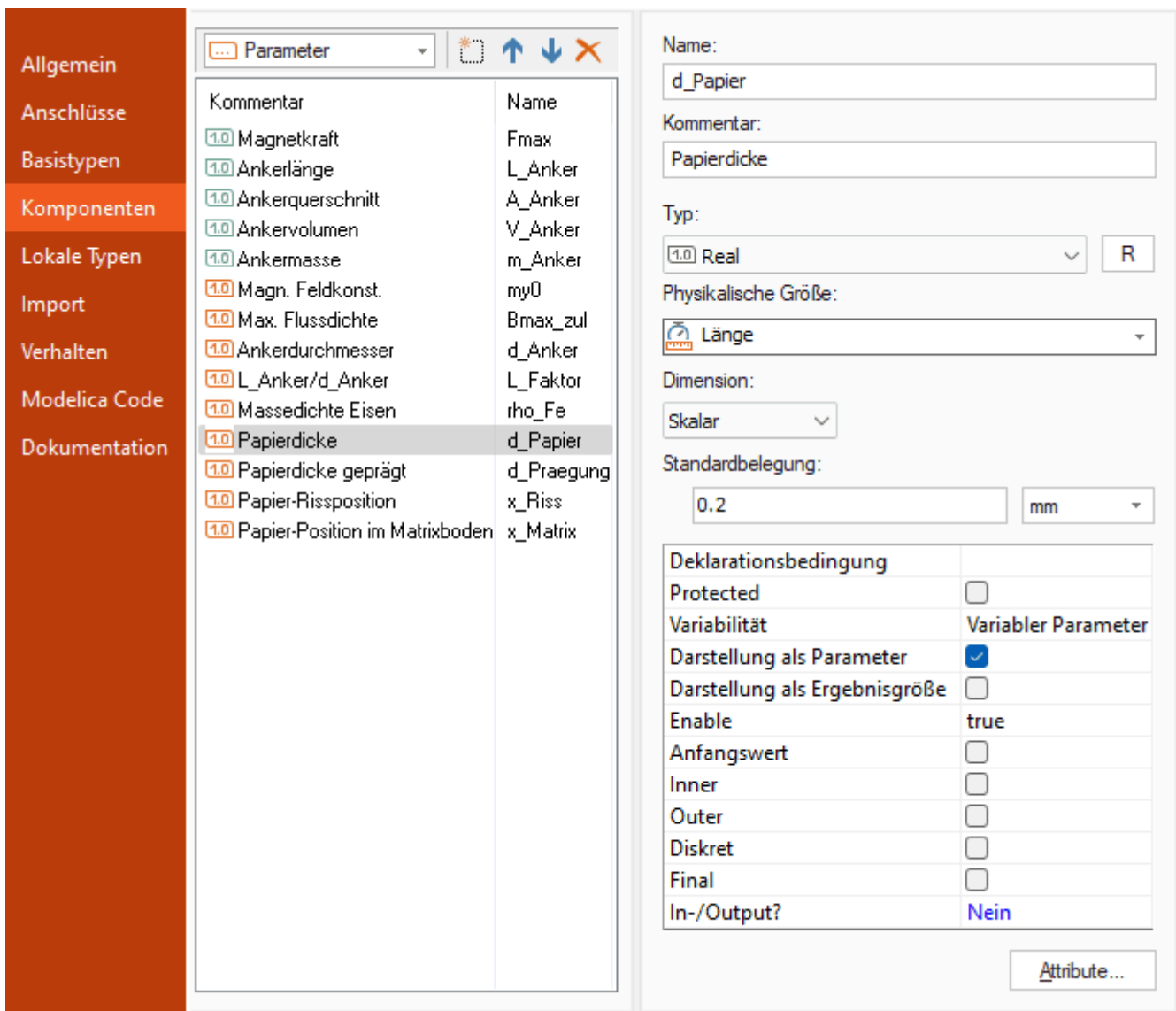
CAD-Daten der Papier- und Matrix-Parameter

Grundlage für die Erfassung der benötigten Kennwerte ist die "Beobachtung" der Nadelbewegung in Bezug auf die Geometrie von Matrize, Papier und Nadel-Ruhelage:

- Bisher haben wir markante Positionen (z.B. Anschlag bei -0,55 mm) einfach als Zahlenwerte in den Parametern der einzelnen Modell-Elemente berücksichtigt.
- Dies erschwert eine automatisierte Änderung solcher Modellparameter, wenn es nachträglich zu konstruktiven Änderungen kommt.
- Deshalb sollen alle relevanten geometrischen Kenngrößen, welche in Bezug zur Nadelbewegung stehen zuerst als Teil der CAD-Daten definiert werden, bevor wir die erforderlichen Bewertungsgrößen für das Optimierungstool ableiten.
- Dazu sind im Elementtyp *CAD_Data* folgende Komponenten als Parameter mittels des *SimulationX-TypeDesigners* zu ergänzen:

Name	Kommentar	Wert	Einheit	Phys. Größe
d_Papier	Papierdicke	0.2	mm	Länge
d_Praegung	Papierdicke geprägt	0.1	mm	Länge
x_Riss	Papier-Rissposition	-0.39	mm	Weg
x_Matrix	Papier-Position im Matrixboden	-0.65	mm	Weg

Hinweis: Die physikalische Größe "**Weg**" findet man unter **Mechanik > Mechanik (translatorisch)**. Der "**Weg**" repräsentiert im Unterschied zur "**Länge**" eine Position auf der Koordinatenachse:



Achtung: Wahrscheinlich kommt es nach dem Fertigstellen des *TypeDesigners* zu dem im vorherigen Abschnitt beschriebenen Lizenzfehler. Darauf ist dann wie erläutert zu reagieren!

Die nun an zentraler Stelle im CAD-Interface definierten konstruktiven Parameter müssen im Folgenden in alle bisher davon abgeleiteten Modell-Parameter eingespeist werden:

```

Kern.x      : CAD.x_Matrix + CAD.d_Praegung
Anschlag.12: abs (CAD.x_Matrix + CAD.d_Praegung)
Papier.L    : 2 * (Nadel.x0 - CAD.d_Papier)
Riss.a     : CAD.x_Riss

```

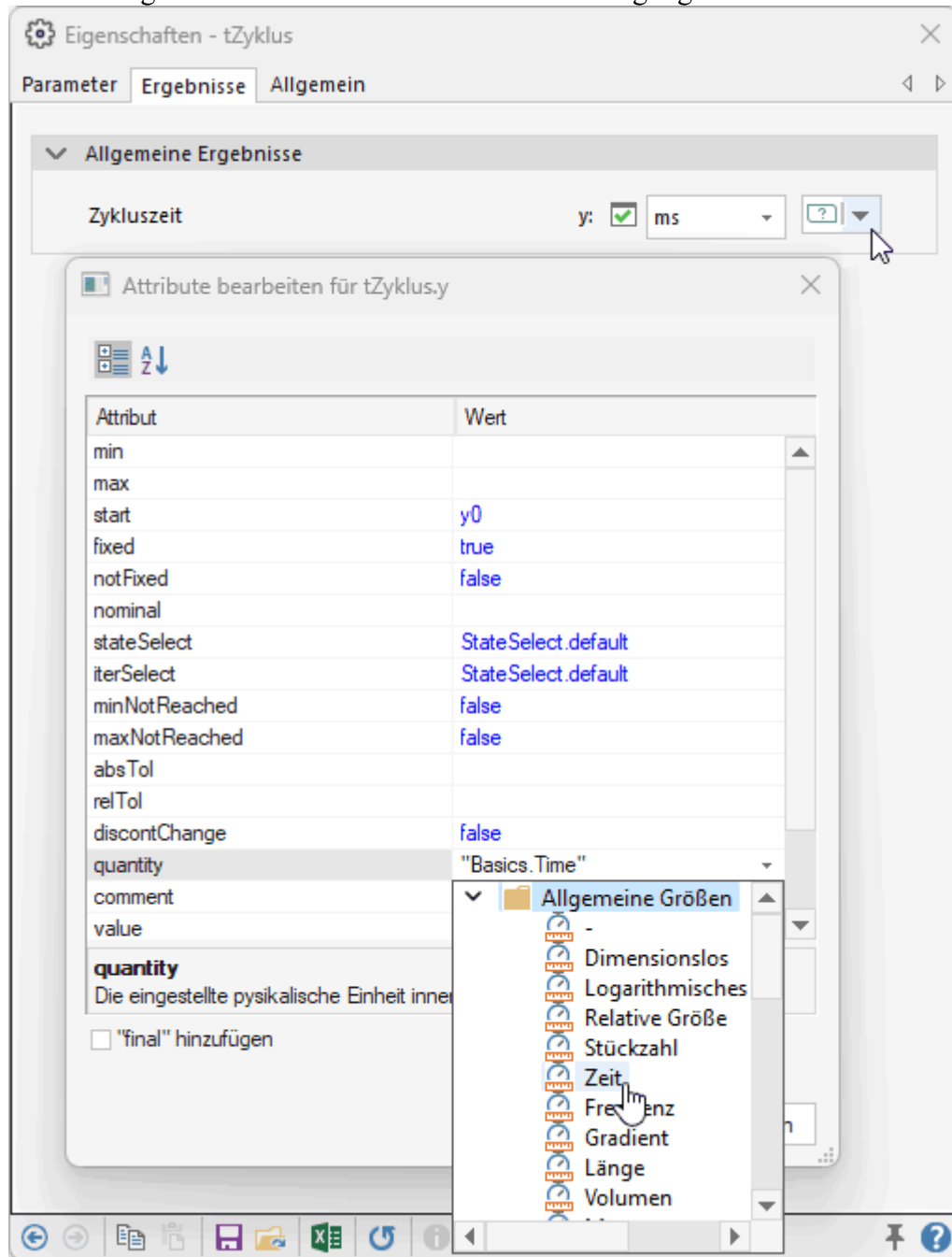
Danach ist anhand der Ergebnisse und Signalverläufe sorgfältig zu überprüfen, ob diese Änderungen bei erneuter Simulation zum exakt gleichen Modellverhalten führen!

Zeit eines Prägezyklus (tZyklus)

In Analogie zur Riss-Erfassung verwenden wir ein ereignisgesteuertes Abtast-Glied, um den Zeitpunkt der Rückkehr der Nadel in die Ruhelage *Nadel.x0* zu messen. Diese Methode hat hier zwei Schwachstellen:

- Wenn der Prägezyklus während der Simulationszeit nicht vollendet wird, ergibt sich aus dem Messprinzip $t_{Zyklus.y}=0$. Dies entspricht der "idealen" Lösung aus der Sicht eines Optimierungstools. Man muss deshalb ausreichend Reserven für die Simulationszeit vorsehen, so dass ein Prägezyklus immer beendet werden kann!

- Infolge der Feder-Vorspannung kann es am Anfang der Simulation kurz nach $t=t_{Start}$ zu unerwünschten $tZyklus$ -Ereignissen kommen, wenn die Nadel sich "numerisch" um winzige Beträge in den Anschlag hineinbewegt. Damit wird der Wert von $tZyklus.y$ näherungsweise auf t_{Start} gesetzt, unabhängig von einem Anfangswert $tZyklus.y0$. Aus diesem Grund darf nicht nur das erste Ereignis berücksichtigt werden!
- Die Parameter sind so zu konfigurieren, dass das Erfassen der Zykluszeit als eine Grundlage für die automatisierte Bewertung des Modellverhaltens unter allen Bedingungen sicher funktioniert!



Ergebnisse von Signalgliedern sind standardmäßig reine Zahlenwerte ohne physikalische Einheit:

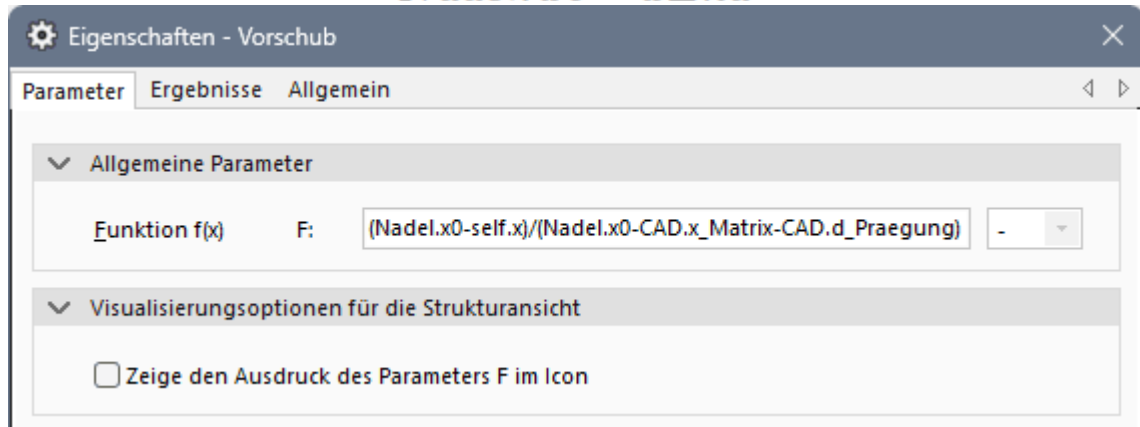
- Insbesondere bei Bewertungsgrößen ist es sinnvoll, die physikalischen Eigenschaften von Signalgrößen im Modell zu berücksichtigen.
- Die Qualität der Ergebnisgröße $tZyklus.y$ ist ein Zeitwert und sollte wegen der Anschaulichkeit in der Maßeinheit **ms** angezeigt werden.
- Deshalb bearbeiten wir die Attribute von $tZyklus.y$ und weisen dieser Größe die Zeit als "quantity" zu.
- Zusätzlich ersetzen wir für diese Ergebnisgröße den Standardwert "Signalausgang" des Kommentars durch eine Erläuterung der Bedeutung (z.B. "**Zykluszeit**").

Max. Nadelvorschub (Praegung)

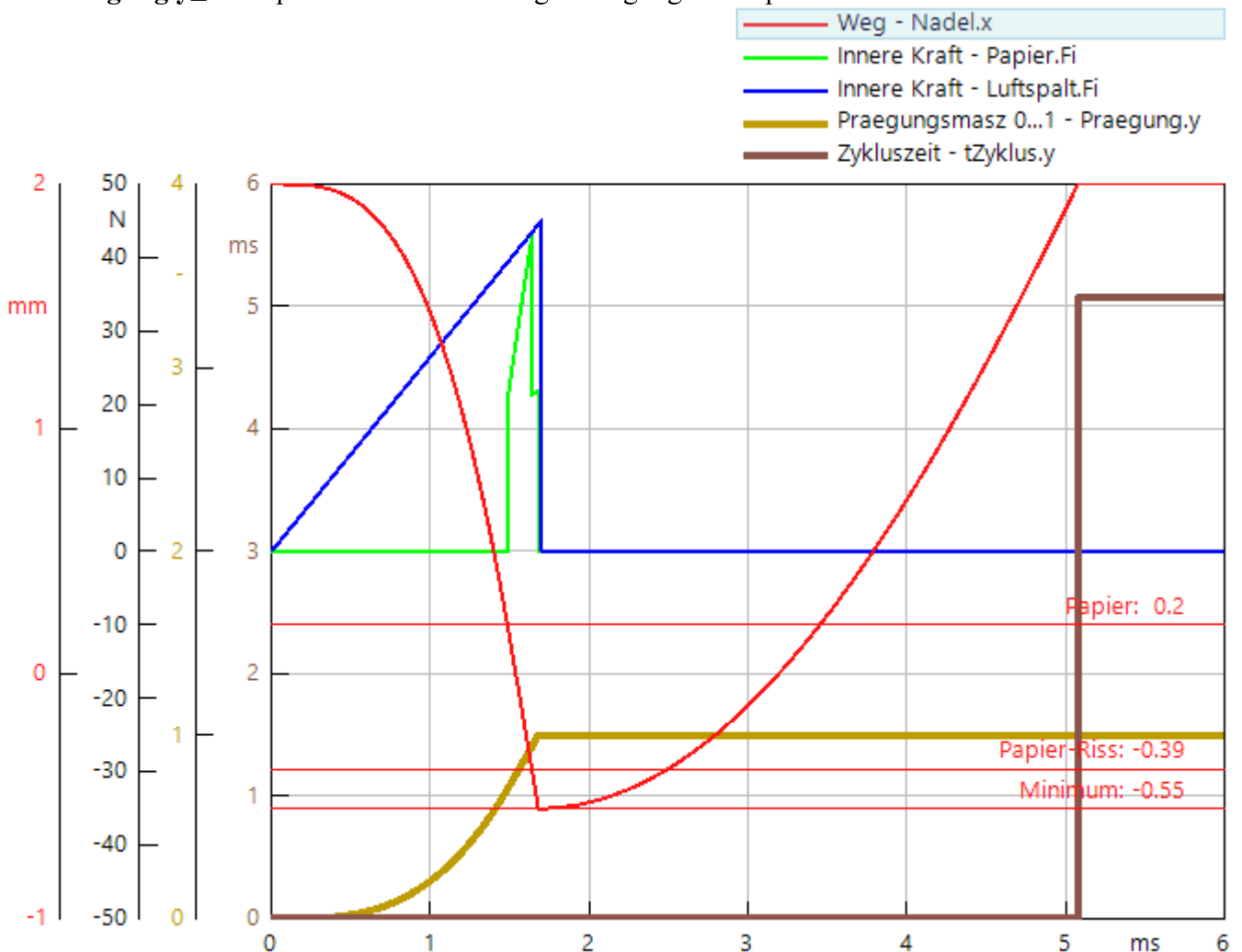
Ein Bewegungszyklus der Nadel ist nur sinnvoll, wenn dabei auch die Präegung des Papiers stattfindet:

- Dafür soll der Maximalwert des auf 0 bis 1 normierten Vorschub-Wertes erfasst werden.
- Den aktuellen normierten Vorschub y der Nadelspitze berechnet man als $f(x)$ mit $x_{End} = \text{Anschlag.l2} = \text{abs}(\text{CAD.x_Matrix} + \text{CAD.d_Praegung})$ zu:

$$f(x) = \frac{\text{Nadel.x0} - x}{\text{Nadel.x0} - x_{End}}$$



- Anschließend registriert man den dabei aufgetretenen Maximalwert mit einem Signalanalyse-Element (aus spezielles Signalglieder) und interpretiert diesen Maximalwert als Maß für die im Bewegungszyklus stattgefundenene Präegung.
- Nur $\text{Praegung.y} \geq 1$ entspricht einer vollständigen Präegung des Papiers:



Funktionelle Ausgangslösung (kompletter Bewegungszyklus der Nadel)

Das Modell des Nadelantriebs werden wir so konfigurieren, dass ein kompletter Prägezyklus stattfindet (wie im vorherigen Signalfensterbild gezeigt). Dabei sollten wir keinen Ehrgeiz darauf verschwenden, durch manuelles Probieren eine besonders kurze Zykluszeit zu erreichen! Das Finden der optimalen Parameter übernimmt für uns die numerische Optimierung.

Jegliche Entwicklung in Natur und Technik basiert auf dem Wechselspiel zwischen Funktion und funktionierender Struktur:



- Die Funktion einer technischen Struktur kann durch Verändern von Parametern nur verbessert werden, wenn die Funktionalität in bewertbarer Form bereits existiert (zumindest andeutungsweise).
- Eine funktionelle Ausgangslösung realisiert diesen Anspruch, auch wenn die Lösung im Sinne der zu erfüllenden Forderungen noch völlig unzureichend ist.
- In Analogie dazu können in der Natur zum Beispiel aus Beinen und/oder Armen durch die Evolution nur Flossen entstehen, wenn damit "irgendwie" geschwommen wird und dies Auswirkung auf die Anzahl der überlebenden Nachkommen hat!

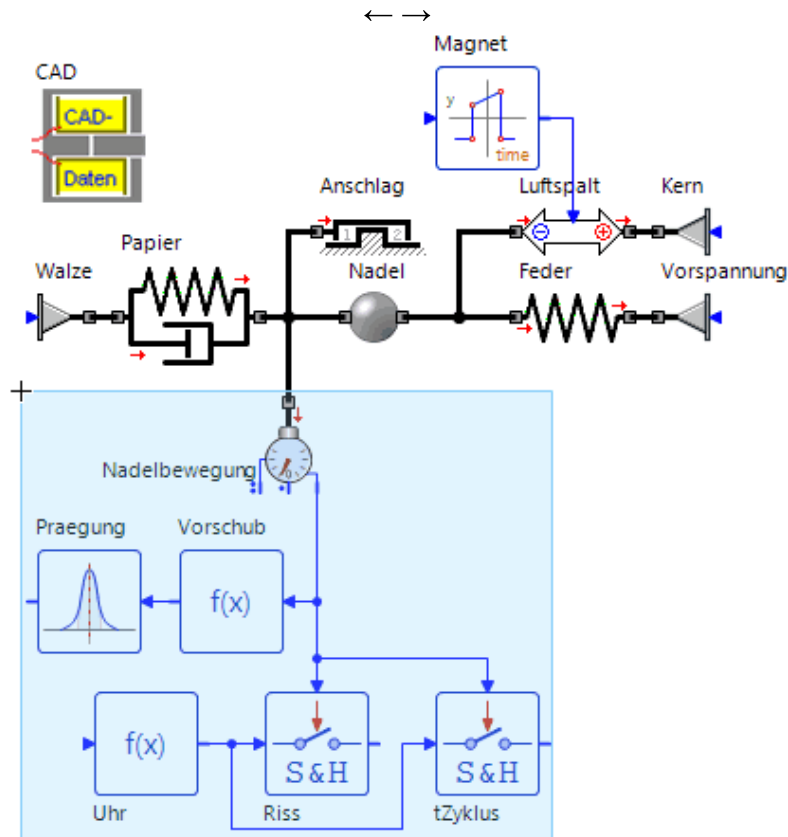
← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Bewertung&oldid=27758“

Software: SimX - Nadelantrieb - Wirkprinzip - Signalprozessor-Compound

Aus OptiYummy

↑



Signalprozessor-Teilmodell (Compound-Definition)

□

Teilmodelle

Die Komplexität bei der Modellierung eines Gerätes oder einer Baugruppe ist nur durch Gliederung in "funktionelle Teilsysteme" beherrschbar:

- Modelle von Teilsystemen werden auch "Teilmodelle" genannt.
- Die Abgrenzung von Teilsystemen erfolgt überwiegend nach funktionellen Aspekten:
 - Es sollten funktionelle Einheiten entstehen, welche sowohl als reale Objekte als auch als Teilmodelle selbstständig betrieben werden können.
 - Die Abgrenzung von Teilsystemen ist nicht eindeutig, aber oft identisch mit der geometrischen Abgrenzung der Funktionseinheiten.
- In unserem Beispiel ist es sinnvoll, den Bewegungssensor und alle damit verbundenen Signalglieder zu einem Signalprozessor-Teilmodell zusammenzufassen (blau markiert). Im realen Versuchsmuster würde man die gleiche Funktionalität durch ein Mikrocontroller-Programm nachbilden, welches aus dem eingespeisten Nadel-Positionen die geforderten Kennwerte berechnet.

Innerhalb von SimulationX kann man Teilmodelle bilden, indem man innerhalb einer Modellstruktur beliebige Elemente mitsamt ihrer Verbindungen mittels des *TypeDesigners* in sogenannte "Compounds" verpackt:

- Dabei entsteht ein neuer Element-Typ, welcher die ausgewählten Elemente und Verbindungen enthält.
- Schnittstellen ermöglichen den Datenaustausch zwischen den Elementen innerhalb des Compounds und der übergeordneten Modellstruktur, in welche das Compound eingebunden wird.
- Die Inhalte eines Compound-Typen können nachträglich mit dem *TypeDesigner* modifiziert werden.

Wichtig:

- Bei der Compound-Bildung besteht die Gefahr der "Zerstörung" unseres erfolgreich aufgebauten Modells.
- Deshalb erzeugen wir mit den Mitteln des Betriebssystems eine Sicherungskopie mit der Bezeichnung "**Etappe1_xx_mit_Signalgliedern.isx**" (mit xx=01..99)
- Die weitere Bearbeitung erfolgt mit der Originaldatei "**Etappe1_xx.isx**".
- Wir notieren uns die bisherigen exakten Ergebniswerte für **tZyklus.y** und **Riss.y**, da nach den folgenden Modelländerungen die gleichen Werte berechnet werden müssen.

Express Edition - Workaround

Beachte:

- Wer die Übung nicht mit der im Funktionsumfang eingeschränkten *SimulationX Express Edition* bearbeitet, kann sich den folgenden Workaround bei der Bildung des Signalprozessor-Compounds ersparen!
- Die Bilder der Anleitung berücksichtigen jedoch die in der *Express Edition* notwendige Modell-Reduktion.

Unser Modell enthält bereits 16 Modell-Elemente und hat damit die maximal zulässige Anzahl innerhalb einer Modellebene im Rahmen der "Express Edition" von SimulationX erreicht:

- Vor dem eigentlichen Zusammenfassen der markierten Elemente erzeugt der *TypeDesigner* innerhalb des aktuellen Modells automatisch die zusätzlich erforderlichen Anschlüsse für die Compound-Schnittstellen zum übergeordneten Modell.
- Diese zusätzlichen Anschlüsse werden mit den bereits markierten Elementen danach zum Compound zusammengefasst. Im Beispiel ist dies der eine mechanische translatorische Anschluss zum Bewegungssensor - > dieser wird als zusätzliche Instanz innerhalb des aktuellen Modells gezählt und damit wird die zulässige Anzahl von 16 überschritten!
- Die Bildung des Compounds ist deshalb nicht mehr möglich und führt zur Fehlermeldung → "**Die Maximale Anzahl der Komponenten wurde erreicht**".
- Erforderlich ist das temporäre Löschen von zwei Modell-Elementen, um diese Hürde zu überwinden! Anscheinend benötigt der *TypeDesigner* vor dem eigentlichen Zusammenfassen ein weiteres "Hilfselement" innerhalb des aktuellen Modells.
- Um die Auswirkung der Compound-Bildung zu verifizieren, soll das Modell nach der erforderlichen Modellreduktion weiterhin ohne Funktionseinschränkung mit dem bereits konfigurierten Kraft-Verlauf betrieben werden:
 - Wir löschen dann den **Magnet**-Impulsgenerator und die **Kern**-Wegvorgabe (die Fehlermeldungen können wir ignorieren!).
 - Im **Luftspalt**-Kraftelement ersetzen wir den **Parameter**-Wert für die Kraft **F** durch eine Formel, welche den gewünschten Dreiecksimpuls der Amplitude **Fmax** und **1,8 ms** Länge generiert:

```
if time/0.0017 <= 1 then CAD.Fmax*time/0.0018 else 0
```

- Eine erneute Simulation des reduzierten Modells muss zu den gleichen Ergebnissen für **tZyklus.y** und **Riss.y** führen!

Erst nach der Definition und erfolgreichen Inbetriebnahme des Compound ergänzen wir wieder die beiden gelöschten Modell-Elemente mit ihren vorherigen Parametern einschließlich des Luftspalt-Elements.

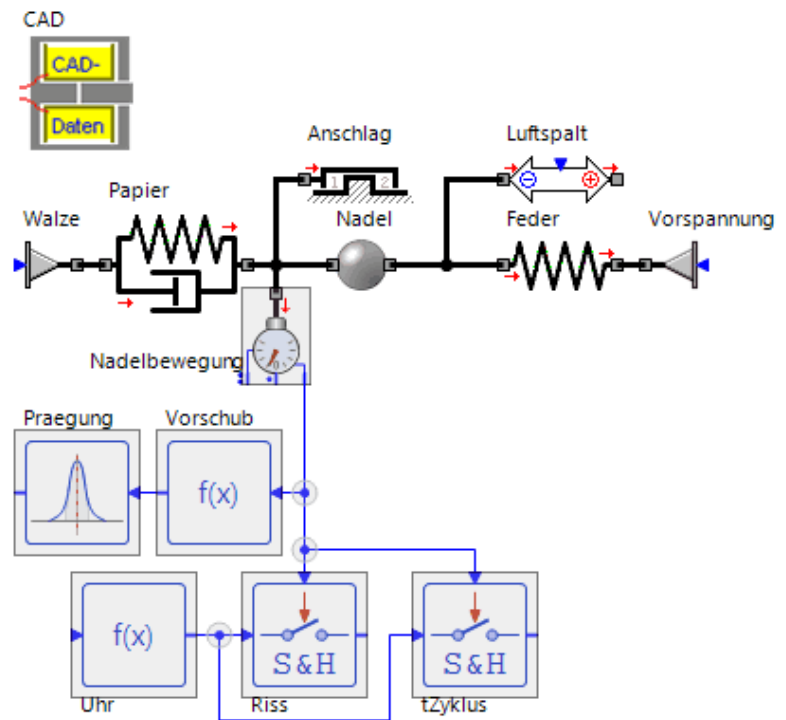
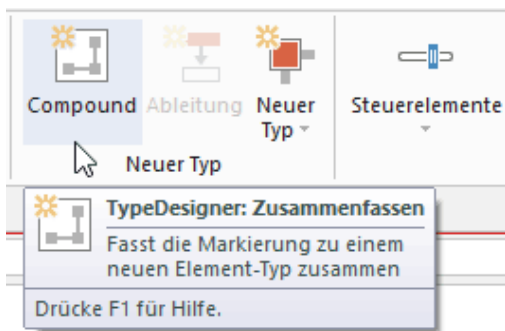
Compound-Definition und -Inbetriebnahme

Zusammenfassung der Bewegungssignalverarbeitung zu einem Teilmodell (Compound)

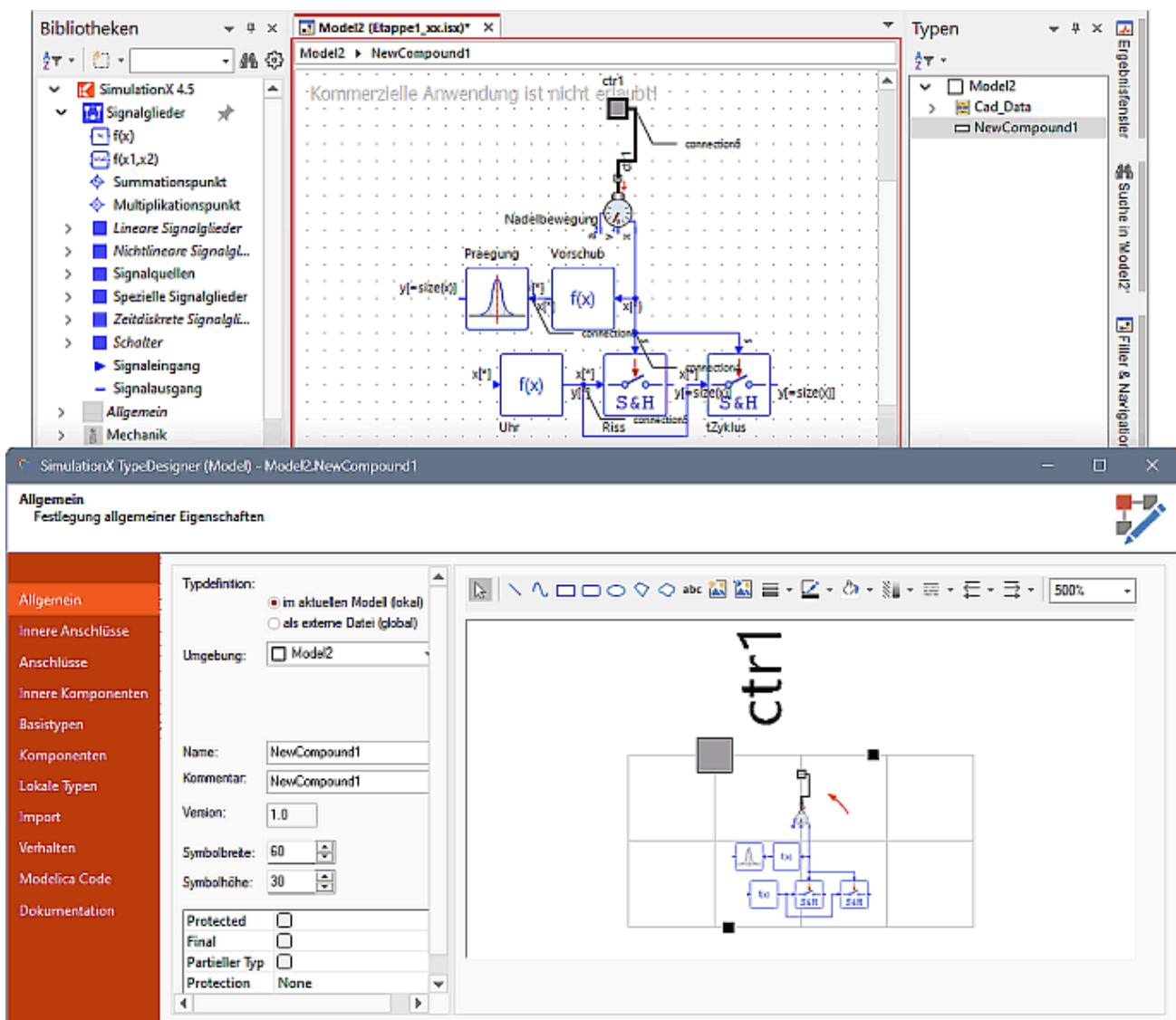
Bevor man über die Multifunktionsleiste den *TypeDesigner* zur Compound-Bildung aufrufen kann, muss man alle Modell-Elemente markieren, welche zusammengefasst werden sollen.

Die gleichzeitige Markierung mehrerer Elemente erreicht man durch Rechteck-Auswahl mittels des Cursors bei gedrückter Maustaste (wie im Titelbild dieses Abschnitts gezeigt). Ist dies auf Grund der Element-Anordnung nicht möglich, kann man Modell-Elemente durch zusätzlich betätigter UMSCHALT-Taste bei der Cursor-Markierung der

Auswahl hinzufügen bzw. auch entfernen:



Nach Aufruf des *TypeDesigners* erscheint dieser im Bearbeitungsmodus für einen neuen Compound-Typ, welcher die markierten Modell-Elemente und deren Verbindungen enthält:



- Standardmäßig wird der neue Compound-Typ lokal im aktuellen Modell erzeugt und erscheint im Typen-Explorer (im Beispiel unter dem bereits vorhandenem CAD_Data-Typ).
- Der grafische Editor des *SimulationX* enthält nun eine Ebene unter dem aktuellen Modell die Compound-Struktur, welche dort auch bearbeitet werden kann.
- Standardmäßig wird für den Compound-Typ ein Symbol der Größe 60x30 mm² erzeugt. Im zugehörigen Grafikeditor wird passend zu dieser Größe ein Strukturbild der zusammengefassten Elemente bereitgestellt.
- Für die Verbindungen zwischen den Compound-Elementen und der Rest der Modellstruktur werden passende Schnittstellen generiert. Im Beispiel ist dies **ctr1** (translatorischer Mechanik-Konnektor), welcher den Anschluss des Bewegungssensors nach außen führt.
- Die auf der linken Seite aufgelisteten Bearbeitungsschritte zur Typ-Definition entsprechen denen, welche wir bereits bei Erstellung des CAD_Data-Typs verwendet haben.

Allgemein

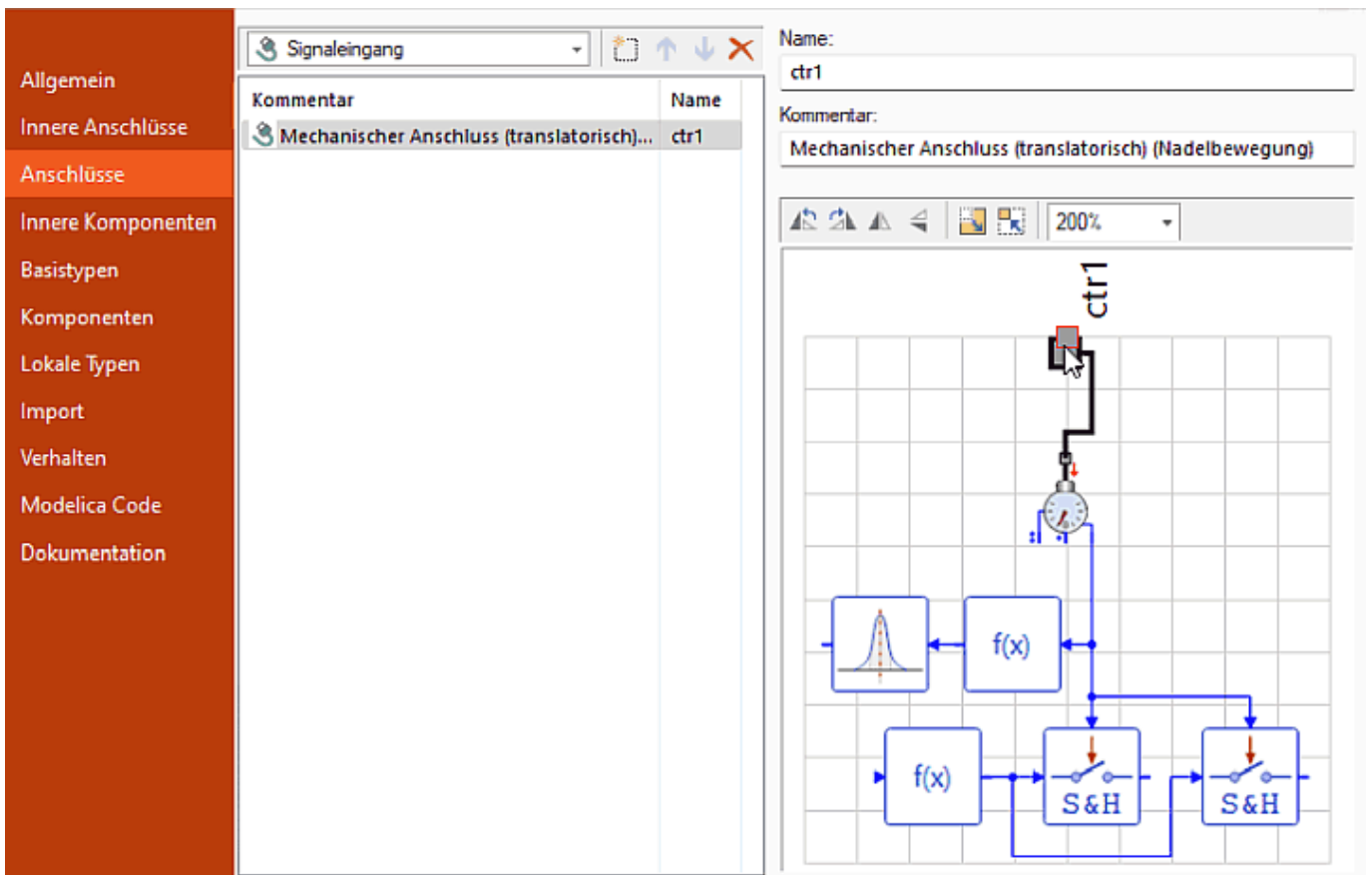
Wir modifizieren hier Namen, Kommentar und Grafiksymboll:

- **Name:** Signalprozessor
- **Kommentar:** Kennwerte der Nadelbewegung
- **Symbol:**
 1. **Breite x Höhe:** 150 Pixel x 150 Pixel (diese Verkleinerung auf ca. 60% ermöglicht Erkennbarkeit)
 2. **Bild-Anpassung:** nach Wahl eines geeigneten Zoom-Faktors im Symbol-Editor (z.B. 200 %) mittels Cursor an den 2 diagonalen Eckpunkten das sehr kleine Bild auf komplette Symbolgröße ziehen (Sensor-Anschluss exakt auf Mitte an der oberen Kante):

Anschlüsse

Der mechanische Anschluss (translatorisch) überträgt die Bewegungsgrößen der Nadelmasse aus der übergeordneten Modellebene in den Bewegungssensor, welcher sich innerhalb des Compounds befindet:

- Da nur dieser eine (automatisch generierte) Anschluss benötigt wird, kann sein Name erhalten bleiben.
- Den zugehörigen **Ctrl1**-Marker muss man mit dem Cursor im Symbol-Editor auf die Mitte der oberen Kanten schieben (deckungsgleich mit Position im Symbol-Bild):



Komponenten-Definition (Parameter und Variable)

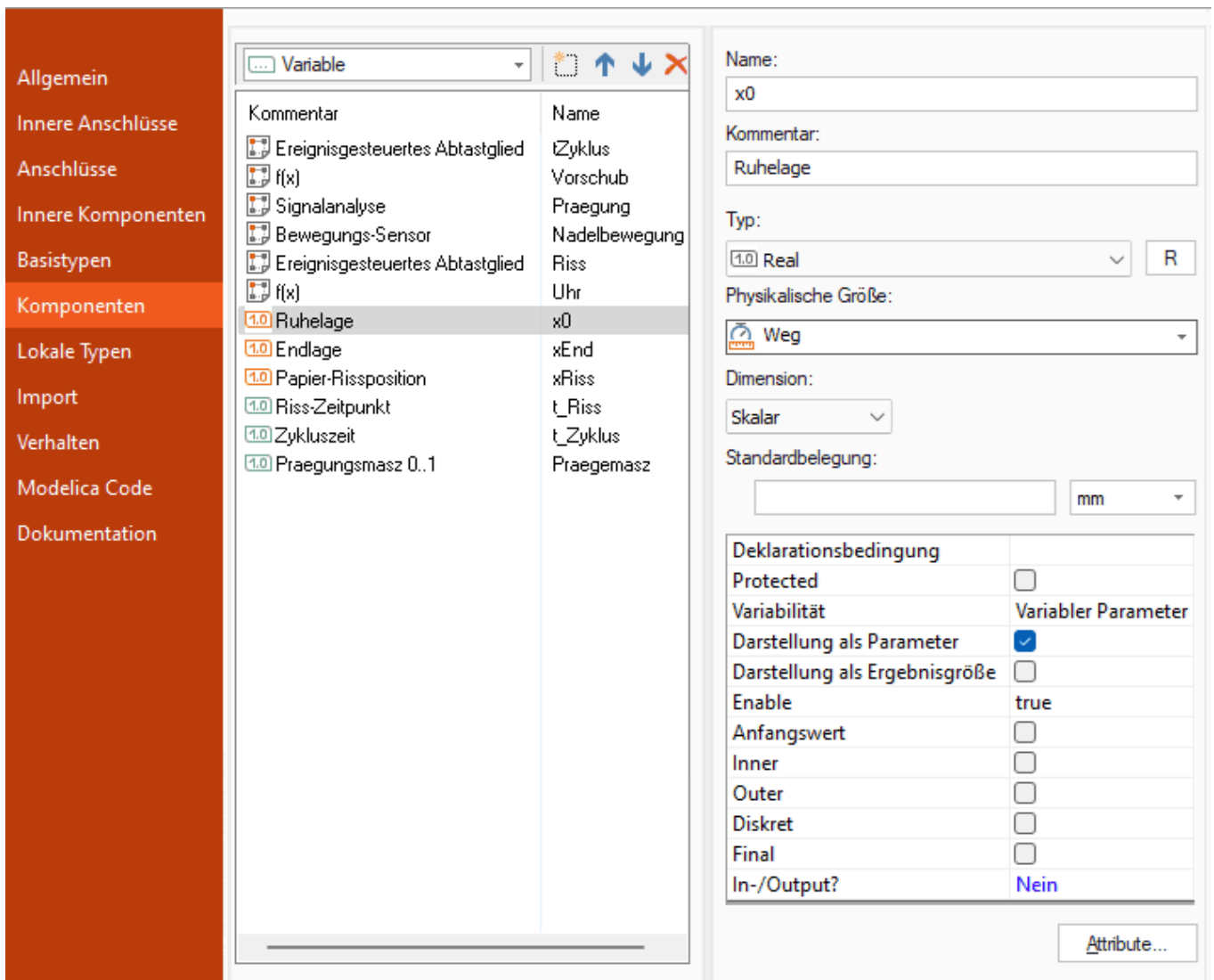
Parameter und Variable (Ergebnisse) bilden die Datenschnittstelle zwischen den Elementen im Compound und der übergeordneten Modellebene.

- **Hinweis:**
Die Namen von Compound-Komponenten dürfen innerhalb des Compound noch nicht anderweitig verwendet worden sein!
- Zur Analyse eines Bewegungszyklus müssen in den Compound die folgenden x-Koordinaten als **Parameter** eingespeist werden, wobei keine Werte als Standardbelegung erforderlich sind:

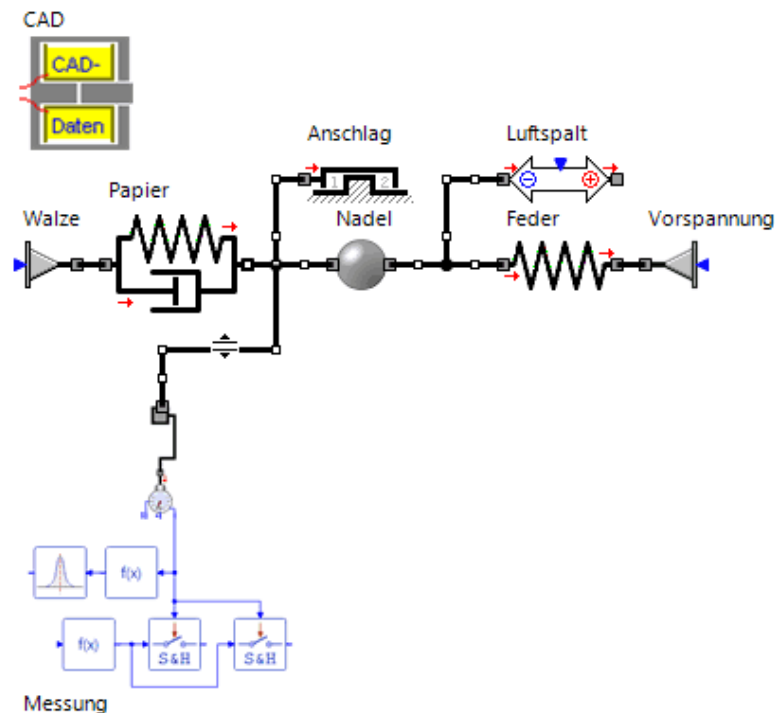
Name	Kommentar	Standardbelegung	Einheit
x0	Ruhelage		mm
xEnd	Endlage		mm
xRiss	Papier-Rissposition		mm

- Die Ergebnisse der implementierten Signalanalyse müssen der übergeordnete Modellebene als **Variable** bereitgestellt werden:

Name	Kommentar	Physik. Größe	Einheit
t_Riss	Riss-Zeitpunkt	Zeit	ms
t_Zyklus	Zykluszeit	Zeit	ms
Praegemasz	Praegungsmasz 0..1	Dimensionslos	-



Mit der Definition der Komponenten ist ein wichtiger Bearbeitungsschritt beim Erstellen des Compound-Typs erbracht. Den bisher erreichten Zwischen-Zustand sichern wir durch "Fertigstellen":



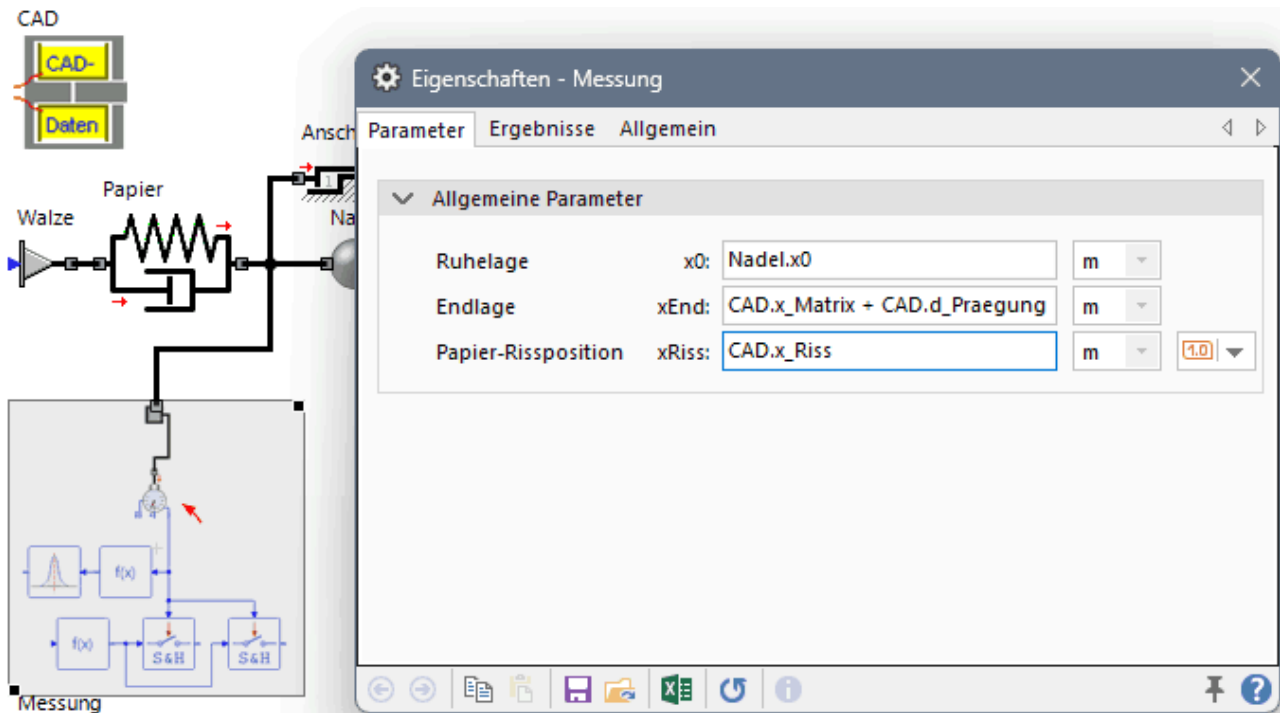
- Für das Compound-Element vergeben wir in der Modellstruktur den Namen "**Messung**" im Modell (Anordnung "unten").
- Wir schieben das neue Element mit dem Cursor an eine günstigere Position.

- Die Anschluss-Verbindung zum Compound-Element müssen wir danach mit Cursor auswählen, um die Lage eines waagerechten Teilstückes korrigieren (Falls es wie im Beispiel zu einer Überdeckung mit dem Papier-Element kam).
- Modell-Datei > **Speichern** realisiert dann die gewünschte Sicherung des erreichten Bearbeitungszustandes auf einem Datenträger.

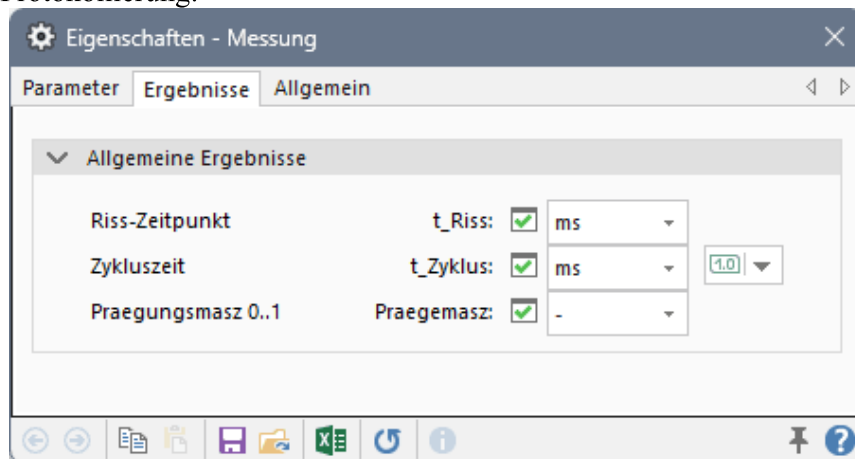
Einbindung des Compounds mittels der Parameter und Ergebnisse in die Modellstruktur

Bevor wir den Compound-Typ weiter bearbeiten, öffnen wir durch Doppelklick auf das Messung-Teilmodell dessen Eigenschaftsdialog:

- In die leeren Parameter-Felder speisen wir die benötigten Kennwerte des Bewegungsablaufes ein:

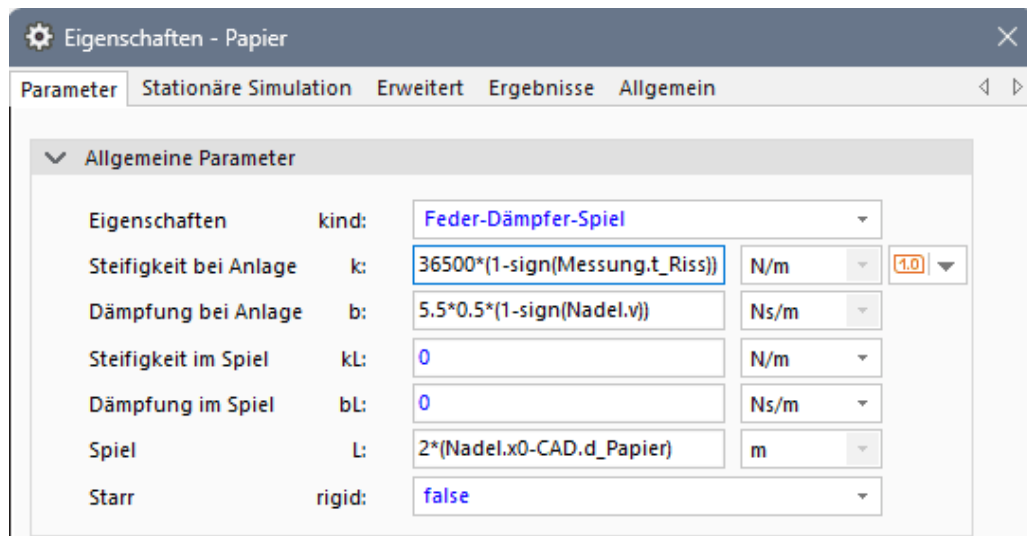


- Damit stehen diese Werte dann zur Parameter-Belegung in den Compound-internen Signalverarbeitungselementen zur Verfügung.
- Wir überprüfen, ob die Ergebnisse wie gewünscht von dem Compound bereitgestellt werden und aktivieren diese für die Signal-Protokollierung:



Von den drei Ergebnisgrößen wird nur der Riss-Zeitpunkt **t_Riss** direkt in einen Element-Parameter benutzt → Papier-Steifigkeit **k**: $36500 \cdot (1 - \text{sign}(\text{Riss.y}))$:

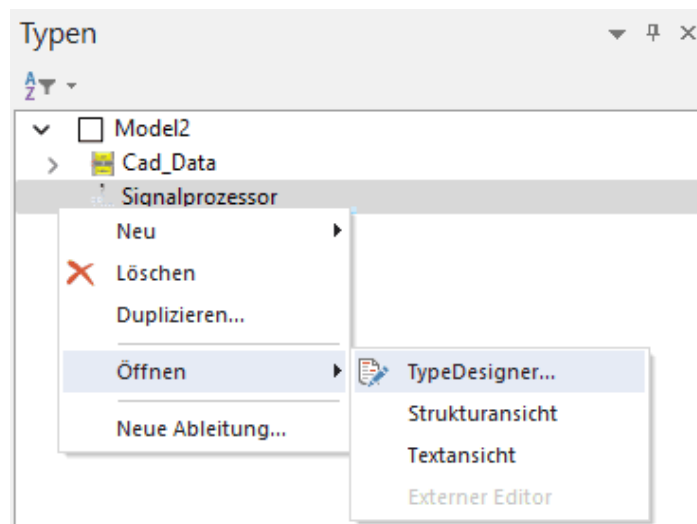
- Der bisherige Bezug auf die Variable **Riss.y** ist im Papier-Element nicht mehr möglich, weil innerhalb des Modells nur auf die Anschlüsse und Komponenten eines Compounds zugegriffen werden kann.
- **Riss.y** ist in der Formel deshalb durch **Messung.t_Riss** zu ersetzen:



Diese Modelländerungen sollte man anschließend wieder speichern!

Einspeisung der Parameter-Werte in die Compound-Elemente (Strukturansicht)

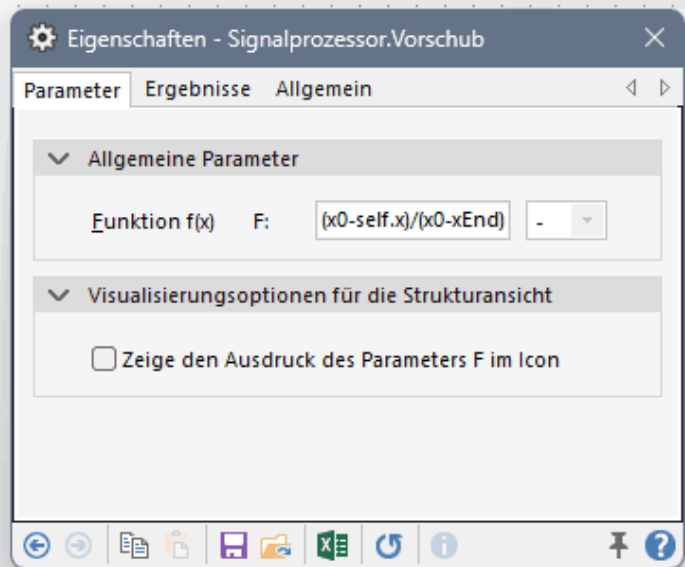
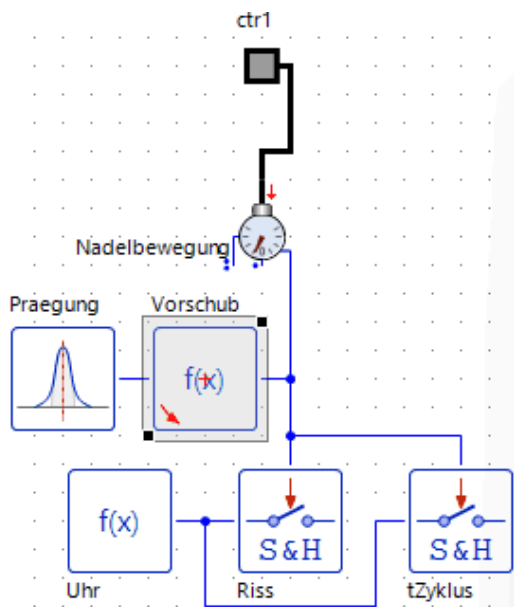
Für das Bearbeiten eines bestehenden Compound-Typen kann man über das Kontext-Menü drei verschiedene Editoren öffnen:



- **Textansicht:** mit Editor für den kompletten Quelltext des Compound-Typen (etwas für Profis und bei Interesse für die Sprachstruktur).
- **Strukturansicht:** öffnet die grafische Struktur der verbundenen internen Modell-Elemente. Falls diese inzwischen geschlossen wurde, öffnen wir sie damit erneut. Anderenfalls wählen wir im Bereich der Strukturansicht anstatt der Modell-Struktur die Registerkarte der noch geöffneten Compound-Struktur.
- **TypeDesigner:** mit den bereits bekannten Bearbeitungsschritten (benutzen wir erst wieder für die Ergebnis-Werte).

Die Bearbeitungsmöglichkeiten in der Strukturansicht des Compound entsprechen denen der übergeordneten Modell-Ebene:

- Doppelklick auf die Struktur-Elemente öffnet deren Eigenschaftsdialog und damit den Zugriff auf Parameter und Ergebnisse.
- Innerhalb des lokalen Compound-Typen kann zwar auf alle Bezeichner der übergeordneten Modell-Ebene direkt zugegriffen werden (z.B. auf "**Nadel.x0**"). Im Sinne der autonomen Betriebbarkeit von Teilmodellen sollte man den Datenaustausch zur übergeordneten Modellebene grundsätzlich nur über die Schnittstellen realisieren!
- Im Vorschub-Element modifizieren wir deshalb die Formel "**(Nadel.x0-self.x)/(Nadel.x0-CAD.x_Matrix-CAD.d_Praegung)**" durch die entsprechenden Compound-Parameter:



- Das Riss-Element erhält als **Grenzwert a: xRiss**.
- Das tZyklus-Element wird mit dem **Grenzwert a: x0** belegt.

Mit geöffneter Compound-Strukturansicht können wir danach den aktuellen Bearbeitungszustand wieder speichern.

Belegung der Ergebnis-Variablen mittels Gleichungen (TypeDesigner-Verhalten)

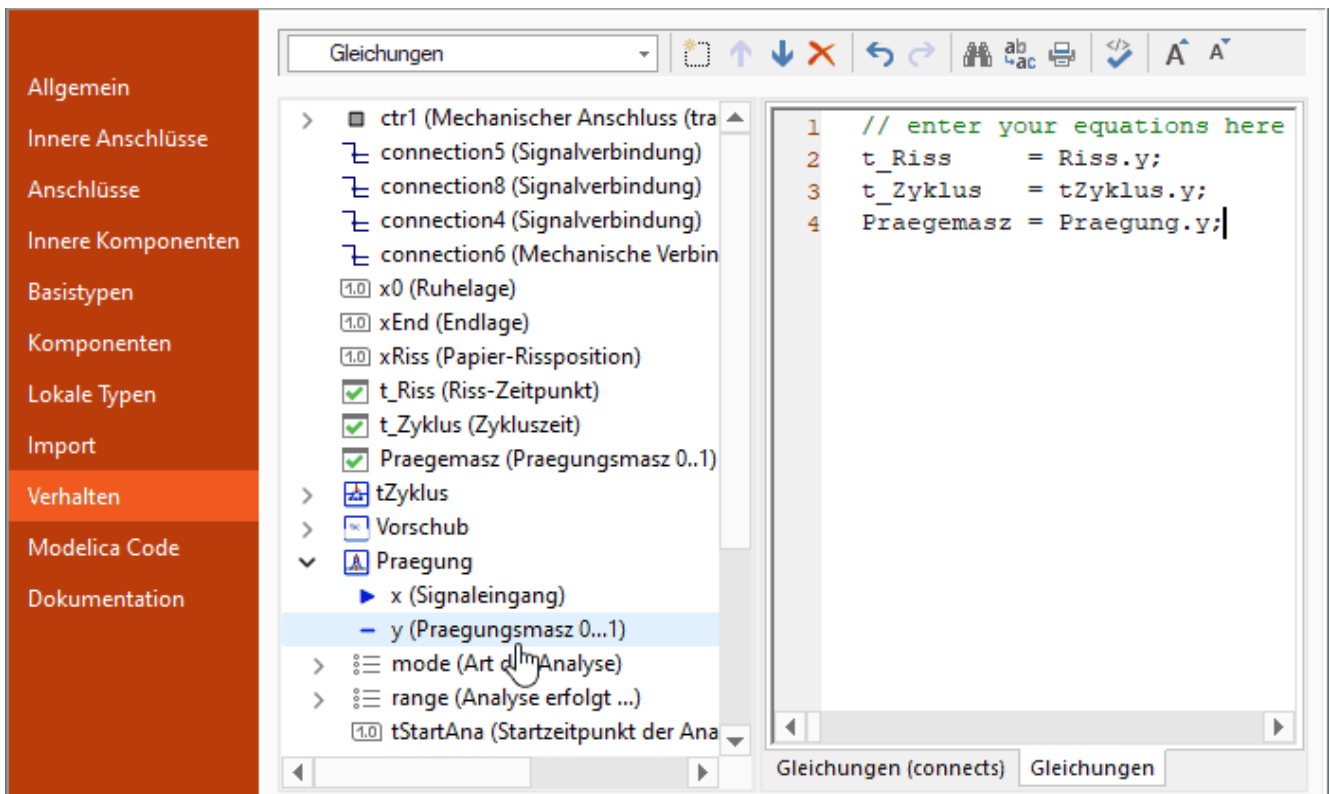
Falls noch nicht geschehen, öffnen wir nun wieder den TypeDesigner, um im Verhalten die Zuweisung der Ergebniswerte zu ergänzen:

```

1 connect (connection6,ctrl1);
2 connect (connection6,Nadelbewegung.ctrl1);
3 connect (connection4,tZyklus.s);
4 connect (connection5,tZyklus.x);
5 connect (connection4,Vorschub.x);
6 connect (connection8,Vorschub.y);
7 connect (connection8,Praegung.x);
8 connect (connection4,Nadelbewegung.x);
9 connect (connection4,Riss.s);
10 connect (connection5,Riss.x);
11 connect (connection5,Uhr.y);

```

- Innerhalb des Verhalten wurde beim Zusammenfassen der markierten Modell-Elemente automatisch ein Gleichungsabschnitt "**connects**" definiert. Dessen Gleichungen beschreiben alle Verbindungen zwischen den inneren Modell-Elementen und zu den Anschlüssen der Compound-Abgrenzung.
- Aktiviert man in der Compound-Strukturansicht die Anzeige der Verbindungen, so kann man die einzelnen connect-Anweisungen gut nachvollziehen.
- Änderungen in der Compound-Strukturansicht bzw. in den connects-Gleichungen werden automatisch synchronisiert, da es sich hier um einen automatisch generierten Gleichungsabschnitt handelt. Deshalb sollte man diesen Abschnitt nicht durch eigene Gleichungen erweitern.
- Wir ergänzen deshalb einen neuen Gleichungsabschnitt für die Zuweisung der Ergebniswerte:



- Beim Erstellen der erforderlichen Gleichungen für die Belegung der Ergebnis-Variablen kann man die benötigten Objekt-Bezeichner mit dem Cursor aus dem eingblendeten Compound-Modellexplorer ziehen.
- Nach dem **TypeDesigner > Fertigstellen** und dem **Speichern des Modells** können wir mittels Simulation das Modellverhalten verifizieren.

Verifizierung und Komplettierung des Modells

Für die Verifizierung des Modells genügt der Vergleich der nun berechneten Werte für **Messung.tZyklus** und **Messung.tRiss** mit den bekannten vorherigen Werten **tZyklus.y** und **Riss.y**:

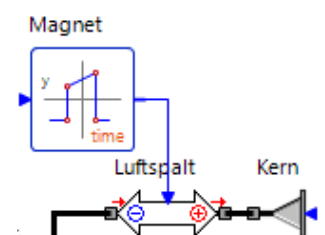
- Insbesondere der Wert für die Zykluszeit hängt vom kompletten, korrekten Bewegungsablauf der Nadel ab.
- Zusätzlich muss man überprüfen, ob weiterhin das korrekte Prägemmaß für die vollständige Präegung ermittelt wird (**Messung.Praegemasz** \approx 1).

Gibt es Fehlermeldungen nach Start der Simulation oder werden falschen Ergebniswerte berechnet, muss man die Fehler finden und beheben:

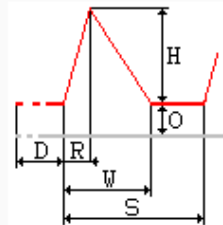
- Fehler in Hinblick auf unbekanntem Bezeichner beruhen im Beispiel meist auf vergessene Parameter-Änderungen (Schreibfehler sollten bereits beim Editieren gemeldet werden).
- Fehlerhafte Ergebnisse resultieren meist aus fehlerhaften oder vergessenen Berechnungsformeln, aber auch aus fehlerhaften Parametern. Hier sollte man bei der Fehlersuche dem "Weg der Daten" rückwärts folgen (beginnend beim falschen Ergebniswert).

Nachdem unser Modell richtig rechnet, kann die bei der Nutzung der Express Edition von *SimulationX* erforderliche Modell-Reduktion rückgängig gemacht werden:

- Den sägezahnförmigen Kraftimpuls realisieren wir wieder mittels einer *Signalquelle - Impulsgenerator* mit der ursprünglichen Konfiguration:



▼ Allgemeine Parameter

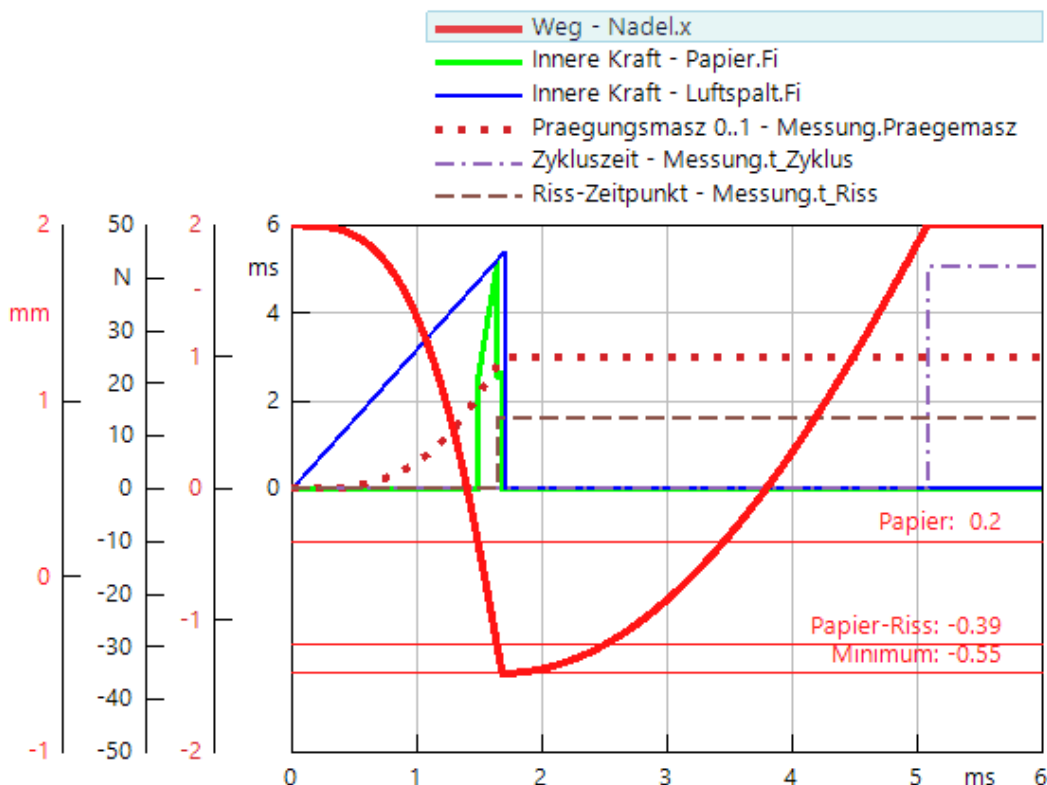


Bezugsgröße	RefVar:	Simulationszeit t [s]	
Kurvenform	CurveTypeVar:	Dreieck	
Anstiegsbreite	R:	0.0018	s
Breite	W:	self.R	s
Höhe	H:	CAD.Fmax	-
Totbereich	D:	0	s
Offset	O:	0	-
Folgeabstand	S:	0.1	s
Anzahl der Wiederholungen	i:	1	
Alternierendes Vorzeichen	PM:	false	

- Im Luftspalt-Element muss die Parameter-Formel wieder durch das Eingangssignal **self.in1** ersetzt werden, damit das Dreieckssignal des Impulsgenerators als Kraftverlauf verwendet wird!
- Abschließend ist die Kern-Wegvorgabe zu ergänzen (Luftspalt=0 mm bei vollständiger Prägung → **Kern.x : CAD.x_Matrix + CAD.d_Praegung**).
- **Wichtig:** Auch diese Modell-Änderung ist in Hinblick auf die Ergebnisse zu verifizieren!

Da alle Signalverarbeitungselemente im Compound "verschwunden" sind, fehlen jetzt ihre in Ergebnis-Fenstern des Modells dargestellten Ergebnisgrößen:

- Bei der Menge der bisher meist zu Testzwecken konfigurierten Ergebnisfenster verliert man die Übersicht!
- Wir konfigurieren deshalb abschließend ein Fenster mit aussagekräftigen Signalverläufen und löschen die restlichen Fenster:



- Das Signalfenster kann jeder individuell gestalten, solange die wesentlichen Signalverläufe dabei ohne großen Aufwand interpretierbar bleiben!

Wichtig:

- Unser Magnetantrieb arbeitet zeitlich im Bereich von wenigen Millisekunden.
- Die im Magnet-Impulsgenerator als Einschaltzeit genutzte **Anstiegsbreite R** verwendet bisher noch die Maßeinheit "**Sekunde**". Dies führt zu unanschaulichen Zahlenwerten auch bei der noch folgenden Ermittlung der optimalen Einschaltzeit.
- Wir wählen deshalb "**ms**" als sinnvollere Einheit für die Anstiegsbreite, dabei erfolgt automatisch eine Umrechnung des bereits eingegebenen Zeitwertes.
- Eine erneute Simulation muss das gleiche Verhalten ergeben!

Damit ist Bildung und Implementierung des Signalanalyse-Teilmodells abgeschlossen und wir können den aktuellen Modellzustand sichern (durch Speichern).

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Signalprozessor-Compound&oldid=27763“

Software: SimX - Nadelantrieb - Wirkprinzip - Experiment

Aus OptiYummy

↑

← →

Versuchsstand: Experiment

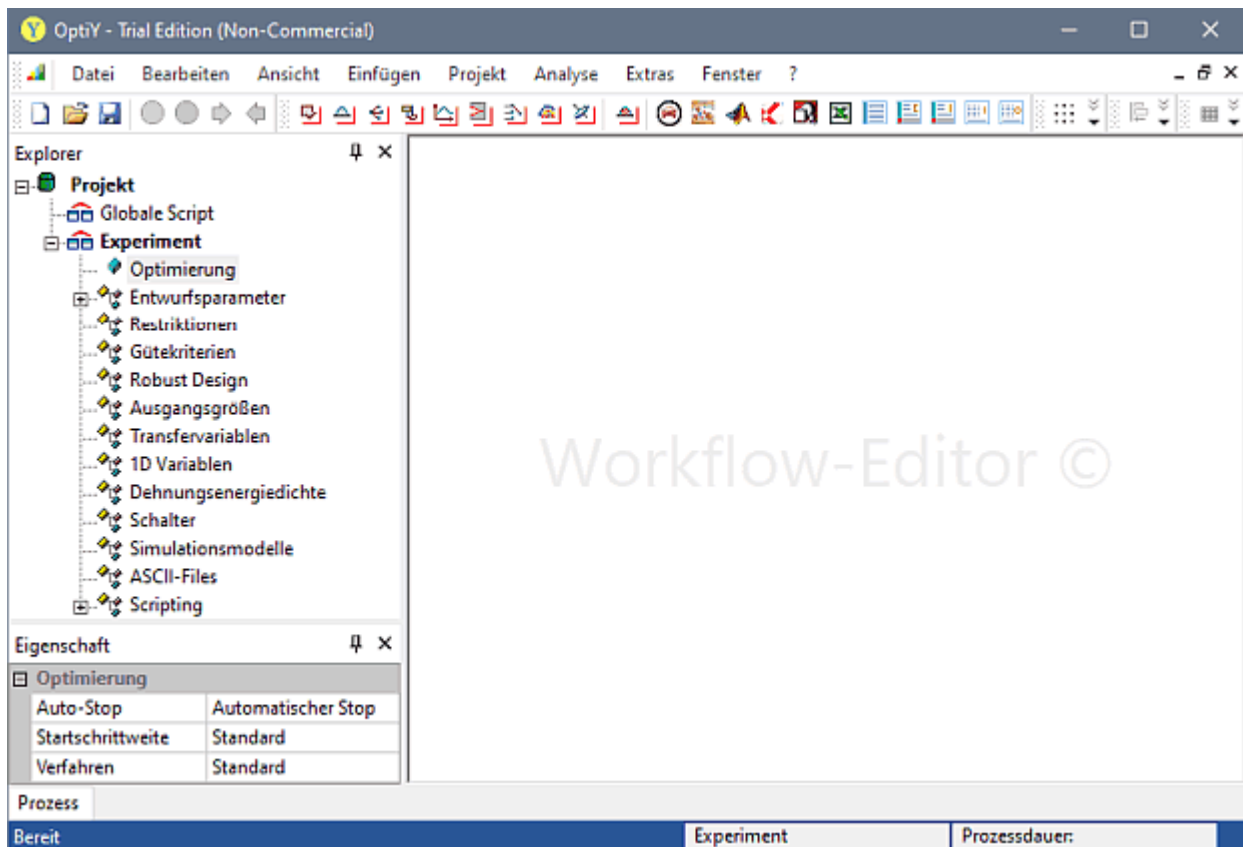
Ein **Experiment** ist durch seine Reproduzierbarkeit gekennzeichnet:

- Unter definierten Bedingungen werden ein oder mehrere Versuche mit einem Versuchsobjekt durchgeführt.
- Wiederholt man ein Experiment auf Basis der beschriebenen Konfiguration, so gelangt man zu den gleichen Aussagen in Bezug auf das Verhalten des Versuchsobjekts.
- Die Reproduzierbarkeit der Ergebnisse ist nicht an den konkret genutzten Versuchsstand gebunden.



Technische Experimente verfolgen meist einen Zweck - sie sind zielorientiert:

- Wir wollen z.B. jetzt zielgerichtet herausbekommen, wie schnell unser Magnetantrieb denn überhaupt werden kann! Dazu benutzen wir das Optimierungstool *OptiY*, welches als selbständigen Programm zu starten ist:



OptiY stellt über den Workflow-Editor Schnittstellen zu unterschiedlichsten Programmsystemen bereit:

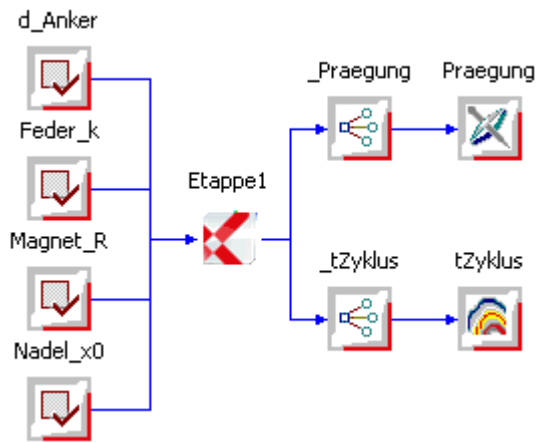
- Bereits implementiert ist die Anbindung von Excel, SimulationX, MapleSim, Matlab, CST Studio Suite und vielen anderen Simulationsprogrammen sowie diversen CAD-Systemen (z.B. Autodesk Inventor,

CATIA, Solid Works) und FEM-Programmen (z.B. ANSYS Workbench, JMAG Express).

- Über eine allgemeine ASCII-File- und COM-Schnittstelle bzw. über ein externes Script wird die Anbindung an beliebige andere CAE-Programme unterstützt. Die Kopplung kann der Nutzer selbst mit wenig Aufwand definieren (wenn er weiß, wie es funktioniert!).

Experimente werden im *OptiY* auf Basis eines Workflows beschrieben (noch nicht durchführen!):

- Unabhängig von konkreten Modellen definiert man die Entwurfsparameter und Bewertungsgrößen, welche für das Optimierungsexperiment von Bedeutung sind. Bei den Bewertungsgrößen unterscheidet man zwischen Restriktionen und Gütekriterien.
- Den abstrakten Elementen (Modelle und Daten) des Workflow-Diagramms ordnet man konkrete Modelle mit ihren Parametern und Ergebnisgrößen zu.
- Benutzt man in einem Experiment mehrere Modelle, so wird ihre Abarbeitungsreihenfolge über den beschriebenen Datenfluss bestimmt.



← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Experiment&oldid=27260“

Software: SimX - Nadelantrieb - Wirkprinzip - Modell

Aus OptiYummy



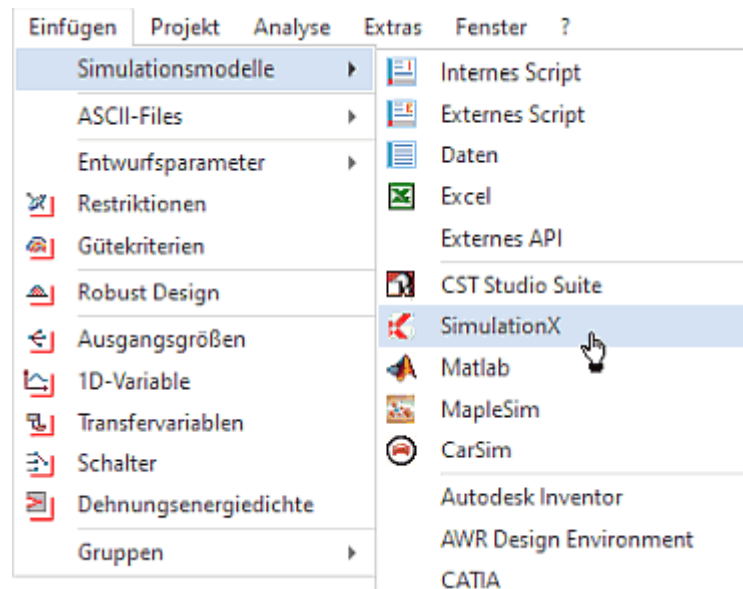
Experiment: Simulationsmodell

Zumindest in den frühen Phasen des Konstruktionsprozesses können noch keine Experimente mit dem zu konstruierenden Objekt selbst durchgeführt werden, weil es noch nicht existiert. So dienen Modelle als Versuchsobjekte, um damit Erkenntnisse für die Lösung der Konstruktionsaufgabe zu gewinnen. Dabei wird es sich meist um numerische Modelle handeln, aber auch materielle Modelle sind denkbar.

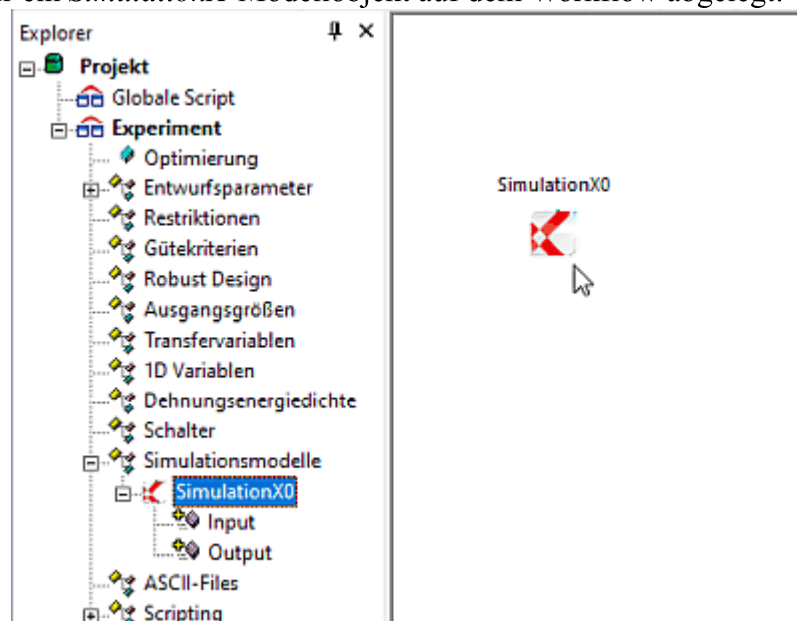


Im Folgenden wird beschrieben, wie man das validierte *SimulationX*-Modell des Nadel-Antriebs in den Experiment-Workflow des Programms *OptiY* einbindet:

- Der Menüpunkt *Einfügen* > *Simulationsmodelle* ermöglicht die Wahl des unterstützten Modell-Typs *SimulationX*:



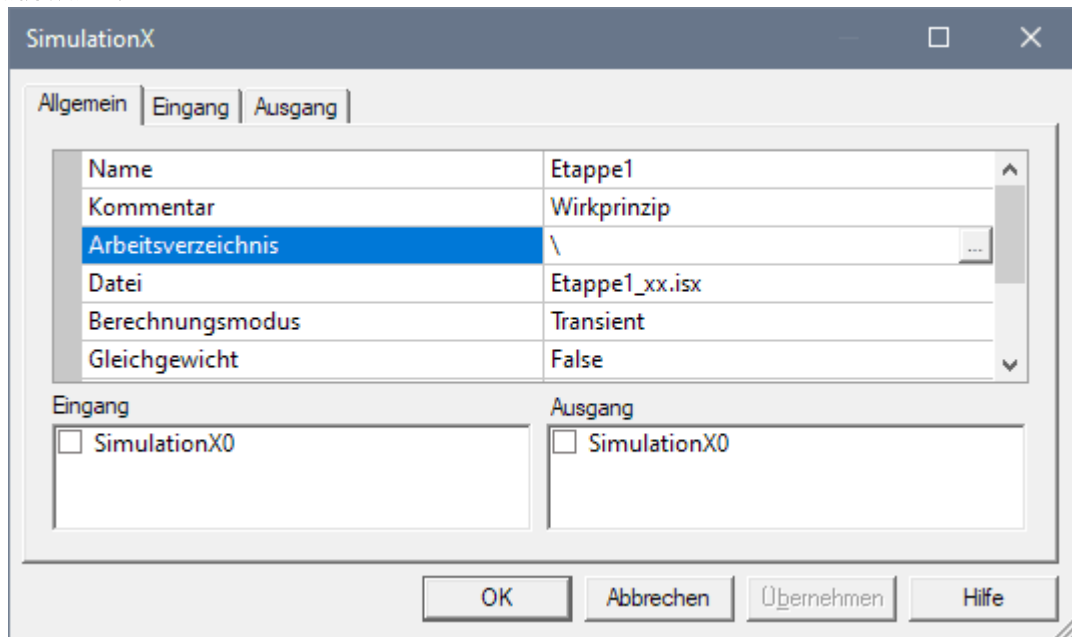
- Welches konkrete Modell sich dahinter verbirgt, wird vorläufig nicht spezifiziert, sondern es wird mittels Mausclick vorerst nur ein *SimulationX*-Modellobjekt auf dem Workflow abgelegt:



- Jedes Modellobjekt enthält Input- und Output-Container (Parameter- und Ergebnis-Interface), welche in der Baumstruktur des Experiments erscheinen.

Die Eigenschaften eines Modellobjekts kann man definieren, nachdem man mittels Doppelklick in den entsprechenden Dialog gelangt ist:

- **Hinweis:** In Hinblick auf einen definierten Modellzustand hat es sich als günstig erwiesen, das *SimulationX* zu beenden, bevor man die *SimulationX*-Projektdatei für das *SimulationX*-Modell im *OptiY*-Workflow auswählt!



- Nach der Auswahl der .isx-Datei wird das *SimulationX* gestartet und das Modell wird darin geöffnet (falls das Modell nicht schon zuvor im *SimulationX* geöffnet war).
- Name und Kommentar sollte man im *OptiY*-Workflow mit sinnvollen Begriffen versehen.
- **Wichtig:** Wir ersetzen den konkreten Arbeitsverzeichnis-Pfad des Modells durch das Zeichen \. Damit wird das Modell im gleichen Pfad erwartet, wie die *OptiY*-Projektdatei. Das *OptiY*-Projekt wird damit zwischen unterschiedlichen Computern portabel!
- Das Zuordnen von Parameter- und Ergebnis-Verbindungen ist noch nicht möglich, weil auf dem Workflow noch keine Datenobjekte definiert wurden. Wir können den *SimulationX*-Einfüge-Dialog deshalb vorläufig mit **OK** beenden.

Konfiguration speichern:

- Mittels **Datei > Speichern** sollte man den aktuellen Zustand des *OptiY*-Projektes sichern (Datei mit der Extension **.opy**).
- Den Dateinamen sollte man im Beispiel möglichst vom Modell übernehmen (**Etappe1_xx.opy**). So erkennt man später auch noch die Zuordnung zwischen Simulationsmodell und Optimierungsprojekt.
- Ein späteres Öffnen der **.opy**-Datei stellt die gesicherte Konfiguration des Projektes wieder her.

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Modell&oldid=27765“

Software: SimX - Nadelantrieb - Wirkprinzip - Entwurfparameter

Aus OptiYummy

↑

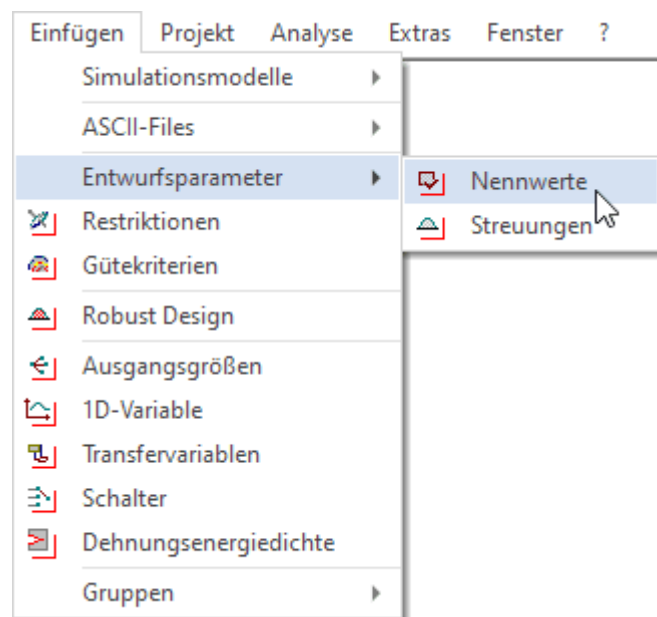
← →

Experiment: Entwurfparameter (Nennwerte)

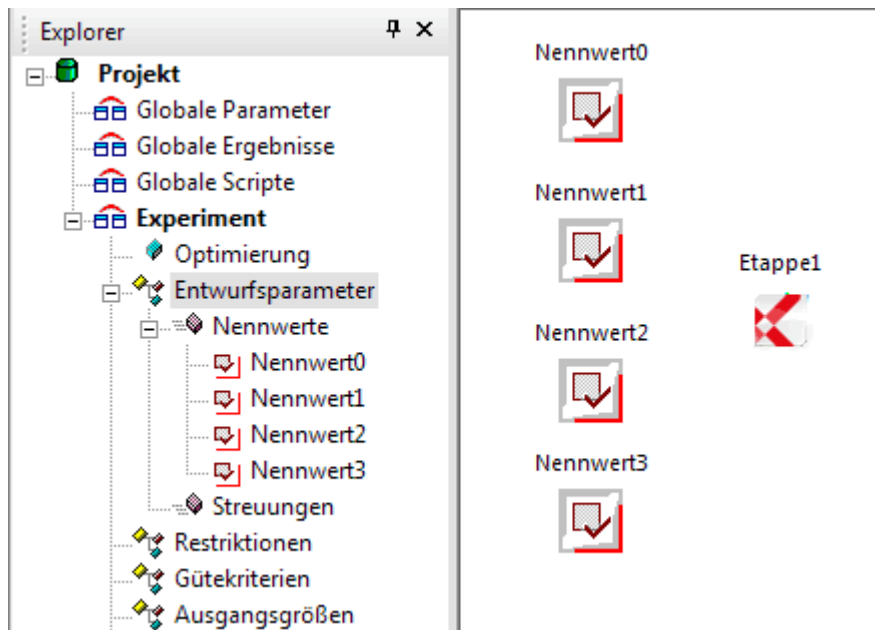
Entwurfparameter sind diejenigen Parameter des Modells, an denen das Optimierungsprogramm Veränderungen vornehmen soll, um eine optimale Lösung für die Optimierungsaufgabe zu finden. In die ersten Experimente wollen wir nur die folgenden vier Parameter einbeziehen:

- **CAD.d_Anker**
- **Feder.k**
- **Magnet.R** (Einschaltzeit)
- **Nadel.x0** (Ruhelage)

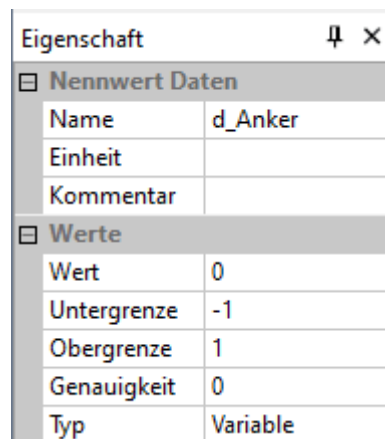
Nun definieren wir die vier Entwurfparameter im Workflow-Editor durch *Einfügen* > *Entwurfparameter* > *Nennwerte*:



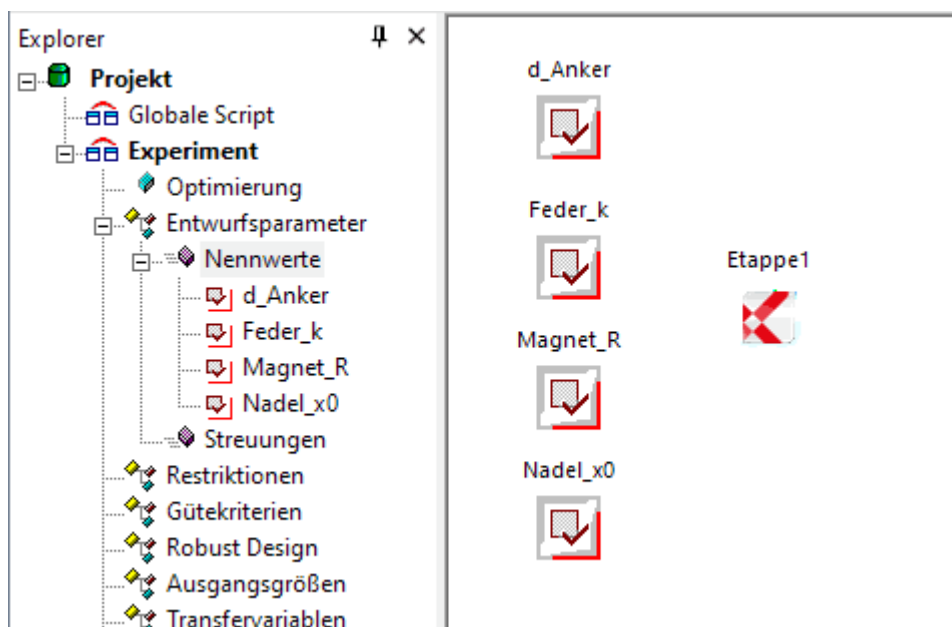
- In einem ersten Schritt erzeugt man dafür mit dem Workflow-Editor abstrakte Daten-Objekte. Diese Nennwert-Objekte kann man mittels Mausklick auf dem Workflow-Desktop ablegen:
 - Hält man dabei die **Strg**-Taste gedrückt, so kann man mehrere nacheinander ablegen.
 - Nach dem Platzieren des letzten Nennwerts beendet man den Vorgang mit der **ESC**-Taste.



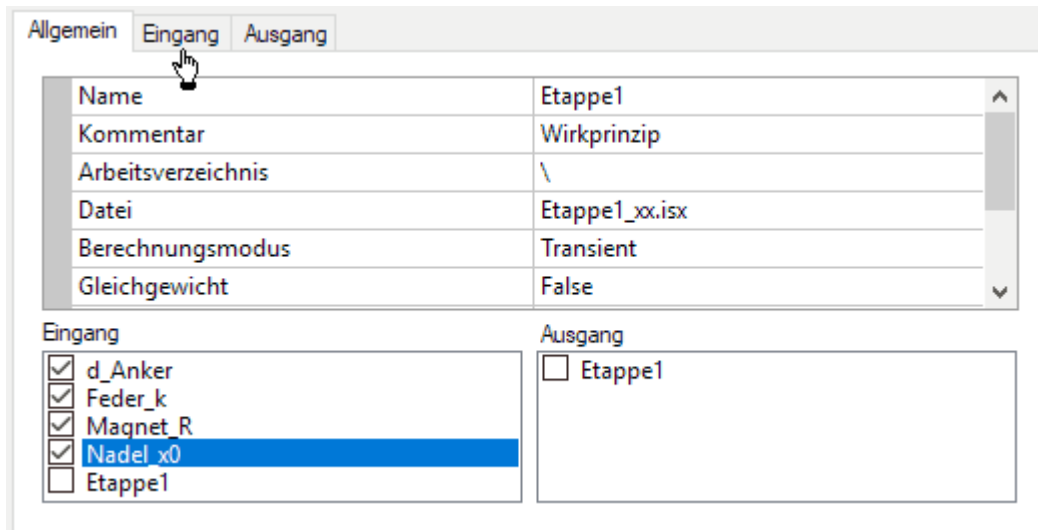
- Markierte Symbole kann man auf dem Workflow verschieben, so dass eine übersichtliche Anordnung entsteht.
- Die Standardbezeichner *Nennwert_x* sollte man durch sinnvolle Namen entsprechend der konstruktiven Entwurfsgrößen versehen:
 - Eigenschaftsdialog über Doppelklick auf Symbol im Workflow-Editor
 - oder über das Eigenschaftsfeld des Nennwerts im *OptiY*-Explorer



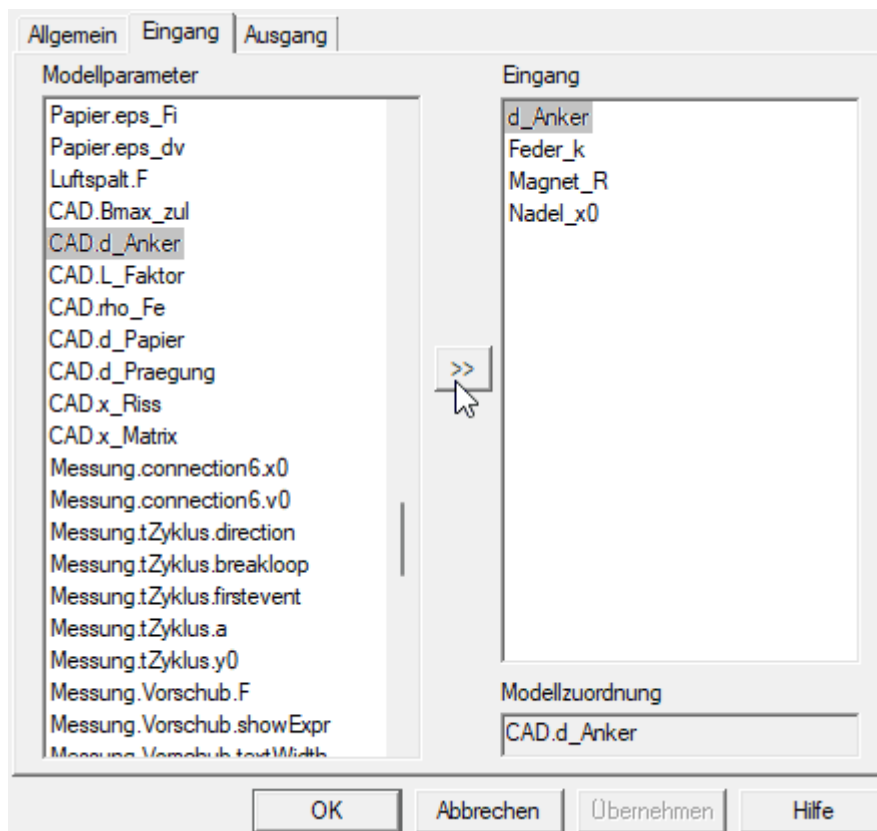
- Hinweise:** Einheiten und Kommentare der Entwurfsparameter werden im *OptiY* automatisch nach der Zuordnung zu den konkreten Parametern des *SimulationX*-Modells überschrieben! Dauerhafte Änderungen an Einheiten und Kommentare sind deshalb nur im *SimulationX*-Modell möglich.



- Die "abstrakten" Entwurfsgrößen sollen nun im Workflow den konkreten Modellparametern zugeordnet werden. Nach Doppelklick auf das *SimulationX*-Objekt erscheint wieder der Dialog zu den Modelleigenschaften:

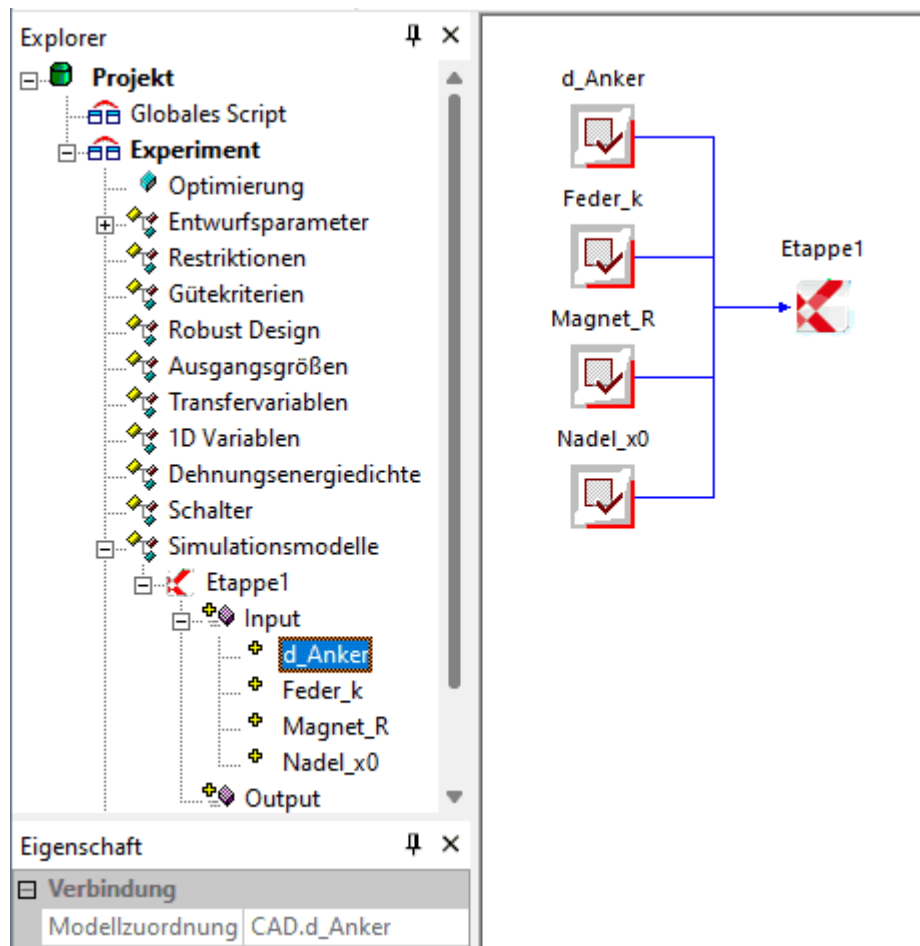


- In der Liste der möglichen Objekte für eine Parameter-Verbindung müssen wir die gewünschten Parameter markieren und wählen danach das Register **Eingang**:



- Leider ist die Liste der *SimulationX*-Modellparameter sehr unübersichtlich, weil alle auf der obersten Modellebene verfügbaren *SimulationX*-Objekte angeboten werden (auch diejenigen aus dem Compound). Es wird keine Filter- oder Sortier-Funktion angeboten werden.
- Eine Verbindung wird realisiert, indem man in den Listen den zu verbindenden Modellparameter und die zugehörige Eingangsgröße markiert. Erst nach Betätigen des Verbindungsbuttons (wie im Bild gezeigt), wird für die gewählte Eingangsgröße die Modellzuordnung hergestellt (erkennbar am Eintrag im Feld "Modellzuordnung" bei gewählter Eingangsgröße!).
- Wichtig:** Man sollte alle markierten Größen konsequent mit Modellparametern verbinden, bevor man den **OK**-Button drückt. Fehlt die Modellzuordnung für einen Eingang, so führt das später zu "unerklärlichem" Verhalten bei der Optimierung, denn die Verbindungslinie wird im *OptiY*-Workflow trotzdem generiert!

Nach Abschluss der Zuordnung erscheinen die Verbindungen im Workflow und können als Eigenschaft der Input-Größen angezeigt werden. Hier sollte man überprüfen, ob alle Zuordnungen korrekt erfolgt sind:



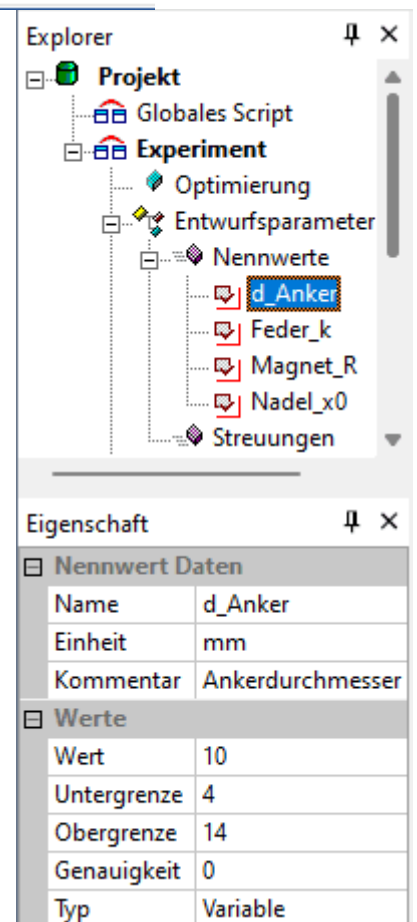
Die Eigenschaften der Entwurfparameter werden auf Basis der aktuellen Modellwerte mit Standard-Annahmen versehen:

Startwert:

- Hier sollte der jeweils im Simulationsmodell für die Ausgangslösung eingestellte **Wert** unter Berücksichtigung der für den Parameter verwendeten physikalischen Einheit verwendet werden (im Beispiel: **d_anker** in [mm]).
- Die Startwerte muss man manuell aus dem *SimulationX*-Modell übertragen (in früheren *OptiY*-Versionen erfolgte diese Übertragung automatisch bei der Verbindung der Nennwerte).

Grenzwerte:

- Die Grenzwerte beschreiben den zulässigen Bereich für die jeweilige Entwurfsgröße.
- Dafür setzen wir technisch-physikalisch sinnvolle Werte ein. Die Grenzen sollte man nicht zu eng wählen, damit die optimale Lösung innerhalb des Bereiches liegt!
- Die Breite des Bereiches wird meist durch das technisch-physikalisch Sinnvolle beschrieben. So ist z.B. ein Ankerdurchmesser von 10 cm wahrscheinlich für unseren Antrieb etwas überdimensioniert. Nach unten wird der Ankerdurchmesser durch die maximal mögliche Magnetkraft beschränkt. 1 mm wird man dabei kaum unterschreiten können.
- Wir verwenden folgende sinnvollen Grenzwerte für den Suchraum:
 - **d_Anker** = (4...14) mm
 - **Feder_k** = (0.5...100) N/mm
 - **Magnet_R** = (0.5...3) ms
 - **Nadel_x0** = (0.2...2) mm



Software: SimX - Nadelantrieb - Wirkprinzip - Restriktionen

Aus OptiYummy

↑

← →

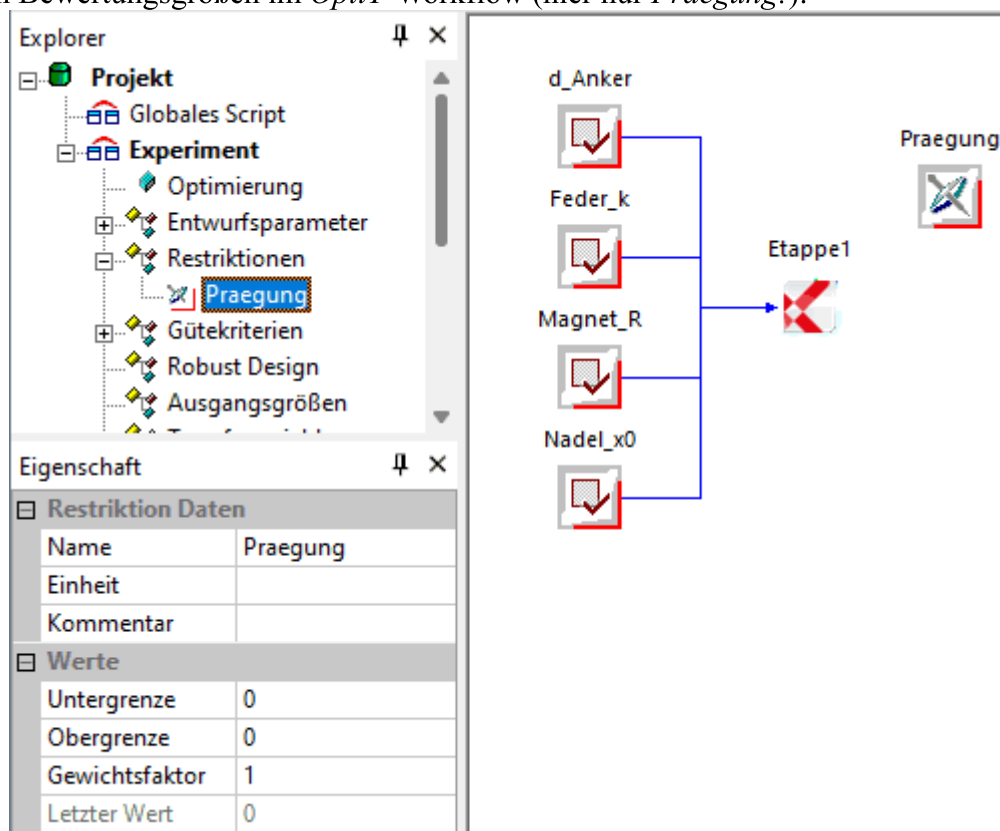
Experiment: Restriktionen (= Forderungen)

Die Erfüllung aller Forderungen (Restriktionen) wird vom Optimierungstool bei der Suche nach der optimalen Lösung mit höchster Priorität verfolgt. Erst wenn alle Forderungen der Aufgabenstellung erfüllt sind, widmet sich das Tool der Erfüllung unserer Wünsche. Wir haben den Wunsch, eine möglichst kurze Zykluszeit zu erreichen und werden in diesem Optimierungsexperiment zwei Forderungen berücksichtigen:

1. das Papier muss geprägt werden und
2. die Zykluszeit darf höchstens **3,6 ms** betragen.

Hinweis: Da der Antrieb in Hinblick auf die minimal mögliche Zykluszeit optimiert wird, ist es nicht erforderlich, die zweite Forderung explizit als Restriktion zu berücksichtigen!

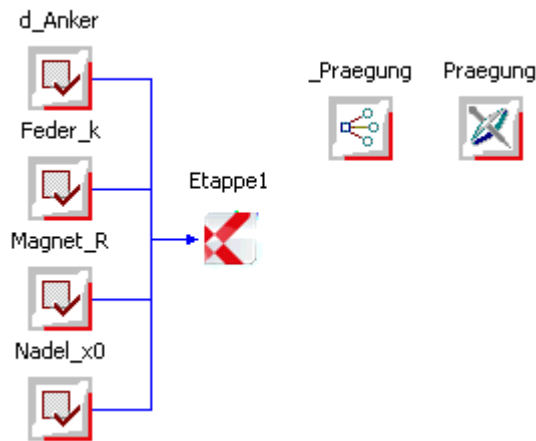
- Mittels **Einfügen > Restriktionen** definiert man für die zu berücksichtigenden Forderungen die erforderlichen Bewertungsgrößen im *OptiY*-Workflow (hier nur *Praegung*!):



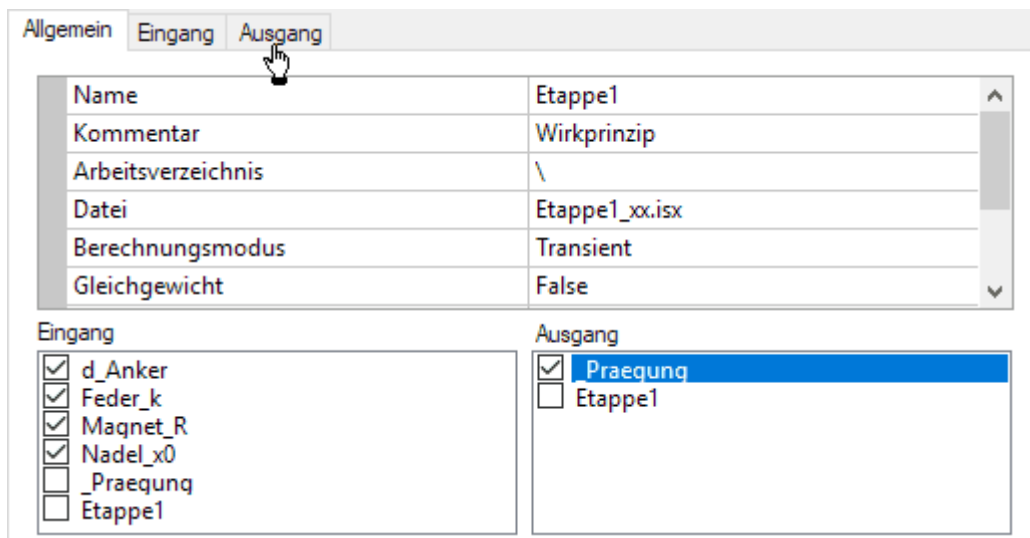
Restriktion Daten	
Name	Praegung
Einheit	
Kommentar	

Werte	
Untergrenze	0
Obergrenze	0
Gewichtsfaktor	1
Letzter Wert	0

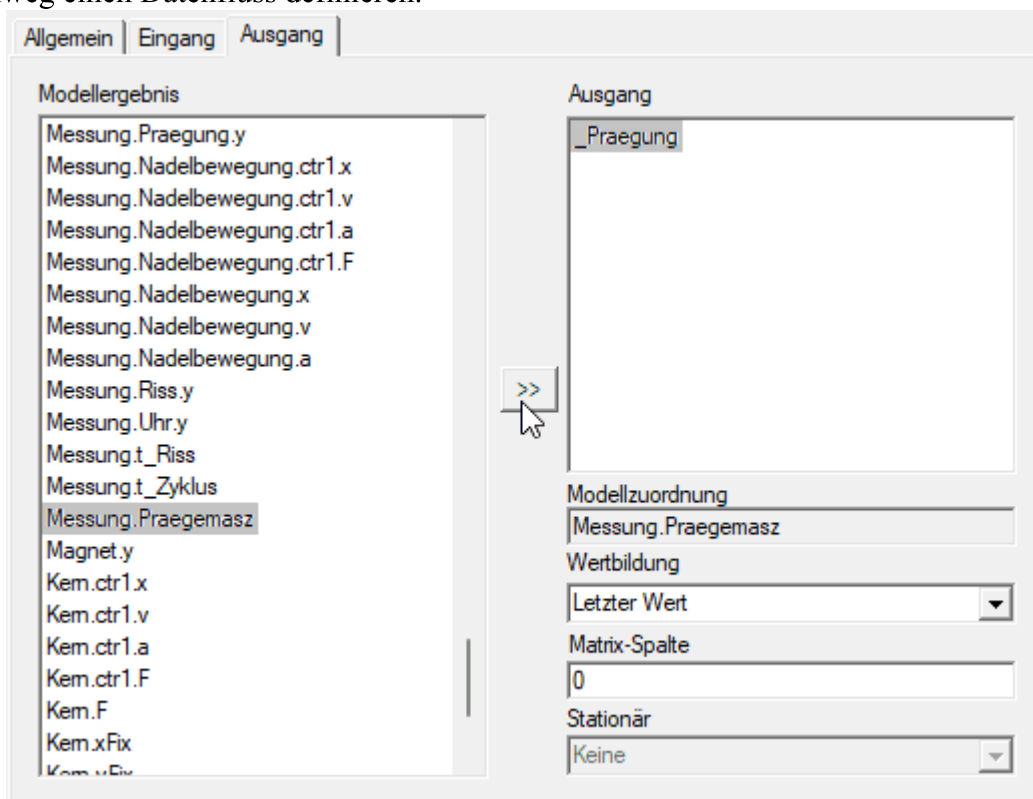
- Bewertungsgrößen (hier Restriktionen) kann man nicht direkt den konkreten Modellgrößen zuordnen. Dafür benötigt man zusätzliche Ausgangsgrößen (**Einfügen > Ausgangsgrößen**):



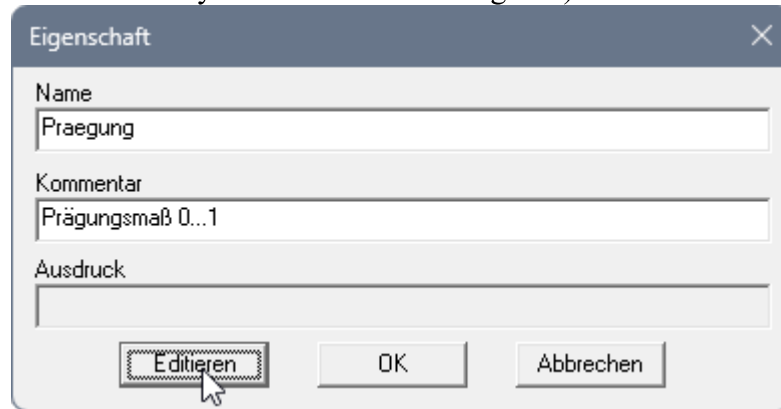
- Da Workflow-Objekte nicht gleiche Namen besitzen sollten, ist es günstig bei den Ausgangsgrößen z.B. in Hinblick auf die zugehörigen Bewertungsgrößen einen Unterstrich voranzustellen.
- Die Ausgangsgrößen kann man nun in Analogie zu den Entwurfparametern mit dem Modell-Objekt verknüpfen:



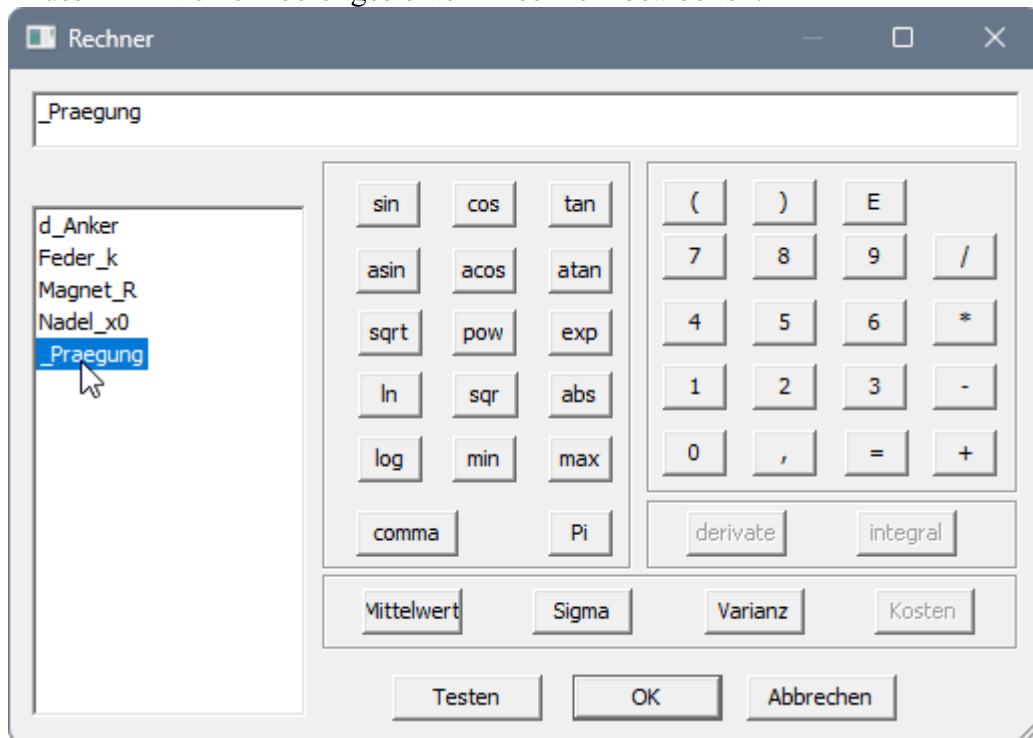
- **Hinweis:** Im Bild erkennt man, dass Ausgangsgrößen auch zum Herstellen von Parameter-Verbindungen genutzt werden können. Das macht im Beispiel keinen Sinn, man kann damit jedoch über mehrere Modelle hinweg einen Datenfluss definieren.



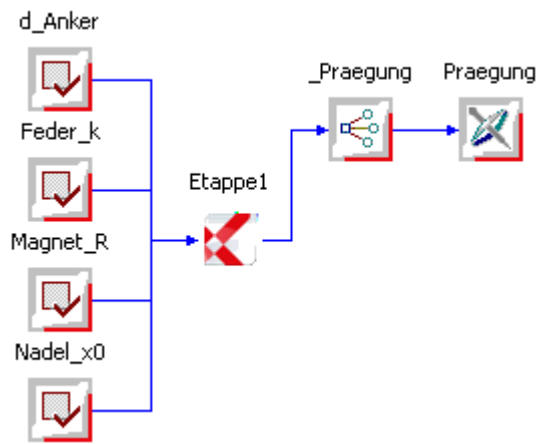
- Die Verknüpfung der Restriktionen mit den Ausgangsgrößen erfolgt über den Eigenschaftsdialog der Restriktionen (Doppelklick auf das Symbol der Restriktionsgröße):



- Da für Restriktionsgrößen keine Kommentare aus dem Modell übernommen werden, kann man einen solchen gleich hier im Eigenschaftsdialog eintragen.
- Der Wert einer Bewertungsgröße kann durch mathematische Verknüpfung aller Input- und Output-Größen des Versuchsstandes gebildet werden.
- Diese Verknüpfung wird in Form eines Ausdrucks notiert. Den Ausdruck kann man nicht direkt eintragen, sondern man muss ihn mit einem bereitgestellten "Rechner" bearbeiten:



- Der "Rechner" bietet den Zugriff auf alle Input- und Output-Objekte des Experiment-Workflows:
 - Nach Doppelklick auf die gewünschte Größe erscheint diese am Ende des aktuellen Ausdrucks.
 - Im Beispiel ist nur die direkte Zuweisung der entsprechenden Output-Größe erforderlich.
 - "Testen" berechnet den aktuellen Wert des Ausdrucks (im Beispiel=0).
 - Beachte:** Auf keinen Fall darf man im Editor den Ausdruck manuell mittels der Tastatur schreiben, weil dann keine Zuordnung zu den gewünschten Workflow-Objekten erfolgt!
- Nach dem Bearbeiten der Bewertungsgröße werden die Verknüpfungen zu den anderen Workflow-Größen visualisiert:



Achtung:

- Bei dem Maß für die *Praegung* handelt es sich um einen Spezialfall einer Restriktion, indem praktisch nur der *Wert=1* als zulässig angesehen wird.
- Um numerische Probleme zu vermeiden, setzt man zwar *Untergrenze=1*, aber die Obergrenze erhält einen etwas größeren Wert (z.B. *Obergrenze=1.1*).

← →

Explorer

- Projekt
 - Globales Script
 - Experiment
 - Optimierung
 - Entwurfparameter
 - Restriktionen
 - Praegung**
 - Gütekriterien
 - Robust Design
 - Ausgangsgrößen
 - Transfervariablen

Eigenschaft

Restriktion Daten	
Name	Praegung
Einheit	
Kommentar	Prägungsmaß 0...1
Werte	
Untergrenze	1
Obergrenze	1.1
Gewichtsfaktor	1
Letzter Wert	0

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Restriktionen&oldid=27766“

Software: SimX - Nadelantrieb - Wirkprinzip - Kriterien

Aus OptiYummy

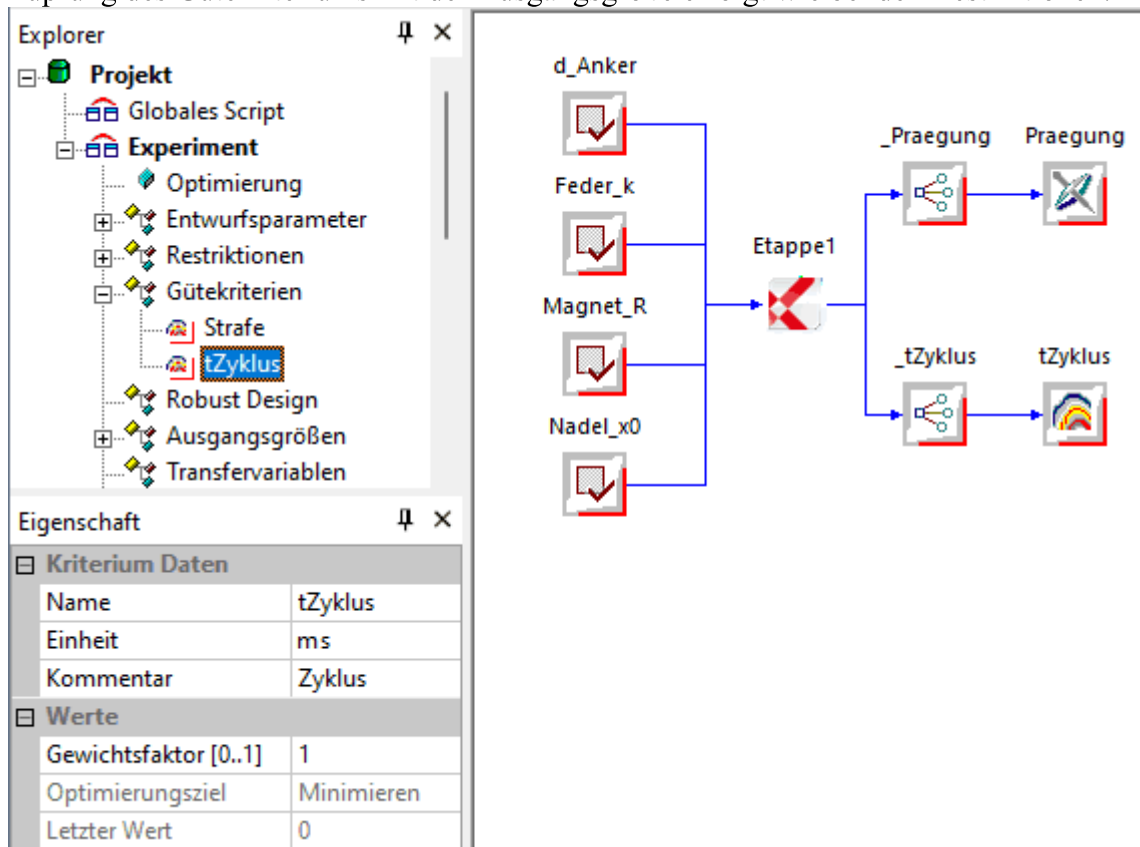
↑

← →

Experiment: Gütekriterien (= Wünsche)

Zulässige Lösungen sind gekennzeichnet durch die Erfüllung aller Forderungen ((hier das "Prägen"). Das Erreichen einer zulässigen Lösung wird das erste Ziel für das Optimierungstool während des Experiments sein. Danach kann es sich das Optimierungstool der Erfüllung von Wünschen widmen, um der idealen Lösung möglichst nahe zu kommen:

- Wir haben in diesem Experiment nur einen Wunsch - die Zykluszeit soll so kurz wie möglich werden.
- Dafür muss ein Gütekriterium in den Experiment-Workflow eingefügt werden. Eine zusätzliche Ausgangsgröße ist auch hier erforderlich, da Gütekriterien nicht direkt einem Modell zugeordnet werden können.
- Die Verknüpfung des Gütekriteriums mit der Ausgangsgröße erfolgt wie bei den Restriktionen.



Maßeinheiten in OptiY:

Im Unterschied z.B. zum *SimulationX* werden physikalische Einheiten im *OptiY*-Workflow nur als Zeichenketten im Sinne eines "Kommentars" berücksichtigt:

1. Von *OptiY* wird der Inhalt dieser Zeichenketten nicht überprüft.
2. Die Maßeinheiten besitzen keinen Einfluss auf die Berechnungen der Zahlenwerte innerhalb des Experiment-Workflows.
3. Werden Maßeinheiten von Workflow-Größen nachträglich geändert, ist man selbst dafür verantwortlich, dass die Maßeinheiten bei Berechnungen innerhalb des Workflows richtig berücksichtigt werden.



Eine Workflow-Größe erhält ihre Maßeinheit (als Zeichenkette) automatisch bei der direkten Zuordnung zu einer anderen Workflow- oder Modell-Größe:

1. Restriktionen und Gütekriterien erhalten keine Maßeinheit, weil keine direkte Zuordnung zu den Ausgabegrößen erfolgt. Es erfolgt eine indirekte Zuordnung über den "Rechner", welcher keine Maßeinheiten berücksichtigt. Die Einheit **ms** für *tZyklus* muss also manuell eingetragen werden!
2. Die Übernahme der Maßeinheiten von Input-/Output-Größen eines Modells kann nur dann automatisch erfolgen, wenn die Übernahme vom Modell-Interface unterstützt wird (z.B. beim *SimulationX*).
3. Die Zahlenwerte der Input-/Output-Größen eines Modells richten sich nach den Vorgaben des Modells (in *SimulationX* z.B. durch Wahl der Einheit einer Länge in mm).

Die Maßeinheiten der Input-/Outgrößen sollte man im *SimulationX*-Modell nicht mehr ändern, wenn das *SimulationX*-Modell bereits in einen *OptiY*-Workflow eingebaut wurde.

1. Veränderte Input-Einheiten führen zu einer fehlerhaften Interpretation der in das Modell eingespeisten Parameterwerte (z.B. Entwurfgröße weiterhin in *Millimeter* und Modellparameter in *Meter*).
2. Veränderte Output-Einheiten bewirken, dass das Modell trotz gleicher Konfiguration andere Zahlenwerte für die Ausgangsgrößen des Workflows bereitstellt.

Hinweis zur "Strafe":

- Unter den Gütekriterien erscheint als separater Eintrag eine "Strafe".
- Die Straf-Funktion wird automatisch aus den gewichteten Restriktionsverletzungen gebildet.
- Das Optimierungstool versucht zuerst eine Lösung für Strafe=0 zu finden. Das entspricht einer im Sinne der Forderungen zulässigen Lösung.
- Erst im Bereich zulässiger Lösungen werden die vom Nutzer definierten Gütekriterien bei der Suche nach der optimalen Lösung berücksichtigt. Dies entspricht einer Erfüllung von Wünschen.
- Man spricht hierbei von "hierarchischer Optimierung".

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Kriterien&oldid=27768“

Software: SimX - Nadelantrieb - Wirkprinzip - Optimierungsverfahren

Aus OptiYummy

↑

← →

Experiment: Optimierungsverfahren

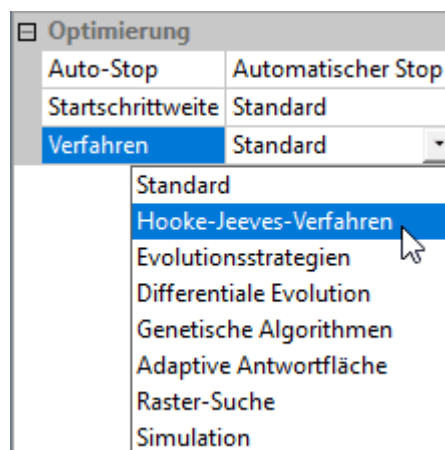
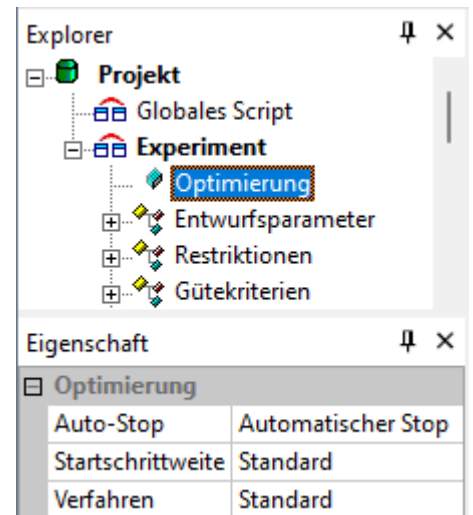
Bisher haben wir unser Entwurfsproblem vor allem inhaltlich in eine Optimierungsaufgabe transformiert. Es fehlt nun noch die Konfiguration der "Numerik", d.h.: "Mit welchem Optimierungsverfahren soll die optimale Lösung gefunden werden?"

Leider ist das "optimale" Optimierungsverfahren und seine "optimale" Konfiguration abhängig von der Optimierungsaufgabe! Ausgehend von einer Standard-Konfiguration ist es meist günstig, einen gewissen Aufwand in den Abgleich des numerischen Verfahrens zu investieren. Das werden wir nun am Beispiel des Nadelantriebs üben:

- OptiY bietet für den unerfahrenen Nutzer ein Standard-Verfahren an. Welches Verfahren sich dahinter verbirgt, ist abhängig von der Anzahl der Entwurfsparameter **P** und Gütekriterien **K**:
 - Evolutionsstrategie bei $K > 1$ oder $P > 9$,
 - Hook-Jeeves-Verfahren für $K < 2$ und $P < 10$

In unserem Beispiel würde also das Hook-Jeeves-Verfahren genutzt. Dieses tastet sich auf der Zielfunktion von der Ausgangslösung schrittweise in Richtung kleinerer Werte "bergab" und endet dort in der Talsohle - in der Hoffnung, dass dies das globale Optimum ist:

- Die Option "Automatischer Stop" soll die Suche beenden, wenn das Minimum erreicht wurde. Das dafür implementierte Kriterium funktioniert meist zuverlässig!
- Um an dieser Stelle zumindest einen Einblick in die Konfiguration eines Optimierungsverfahren zu erhalten, wählen wir das Hook-Jeeves-Verfahren:



- Die individuelle Abtast-Schrittweite für jeden Entwurfsparameter ist in der Standardeinstellung u.a. von deren vorgegebenem Variationsbereich [Untergrenze, Obergrenze] abhängig. Da dies numerisch nicht immer günstig sein muss, sollte man sich den Zugriff auf diese Abtast-Schrittweite ermöglichen, indem man *Startschrittweite=manuell* setzt.

- Die Startschrittweite erscheint dann als zusätzliche Eigenschaft nach Wahl des jeweiligen Entwurfsparameters im OptiY-Explorer.
- Die Abtastschrittweite für den aktuellen Wert des Entwurfsparameters darf innerhalb des vorgegebenen Suchbereiches [Untergrenze .. Obergrenze] nicht zu klein und nicht zu groß werden, damit die damit die Steigung an der Abtaststelle hinreichend genau berechnet werden kann.
- Aus Erfahrung sollte man bei der Bestimmung der Startschrittweite in folgender Reihenfolge vorgehen:

1. Anfangswert der Entwurfsgröße → **Startschrittweite** \approx **Wert/50**
2. Überprüfen dieser Schrittweite für die Obergrenze → **Startschrittweite** \geq **Obergrenze/1000?**
3. Ist die Schrittweite für die Obergrenze zu klein, dann **Startschrittweite** \approx **Obergrenze/1000**

- Dies soll am Beispiel der Rückholfeder verdeutlicht werden, welche mit einem relativ kleinen Anfangswert im Verhältnis zur geschätzten Obergrenze beginnt:

Eigenschaft		⌵ ×	Eigenschaft		⌵ ×
Nennwert Daten			Nennwert Daten		
Name	Feder_k		Name	Nadel_x0	
Einheit	N/mm		Einheit	mm	
Kommentar	Steifigkeit		Kommentar	Anfangsweg	
Werte			Werte		
Wert	0,761544		Wert	2	
Startschrittweite	0.1		Startschrittweite	0,04	
Untergrenze	0.5		Untergrenze	0,2	
Obergrenze	100		Obergrenze	2	
Genauigkeit	0		Genauigkeit	0	
Typ	Variable		Typ	Variable	

- Hier ergibt sich bei der beschriebenen Vorgehensweise eine Startschrittweite von **0.1 N/mm** (**1/1000** von **100 N/mm**). Auch wenn die Abtastschrittweite für den Anfangswert etwas zu groß ist, bei Bedarf verkleinert das Optimierungsverfahren die Abtastschrittweite bei mangelhafter Konvergenz!
- Analog dazu sind die Abtastschrittweiten für die anderen Entwurfsgrößen zu wählen, wie dies beispielhaft für **Nadel_x0** gezeigt ist.
- **Hinweis:** Ergeben sich laut Berechnung "krumme" Werte für die Startschrittweite, kann man auf nächstliegende anschauliche Werte runden (da es sich sowieso nur um grobe Schätzwerte handelt!).

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Optimierungsverfahren&oldid=27301“

Software: SimX - Nadelantrieb - Wirkprinzip - Visualisierung

Aus OptiYummy

↑

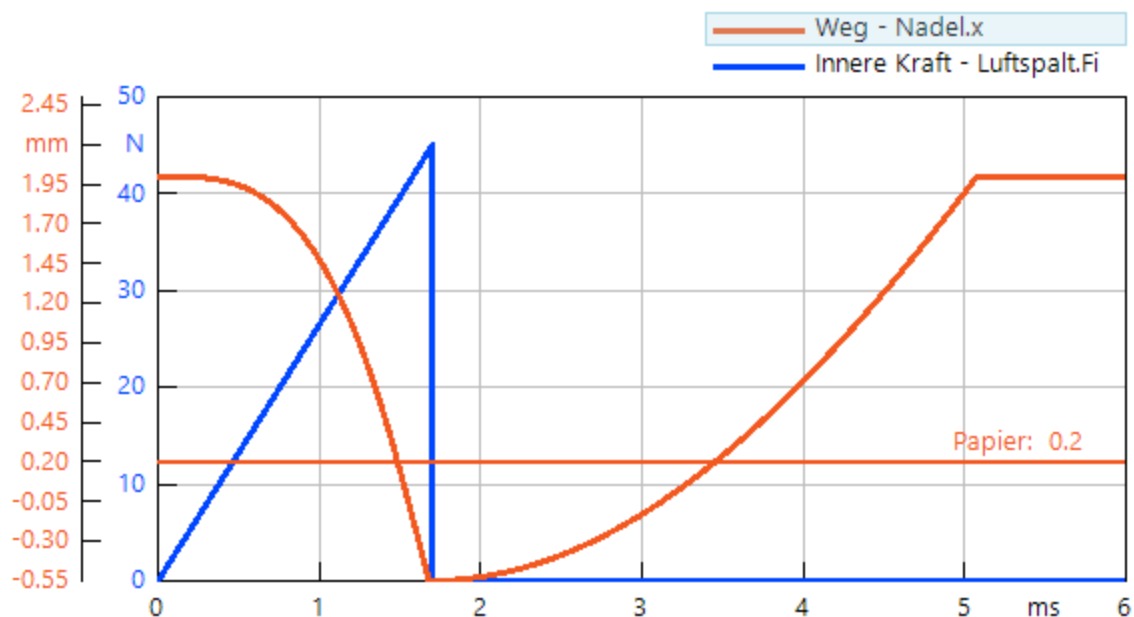
← →

Experiment: Visualisierung

Um das Optimierungsexperiment verfolgen und beurteilen zu können, muss man alle dafür wesentlichen Informationen in geeigneter Form visualisieren. Dazu gehören auch Signalverläufe des Simulationsmodells:

1. SimulationX-Modell

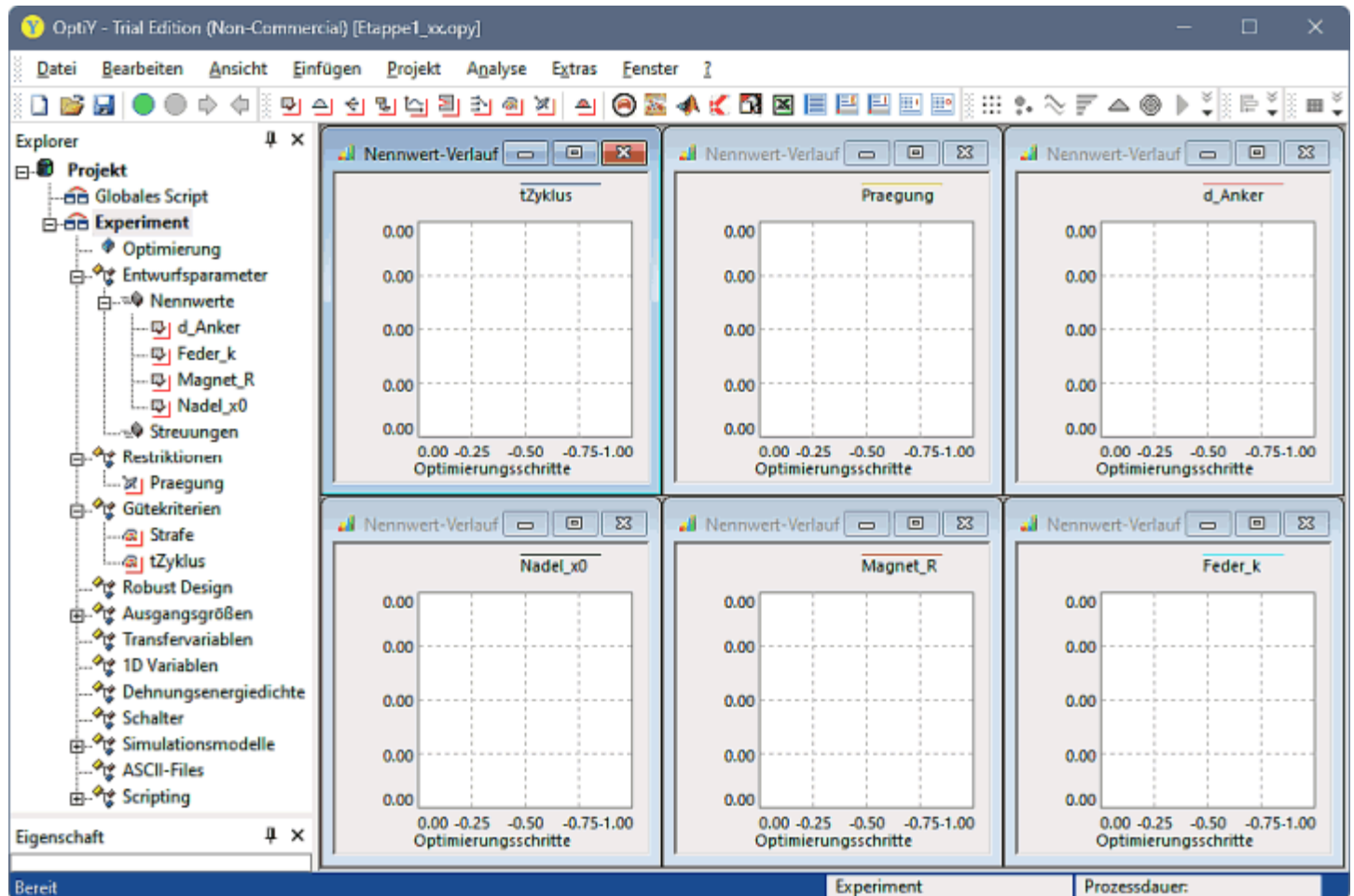
- Während des Optimierungsexperimentes werden die Entwurfparameter für das Antriebsmodell in Abhängigkeit vom konfigurierten Optimierungsverfahren verändert.
- Im *OptiY* kann man nur die Output-Größen des Modells als Ergebniswerte des aktuellen Simulationslaufes darstellen. Ob das zeitliche Verhalten des Modells wirklich sinnvoll ist, wird damit nicht unbedingt sichtbar.
- Im *SimulationX* sollte man möglichst ein Ergebnisfenster konfigurieren, welches alle zur Beurteilung des Modellverhaltens relevanten Signalverläufe enthält. Das sind z.B. die folgenden Signale:



- Alle nicht mehr benötigten Signalfenster sollte man "zuklappen" (wegen der Übersichtlichkeit und dem Zeitbedarf für das Zeichnen!).
- Für die zur Experiment-Visualisierung gewählten Signale sollte man feste Grenzen für die Koordinatenachsen einstellen, um ein ständiges Flackern der Darstellung zu vermeiden.
- Das Fenster mit den Simulationssignalen sollte so auf dem Bildschirm platziert werden, dass es neben dem *OptiY*-Fenster sichtbar bleibt.
- **Achtung:** Die Simulationszeit *tStop* muss so groß gewählt werden, dass jeder Simulationslauf sicher einen kompletten Prägezyklus berechnet. Im Beispiel genügt ein gewisser Zeit-Zuschlag zur Ausgangslösung.
- **Wichtig:** Damit die Änderungen im *SimulationX*-Modell erhalten bleiben, muss man dieses speichern!

2. OptiY-Experiment

- Zumindest bei einer überschaubaren Anzahl von Experiment-Größen sollte man alle variablen *Entwurfsparameter*, *Restriktionen* und *Gütekriterien* per *Drag & Drop* vom Explorer in den *OptiY*-Grafik-Bereich ziehen:
 - Es ist günstig, zuvor das Fenster des Workflow-Editors zu schließen.
 - Es ist auch möglich, mehrere Größen in einem Diagramm darzustellen, wenn man sie in ein vorhandenes Fenster zieht.
 - Die Fenster mit den Nennwert-Verläufen sollte man nebeneinander anordnen:



3. Konfiguration speichern

Spätestens jetzt sollte man die aktuelle Konfiguration des *OptiY*-Versuchsstandes mal wieder als *.opy*-Datei speichern!

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Visualisierung&oldid=27773“

Software: SimX - Nadelantrieb - Wirkprinzip - Lokale Suche


Aus OptiYummy

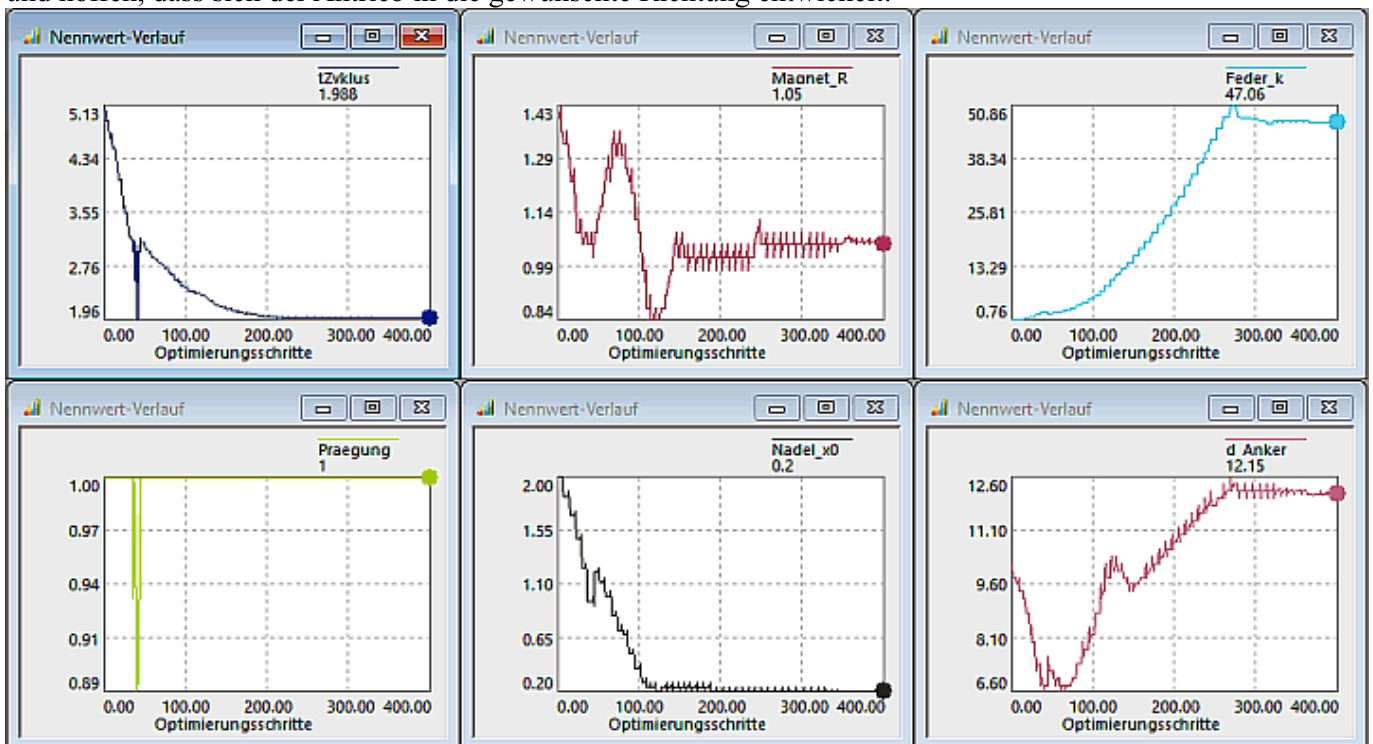
↑


← →

Experiment: Lokale Suche (Hooke-Jeeves)


Wichtig: Das *SimulationX* sollte jetzt mit dem soeben erfolgreich simulierten Modell geöffnet sein!

- Nach so viel Vorbereitung wollen wir nun im *OptiY* beherzt den Start-Button  für das Experiment drücken und hoffen, dass sich der Antrieb in die gewünschte Richtung entwickelt:



- Wenn keine Verbesserung der Ergebnisse mehr zu erwarten ist, können wir das Experiment stoppen  (im Beispiel nach 400 Optimierungsschritten).
- Die Verkürzung der Zykluszeit wird vor allem erreicht durch eine Vergrößerung der Steifigkeit der Rückholfeder mit entsprechender Anpassung der anderen Antriebsparameter.
- Man kann sich die exakten Werte eines Optimierungsschrittes anzeigen lassen. Nach Doppelklick mit der linken Maustaste auf den gewünschten Kurven-Punkt wird dieser Punkt auf allen Kurven markiert und es werden die zugehörigen Werte eingeblendet. Ein Doppelklick abseits jeglicher Kurve blendet die Markierungen wieder aus.

Hinweise zur Fehlersuche:


Erkennt man ein fehlerhaftes Verhalten des Modells während der Optimierungsrechnung, so muss man das Experiment stoppen :

- Im *SimulationX* den Simulationslauf auf den Anfang zurücksetzen, damit ein definierter Beginn des nächsten Simulationslaufes stattfindet.
- Das Modellverhalten kann man dann im *SimulationX* mit den aktuell vom *OptiY* eingespeisten Nennwerten untersuchen.

Korrektur von Fehlern im *SimulationX*-Modell bei aktivem *OptiY*-Versuchsstand kann zum Absturz von *OptiY* führen, da die Verknüpfungen zwischen beiden Programmen dadurch fehlgeleitet werden:

- Vor der Fehler-Korrektur muss man *OptiY* beenden. Dabei bleibt das im *SimulationX* benutzte Modell offen. Da es mit den Parametern des letzten Simulationslaufes belegt ist, sollte man es schließen (ohne speichern!).
- Lädt man das Modell erneut im *SimulationX*, so besitzt es die Ausgangslösung als Parametersatz.
- Nach Korrektur der Modellfehler speichert man den neuen Modellzustand und schließt das Modell.
- In *OptiY* steht nach erneutem Start das korrigierte Modell im Experiment-Workflow zur Verfügung.

Typische Fehler-Situationen:

- **Das Modell reagiert unglaublich auf Nennwert-Änderungen:**
 - Hat z.B. die Änderung des Ankerdurchmessers keinen Einfluss auf die Bewegung, so fehlt die Verbindung zwischen dem *OptiY*-Workflow und dem Modellparameter oder etwas kann im Modell nicht stimmen!
 - Erforderliche Modell-Änderungen muss man wie zuvor beschrieben vornehmen.
- **Es wird keine Verbesserung der Anfangslösung erreicht:**
 1. **Das Modell rechnet zu ungenau**, so dass durch die Abtastschritte kein Anstieg auf der Gütefunktion erkannt werden kann (Numerisches Rauschen):
 - Wechseln in das *SimulationX*.
 - Öffnen des Einstellungsdialogs für die Simulation.
 - Verringern von *absTol* und *relTol* z.B. um 1 Zehnerpotenz auf **1E-6**. (Achtung: höhere Genauigkeiten führen teilweise zu wesentlich längeren Rechenzeiten!).
 - Verringern von *dtMin* auf z.B. **1E-12** (wesentlich kleinere Werte können zu numerischen Problemen führen!).
 - In *OptiY* die Optimierung fortsetzen.
 2. **Ungünstige Abtastschrittweiten** für Entwurfsgrößen:
 - Die Startschrittweite für die Abtastung wurde im Beispiel zu 1/50 des Startwertes (oder größer) gebildet.
 - Reagiert das Modellverhalten sehr empfindlich auf eine Änderung des betreffenden Entwurfsparameters, so kann hier ein kleinerer Wert sinnvoll sein.
 - Umgekehrt kann es auch sinnvoll sein, diese Startschrittweite zu vergrößern, falls die geringen Auswirkungen der Änderung ansonsten im numerischen Rauschen untergehen.
 - Optimierung mit der veränderten Abtastschrittweite fortsetzen .
- **Entwurfsparameter konvergiert gegen seinen Grenzwert:**
 - Falls der Grenzwert nicht infolge einer Forderung festgelegt wurde (z.B. max. Einbaumaß) oder nicht der Physik widerspricht (z.B. negative Einschaltzeiten), sollte man die Grenze des Entwurfsparameters ändern.
 - Ein Rücksetzen und erneutes Starten der Optimierung führt dann mit großer Wahrscheinlichkeit zu einer besseren Lösung.
- **Die Lösung verschlechtert sich konsequent:**
 - Es werden im Modell fehlerhafte Bewertungsgrößen berechnet oder
 - es wurden die Forderungen der Aufgabenstellung fehlerhaft in Form der Restriktionen oder Gütekriterien definiert.
- **Konvergenz zum Optimum vorzeitig beendet:**
 - Bleiben nach einer gewissen Zeit alle Nennwerte konstant, so kann man noch nicht sicher sein, wirklich ein Optimum erreicht zu haben.
 - Im Beispiel wird die Zielfunktion um das Optimum so flach, dass aufgrund des numerischen Rauschens das Optimierungsverfahren keine Richtungsinformation mehr gewinnen kann.
 - Man sollte in jedem Fall versuchen, eine Optimierung ausgehend von den gleichen Anfangswerten, mit einer höheren Rechengenauigkeit durchzuführen (*absTol* und *relTol*).
 - Erreicht man damit den gleichen Bestwert, so handelt es sich mit großer Wahrscheinlichkeit um eine optimale Lösung.
 - Ist dies nicht der Fall, kann man die Rechengenauigkeit weiter erhöhen (falls die Berechnungszeit dies noch zulässt!).

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Lokale_Suche&oldid=27775“

Software: SimX - Nadelantrieb - Wirkprinzip - Auswertung

Aus OptiYummy

↑

← →

Experiment: Lokale Suche (Auswertung)

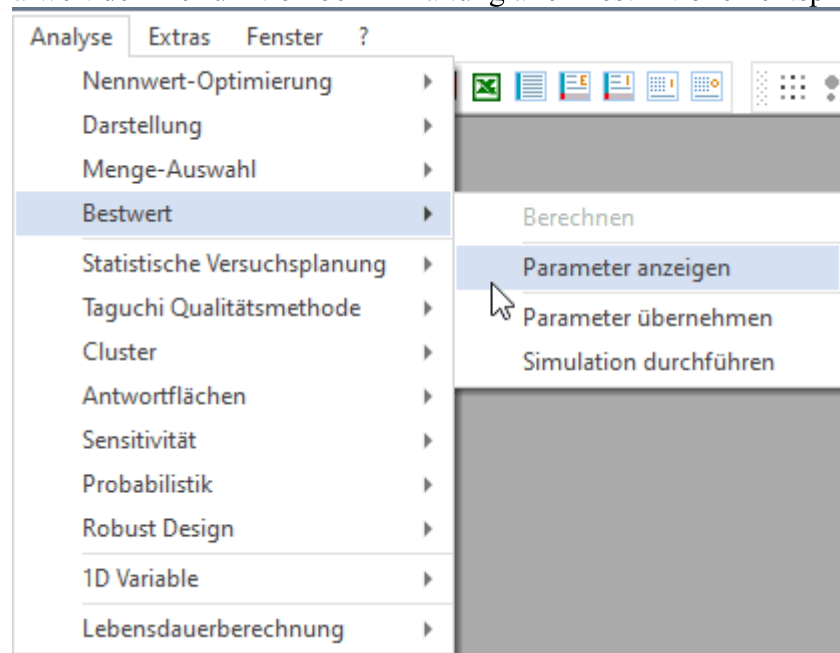
Experimente sollen dem Erkenntnisgewinn dienen. *OptiY* bietet eine Menge unterschiedlichster Analyse-Möglichkeiten für die Auswertung der im Optimierungsexperiment anfallenden Ergebnisdaten:

1. Während der Experimentdurchführung

- Die verfügbare Computerhardware bietet während des Optimierungsprozesses noch Zeit zum Nachdenken. Durch Beobachtung der schrittweisen Entwicklung der Entwurfparameter zieht man Schlussfolgerungen zu ihrem Einfluss auf das Modellverhalten.
- Man erkennt, durch welche Änderungen man recht einfach eine Verbesserung des Modellverhaltens erzielen kann (z.B. Ruhelage der Nadel in Richtung Papier verschieben).
- Man erkennt, ab welcher Güte des Verhaltens kaum noch Verbesserungen zu erwarten sind.

2. Nach dem Optimierungslauf

- Wurde eine hinreichende Verbesserung der Ausgangslösung erreicht, existieren mehrere Möglichkeiten zur Übernahme einer optimalen Lösung:
- **Aktueller Bestwert:** Bestwert ist die Kombination von Entwurfparametern, welche dem bisher erreichten Minimalwert der Zielfunktion bei Einhaltung aller Restriktionen entspricht.



1. Parameter anzeigen:

- zeigt in Form einer Tabelle die bisher beste Kombination aller Nennwerte
- diese Anzeige ändert weder im Modell noch im Experiment die Belegung der Nennwerte

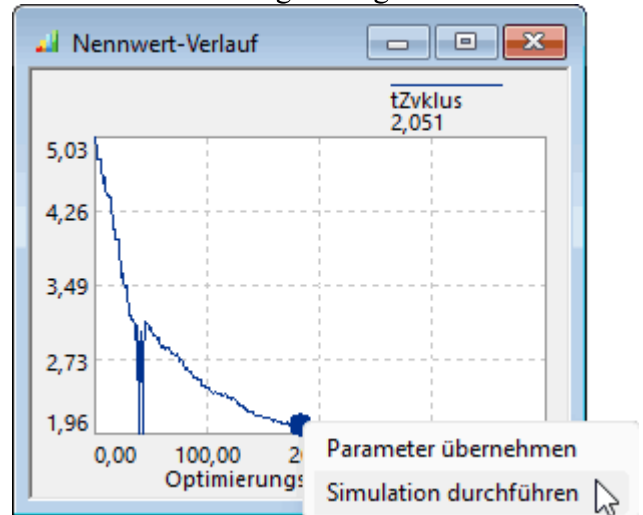
2. Simulation durchführen:

- mit den Entwurfparametern des Bestwerts wird ein Modell-Lauf durchgeführt
- danach stehen diese Werte als aktuelle Modellparameter im *SimulationX*
- die Simulation ändert nicht den aktuellen Wert der Nennwerte im *OptiY*-Experiment

Name	Werte	Einheit	Kommentar
d_Anker	12,15	mm	Ankerdurchmesser
Feder_k	47,0663	N/mm	Steifigkeit
Magnet_R	1,05875	ms	Anstiegsbreite
Nadel_x0	0,2	mm	Anfangsweg
Praegung	1		Prägungsmaß 0...1
tZyklus	1,98811	ms	Zyklus

3. Parameter übernehmen:

- die Entwurfsparameter des Bestwerts sind danach Bestandteil der Modellparameter des Experiments
 - und werden gleichzeitig zum neuen Startwert für die weitere Optimierung
 - **Hinweis:** Im Normalfall sollte man die Übernahme des Bestwertes vermeiden! Die bisherige Ausgangslösung würde damit überschrieben und es ist nicht mehr direkt nachvollziehbar, wie man zur neuen Lösung kam.
- **Ausgewählter Optimierungsschritt:**
 - **Hinweis:** Der numerisch ermittelte Bestwert muss nicht unbedingt eine optimale technische Lösung darstellen! In unserem Beispiel deutet sich bereits dieses grundlegende Problem an:
 - Am Ende der Optimierung wird nur noch eine unwesentliche Verbesserung der Zykluszeit erreicht.
 - Dafür werden aber alle Entwurfsparameter soweit es geht, an die Grenzen des Möglichen gestellt.
 - Im Sinne einer robusten Lösung ist dies zumindest durch Analysen zu hinterfragen!
 - Nach Markieren eines interessierenden Lösungsschrittes gelangt man über die rechte Maustaste in das erforderliche Kontext-Menü für den Zugriff auf die Entwurfsparameter dieses Schrittes.
 - Die Wirkung der Menü-Funktionen ist analog zur Bestwert-Behandlung.
 - Auch hier sollte man auf die Übernahme der Entwurfsparameter als neue Anfangslösung für das Optimierungsexperiment verzichten!
 - Die Nennwerte der interessierenden Workflow-Größen aller Optimierungsschritte kann man sich über **Analyse > Nennwert-Optimierung > Nennwert-Tabelle** auflisten lassen:



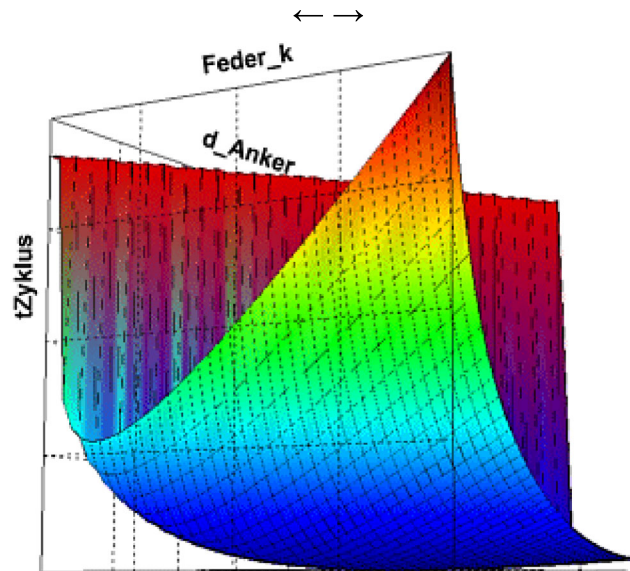
No	d_Anker	Feder_k	Magnet_R	Nadel_x0	Praegung	tZyklus	Strafe	Status
30	8	1,853772	1,12	1,44	1,00000005	3,47639512	0	Ok
31	7,2	2,253772	1,05	1,24	1,00000006	3,20441937	0	Ok
32	7,4	2,253772	1,05	1,24	1,00000006	3,19021528	0	Ok
33	7,4	2,353772	1,05	1,24	1,00000006	3,16670938	0	Ok
34	7,4	2,353772	1,085	1,24	1,00000006	3,13609025	0	Ok
35	7,4	2,353772	1,085	1,28	1,00000006	3,15564224	0	Ok
36	7,4	2,353772	1,085	1,2	1,00000006	3,11709508	0	Ok
37	6,8	2,853772	1,05	0,96	0,927616616	2,93569832	0,523935426	Ok
38	7	2,853772	1,05	0,96	0,			Ok
39	6,6	2,853772	1,05	0,96	0,			Ok
40	6,8	2,953772	1,05	0,96	0,9194038	2,888292	0,64954742	Ok
41	6,8	2,753772	1,05	0,96	0,935817091	2,9817261	0,41194458	Ok

- Neben den Entwurfsparametern werden auch alle anderen Workflow-Größen aufgelistet (z.B. Ausgangsgrößen, Gütekriterien und Restriktionen).
- Zusätzlich erscheint der jeweilige Wert der Strafe für eventuelle Verletzung von Restriktionen.
- Der Statuswert enthält die Informationen zum erfolgreichen Abschluss eines Optimierungsschrittes (im Beispiel ist alles "Ok").
- Analog zu den Nennwert-Verläufen steht in der Nennwert-Tabelle das Kontext-Menü für jeden Optimierungsschritt zur Verfügung. Falls man sich dafür interessiert, was im Detail bei nicht vollständigem Prägen passiert, kann man einen entsprechenden Optimierungsschritt auswählen und die Simulation mit diesen Parametern starten.

Software: SimX - Nadelantrieb - Wirkprinzip - Guetefunktion

Aus OptiYummy

↑



Experiment: 3D-Gütefunktion (Rastersuche)

□

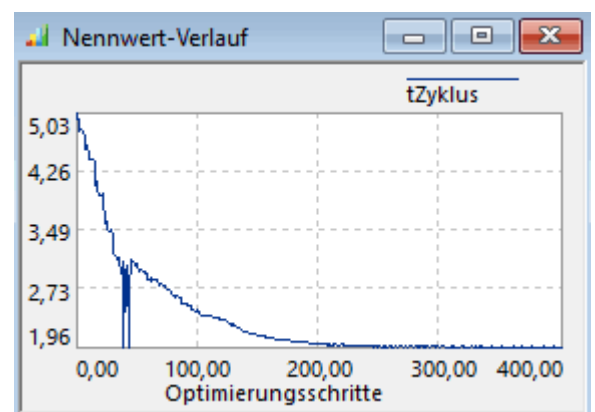
Komplette Guetefunktion im Suchraum mit globalem Optimum

Das Hooke-Jeeves-Verfahren konvergiert durch ständige Abwärtsbewegung auf der Gütefunktion zum nächstgelegenen Minimum:

- Beim erreichten "Bestwert" handelt es sich zumindest um ein lokales Minimum auf der Gütefunktion.
- Man kann sich dabei jedoch nie sicher sein, ob nicht doch noch Kombinationen für die Entwurfsparameter existieren, welche zu einer besseren Lösung führen (möglichst zum globalen Minimum der Gütefunktion).
- Es wäre günstig, wenn man die Topografie der Gütefunktion in Analogie zu einer Landkarte kennen würde. Für unser Optimierungsproblem wollen wir dies in vereinfachter Form realisieren.

Die Gütefunktion wird in unserem Beispiel nur durch das Gütekriterium *tZyklus* gebildet. Während der Optimierung wird als Nennwert-Verlauf praktisch ein Höhenprofil des auf der Oberfläche der Gütefunktion zurückgelegten Pfades abgebildet. Der Wert der Zykluszeit ist bisher eine Funktion von den vier Entwurfsparametern:

- *d_Anker*
- *Feder_k*
- *Magnet_R*
- *Nadel_x0*



Geometrisch handelt es sich bei der kompletten Gütefunktion um eine 5D-Hyperfläche, welche für uns 3D-Lebewesen schwer vorstellbar ist. Das Optimierungsproblem soll deshalb auf zwei Entwurfsparameter zurückgeführt werden. Wir erhalten dann bei hinreichend feiner Abstastung dieser Gütefunktion eine

anschauliche 3D-Fläche.

Reduktion des Optimierungsproblems auf zwei Entwurfparameter

Für die Reduktion des Optimierungsproblems müssen wir Änderungen am Simulationsmodell vornehmen. Dabei werden wir Bezug auf die bisherige Optimallösung nehmen:

- Die optimalen Entwurfparameter übertragen wir mittels *Analyse > Bestwert > Simulation durchführen* in das *SimulationX*-Modell.
- Im *SimulationX* speichern wir danach die Modell-Datei, um diesen Zustand zu sichern.
- Wir beenden das *OptiY*. Dabei wird das *SimulationX*-Modell nicht geschlossen.
- Nun können wir im *SimulationX*-Modell die erforderlichen Änderungen vornehmen.

Bei der Reduktion der Zahl der Entwurfparameter sollen uns folgende Vorüberlegungen helfen:

■ **Nadel_x0=0.2 mm**

Die bisherigen Experimente zeigten, dass es am günstigsten ist, die Nadelspitze in der Ruhelage direkt auf der Papieroberfläche zu platzieren. Der Wert dieses Entwurfparameters kann also konstant auf diesen Wert gesetzt werden! Wir setzen im *SimulationX*-Modell *Nadel.x0=0.2 mm*, was durch die optimalen Entwurfparameter im Rahmen der numerischen Genauigkeit näherungsweise bereits erfolgt sein sollte.

■ **Magnet_R - Abschaltung, wenn geprägt**

Die Einschaltdauer der Magnetkraft muss vom Optimierungsverfahren kontinuierlich an die aktuelle Konfiguration von Ankerdurchmesser, Nadel-Ruhelage und Rückholfeder angepasst werden. Nur so kann für die aktuelle mechanische Konfiguration ein möglichst schnelles und sicheres Prägen des Papiers erreicht werden.

Diese Anpassung der Einschaltzeit kann man durch eine Änderung des Simulationsmodells automatisieren, indem man die Magnetkraft in Abhängigkeit vom Prägezustand steuert.

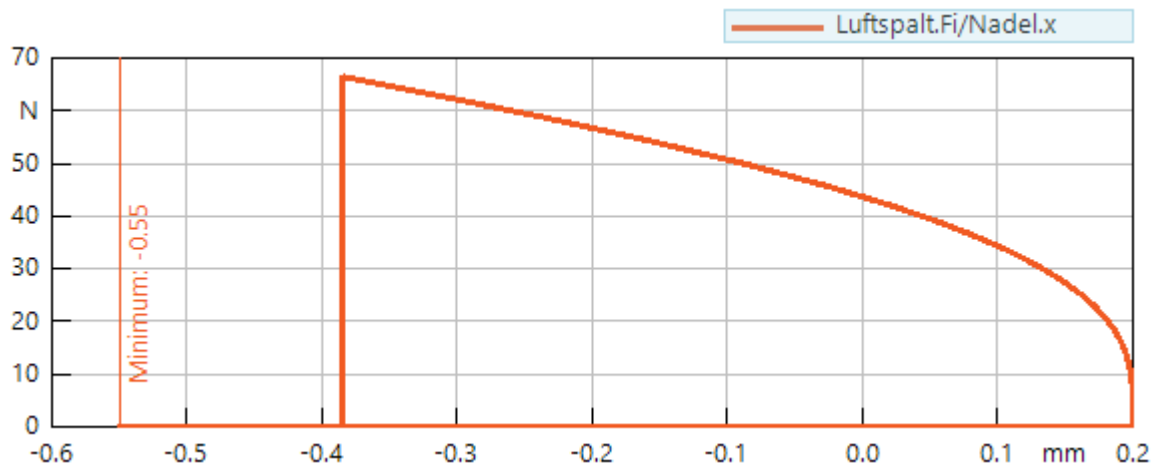
■ **tZyklus=f(d_Anker, Feder_k)**

Die übrig bleibende 3D-Gütefunktion wird wahrscheinlich das globale Optimum enthalten. Die Reduzierung der Dimensionen wird durch Berücksichtigung bekannter Abhängigkeiten im Simulationsmodell bzw. in der Experiment-Konfiguration erreicht.

Automatische Abschaltung der Magnetkraft

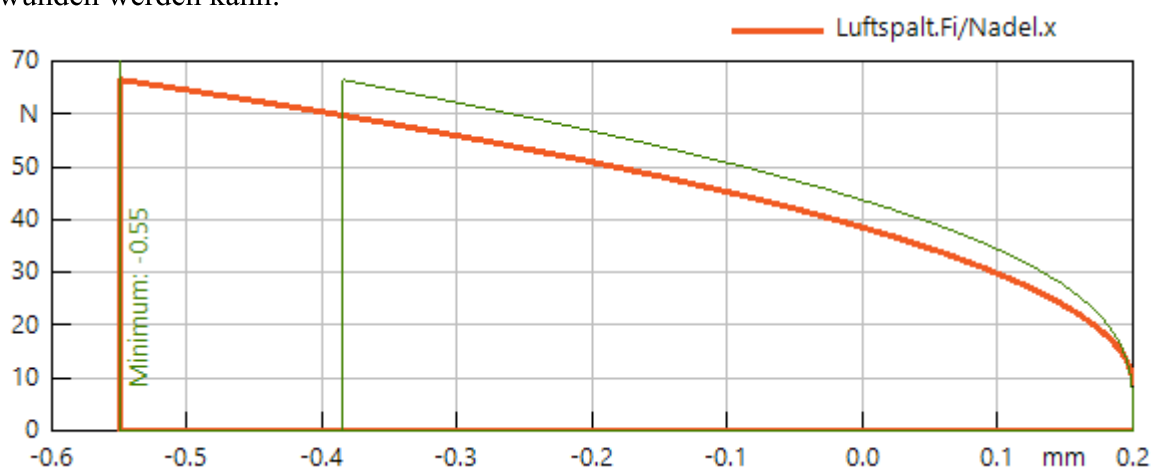
Das für die Abschaltung benötigte Status-Signal steht uns mit *Messung.Praegemasz* im Modell bereits zur Verfügung. Problematischer ist die hinreichend genaue Berücksichtigung von *Fmax*, hier sind einige Tricks erforderlich:

- *Fmax* wurde bisher als Amplitude eines Sägezahn-Dreiecks vorgegebener Länge berechnet.
- Die Kraft soll nun automatisch abgeschaltet werden, wenn das Papier komplett geprägt ist (*Messung.Praegemasz* ≥ 1).
- Da bei dieser automatischen Abschaltung der Abschaltzeitpunkt vorher unbekannt ist, kann dafür der Dreiecksimpuls nicht mehr genutzt werden (Endwert von *Fmax* kann damit nicht realisiert werden!).
- Deshalb wollen wir den Verlauf $F=f(x)$ für die optimal angepasste Einschaltdauer näherungsweise nachbilden. Den erforderlichen Verlauf sehen wir in der Darstellung Luftspalt.Fi=f(Nadel.x) → beide Signale in ein Ergebnisfenster ziehen und $y(x)$ -Darstellung wählen:
 - Der optimale Zeitpunkt für die Kraftabschaltung ist unmittelbar nach der Rissposition $x=-0.39 \text{ mm}$.



Die Nadel fliegt dann auf Grund ihrer Masseträgheit noch bis zum Anschlag und prägt das Papier komplett. Allerdings besteht die Gefahr, dass die kinetische Energie nicht ganz ausreicht und deshalb das Papier nicht richtig geprägt wird. Deshalb werden wir die automatische Abschaltung der Magnetkraft erst vornehmen, wenn $Messung.Praegemasz=1$ erreicht wird.

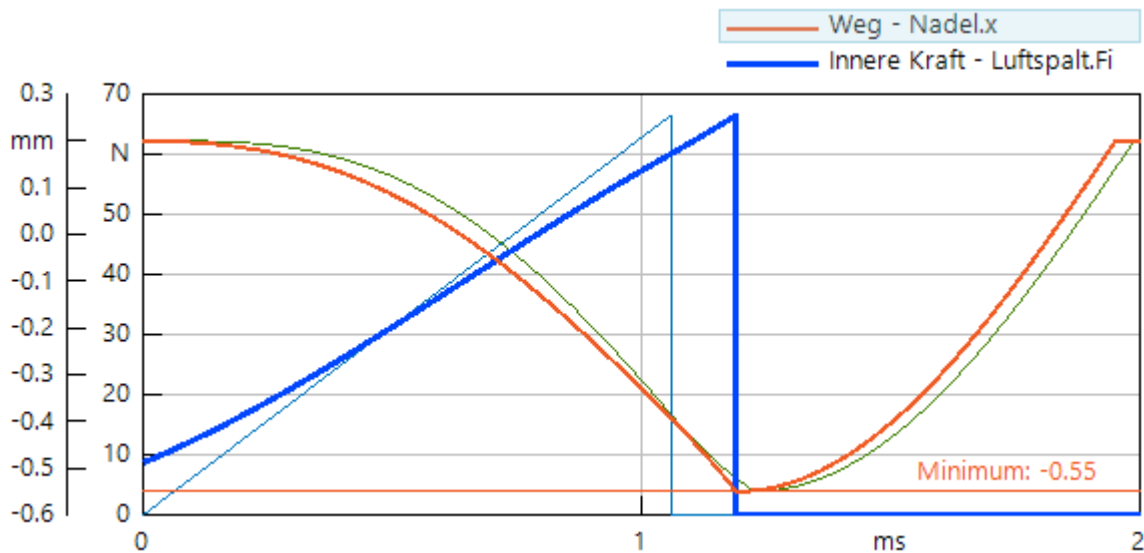
- Die Form der Funktion $F_{magn}=f(x)$ entspricht weitestgehend einer quadratischen Funktion mit dem Wert F_{max} zum Abschaltzeitpunkt. Das kann man bei Vernachlässigung der nichtlinearen Papierkräfte für dieses einfache Feder-Masse-System auch analytisch nachvollziehen.
- Man muss dabei beachten, dass sich die Nadel erst bewegen kann, wenn die Vorspannkraft der Feder überwunden wird. Die Vorspannkraft von ca. 2 N ist im Vergleich zu ca. $F_{max}=50$ N relativ gering (bei einer Masse von 10 g und einer Beschleunigung von 20 g).
- Die Größe $Messung.Praegemasz$ ändert sich beim Vorschub der Nadel von 0 auf 1. Dies soll genutzt werden, um den erforderlichen quadratischen Verlauf von $Luftspalt.F$ näherungsweise nachzubilden. Die Anfangskraft für die Überwindung der Vorspannung wurde im Beispiel auf ca. 15% von F_{max} gesetzt, damit zum Zeitpunkt $t=0$ die Federvorspann-Kraft mit Sicherheit überwunden werden kann:



Blass hinterlegt erscheint im obigen Bild der weg-abhängige Kraftverlauf des Dreieck-Impulses als eingefrorene Kurve ❄️. Die blaue Kurve entspricht der folgenden Formel, welche im Parameterfeld des Luftspalt-Elements eingetragen wurde. Der Ausdruck $\text{sign}(1-\text{floor}(Messung.Praegemasz))$ springt von Eins auf Null, wenn $Praegemasz \geq 1$ wird. Der Faktor 0.87 wurde gewählt, damit der Endwert von F_{max} beim Abschalten erreicht wird. $\text{floor}(x)$ gibt den größten ganzzahligen Wert zurück, der nicht größer als x ist.

$$0.87 * CAD.F_{max} * (0.15 + \sqrt{Messung.Praegemasz}) * \text{sign}(1 - \text{floor}(Messung.Praegemasz))$$

- Das Verhalten des Magnet-Antriebs unterscheidet sich mit diesem Kraftansatz etwas vom optimal eingestellten bisherigen Dreiecksverlauf $F_{magn}=f(t)$. Qualitativ verhält sich der Antrieb jedoch ähnlich:



- Das Modell soll nun so modifiziert werden, dass in Abhängigkeit von einem dimensionslosen Parameter **Abschaltung** (0=Dreieck / 1=Auto) vom Typ=Real eine angepasste Berechnung von *Luftspalt.F* erfolgt:
 - Wir ergänzen in *CAD_Data* den Parameter *Abschaltung* mit entsprechendem Kommentar.
 - Die Standard-Belegung sollte dem Dreiecks-Impuls entsprechen
 - Den Impulsgenerator *Magnet* verändern wir nicht. Er wird also weiterhin nur Dreiecksimpulse mit der vorgegebenen Amplitude *Fmax* erzeugen.
 - Die Umschaltung zwischen Dreiecksimpuls und Rechteckverlauf soll im Element *Luftspalt* für den Parameter *F* erfolgen. Der erste Summand entspricht dem eingespeisten Dreiecksimpuls, der zweite Summand der Ersatzfunktion. In Abhängigkeit des Wertes von *CAD.Abschaltung* (0 oder 1) wirkt nur einer der beiden Summanden:

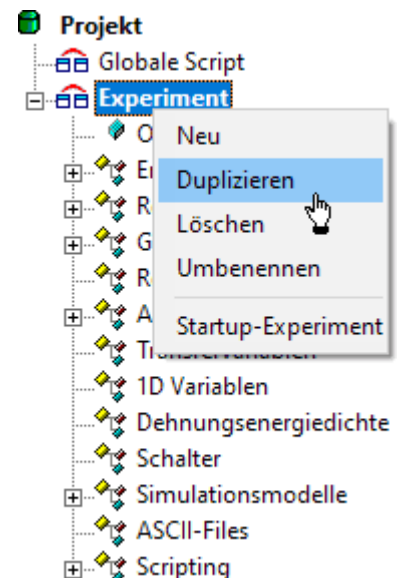
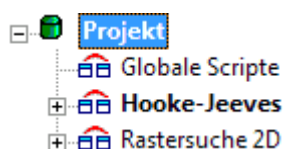
```
self.in1*(1-CAD.Abschaltung) +
0.87*CAD.Fmax*(0.15+sqrt(Messung.Praegemasz))*sign(1-floor(Messung.Praegemasz))*CAD.Abschaltung
```

- Die sichere Funktion beider Betriebsmodi sollte man unbedingt überprüfen!
- Achtung:**
 - Wir erhöhen die Simulationszeit auf **tStop=10 ms**, um auch langsame Prägezyklen vollständig zu berechnen
 - Um Probleme mit unserem bereits konfigurierten *OptiY*-Experiment zu vermeiden, setzen wir **Abschaltung=0**, bevor wir das Modell speichern. Damit ist im *OptiY* standardmäßig der bisherige Betriebsmodus mit Impulsgenerator wirksam.

Inbetriebnahme eines neuen OptiY-Experiments zur Rastersuche

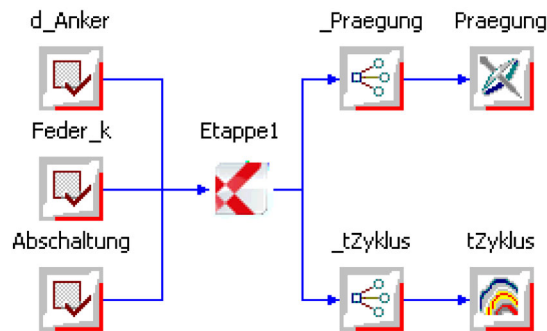
OptiY bietet die Möglichkeit, für ein Projekt mehrere Experimente zu verwalten. So können wir die bisherigen Einstellungen und Ergebnisse beibehalten und als Ausgangspunkt für ein neues Experiment nutzen:

- Nach dem **Duplizieren** des Experiments erscheint im *OptiY*-Explorer die Kopie als **Experiment2**.
- Dieses weitere Experiment besitzt am Anfang die gleiche Konfiguration wie das Original. Nur die Anzeigefenster muss man neu definieren.
- Verwaltet man mehrere Experimente in einem Projekt, so sollte man mittels **Umbenennen** dafür sinnvolle Namen vergeben:
 - Das bisherige Experiment könnte man z.B. nach dem verwendeten Optimierungsverfahren **Hooke-Jeeves** nennen.
 - Für das neue Experiment bietet sich die Bezeichnung **Rastersuche 2D** an:



Für jedes Experiment wird von *OptiY* ein separater Workflow verwaltet, so dass darin unabhängig voneinander Änderungen vorgenommen werden können:

- Zuerst selektieren wir die *Rastersuche 2D* über das Kontextmenü als **Startup-Experiment**, damit wir es bearbeiten können.
- Wir löschen im zugehörigen Workflow die nicht mehr benötigten Entwurfparameter:



- Während der Rastersuche muss die Magnetkraft-Erzeugung mit *Abschaltung=1* versehen werden:
 - Diesen Wert könnte man im *SimulationX*-Modell setzen. Das hat jedoch den Nachteil, dass der Wert für alle Experimente wirkt. Übersieht man diesen Effekt, gelangt man in anderen Experimenten ganz schnell zu fehlerhaften Resultaten!
 - Wir ergänzen deshalb einen Entwurfparameter *Abschaltung*, den wir dann mit dem Startwert=1 versehen, aber konstant halten.
 - *Abschaltung* ordnen wir im Workflow dem Modellparameter *CAD.Abschaltung* zu.
 - **Hinweis:** Falls die Modellparameter für die Zuordnung des Eingangs nicht angezeigt werden, muss die Datei des Modells für *Etappe1* erneut geöffnet werden!
- Die Grenzen der beiden "echten" Entwurfparameter muss man so groß wählen, dass das globale Optimum im aufgespannten Bereich enthalten ist:
 - Ein zu großer Bereich verhindert ein hinreichend dichtes Abtasten der Oberfläche wegen ausufernder Rechenzeit.
 - Orientieren sollte man sich an technisch-physikalisch sinnvollen Größen für die Parameter.
 - Meist kann man sich günstigen Grenzwerten für die Rastersuche nur iterativ nähern!
- Im Sinne einer effektiven Nutzung der knappen Übungszeit werden hier günstige Grenzwerte vorgegeben:
 - **d_Anker = 6...16 mm**
 - **Feder_k = 2...160 N/mm**
- Als Optimierungsverfahren wählen wir die Rastersuche mit 900 Abtastschritten:

Nennwert Daten	
Name	Abschaltung
Einheit	-
Kommentar	0=Dreieck / 1=Auto
Werte	
Wert	1
Typ	Konstante

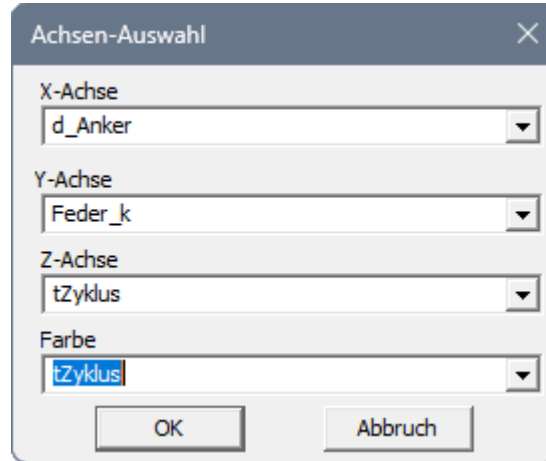
Eigenschaft	
Optimierung	
Auto-Stop	Manueller Stop
Optimierungsschritte	900
Startschrittweite	Standard
Verfahren	Raster-Suche

- Der Bereich jedes Entwurfparameters wird dabei in gleichmäßigen Abständen an 30 Punkten abgetastet.
- 30x30=900 Simulationsläufe sind auf modernen PC ein günstiger Kompromiss zwischen Rechenzeit und Feinheit der Abtastung.
- Der Rechenzeitbedarf steigt quadratisch mit der Feinheit der Rasterung, deshalb sollte man sich auf älterer Hardware mit 20x20=400 Simulationsläufen zufrieden geben!

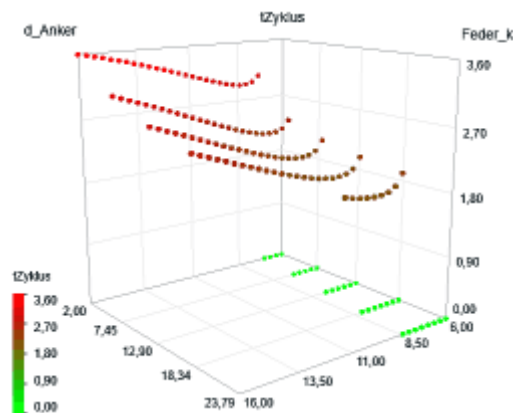
Hinweis:



In der Simulationssteuerung von *SimulationX* muss **tStop** größer gewählt werden, als der längste Prägezyklus im Suchraum! Im Beispiel sollte **tStop=10 ms** ausreichen.

- Das 3D-Diagramm für das Gütekriterium $t_{\text{Zyklus}}=f(d_{\text{Anker}}, Feder_k)$ kann man bereits vor dem Start der Optimierung öffnen (*Analyse > Darstellung > 3D-Darstellung*). Der Verlauf der "Farbe" soll durch den Wertebereich von t_{Zyklus} bestimmt werden:

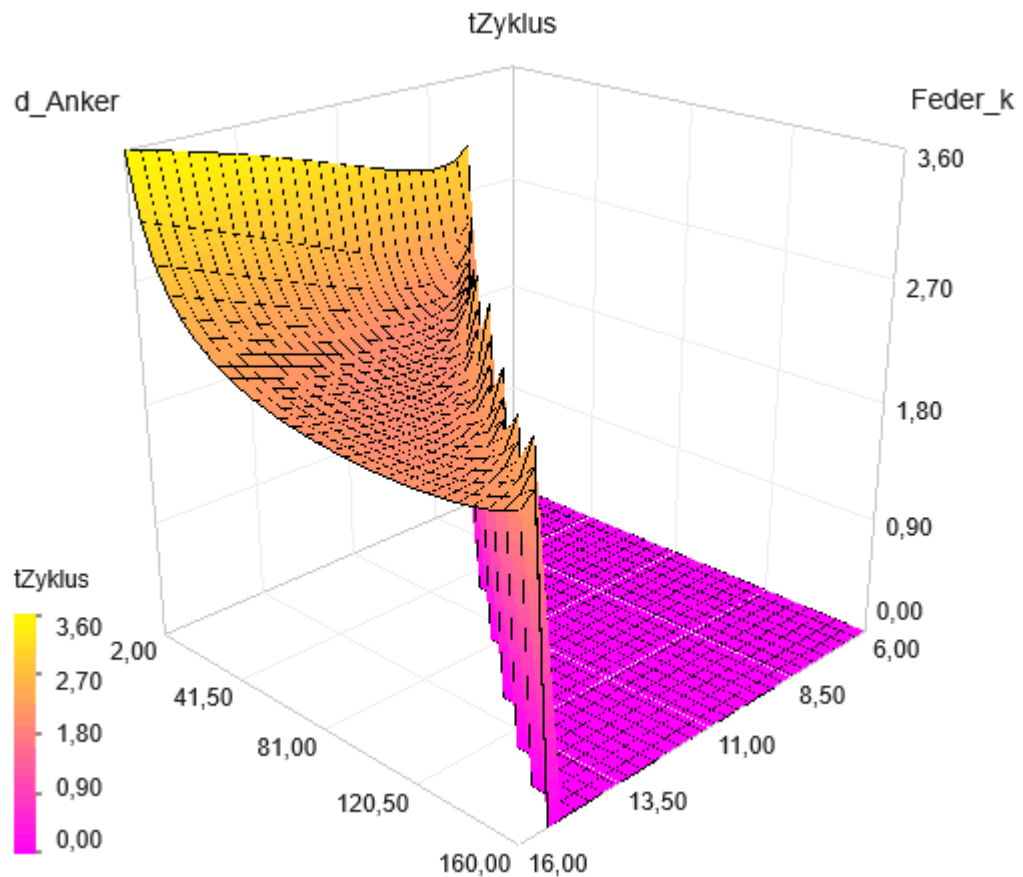


- Nach Start der Optimierung baut sich Stück für Stück die 3D-Darstellung im Diagramm auf.
 - Ein Mausklick auf das 3D-Diagramm ermöglicht im Eigenschaftsfeld die Konfiguration der Darstellung.
 - Während der Berechnung ist es günstig, nur die Punkte innerhalb des Rahmen darzustellen (Linien und Flächen ausblenden):



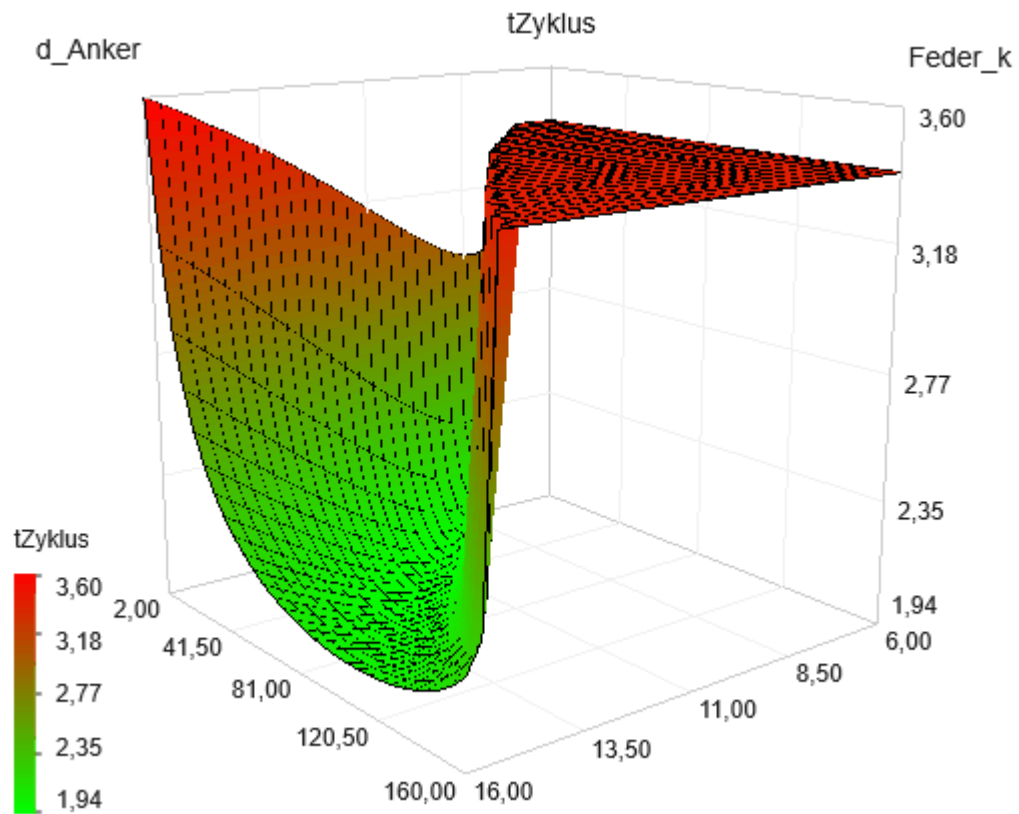
Eigenschaft	
3D Darstellung	
Rahmen	<input checked="" type="checkbox"/>
Punkte	<input type="checkbox"/>
Linien	<input checked="" type="checkbox"/>
Fläche	<input checked="" type="checkbox"/>
Legende	<input checked="" type="checkbox"/>
Radius	5
Max-Farbe	 255; 255; 0
Min-Farbe	 255; 0; 255
Gleitkommastelle	2
E-Format	<input type="checkbox"/>
Auto-Skalierung	<input checked="" type="checkbox"/>

- Anderenfalls entsteht infolge der unvollständigen Grafik ein Wirrwarr von Linien- und Flächenelementen.
- Nach Ende der Berechnung hat sich im Beispiel die Flächendarstellung mit eingeblendeten Linien als anschaulich erwiesen.
- Die Standardfarben für Max- und Min-Farbe wurden im Beispiel durch Gelb und Violett ersetzt, um einen kontrastreicherer Farbverlauf zu erhalten.
- Mit dem Maus-Cursor lässt sich die Darstellung drehen (Einstellen einer günstigen Ansichten):



Interpretation und Skalierung der Gütefunktion

- Die Magnetkraft schaltet erst nach erfolgreichem Prägen des Papiers ab:
 - Reicht die Kraft dafür nicht aus, so kehrt die Nadel nicht mehr in die Ruhelage zurück, weil die Kraft nicht abschaltet.
 - In diesem Fall wird auf Grund des Messprinzips $tZyklus=0\text{ s}$ ermittelt, obwohl praktisch $tZyklus=unendlich$ ist!
- Die Z-Achse (tZyklus) des 3D-Diagramms wird automatisch zwischen Minimalwert (0 s) und Maximalwert (langsamster Prägezyklus) skaliert:
 - $tZyklus=0$ repräsentiert den verbotenen Bereich, in dem keine Prägung stattfindet (=Restriktionsverletzung)
 - Die "Zacken" an der Grenze des zulässigen Lösungsbereichs resultieren aus dem Abtast-Raster.
- Um die abgebildete Gütefunktion besser an die "natürlichen" Gegebenheiten anzupassen und gleichzeitig eine bessere Skalierung in Z-Richtung zu erhalten, genügt eine einfache Modell-Änderung:
 - **Wichtig:** Vor der Modelländerung *OptiY* schließen (mit Speichern) und *SimulationX* schließen (OHNE Speichern!)
 - Wir setzen im *SimulationX*-Modell in der Strukturansicht des **Signalprozessor-Compound** für den Anfangswert **tZyklus.y0=0.0036**. Das geänderte Modell muss wieder gespeichert werden.
 - Nach erneut durchgeführter Rastersuche im *OptiY* erscheint dieser Wert von **3,6 ms** beim Nichtprägen.
 - Man erkennt dann im Bild, dass für diesen Zustand die Zykluszeit steil ansteigt (praktisch gegen Unendlich geht).
 - Dieses Plateau markiert gleichzeitig die Forderung für die maximal zulässige Zykluszeit.
 - Für die Farb-Legende hat sich folgende Zuweisungen als "selbsterklärend" erwiesen: **rot = schlechte Werte** / **grün = gute Werte**.
 - Wir erhalten damit eine günstigere Darstellung der Gütefunktion $tZyklus=f(d_Anker, Feder_k)$:



Das globale Optimum ("Bestwert") liegt in einer flachen Mulde unweit der Grenze zum "Nichtprägen":

Name	Werte	Einheit	Kommentar
d_Anker	11,5172	mm	Ankerdurchmesser
Feder_k	45,5862	N/mm	Steifigkeit
Abschaltung	1	-	0=Dreieck / 1=Auto
Praegung	1		Prägungsmaß 0...1
tZyklus	1,9392	ms	Zyklus

Daraus kann man folgende Schlussfolgerungen ziehen:

1. Insbesondere die Federkonstante kann infolge der flachen Mulde in einem weiten Bereich verändert werden, ohne dass sich die Zykluszeit in relevanten Größenordnungen ändert. Man kann sich bei der Wahl eines konkreten Wertes an der technischen Realisierbarkeit orientieren.
2. Zur gewählten Federsteife kann man einen geeigneten Ankerdurchmesser wählen, der einen hinreichend schnellen Prägezyklus ermöglicht.
3. Die Kombination der gewählten Entwurfsparameter sollte für eine optimale Lösung möglichst weit entfernt von der Restriktionsgrenze liegen, um trotz aller Toleranzen einen sicheren Betrieb des Antriebs zu gewährleisten.

Hinweis: Der Anfangswert $0,0036$ von $tZyklus.y0$ wird nur wirksam, wenn man den Kraftmodus "Abschaltung=1" benutzt! Ansonsten beginnt die Magnetkraft bei Null und die Nadel wird durch die Feder-Vorspannung in den Anschlag der Ruhelage gedrückt. Dies wird als erstes Ereignis registriert und dafür eine Zeit nahe Null für $tZyklus.y$ übernommen.

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Guetefunktion&oldid=27780“

Software: SimX - Nadelantrieb - Wirkprinzip - Zielfunktion

Aus OptiYummy

↑

← →

Experiment: Hierarchische Optimierung

□

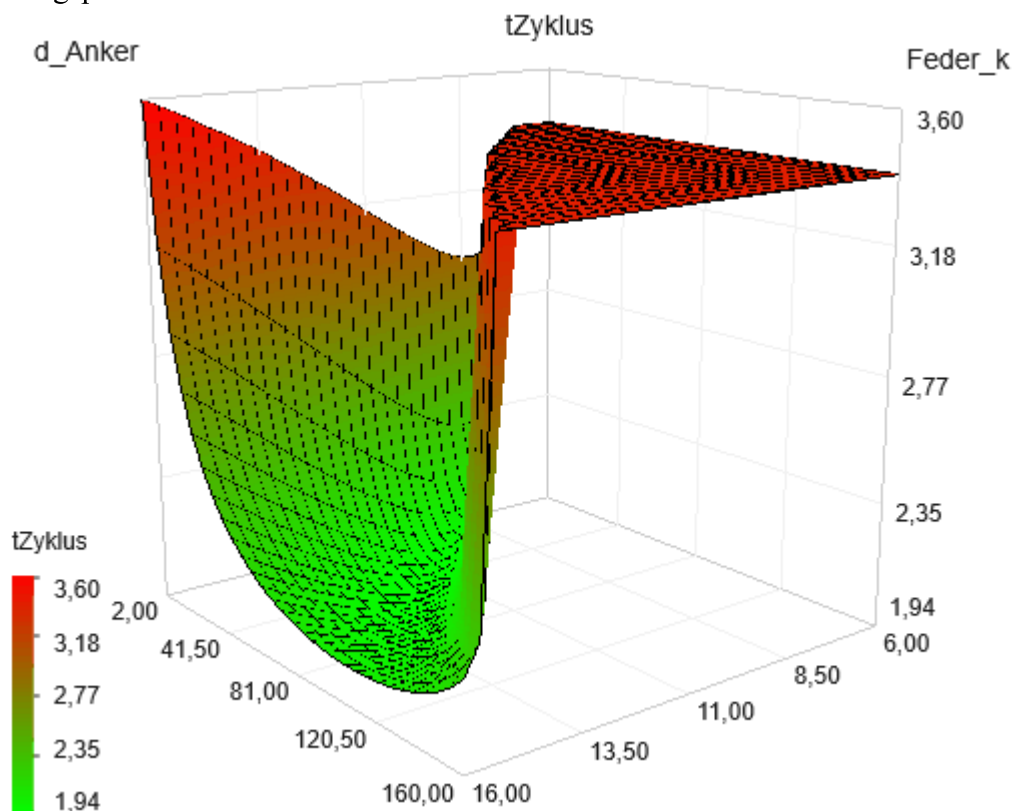
Globale Suche

Die zuvor angewandte Rastersuche gehört zu den globalen Suchverfahren. Damit werden Informationen über den gesamten aufgespannten Suchraum gewonnen. Solche Informationen zu den Eigenschaften der Gütefunktion sind äußerst nützlich für den Optimierungsprozess:

- Existieren mehrere lokale Optima?
- Existieren voneinander getrennte Bereiche zulässiger Lösungen?
- Liegen Optima direkt an Grenzen zum Bereich unzulässiger Lösungen?
- Ist die Gütefunktion numerisch kritisch (unstetig, verrauscht, Polstellen usw.)?

Globale Suche setzt meist eine Reduktion des Suchraums voraus (Rechenzeit!):

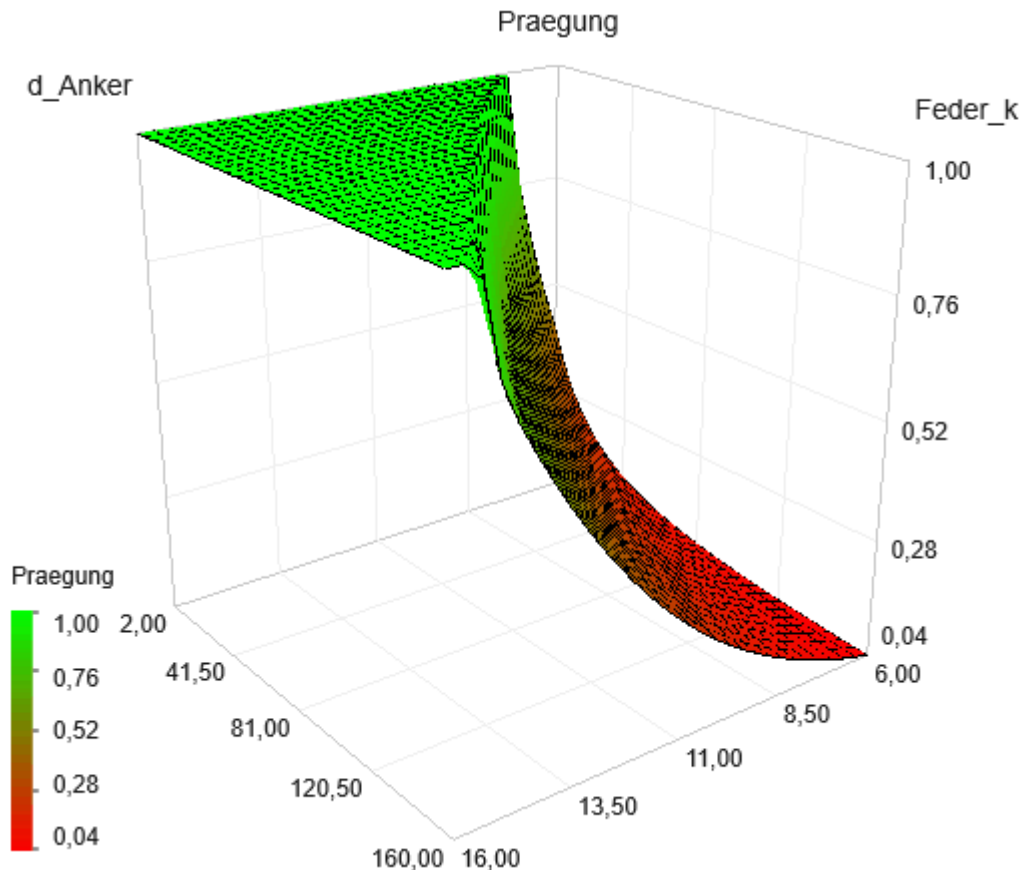
- Ausnutzung von Abhängigkeiten zwischen den Entwurfparametern.
- Berücksichtigung von bereits erarbeitetem Wissen zum Optimalwert einzelner Entwurfparameter.
- Die erforderliche Reduktion auf 2 bzw. maximal 3 Entwurfparameter gelingt nur in den ersten Etappen eines Optimierungsprozesses.



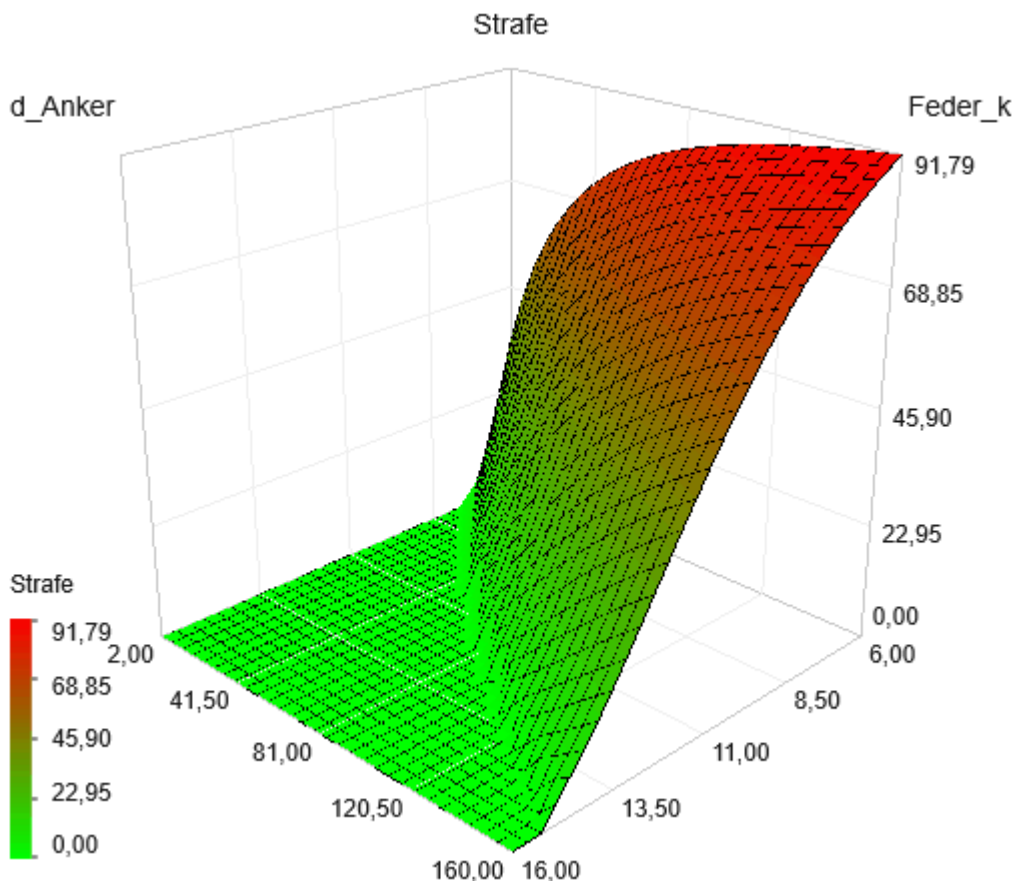
Die gesamte Gütefunktion wie im obigen Bild bekommt man bei praktischen Optimierungsproblemen eher selten zu Gesicht. Deshalb sollte man zumindest ein Gefühl dafür entwickeln, wie ausgehend von einem Startpunkt mit lokalen Suchverfahren eine optimale Lösung auf der "unsichtbaren" Gütefunktion gefunden wird:

- Wir hatten bereits festgestellt, dass *innerhalb* des Suchraums zwei unterschiedliche Bereiche existieren:
 - Bereiche mit zulässigen Lösungen (alle Forderungen werden erfüllt).
 - Bereiche unzulässiger Lösungen (es werden Restriktionen verletzt).
- Zusätzlich existiert noch der Bereich *außerhalb* des Suchraums:
 - Enthält alle Lösungen, von denen mindestens ein Entwurfsparameter außerhalb der festgelegten Grenzen liegt.
 - Mit Ausnahme der Rastersuche können bei anderen Optimierungsverfahren während des Suchprozesses auch Lösungen außerhalb des Suchraums generiert werden! Für diese Lösungen wird dann jedoch kein Simulationslauf durchgeführt.

Die 3D-Darstellung der Restriktionsgröße *Praegung* zeigt die Abgrenzung zwischen zulässigen (=1) und unzulässigen Lösungen (<1) im Suchraum:



Zusätzlich wird automatisch eine Straffunktion als Bestandteil der Gütekriterien generiert, wenn Restriktionen definiert wurden. In der 3D-Darstellung der Straffunktion bilden sich die Lösungsbereiche wie folgt ab (*Strafe=0* sind die zulässigen Lösungen):



Restriktionsverletzungen werden dabei nach folgendem Prinzip mit "Strafpunkten" bestraft:

- Für jede Restriktionsgröße wird die Distanz zum zulässigen Wertebereich berechnet (UG=untere Grenze / OG=obere Grenze):

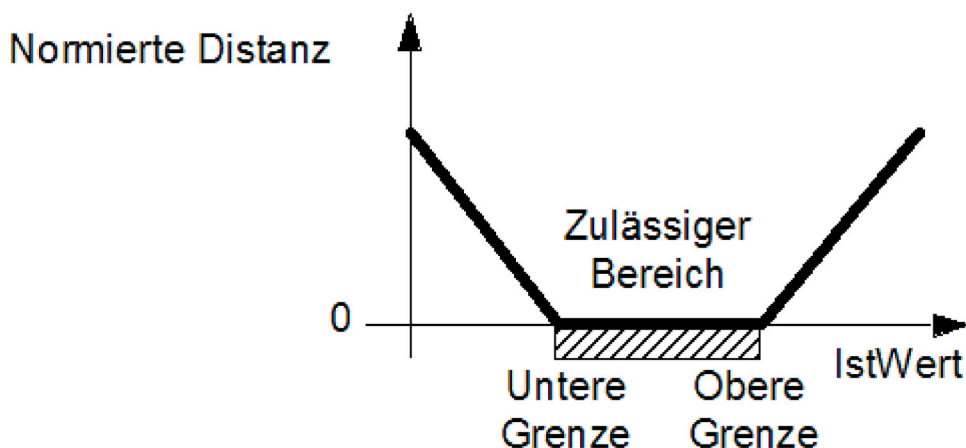
$Distanz = 0$: wenn $UG \leq \text{Istwert} \leq OG$

$Distanz = \text{abs}(\text{Istwert} - OG)$: wenn $\text{Istwert} > OG$

$Distanz = \text{abs}(\text{Istwert} - UG)$: wenn $\text{Istwert} < UG$

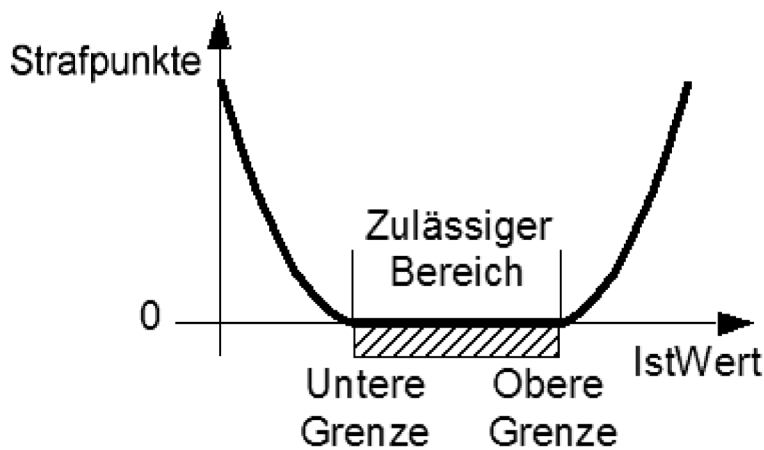
- Es erfolgt eine Normierung der Distanz auf die Bereichsbreite:

$$NormDistanz = Distanz / \text{Bereich} \text{ mit } \text{Bereich} = \text{abs}(OG - UG)$$



- Die normierte Distanz repräsentiert die Stärke der zugehörigen Restriktionsverletzung und kann als Maß für die "Bestrafung" des Modellverhaltens dienen.
- Ein quadratischer Ansatz für die Berechnung der Strafpunkte aus der normierten Distanz wichtet den Grad der jeweiligen Restriktionsverletzung:

$$\text{Strafpunkte} = \text{Wichtung} \cdot \text{NormDistanz}^2$$



- Der Wert der "Strafe"-Funktion ergibt sich als Summe der Strafpunkte aller Restriktionen.

Hierarchische Zielfunktion

OptiY nutzt die vom Nutzer formulierten Gütekriterien als Bestandteil einer "hierarchischen" Zielfunktion:



- Um zwei Lösungen miteinander vergleichen zu können, muss für jede dieser Lösungen ein Zielfunktionswert nach der gleichen Zielfunktion gebildet werden. Der Zielfunktionswert entspricht der "Güte" oder "Fitness" einer Lösung.
- *OptiY* verwendet dafür drei verschiedene Zielfunktionen
 1. Straf-Funktion für die Begrenzung des Suchraums (Tool-interne "unsichtbare" Größe)
 2. Straf-Funktion für Restriktionsverletzungen (zusätzliches Gütekriterium "Strafe")
 3. Gütekriterien für zulässige Lösungen (vom Nutzer definiert)
- Die 1. Zielfunktion hat eine höhere Priorität als die 2. Zielfunktion. Diese hat eine höhere Priorität als die 3. Zielfunktion.
- Für den Vergleich zweier Lösungen wird die zuständige Zielfunktion mit der höchsten Priorität benutzt, z.B.:
 1. Liegt mindestens eine Lösung außerhalb des durch die Grenzen der Entwurfparameter aufgespannten Suchraums, so wird die 1. Zielfunktion benutzt.
 2. Gibt es bei mindestens einer Lösung noch Restriktionsverletzungen, so wird die 2. Zielfunktion benutzt ("Strafe").
 3. Nur wenn beide Lösungen im zulässigen Bereich liegen, werden die nutzerdefinierten Gütekriterien für die Bewertung benutzt.
- Der Wert von "Strafe" ist Null, wenn keine Restriktionen verletzt sind (Bereich zulässiger Lösungen im zulässigen Suchraum).
- Im Bereich unzulässiger Lösungen steigt der Wert der Straffunktion quadratisch mit dem Grad der Restriktionsverletzung, wie es zuvor erläutert wurde.

Lokale Suche - Visualisierung von Suchpfaden

Aufbauend auf dem Experiment *Rastersuche 2D* definieren wir durch *Duplizieren* ein neues Experiment *Suchpfade*:

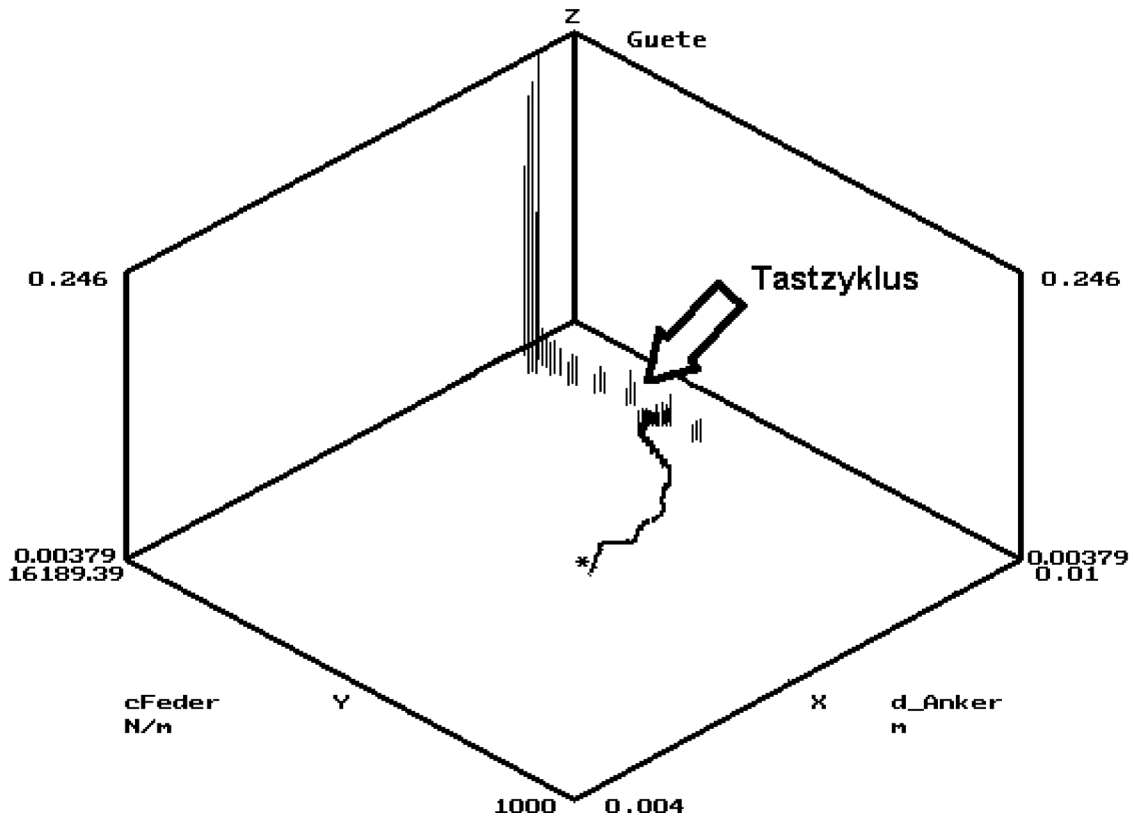
- Zur Optimierung wählen wir das Hooke-Jeeves-Verfahren mit:
 - **Startschrittweite = Manuell**
 - **Auto-Stop = Automatischer Stop**

- Ausgehend von unterschiedlichen Startpunkten wollen wir den Pfad des Verfahrens im Suchraum beobachten.

Hooke-Jeeves-Verfahren

Die Wirkungsweise eines lokalen Suchverfahrens, welches die 1. Ableitung am aktuell erreichten Ort durch Abtastung der Zielfunktion ermittelt, soll am Beispiel des gewählten **Hooke-Jeeves-Verfahrens** erläutert werden:

- Die Steigung der Zielfunktion am Punkt der aktuell erreichten Lösung wird über kleine Tastschritte ermittelt. Die genutzte Tastschrittweite muss dabei der Oberfläche der Zielfunktion angepasst sein.
- In Richtung des ermittelten steilsten Abstiegs wird anschließend ein größerer Schritt ausgeführt:



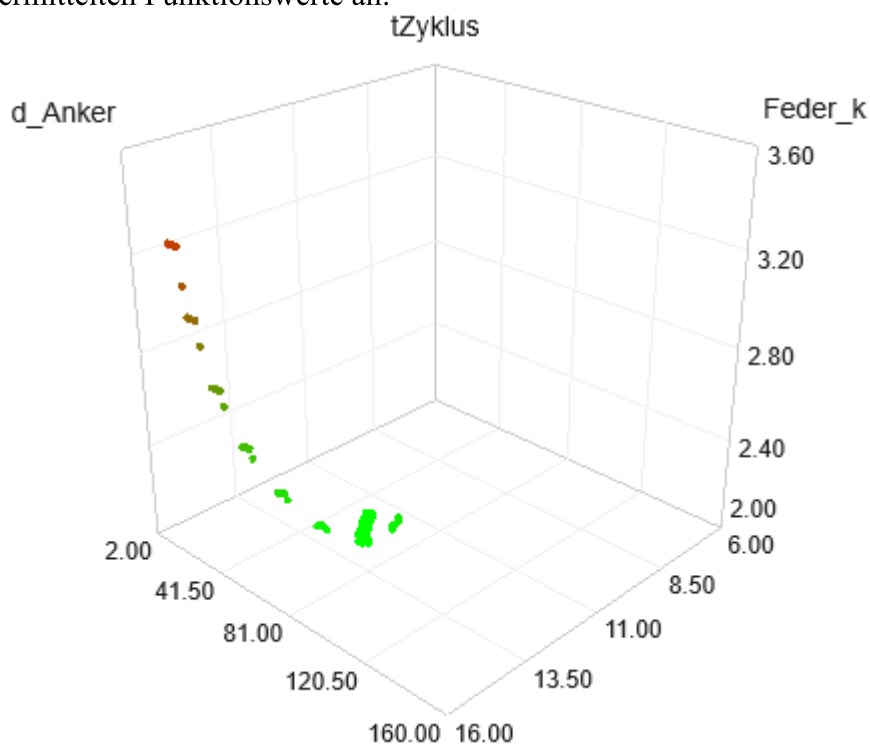
- Die Tastschritte werden gleichzeitig zur Verbesserung der Lösung genutzt. Beim Tastzyklus wird von einem Startvektor aus nacheinander in jede Koordinatenrichtung ein diskreter Suchzyklus durchgeführt. Führt dabei der Tastschritt in einer Richtung nicht zum Erfolg (Verbesserung des Gütewertes), so wird zusätzlich die entgegengesetzte Richtung abgetastet.
- Nach einer Tastphase erfolgt die Extrapolation. In Richtung des ermittelten steilsten Abstieges wird ein größerer Schritt ausgeführt. Ein erneuter Extrapolationsschritt ist immer doppelt so groß wie der vorherige, solange dabei eine Verbesserung des Zielfunktionswertes erreicht wird. Schießt das Verfahren über den tiefsten Punkt hinaus oder in einen verbotenen Bereich, erfolgt eine schrittweise Verringerung der Extrapolationsschrittweite.
- Das Hooke-Jeeves-Verfahren regelt die anfängliche Tastschrittweite herunter, wenn längere Zeit keine Verbesserung des Zielfunktionswertes mehr erreicht werden konnte. Meist gelangt dann diese Schrittweite in Größenordnungen der stochastischen Rauigkeiten der Zielfunktion und das Optimierungsverfahren bleibt "hängen", weil die partiellen Ableitungen total verfälscht werden (die reale Steigung wird nicht mehr erkannt!).

Suchpfad ohne Restriktionsverletzungen

- Wir konfigurieren erneut eine 3D-Darstellung für $t_{Zyklus} = f(d_Anker, Feder_k)$.
- Zuerst starten wir in der oberen Ecke der Gütefunktion mit Startwerten, welche ein sicheres Prägen ermöglichen:
 - **d_Anker=15 mm** mit Startschrittweite=0,2 mm
 - **Feder_k=5 N/mm** mit Startschrittweite=2 N/mm

- Die Anfangswerte für die Entwurfparameter liegen nicht direkt an der Grenze des Suchraumes, um die lokale Abtastung der Steigung am Rand nicht zu behindern.
- Die Startschrittweiten sind in Bezug auf die Suchraumgrenzen relativ zueinander ungefähr gleich groß. Sie wurden so gewählt, dass man optisch die einzelnen Stützstellen der lokalen Abtastung im 3D-Diagramm unterscheiden kann.
- Während der Lösungssuche erfolgt standardmäßig eine automatische Skalierung der Koordinatenachsen unseres 3D-Diagramms (nur **Punkte** zur Darstellung wählen!):
 - Um eine bessere Vergleichbarkeit mit den 3D-Flächen der vorherigen Rastersuche zu erreichen, setzen wir nach dem Erreichen des Optimums für das Diagramm *Auto-Skalierung=False*.
 - Für die X- und Y-Achse tragen wir die Grenzen des kompletten Suchbereichs ein.
 - Die Grenzen der Z-Achse passen wir entsprechend der individuell ermittelten Funktionswerte an:

Eigenschaft		↑	×
3D Darstellung			
Rahmen	<input checked="" type="checkbox"/>		
Punkte	<input checked="" type="checkbox"/>		
Linien	<input type="checkbox"/>		
Fläche	<input type="checkbox"/>		
Legende	<input type="checkbox"/>		
Radius	5		
Max-Farbe	■	255; 0; 0	
Min-Farbe	■	0; 255; 0	
Gleitkommastelle	2		
E-Format	<input type="checkbox"/>		
Auto-Skalierung	<input type="checkbox"/>		
X-Achse			
Name	d_Anker		
Min	6		
Max	16		
Y-Achse			
Name	Feder_k		
Min	2		
Max	160		
Z-Achse			
Name	tZyklus		
Min	2		
Max	3.4		



- Während der Suche wird der Bereich zulässiger Lösungen im Beispiel nicht verlassen:
 - Das Optimierungsverfahren bewegt sich deshalb näherungsweise entlang des steilsten Abstiegs der Gütefunktion $tZyklus=f(d_Anker, Feder_k)$ zum flachen und lang gestreckten Talgrund.
 - In der Talmulde erfolgt eine Neuorientierung für die Suchrichtung mit einer Drehung von ca. 90° und das Verfahren wandert in der Mulde bis zum tiefsten Punkt (Bestwert).

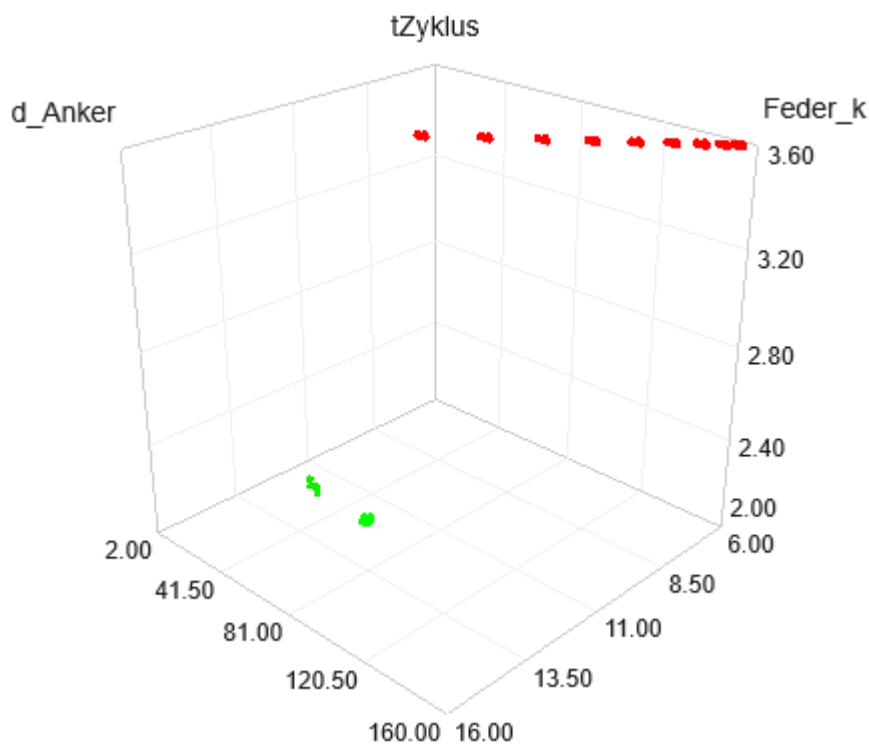
Suchpfad beginnend mit Restriktionsverletzungen

Mit den Startwerten (ohne Änderung der Startschrittweiten)

- d_Anker=6,2 mm** und
- Feder_k=155 N/mm**

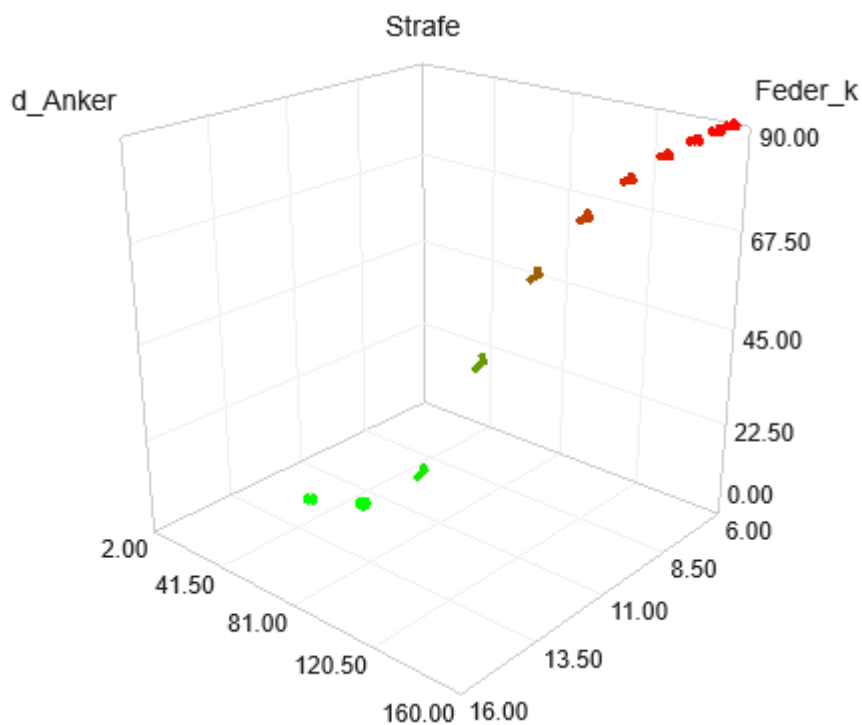
konfigurieren wir einen Antrieb, der sich fast nicht mehr bewegt (*Praegung* ca. 0.05).

Anhand des Pfades auf der Funktion $tZyklus=f(d_Anker, Feder_k)$ kann man noch nicht direkt erkennen, wie das Optimierungsverfahren den zulässigen Bereich der Lösungen findet:

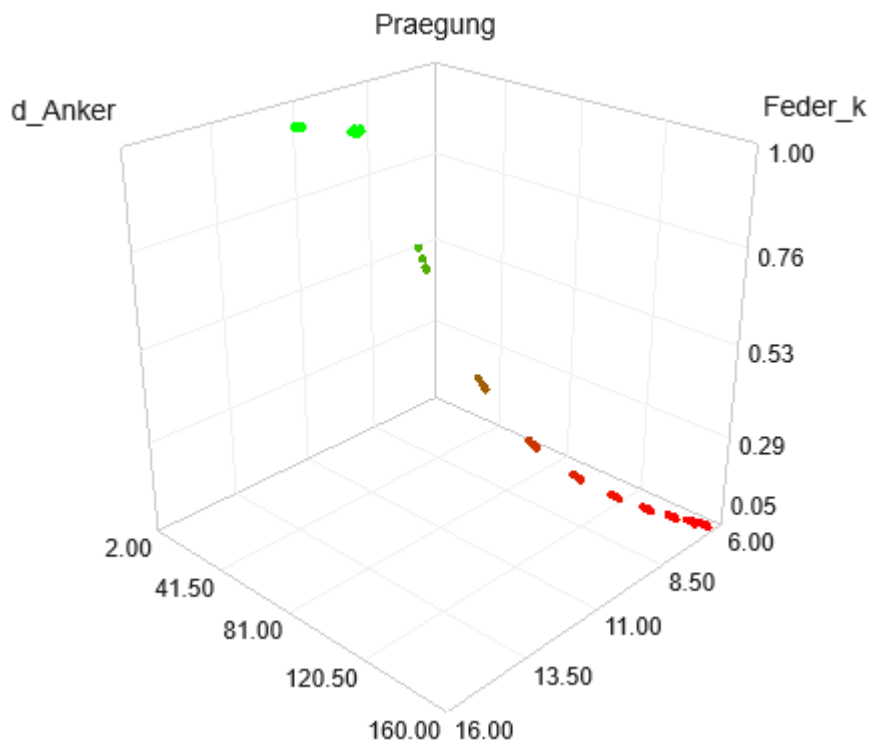


- Das Optimierungsverfahren erreicht den zulässigen Bereich der Lösungen in der flachen Mulde der Gütefunktion.
- In der Mulde wird letztendlich wieder die gleiche Lösung mit minimaler Zykluszeit gefunden. Die Abweichung zwischen den ermittelten Bestwerten bewegt sich im Bereich des numerischen Rauschens.

Das Prinzip der hierarchischen Optimierung wird deutlicher, wenn man sich die 3D-Darstellung von $Strafe=f(d_Anker, Feder_k)$ anschaut:



- Infolge der Restriktionsverletzung $Praegung < 1$ ist der Wert der Straffunktion zu Beginn der Optimierung nicht Null.
- Deshalb erfolgt die Optimierung in der ersten Phase nur auf der Straffunktion als Zielfunktion. Es wird näherungsweise der steilste Abstieg zum Gebiet mit $Strafe=0$ gesucht. Dabei steigt das erreichte Prägungsmaß stetig an:



- Erst wenn die Lösungssuche im zulässigen Bereich angekommen ist, erfolgt die Umschaltung der Zielfunktion auf das Gütekriterium *tZyklus*.
- Unabhängig vom Startpunkt sollte die lokale Suche des Hooke-Jeeves-Verfahrens im Beispiel immer das globale Optimum als Bestwert finden:

Name	Werte	Einheit	Kommentar
d_Anker	11.5875	mm	Ankerdurchmesser
Feder_k	47.8125	N/mm	Steifigkeit
Abschaltung	1	-	0=Dreieck / 1=Auto
Praegung	1		Prägungsmaß 0...1
tZyklus	1.93875	ms	Zyklus

Beachte: Für die Einsendung der Lösung zur 1. Etappe ist diese Konfiguration des Suchpfad-Experimentes zu speichern!

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Zielfunktion&oldid=27783“

Software: SimX - Nadelantrieb - Wirkprinzip - Ergebnisse

Aus OptiYummy

↑

← →

Ergebnisse der 1. Etappe

In dieser ersten Bearbeitungsetappe wurden mit einem sehr stark idealisierten Modell die funktionsbestimmenden Strukturen des Prägenadel-Antriebs nachgebildet. Die durchgeführten Experimente sprechen nicht gegen die Verwendung eines Elektromagneten. Die unter den idealisierten Bedingungen erreichbare Zykluszeit liegt weit unter den laut Aufgabenstellung geforderten **3,6 ms**.

Jeder **Teilnehmer** $xx=(01\dots99)$ der Lehrveranstaltung "**Optimierung**" hat folgende individuellen Ergebnisse unter Verwendung des **Längenfaktors** $CAD.L_Faktor=1.xx$ erzielt:

1. Bestwert

1. Zykluszeit
2. Ruheposition der Nadelspitze
3. Ankerdurchmesser
4. Elastizitätskonstante der Rückholfeder
5. Wert des Vorspannweges s_0 (in μm) der Rückholfeder ("Dehnung" in Nadel-Ruhelage)
6. Einschaltzeit des Magneten

2. Zielfunktionsgrafiken in den gespeicherten Experimenten

1. 3D-Flächen der Rastersuche (*tZyklus, Praegung, Strafe*)
2. Suchpfade der lokalen Suche (*tZyklus, Praegung, Strafe*)

Hinweise für die Ergebnis-Aufbereitung:

1. Die geforderten Werte der optimalen Lösungen (Bestwert) sind in einer Text-Datei zu notieren.
2. Das Simulationsmodell (.isx-Datei) ist mit dem Bestwert zu konfigurieren (mit "Abschaltung=0"). Die Zeitverläufe der Nadelbewegung, der Magnetkraft und der Kraftkomponenten des Papiers müssen im Modell in anschaulicher Form in Ergebnisfenstern dargestellt werden ("sinnlose" Ergebnis-Fenster, welche im Laufe der Modell-Entwicklung zusätzlich entstanden, sind zu löschen!).
3. Das *OptiY*-Projekt (.opy-Datei) muss in allen Experimenten aussagekräftige Nennwert-Verläufe bzw. 3D-Darstellungen enthalten. Diese müssen ohne erneute Experiment-Durchführung betrachtbar sein.
4. Es ist zu überprüfen, dass die als Anhang der E-Mail einzusendenden Dateien **Etappe1_xx.isx** und **Etappe1_xx.opy** portabel sind (lauffähig auf einem anderen PC in einem beliebigen Ordner!).
5. **Achtung:** Alle Dateien sind in ein Archiv (z.B. .zip) zu packen, um die Verwaltung der eingesandten Lösungen zu erleichtern.

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Wirkprinzip_-_Ergebnisse&oldid=27784“