

Software: SimX - Nadelantrieb - Vorbereitung

Aus OptiYummy

↑

← →

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

*Es irrt der Mensch solange er strebt.
- Johann Wolfgang von Goethe -
(Faust - Prolog im Himmel)*

Präge-Baugruppe für Brailleschrift-Drucker (erste Phasen des Entwurfsprozesses):

- Aufgabenstellung
- Aufgabenpräzisierung
- Konzeptphase

Modelle in der Konzeptphase:

- Besonderheiten und Zielstellung
- Installation der verwendeten Software (*SimulationX* & *OptiY*)

Einstieg in die Modellbildung (Dynamik-Simulation)

- Wichtig: Nutzung der teilnehmer-spezifischen Modelldatei!
- Antriebsmechanik (Präge-Nadel mit Rückholfeder) zur Einarbeitung in *SimulationX*



Einzusendende Ergebnisse (für den Erhalt 1 Bonus-Punktes):

- Teilnehmer der Lehrveranstaltung **Optimierung** laden ihre vollständig konfigurierte Modell-Datei **Etappe1_xx.isx** (xx=Teilnehmer-Nummer 01...99) im zugehörigen Opal-Kurs hoch.
- Das Modell muss in der bereitgestellten, teilnehmer-spezifischen Modelldatei entwickelt worden sein.
- Einsendeschluss ist die Nacht vor der ersten Übungsetappe. Die Nacht endet um 10:00 Uhr!

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Vorbereitung&oldid=27716“

Software: SimX - Nadelantrieb - Vorbereitung - Aufgabenstellung

Aus OptiYummy

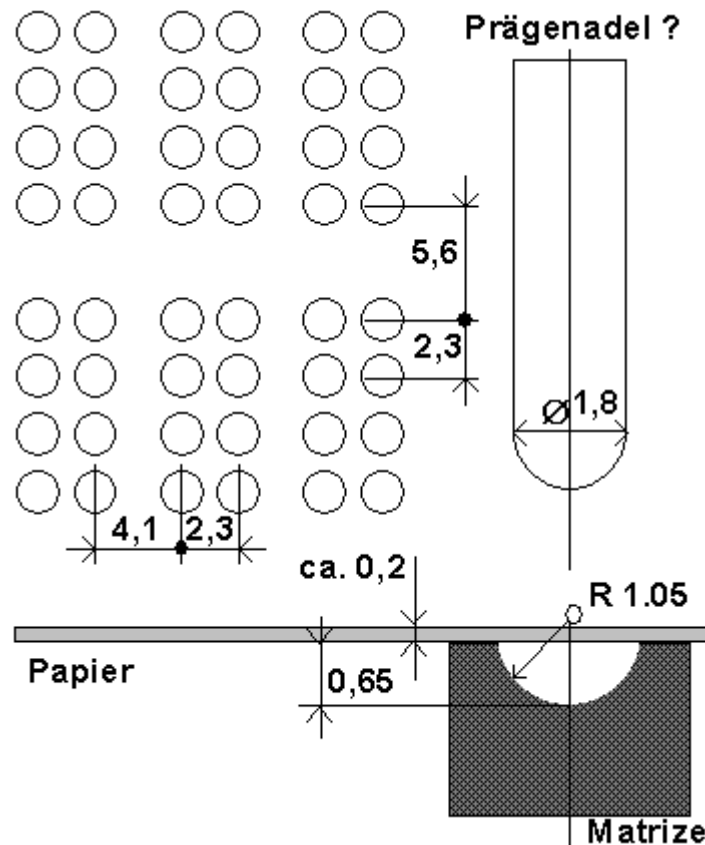
↑

← →

Präge-Baugruppe für Brailleschrift-Drucker



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:



Um eine gute haptische Lesbarkeit zu erzielen, soll mit dem zu entwickelnden Drucker die Prägung der Schriftzeichen mechanisch in das auch für manuelles Prägen verwendete Blindenschriftpapier erfolgen (Papierstärke 100-250 g/m²).

Dabei soll eine Prägeschwindigkeit von mindestens 130 Schriftzeichen pro Sekunde erreicht werden, um nach Einschätzung des Marketings im Vergleich zu Mitbewerbern auf dem Markt bestehen zu können.

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Vorbereitung_-_Aufgabenstellung&oldid=26884“

Software: SimX - Nadelantrieb - Vorbereitung - Aufgabenpraezisierung

Aus OptiYummy

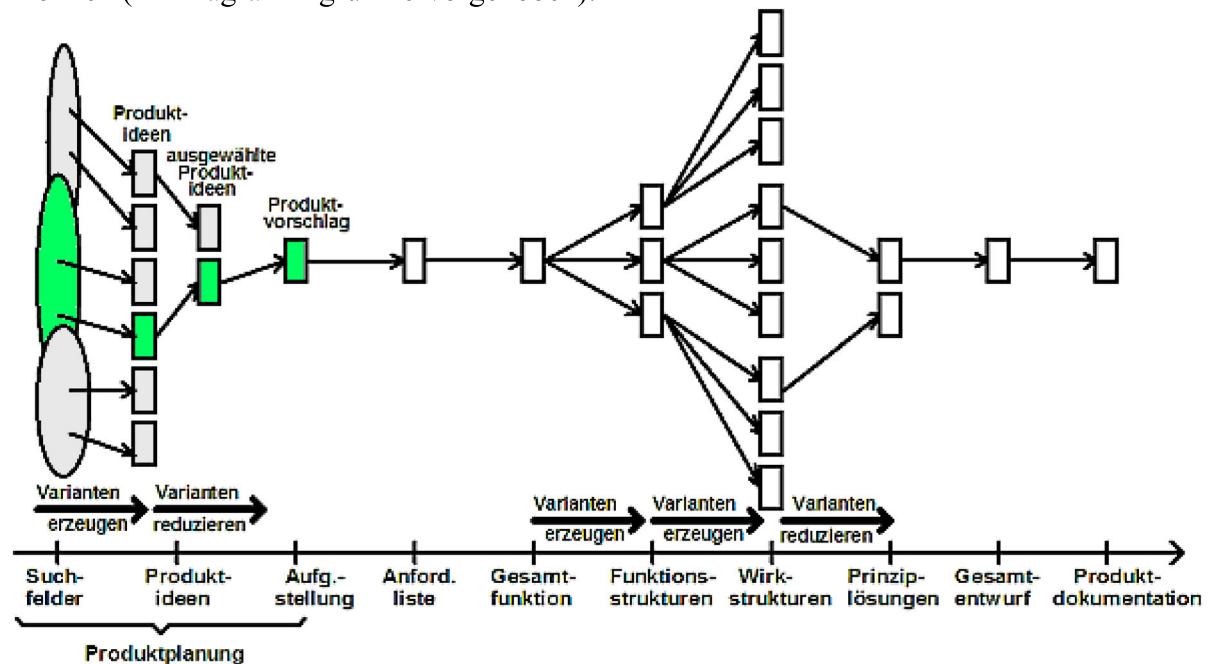
↑

← →
Aufgabenpräzisierung

Einordnung in den konstruktiven Entwicklungsprozess

Die Aufgabenstellung zur Entwicklung einer "Präge-Baugruppe für einen Brailleschrift-Drucker" wurde abgeleitet aus einem Projekt der firmeninternen Produktplanung zur Entwicklung eines hochwertigen Blindenschriftdruckers:

- Die **Produktplanung** ist dem Konstruktionsprozess vorgelagert. Das Ziel der Produktplanung besteht in der Formulierung "optimaler" Aufgabenstellungen für den eigentlichen Konstruktionsprozess.
- Wie auch im im nachfolgenden Konstruktionsprozess bedient man sich während der Produktplanung mehr oder weniger bewusst des Grundprinzips der natürlichen Evolution "**Varianten-Generierung** und **Selektion**", um im Sinne der "Gewinnmaximierung der Firma" zu optimalen Produktvorschlägen (Aufgabenstellungen) zu gelangen.
- Letztendlich sollen nur diejenigen Produktvorschläge im Konstruktionsprozess weiter bearbeitet werden, welche unter Berücksichtigung aller bekannten Aspekte zu einem auf dem Markt erfolgreichen Produkt führen können (im Diagramm grün hervorgehoben):



- Dieses Prinzip der Varianten-Generierung mit anschließender Selektion der besten Varianten (im Sinne der Gütekriterien) ist das universelle Optimierungsprinzip, auf dem der gesamte Evolutionsprozess in der Natur basiert.
- In der Technik "züchtet" der Ingenieur mit diesem Optimierungsprinzip im übertragenen Sinne die geforderten optimalen Ergebnisse:

1. Varianten-Erzeugung mit Hilfe von **Kreativitätstechniken**
2. Varianten-Selektion mit Hilfe von **Auswahl- und Bewertungsmethoden**

- Der gesamte Entwicklungsprozess ist durch ein ständiges Erzeugen und Auswählen von Varianten gekennzeichnet.

Aufgabenstellungen besitzen unterschiedliche Grade an Vollständigkeit, da sie sich auf Hauptmerkmale der angestrebten Lösung konzentrieren (Funktion, Fertigung, Kosten, ...):

- Idealer Weise liegen Aufgabenstellungen in Form eines **Lastenheftes** vor (was in unserem Übungsbeispiel nicht der Fall ist!).
- Die Aufgabenstellung stellt das Input-Dokument für den Konstruktionsprozess dar, welcher nach der vorgelagerten Produktplanung aktiv wird:

Produktplanung (vorgelagert)

: Erarbeiten von Produkt-Ideen
(in Form von Aufgabenstellungen)



AUFGABENSTELLUNG
(Lastenheft)

1. Aufgabenpräzisierung

: Erarbeiten der Anforderungsliste (Pflichtenheft)

2. Konzeptphase

: Festlegen der Prinziplösung

3. Entwurfsphase

: Festlegen der Gestalt

4. Ausarbeitungsphase

: Festlegen der Herstellungstechnologie



PRODUKT-
DOKUMENTATION

Anforderungsliste als zentrales Dokument der Aufgabenpräzisierung

Aufgabenpräzisierung ist die erste Phase des konstruktiven Entwicklungsprozesses:

- Aus der Sicht des Aufgaben-Bearbeiters (Konstrukteur, Entwickler) erfordert die übergebene Aufgabenstellung ("Lastenheft") immer einer Ergänzung durch zusätzliche Informationen → das nennt man "Aufgabenpräzisierung":
- Die "Anforderungsliste" ist das zentrale Dokument dieser Phase und wird vom Aufgaben-Bearbeiter erstellt.
- Sie stellt ein Verzeichnis aller "Anforderungen" an die Lösung in der Sprache der konstruierenden Abteilungen dar. Begriffe der "Alltagssprache" des Aufgabenstellers (z.B. des Kunden) müssen dafür häufig erst in Begriffe einer eindeutigen technischen Spezifikation "übersetzt" werden!

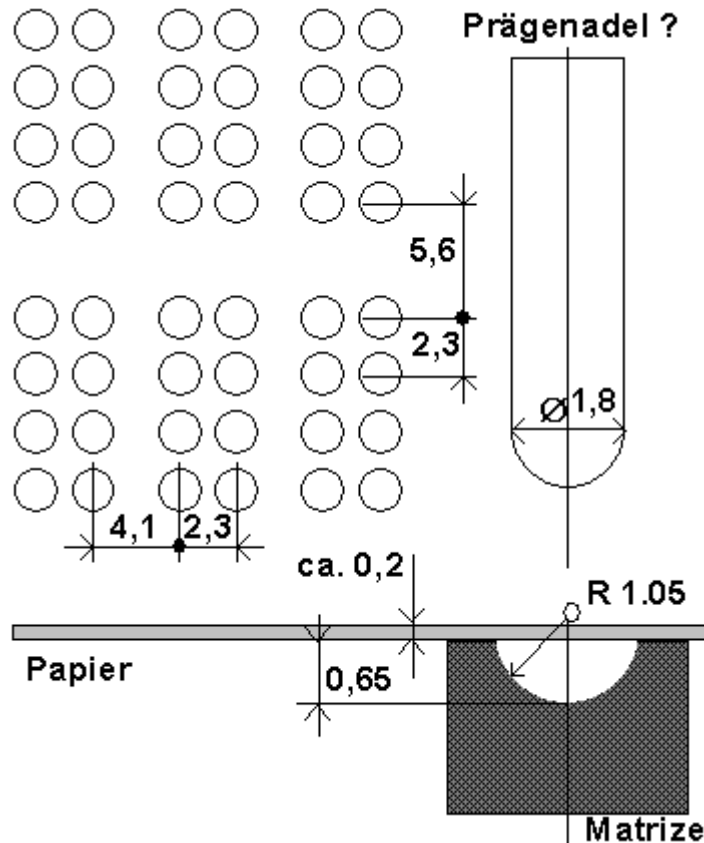
Anforderungen werden unterteilt in Forderungen und Wünsche:

- **Forderungen:**
 - Das sind die Anforderungen, welche unter allen Umständen erfüllt werden müssen (sonst wird die Lösung nicht akzeptiert!).
 - Sind immer gekennzeichnet durch quantitative Vorgaben:
 - z.B. Druckgeschwindigkeit ≥ 130 Zeichen/Sekunde (direkter Vorgabewert),
 - Qualitätsforderungen lt. Norm (z.B. Spritzwassergeschützt) (indirekte Vorgabewerte entsprechend der Norm)
- **Wünsche:**
 - Das sind die Anforderungen, welche nach Möglichkeit berücksichtigt werden sollen.
 - Meist qualitative Vorgaben (im Sinne "möglichst klein, schnell, geräuscharm, ...").
 - Eventuell wird dafür ein begrenzter Mehraufwand akzeptiert.
 - Wünsche sind zu Wichten nach hoher, mittlerer und geringer Bedeutung

Aufgabenpraezisierung fuer das Uebungsbeispiel

Unteraufgabe für den Übungskomplex aus der Aufgabenstellung "Präge-Baugruppe für einen Brailleschrift-Drucker":

- Es ist zu untersuchen, ob mit einem Elektromagneten ein Antrieb für eine Prägenadel realisierbar ist, sodass mindestens 130 Braillezeichen/s geprägt werden können:

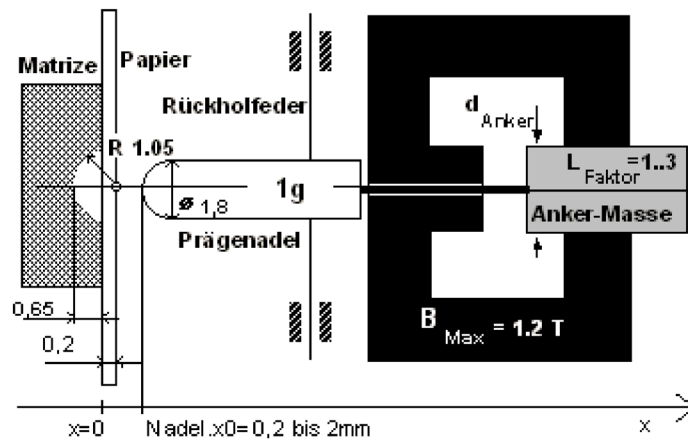


- Mit einer einzelnen Nadel müsste man dafür ca. 1000 Punkte/s prägen. Eine Recherche zu Magnetantrieben ergab, dass die erforderliche Zykluszeit kaum realisierbar ist. In der zu untersuchenden Antriebsvariante ist deshalb ein Prägekopf mit einer Spalte aus 4 Präge-Nadeln vorgesehen.
- Jede der vier Nadeln erhält einen eigenen Elektro-Magneten als Antrieb, welcher mindestens 260 Punkte/s prägen muss. Für einen Prägezyklus stehen damit $1/260 \text{ s} = \text{ca. } 3,8 \text{ ms}$ zur Verfügung. Mit einem Sicherheitszuschlag von 0,2 ms sollte ein Nadel-Antrieb einen kompletten Prägezyklus in 3,6 ms vollenden können.
- Für solch einen einzelnen Nadel-Antrieb ist eine grundlegende Dimensionierung so vorzunehmen, dass daraus die konkreten Fertigungsunterlagen für ein Versuchsmuster abgeleitet werden können.

Anforderungsliste für Präge-Nadelantrieb:

Forderungen

- Vorgaben zum Wirkprinzip und zur Geometrie:



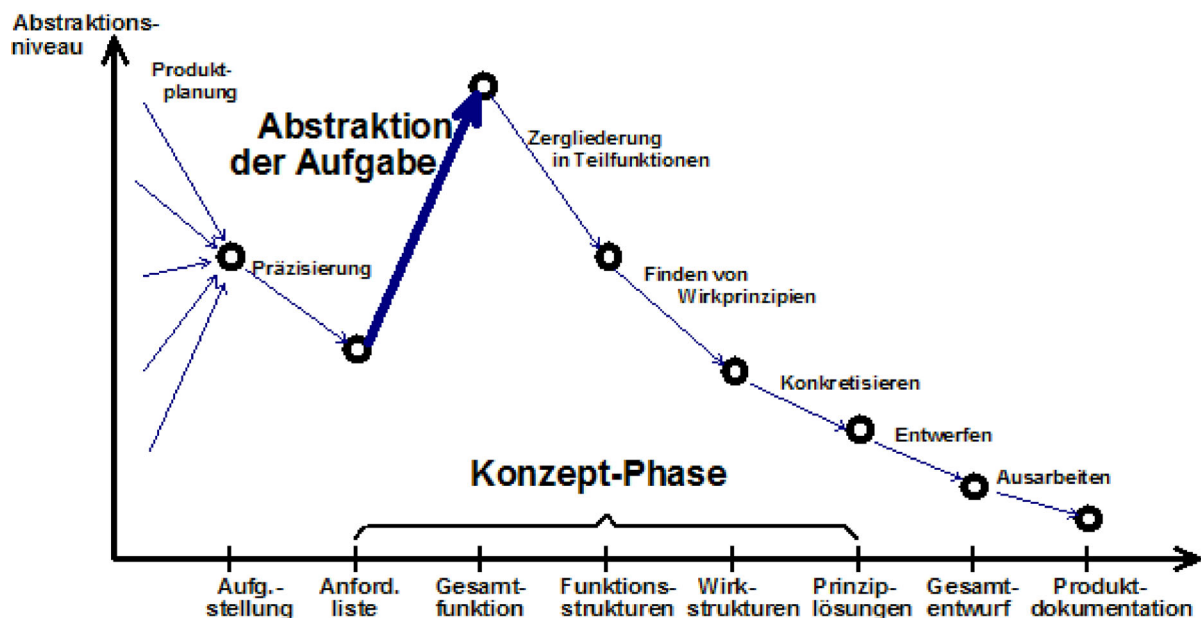
- Dimensionierung als "idealisierter" Topfmagnet.
- Energieversorgung: Gleichspannung **24 V** mit max. **1,5 A**
- Die Prägenadel soll starr mit dem Anker des Topfmagneten verbunden sein.
- Masse einer Präge-Nadel: in den Simulationen soll ein Schätzwert von ca. **1 g** verwendet werden (bis zur endgültigen Dimensionierung).
- Die Wiederherstellung einer stabilen Nadel-Ruheposition nach dem Prägen ist durch eine vorgespannte "Rückholfeder" zu realisieren (Stoßbelastung in Ruhe max. **20 g** → Erdbeschleunigung: $g=9,81 \text{ m/s}^2$)
- Zeit für einen Prägezyklus: **≤ 3,6 ms** (bei 130 Zeichen/s)
- Außendurchmesser des E-Magneten: **≤ 20 mm**
- Standardpapier: 180 g/m^2 mit einer Papierdicke von **0,2 mm**
- Spezialpapiere: im Bereich von **100 g/m^2** bis **250 g/m^2** (Papierdicken von **0,1 mm** bis **0,3 mm**)
- Nach dem Prägen in der **0,65 mm** tiefen Matrize soll das Papier an dieser Stelle nur noch eine Dicke von **0,10 mm** besitzen (Realisierung durch "Nadel-Anschlag")

Wünsche (gewichtet nach Reihenfolge)

1. möglichst kleine Zykluszeit (um mehr Reserven für "unvorhergesehene" Prozesse zu erhalten)
2. möglichst robuste Funktion im Rahmen zu erwartender Toleranzen (Fertigung, Material und Umgebungsbedingungen)
3. möglichst kleines Magnetkreis-Volumen (Bauraum-Minimierung)

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Vorbereitung_-_Aufgabenpraезisierung&oldid=27726“



- Der Abstraktionsschritt dient zum Lösen von Vorfixierungen und zum Befreien von konventionellen Vorstellungen.
- Dafür gibt es ausgehend von der präzisierten Aufgabenstellung eine zweckmäßige Vorgehensweise mit aufeinander folgenden Stufen der Abstraktion:
 1. Wünsche weglassen;
 2. Forderungen weglassen, welche die Funktion nicht unmittelbar betreffen;
 3. quantitative in qualitative Angaben umsetzen;
 4. redundante Angaben zusammenfassen;
 5. Problem lösungsneutral formulieren.

Diese Abstraktion führt zum **Wesen der Aufgabenstellung**, welches **in Form eines Satzes** zu notieren ist:

- Betrachtet man unser Übungsbeispiel, so besteht unabhängig von konkreten Anforderungen das Wesen dieser Aufgabe im

"Erzeugen von Braille-Schrift auf Papier"

- Damit ist die Aufgabe weitestgehend lösungsneutral formuliert und erleichtert das Finden und Verfolgen neuer, ungewöhnlicher Varianten.

Abstraktion:

- Distanzieren von konkreten Lösungsvorstellungen, hin zu idealisierter Funktion.
- Auf dem Niveau von Symbolen, Zeichen bzw. allgemeinen Begriffen arbeitet man gleichzeitig mit allen potentiellen Lösungen.

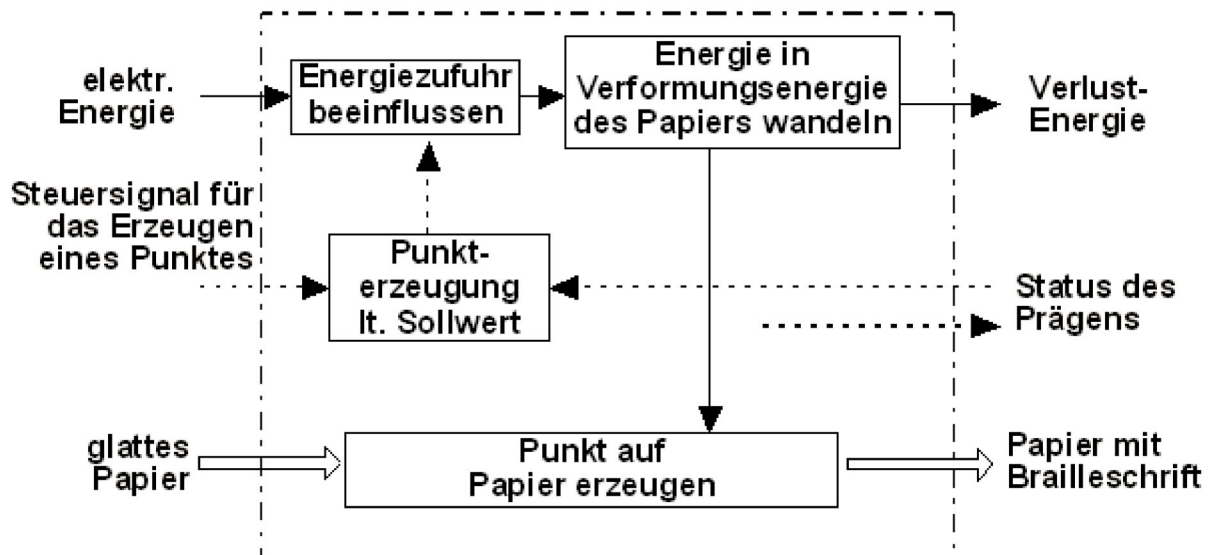
Konkretisierung:

- Realisieren von idealisierter Funktion durch unterschiedlichste Funktionsstrukturen.

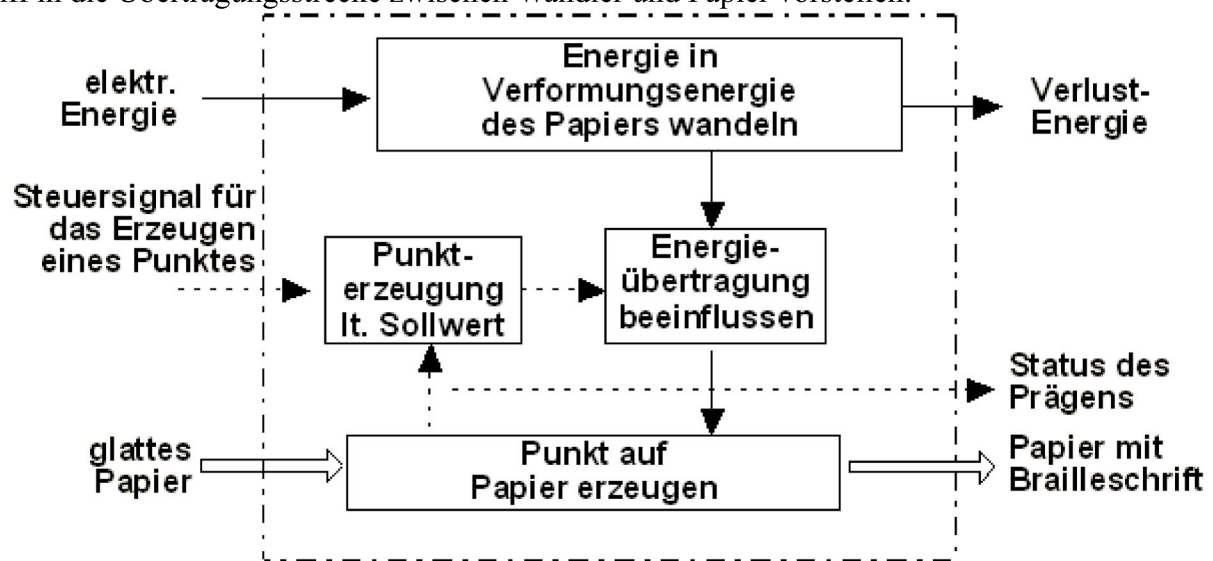
Strukturvarianten:

Am Beispiel des Blindenschriftträgers soll angedeutet werden, wie beim Zergliedern in Teilfunktionen zwangsläufig Strukturvarianten entstehen:

- Falls man einen Regelkreis für das Erzeugen des Punktes im Papier vorsieht, so kann man als eine Variante ein Stellglied für die elektrische Energiezufuhr vorsehen:



- Statt der Beeinflussung der Energiezufuhr für den Wandler kann man sich als weitere Variante auch einen Eingriff in die Übertragungsstrecke zwischen Wandler und Papier vorstellen:

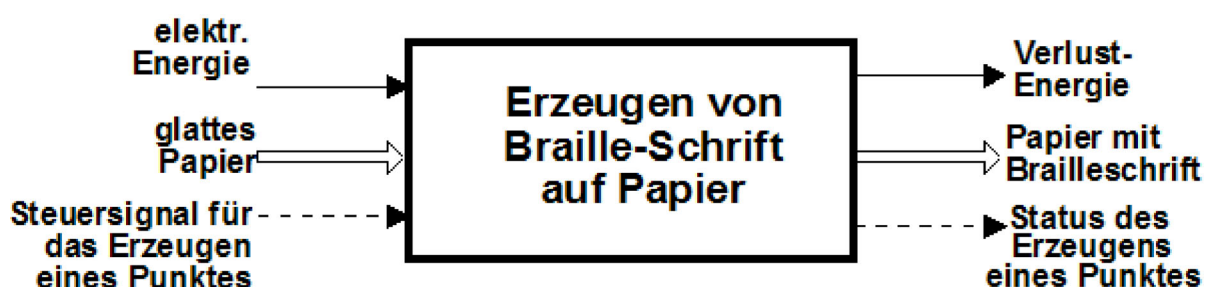


- Das Verhalten von Strukturvarianten kann nur sehr begrenzt numerisch simuliert werden (z.B. Zeitabläufe).
- Funktionsstrukturen können von Experten bewertet werden (z.B. Realisierbarkeit, Einfachheit, Neuheitsgrad, ...).

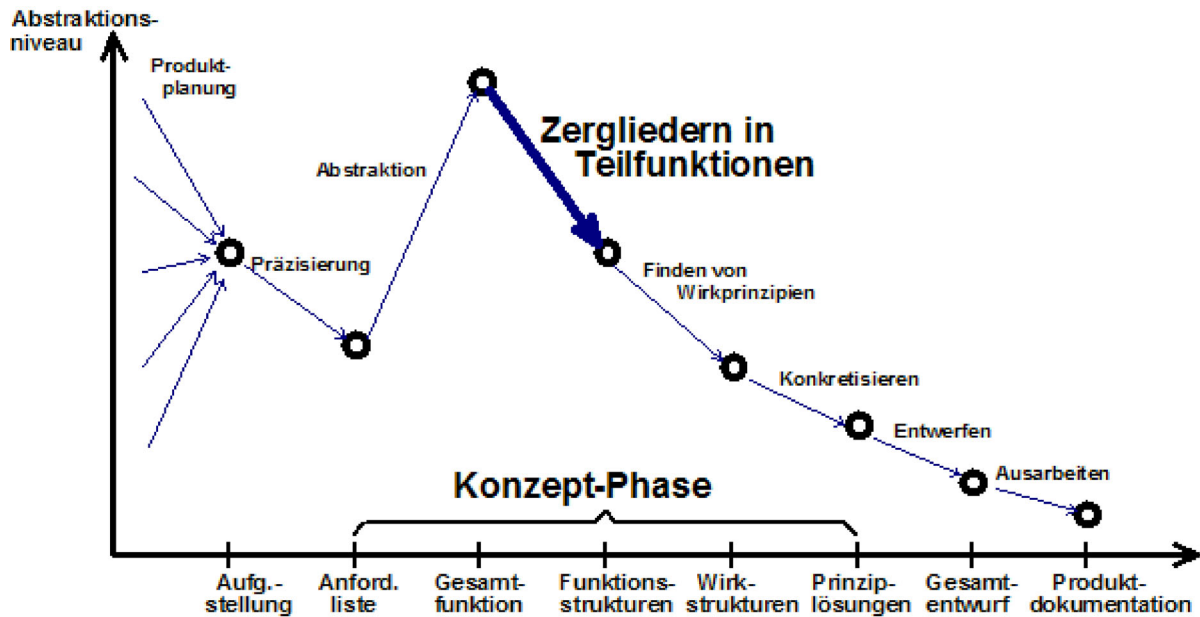
Entwickeln von Funktionsstrukturen

Ist man durch Abstraktion zum Kern der Aufgabenstellung vorgedrungen, so entspricht dieser Aufgaben-Kern der **Gesamtfunktion**, welche durch die Lösung zu realisieren ist. Die nächsten Lösungsschritte innerhalb der Konzeptphase bewegen sich auf dem Niveau von Funktionsstrukturen:

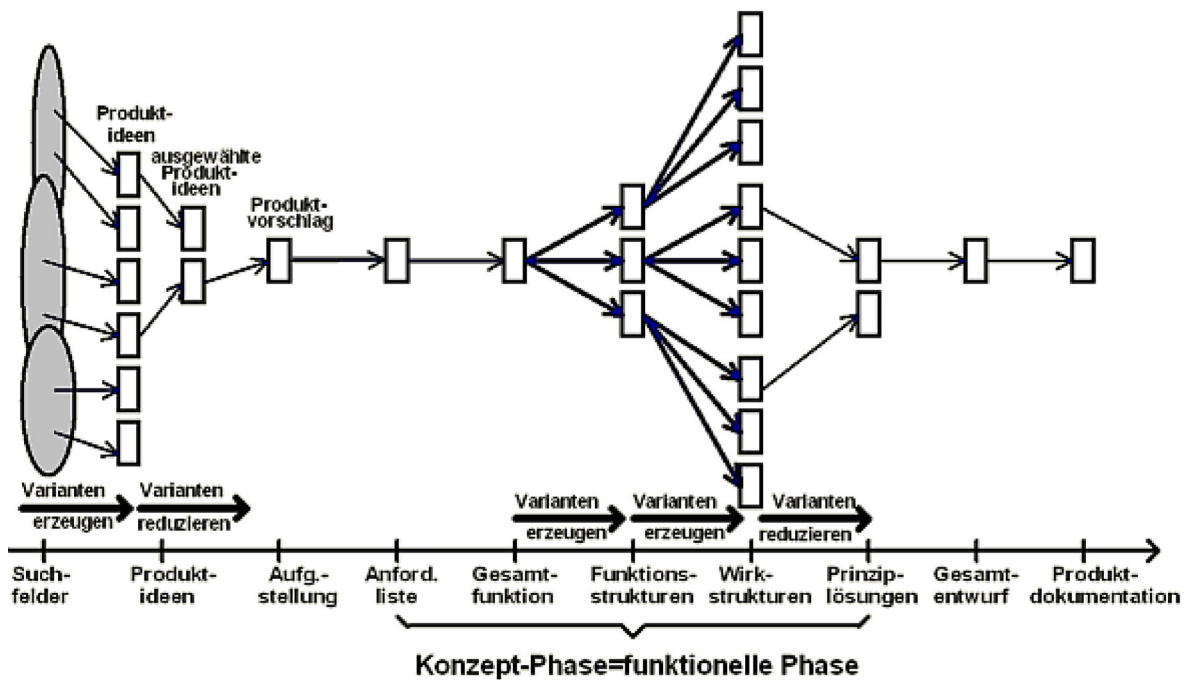
- **Funktionsstruktur** nennt man ein Blockschaltbild mit Stoff-, Energie- und Informationsfluss zur Darstellung des funktionellen Zusammenhangs zwischen Eingangs- und Ausgangsgrößen eines (meist technischen) Systems.
- **Gesamtfunktion** (=Kern der Aufgabenstellung) am Beispiel des Blindenschrift-Prägers:



- **Zergliedern in Teilfunktionen** erfolgt schrittweise ausgehend von der Gesamtfunktion. Dies verringert wieder das Abstraktionsniveau der Lösung:

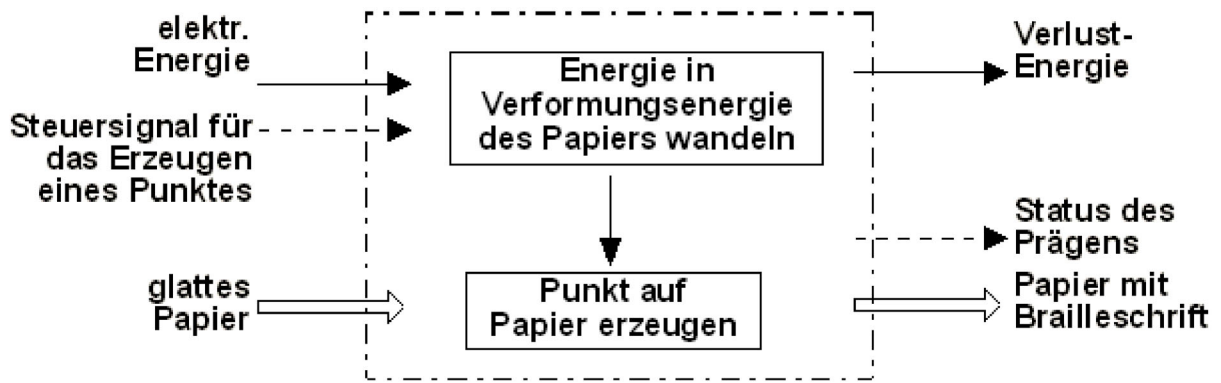


- **Funktionsstruktur-Varianten** entstehen zwangsläufig bei der Zergliederung in Teilfunktionen, weil eine Gesamtfunktion praktisch nie eineindeutig nur einer Struktur von Teilfunktionen zugeordnet werden kann:



Lösungsneutralität:

- **Vermeidung vorzeitiger Fixierung auf konkrete Lösungen**, indem man beim Zergliedern der Gesamtfunktion in Teilfunktionen möglichst lange allgemeine Begriffe verwendet (z.B.: *wandeln, ändern, verknüpfen, leiten, speichern*).
- Lösungsneutralität bedeutet dabei "frei von konkreter physikalisch-technischer Realisierung" (reine Funktion ohne das "Wie!"), z.B.:



- Die Lösungsneutralität gelingt nur bis zu einem gewissen Grad, da man bereits auf dem Abstraktionsniveau der Funktionsstrukturen Vorstellungen zu Wirkprinzipien und Gestaltung entwickelt.

Varianten von Wirkstrukturen

- Die abstrakten Teilfunktionen in den Funktionsstrukturen müssen als Nächstes mit Leben erfüllt werden.
- Hierbei geht man wieder vorsichtig zu Werke, indem man noch nicht nach konkreten stofflich-geometrischen Realisierungen sucht.
- Grundlage sowohl der technischen Konkretisierung als auch der Bildung numerischer Modelle bilden die Effekte:

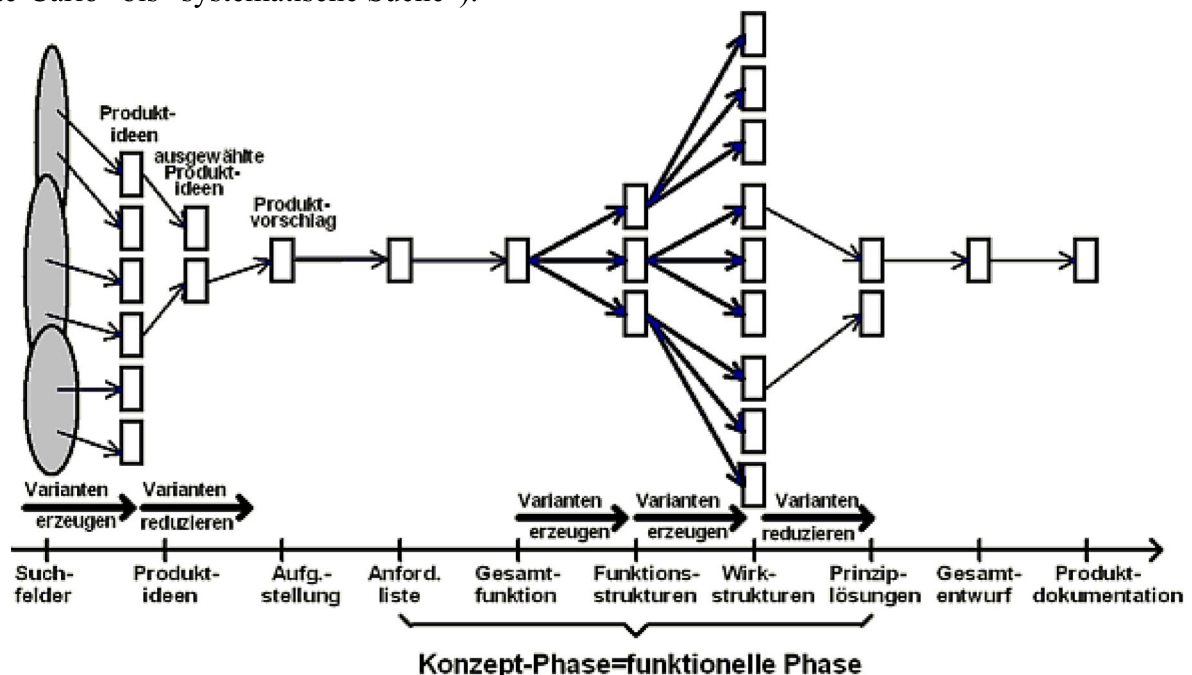
Effekt - Gesetz (Grundzusammenhang) zur Beschreibung eines physikalischen, chemischen, technischen o.ä. Geschehens

- Man untersucht als erstes, mit welchen Wirkprinzipien man die Teilfunktionen der Hauptfunktion realisieren kann:

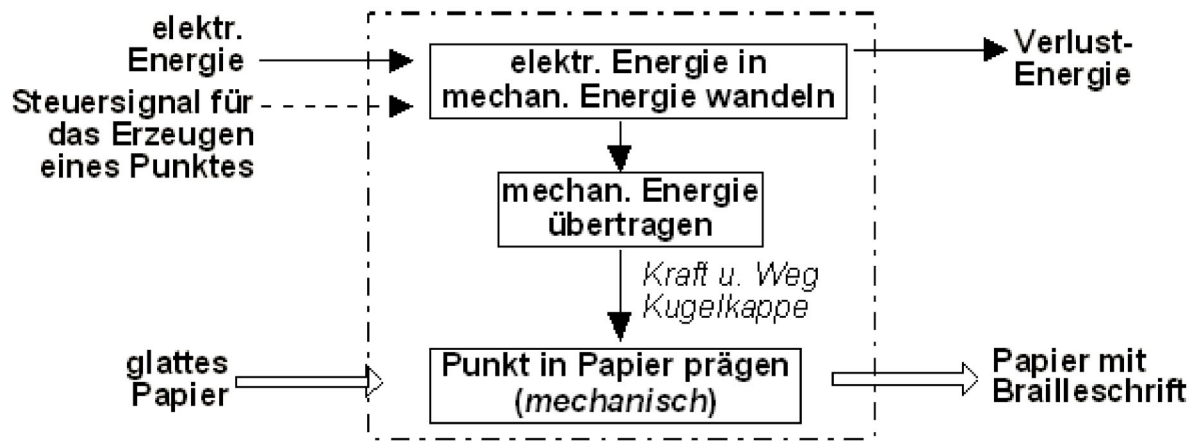
Wirkprinzip - Mittels eines Effektes bzw. geometrisch-stofflicher Merkmale realisierte Teilfunktion.

Wirkstruktur - Verknüpfung von Wirkprinzipien mehrerer Teilfunktionen zum Erfüllen der Gesamtfunktion.

- Für die **Ideenproduktion** bei der Varianten-Auffächerung gibt es eine Menge von Methoden (von "Monte-Carlo" bis "systematische Suche"):



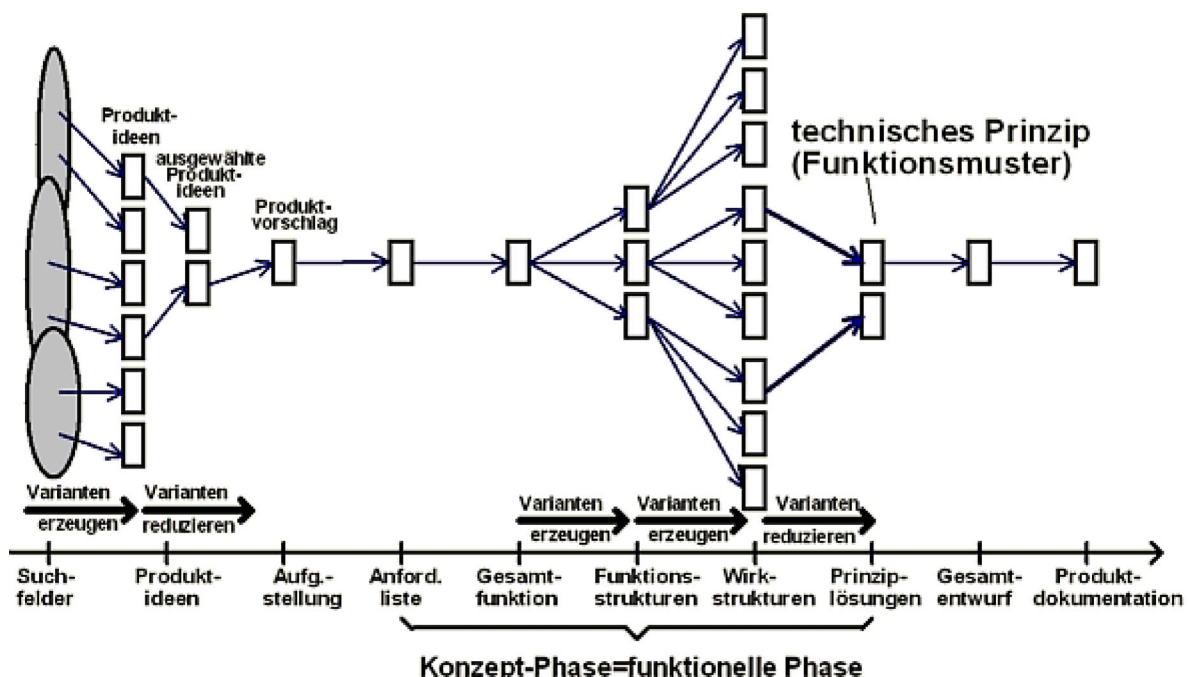
- Man erhält Wirkstruktur-Varianten, indem man für jede Teilfunktion unterschiedliche Wirkprinzipien ansetzt und diese kombiniert (Beispiel "Blindenschrift-Präger"):



- **Elektrische Energie -> mechanische Energie:** E-Magnet, Motor, Piezowandler, Elektrostatisch, ...
 - **mechanische Energie übertragen (und speichern):** Hebel, Gelenkmechanismus, Elastischer Körper, Schwungmasse, ...
 - **Punkt in Papier prägen (mechanisch):** Kugelkappe linear bewegt; Kugelkappe auf Papier abgerollt
- Wirkprinzipien und deren Verknüpfungen zu Wirkstrukturen können auf der Basis ihrer Effekte bereits numerisch im Rahmen eines Systemmodells simuliert und optimiert werden!

Prinziploesung

- Durch sinnvolle Kombination von Wirkprinzipien kommt man zu einer Vielfalt von Wirkstrukturen (z.B. für das mechanische Prägen von Blindenschrift).
- Diese Vielfalt muss danach auf der Basis von **Auswahl- und Bewertungsmethoden** reduziert werden (möglichst auf eine optimale Variante):



- Da Musterexperimente sehr teuer und auch zeitaufwendig sind, sollte man dabei soweit wie möglich auf die numerische Simulation und Optimierung zurückgreifen.
- Am Ende der Konzeptphase steht in der Gerätetechnik das technische Prinzip im Sinne eines ersten Funktionsmusters. Die Gestaltung der Baugruppen muss zumindest in Form von Skizzen soweit vorangeschritten sein, dass durch die Werkstatt das Funktionsmuster gebaut werden kann.
- Der Trend geht dabei weg von der Handskizze und hin zur Nutzung von parametrisierten 3D-CAD-Modellen für diesen ersten Grobentwurf, da darauf aufbauend sowohl das Rapid Prototyping als auch die eigentliche Entwurfsphase nahtlos anschließen können.

- Die funktionelle Phase endet nach dem Nachweis der prinzipiellen Funktionsfähigkeit mit der Freigabe zum Entwerfen.
- **Beachte:** Im Rahmen der Aufgabenpräzisierung erfolgten beim Übungsbeispiel bereits grundlegende Festlegungen zur Wirkstruktur für das zu realisierende Funktionsmuster!
 - Elektrische Energie -> mechanische Energie: **E-Magnet** (als Topfmagnet)
 - mechanische Energie übertragen (und speichern): **Prägenadel** (starr mit Magnetanker verbunden) und **Rückholfeder**
 - Punkt in Papier prägen (mechanisch): **Kugelkappe** (als "Spitze" der Prägenadel) und **Matrize**

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Vorbereitung_-_Konzeptphase&oldid=27723“

Software: SimX - Nadelantrieb - Vorbereitung - Modelle in Konzeptphase

Aus OptiYummy

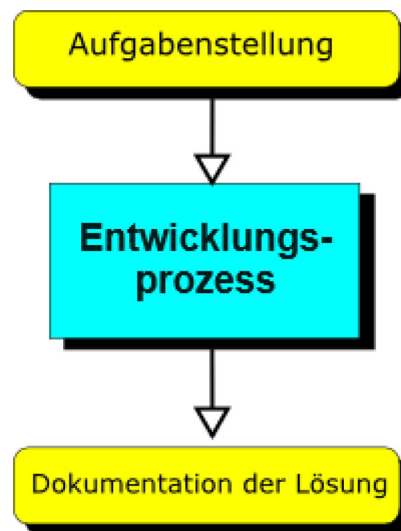
↑

← →

Modelle in der Konzeptphase - Besonderheiten und Zielstellung

Einordnung des Uebungsbeispiels in den Konstruktionsprozess

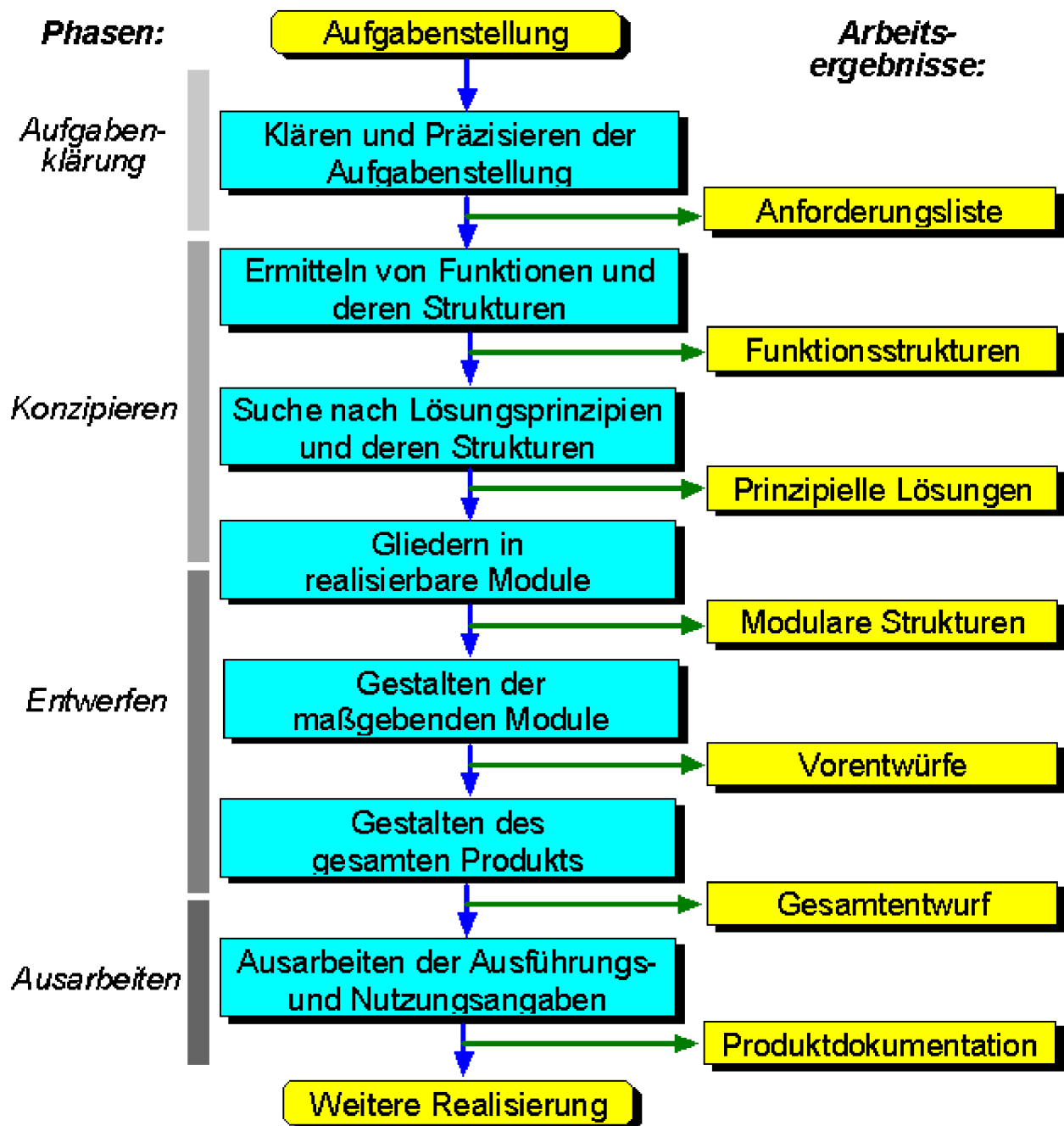
Der in Forschung/Entwicklung tätige Ingenieur soll Lösungen für technische Probleme finden. Der konstruktive Entwicklungsprozess ("Konstruktionsprozess") umfasst dabei den Abschnitt zwischen einer bereits vorliegenden Aufgabenstellung und der fertigen Produkt- bzw. Prozessdokumentation als Ergebnis der Aufgabenlösung:



- **Entwerfen** ("Entwickeln") bedeutet in der Technik, ausgehend von einer Aufgabenstellung das zu entwickelnde Objekt (z.B. ein Produkt oder einen technologischen Prozess) "vorausdenken".
- Als Ergebnis eines technischen Entwurfsprozesses liegt eine "Beschreibung" des zu entwickelnden Objektes und bestimmter Anforderungen an seine Realisierung vor - nicht das "Objekt" selbst!

Entwurfsprozesse als zielgerichtete geistige und schöpferische Leistungen besitzen eine sehr individuelle Ausprägung. Diese Ausprägung erfolgt sowohl durch die Organisationen in denen diese Prozesse ablaufen, als auch durch die konkreten Bearbeiter einer Entwurfsaufgabe:

- Speziell auf dem Gebiet der Konstruktion bemüht sich seit über einem halben Jahrhundert die Konstruktionstechnik als Technikwissenschaft, die firmenspezifische Organisation von Konstruktionsprozessen zu verallgemeinern.
- Aus diesen Bemühungen resultierte z.B. 1973 die generelle Vorgehensweise im Konstruktionsprozess nach der Richtlinie **VDI 2221**:



Beachte:

- "Entwerfen" beschreibt innerhalb des Konstruktionsprozesses die Phase des maßlichen Gestaltens, welche sich an die Konzeptphase anschließt.
- "Entwurfsprozess" steht umgangssprachlich aber auch als Synonym für den gesamten "konstruktiven Entwicklungsprozess" oder kurz "Konstruktionsprozess".

Alle Übungsetappen zum Beispiel "Prägenadel-Antrieb" beziehen sich auf den Abschnitt "**Suche nach Lösungsprinzipien und deren Strukturen**" innerhalb der Konzeptphase. Innerhalb dieser Übungsetappen stehen dabei zwei wesentliche Aspekte im Mittelpunkt:

- Wie kommt man von einer vorgegebenen Funktionsstruktur zu einer optimalen Wirkstruktur?
- Wie gewinnt man aus einer Wirkstruktur für die geometrisch-stoffliche Grobdimensionierung einer "prinzipiellen Lösung" die benötigten Entwurfparameter?

Besonderheiten für Modelle in der Konzeptphase

Modelle sind Ersatzobjekte, welche man zur Gewinnung von Erkenntnis über ein Originalobjekt benutzt:

1. Ein Modell ist immer ein Abbild von einem "Original" (d.h. von "etwas anderem"). Bei einem "Original" kann es sich auch um ein Modell handeln!
2. Ein Modell bildet nur diejenigen Eigenschaften des Originals ab, die dem Modellbenutzer wichtig erscheinen:
 - Bestimmte Eigenschaften eines Modells werden als Eigenschaften des Originals interpretiert.
 - Es gibt immer Eigenschaften eines Modells, die keinen Bezug zum Original besitzen.
3. Ein Modell wird pragmatisch und zweckorientiert angewendet:
 - "Minimalmodell" → mit möglichst wenig Modell hinreichend viel Erkenntnis!
 - Erlaubt ist, was im Sinne des angestrebten Erkenntnisgewinns nützt.

Innerhalb des konstruktiven Entwicklungsprozesses entsteht nur eine "Anleitung", wie das angestrebte Produkt (das "Original") zu realisieren ist! Damit ist der Entwickler zwangsläufig auf die Verwendung von "Modellen" unterschiedlichster Natur angewiesen, z.B.:

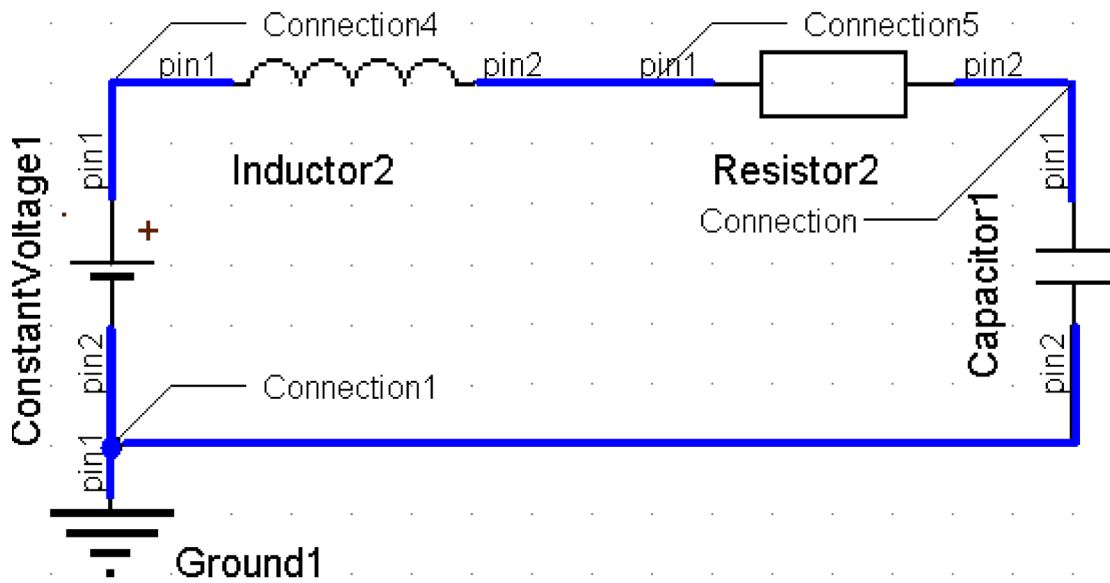
- Wissen über den Objektbereich mit Fachliteratur und Standards/Normen als externe Wissensspeicher → Basis für Gedankenexperimente
- Blockschaltbilder und Skizzen → externe Hilfsmittel für Gedankenexperimente
- Dimensionierungsformeln (Papier & Bleistift, Taschenrechner, Dimensionierungstools in CAD-Programmen)
- Computermodelle (CAD, Finite Elemente Methode, Netzwerk-Analogien, Computer-Algebra)
- Versuchsmuster im Labor (Teile, Baugruppen, Gerät - reduziert auf bestimmte Aspekte: z.B. als Prototyp).

Klassische Methoden mit "Kopf, Hand, Stift und Papier" sind weiterhin wertvolle Hilfsmittel im Erkenntnisprozess. Der Trend geht jedoch auf Grund des Automatisierungspotentials eindeutig in Richtung der verstärkten Nutzung von Computermodellen:

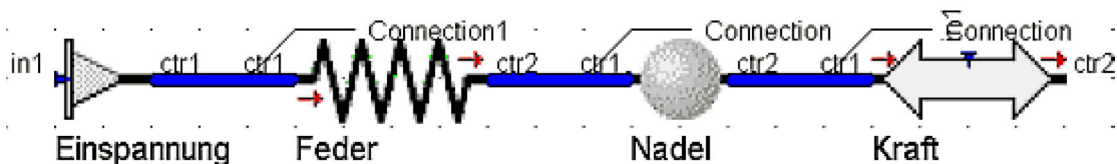
- Im Rahmen von Konstruktionsprozessen denkt man dabei sofort an CAD-Programme mit ihren vielfältigen Analyse-Möglichkeiten.
- CAD-Modelle erfordern jedoch zumindest Informationen zur Grobgeometrie bzw. zur Geometrie der Schnittstellen von Baugruppen und Bauteilen des zu konstruierenden Produktes.
- Solche Geometrie-Informationen stehen zu Beginn der Konzeptphase nur fragmentarisch als Bestandteil der Anforderungsliste zur Verfügung.

Einen Ansatzpunkt für physikalisch fundierte Modellbildung bieten die in der Konzeptphase entwickelten Wirkstrukturen mit ihren physikalischen Effekten:

- In diesem Abschnitt der Konzeptphase liegt der Schwerpunkt auf der sogenannten Netzwerkmodellierung, wie sie für "**elektrische Netzwerke mit konzentrierten Elementen**" bekannt ist:



- Durch Anwendung von Analogiebeziehungen ist es möglich, solche Netzwerkmodelle für unterschiedlichste physikalische Domänen zu entwickeln (Wärme, Fluidtechnik, Magnetfelder, Regelungstechnik, ...) → z.B. Mechanik:



- Dafür hat sich der Begriff der "physikalisch-objektorientierten Modellierung" etabliert.
- Diese "Netzwerkmodellierung mit konzentrierten Elementen" repräsentiert die natürlichste Weise der physikalischen Verhaltensbeschreibung, da sie direkt auf den physikalischen Effekten basiert.
- Als Modellierungssprache hat sich dafür **Modelica** als Quasi-Standard durchgesetzt. Für diese objektorientierte "Programmiersprache" existieren unterschiedliche grafische Benutzer-Oberflächen mit umfangreichen Bibliotheken von Modellelementen.
- Mit dieser grafischen "Schaltungssicht" ist eine sehr intuitive Modellierung komplexer technischer Systeme und deren Simulation im Zeitbereich möglich.
- Im Rahmen des Übungsbeispiels verwenden wir "**SimulationX**" als einer der ausgereiftesten grafischen Entwicklungsumgebungen für die physikalisch-objektorientierten Modellierung. Die freie Verfügbarkeit einer eingeschränkten *Express Edition* für nichtkommerzielle Zwecke ermöglicht die Bearbeitung der Übungsetappen auch auf privaten Computern.

Zielstellung der Modellnutzung in der Konzeptphase

Es existieren zwei grundsätzliche Zielstellungen für die Entwicklung von Modellen beim Dimensionieren einer Prinziplösung ("Versuchsmuster") auf Grundlage einer Wirkstruktur:

1. **Optimale Dimensionierung der Prinziplösung** bei Berücksichtigung aller relevanten Effekte ("Hauptziel").
2. **Forderungen physikalisch erfüllbar?** -> Auswirkung der schrittweise zu berücksichtigenden physikalischen Effekte auf die Funktion (bei jeweils optimaler Dimensionierung!).

Die Modellentwicklungsumgebung besitzt bereits die erforderlichen numerischen Verfahren für die Simulation des zeitlichen Verhaltens konkret parametrisierter Modelle:

- Auf dieser Basis sind einfache Analysen durch Verändern effekt-bestimmender Parameter möglich und damit auch Erkenntnis zur Auswirkung einzelner physikalischer Effekte auf das Verhalten.
- Es fehlen jedoch z.B. im Programm *SimulationX* geeignete Tools der statistischen Versuchsplanung (zur detaillierten Toleranzanalyse) und der Optimierung (zur optimalen geometrisch-stofflichen Dimensionierung).

- Wir nutzen deshalb im Rahmen der Übungsetappen als zusätzliches Programm **OptiY**, welches die im *SimulationX* fehlende Funktionalität zur Verfügung stellt. Die freie Verfügbarkeit einer einschränkten *Trial Edition* ermöglicht ebenfalls die Bearbeitung der Übungsetappen auf privaten Computern.

Im Ergebnis der Konzeptphase muss eine geometrisch-stoffliche Dimensionierung für eine Prinziplösung vorliegen:

- Grundlage dieser Dimensionierung sollten überwiegend hinreichend genaue Minimalmodelle zur Simulation und Optimierung bilden.
- Praktische Experimente sollten bei der Prototyp-Entwicklung aus Kostengründen auf die Gewinnung erforderlicher Modellparameter beschränkt bleiben.
- Auf Grundlage der praktischen Messungen am Versuchsmuster erfolgt die Validierung der verwendeten Modelle → eventuell muss erneut ein Versuchsmuster mit verbesserten Modellen dimensioniert werden!
- Entsprechen die Ergebnisse den Erwartungen, kann man die erarbeitete Prinziplösung mit parallel entwickelten Prinzipvarianten vergleichen.

Mit der Entscheidung für eine "optimale" Prinziplösung verlässt man dann die "Konzeptphase" und kann sich in der anschließenden "Entwurfsphase" der schrittweisen geometrisch-stofflichen Dimensionierung der endgültigen Lösung widmen.

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Vorbereitung_-_Modelle_in_Konzeptphase&oldid=27724“

Software: SimX - Nadelantrieb - Vorbereitung - Software-Installation

Aus OptiYummy

↑

← →

Installation der verwendeten Software (SimulationX & OptiY)

Wichtig:

- Die benötigte Windows-Software existiert nur als 64-Bit-Version und erfordert deshalb ein 64-Bit Windows-Betriebssystem.
- Den Zugang zum Download der zu verwendenden Software-Versionen erhalten die eingeschriebenen Teilnehmer der Lehrveranstaltung "**Optimierung**" über den Opalkurs.
- Durch Verwendung einheitlicher Software-Versionen soll die Fehlersuche und -behebung bei eventuell auftretenden versionsbedingten Fehlern vereinfacht werden.
- Für weitere Interessierte wird zur Beschaffung der erforderlichen Software auf die entsprechenden Internetseiten der beiden Firmen verwiesen: **ESI Academy portal** und **OptiY** → Man beachte, dass **OptiY** eventuell die neueste **SimulationX**-Version noch nicht unterstützt, weil die erforderliche Schnittstellen-Anpassung noch nicht erfolgte!
- Existieren privat nicht die erforderlichen technischen Voraussetzungen für die Installation der benötigten Software, müssen Teilnehmer der Lehrveranstaltung "**Optimierung**" rechtzeitig den Betreuer darüber informieren, damit der Zugang zum zugehörigen PC-Pool bereits vor der ersten Übungsetappe ermöglicht werden kann!

SimulationX-Installation und Programmstart

Die Installation von *SimulationX* beginnt nach dem Starten einer ca. 1 GByte großen EXE-Datei (z.B. SimulationX-4.5.1.76322.exe):

- **Wichtig:** Erfolgte der Download in Form einer ZIP-Datei, so muss die enthaltene EXE-Datei zuvor entpackt werden (EXE-Datei nicht direkt im ZIP-Ordner starten!).
- Nach dem Start der EXE-Datei ist die unter Windows übliche Zustimmung für die Installation erforderlich.

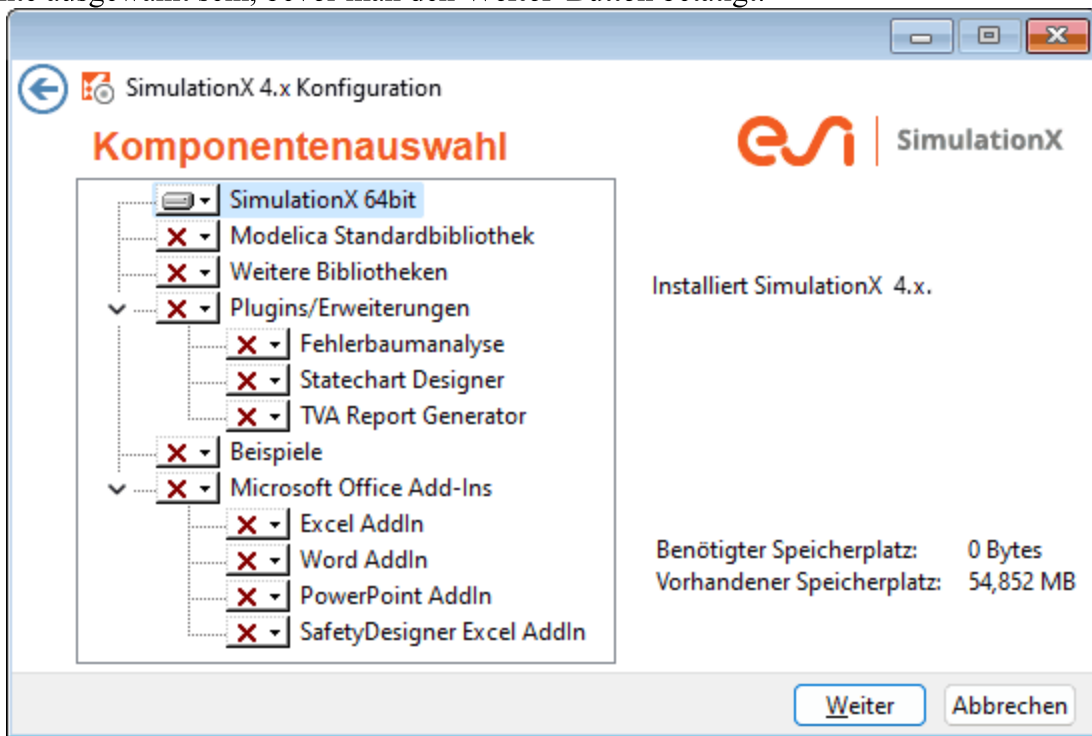
Das *SimulationX*-Programm wurde im Verlauf seiner Entwicklung im Funktionsumfang immer mächtiger:

- Im Rahmen der Übungsetappen für das Beispiel "Präge-Nadelantrieb" benötigen wir nur die Grundfunktionalität von *SimulationX*.
- Zur Erhöhung der Übersichtlichkeit beziehen sich die Übungsanleitungen auf die Programm-Oberfläche einer entsprechend angepasste Installation.
- Es wird empfohlen, auf dem privatem PC solch eine angepasste *SimulationX*-Installation vorzunehmen.

Empfohlene angepasste SimulationX-Installation

- Die Übungsanleitung bezieht sich auf die deutsche Sprachversion → **Select Setup Language -> German**.
- Bei Bedarf erfolgt automatisch die Installation und Konfiguration von "**Visual C++ Redistributable**" (Laufzeitkomponenten von Visual C++-Bibliotheken).

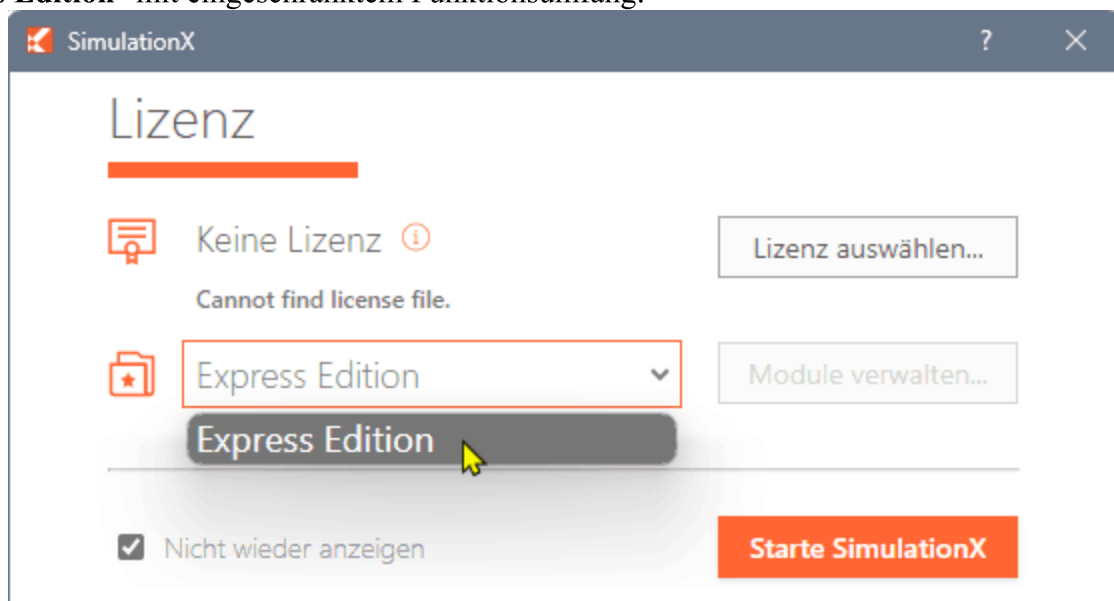
- Entscheidend ist dann der Schritt der **angepassten Installation** - hier sollte nur "SimulationX" als Komponente ausgewählt sein, bevor man den **Weiter**-Button betätigt:



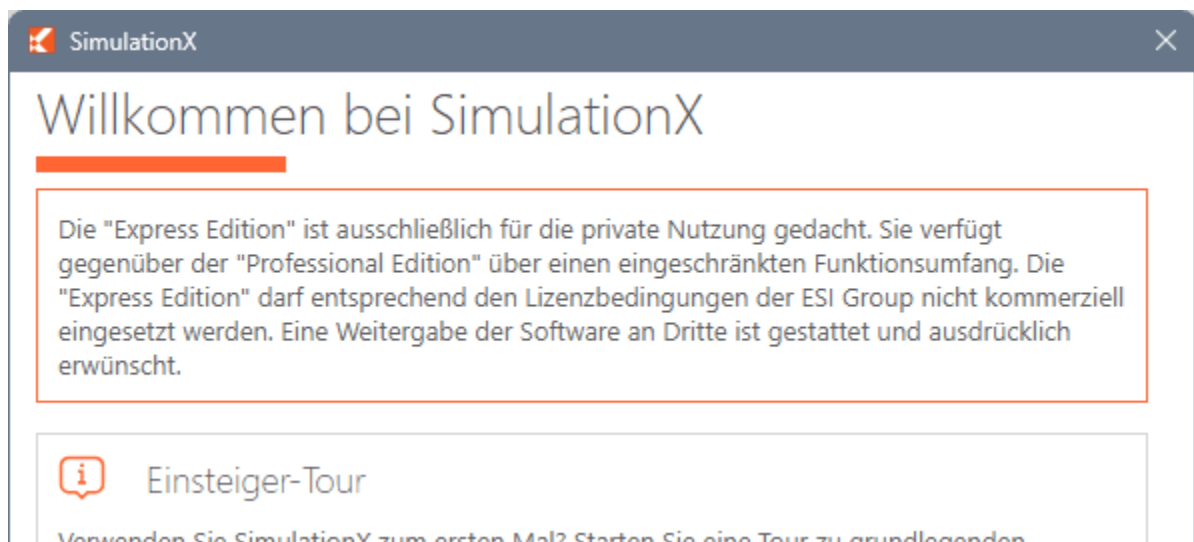
- Das angebotene Zielverzeichnis möglichst nicht ändern → z.B. "**C:\Program Files\ESI Group\SimulationX 4.x**"
- Die Verknüpfungen auf Desktop und im Startmenü sollten für alle Nutzer des Computers erstellt werden.
- Die eigentliche Installation dauert dann sehr lange (mehrere Minuten) und man darf dabei nicht die Geduld verlieren!

Nachdem die Installation erfolgreich beendet ist, kann man das *SimulationX* starten:

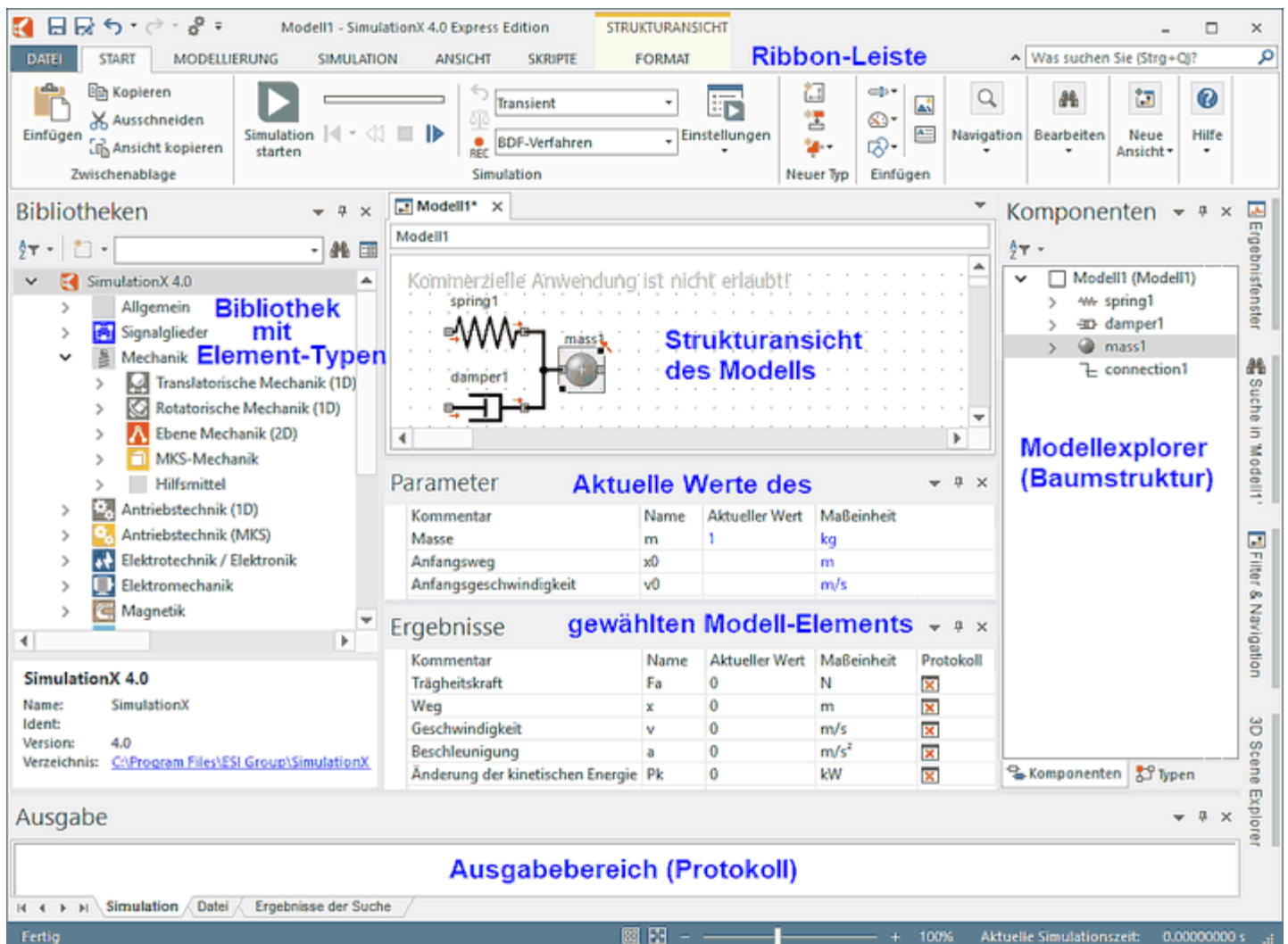
- Beim ersten Programmstart erfolgt zuerst eine Abfrage in Hinblick auf die an die Entwickler zu übertragenden Nutzungsdaten.
- Danach muss man die zu verwendende *SimulationX*-Lizenz auswählen. Im Rahmen der Übung genügt die "**Express Edition**" mit eingeschränktem Funktionsumfang:



- Die private Nutzung und Weitergabe der Software als "Express Edition" ist laut Lizenzbedingungen zulässig bzw. sogar erwünscht, wie die eingeblendete "Willkommen-Seite" zeigt:

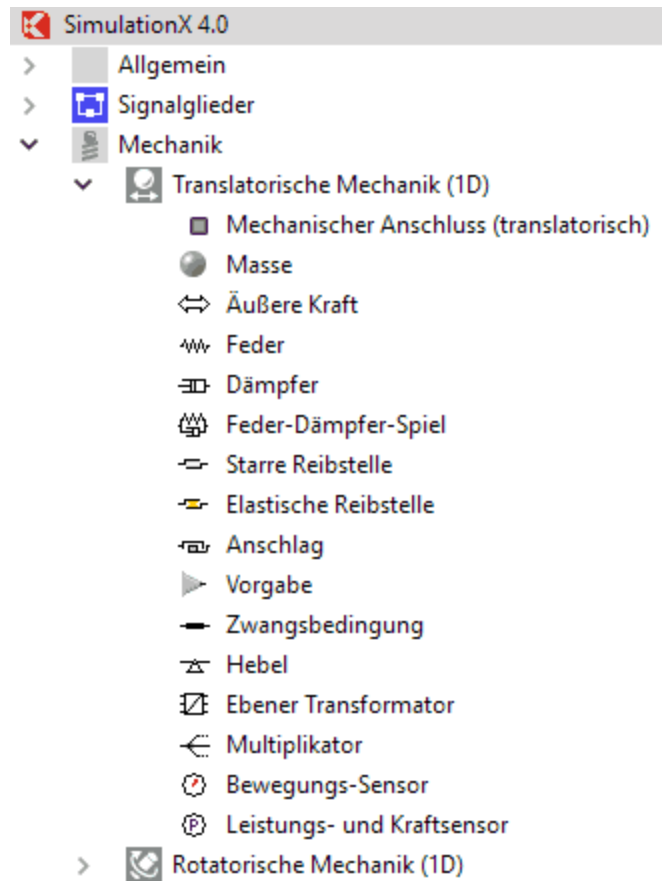


Die Programmoberfläche ab Version 4.0 hat in der Grundeinstellung folgende wesentlichen Bereiche:



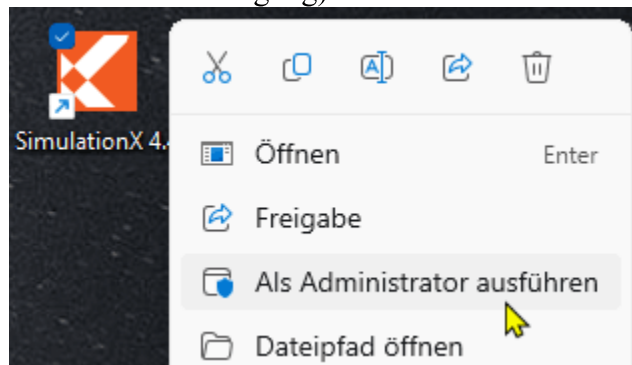
- In der **Leiste der Modellbibliothek** findet man ein reichhaltiges Angebot an Elemente-Typen aus allen möglichen Fachgebieten.
- In den **Bereich für die Strukturansicht des Modells** kann man diese Elemente einfach mit dem Cursor ziehen:
 - Wie in einem "echten" Versuchsstand kann man die Elemente miteinander verbinden.
 - Zum Verbinden nutzt man wieder den Cursor.
 - Das gelingt aber nur, wenn die Element-Anschlüsse zueinander passen (auch in der wirklichen Welt funktioniert eine elektrische Leitung meist nicht besonders gut als Wasserleitung!)

- Im **Modellexplorer** wird das aufgebaute Modell als Baumstruktur dargestellt:
 - Die Form dieser Darstellung entspricht der Ordneransicht im Windowsexplorer.
 - In dieser Baumstruktur werden die Elemente und die Verbindungen abgebildet.
- Die **Werte des aktuell ausgewählten Modell-Elements** erscheinen in einem separaten Bereich (Auswahl des Elements in der Strukturansicht oder in der Baumstruktur)
- Der **Ausgabebereich** dient dem Protokoll, so erscheinen dort z.B. Fehlermeldungen bei der Arbeit mit dem Modell (hoffentlich nicht!).



Wichtig:

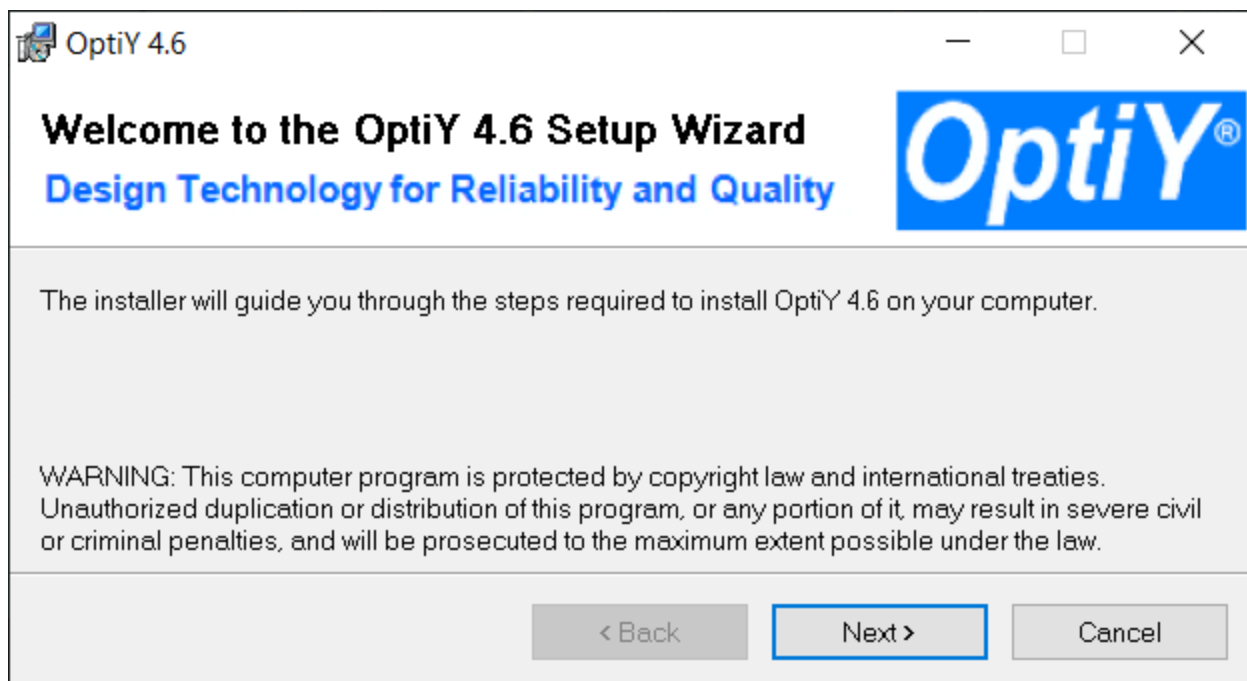
- Das *SimulationX* wird zum Zwecke der optimalen Dimensionierung der Entwurfparameter nicht nur manuell bedient, sondern muss automatisiert vom zusätzlich genutzten Optimierungsprogramm "ferngesteuert" werden.
- Trotz eingeschränkter Benutzerrechte muss dafür sichergestellt sein, dass alle notwendigen Eintragungen im Windows-Betriebssystem erfolgt sind.
- Es kann sein, dass dies unter den aktuellen Software-Bedingungen bereits gewährleistet ist.
- Man sollte aber trotzdem das *SimulationX* einmalig "**als Administrator ausführen**" (über "Kontextmenü auf Programm-Icon" und anschließende Bestätigung):



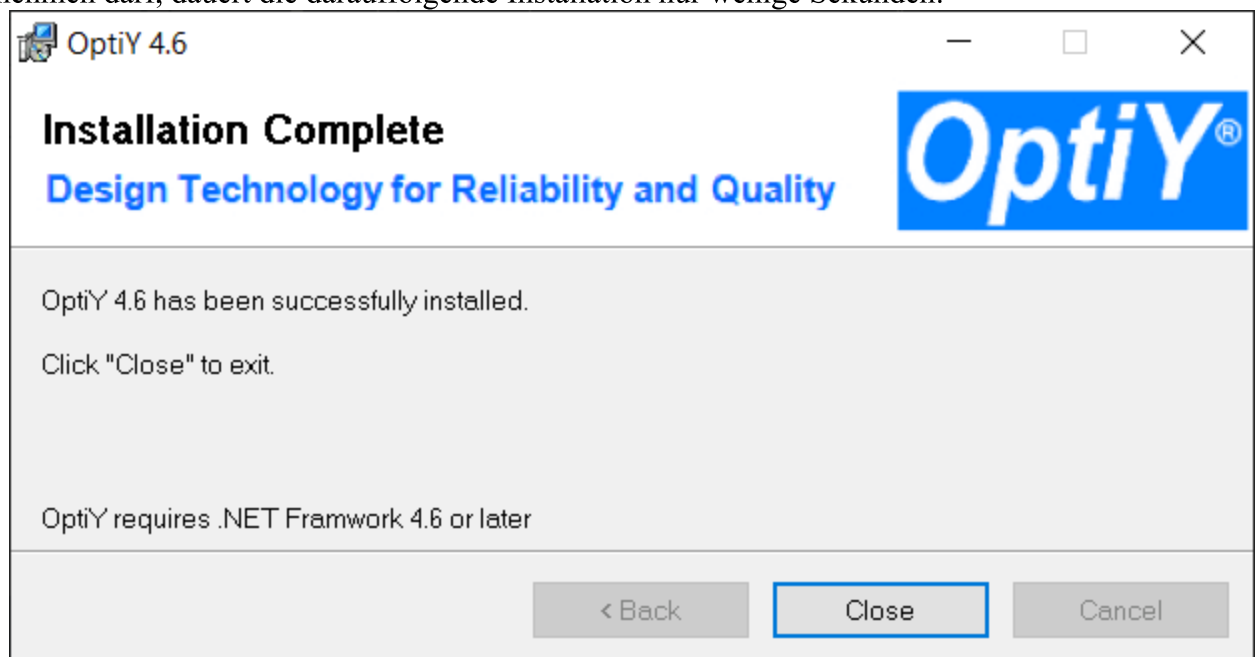
- Anschließend genügen in jedem Fall die normalen Benutzerrechte für das Zusammenspiel zwischen *SimulationX* und *OptiY*.

OptiY-Installation und Programmstart

- Ein "**Setup Wizard**" organisiert die Installation von *OptiY* nach dem Starten einer reichlich 0,2 GByte großen MSI-Datei:

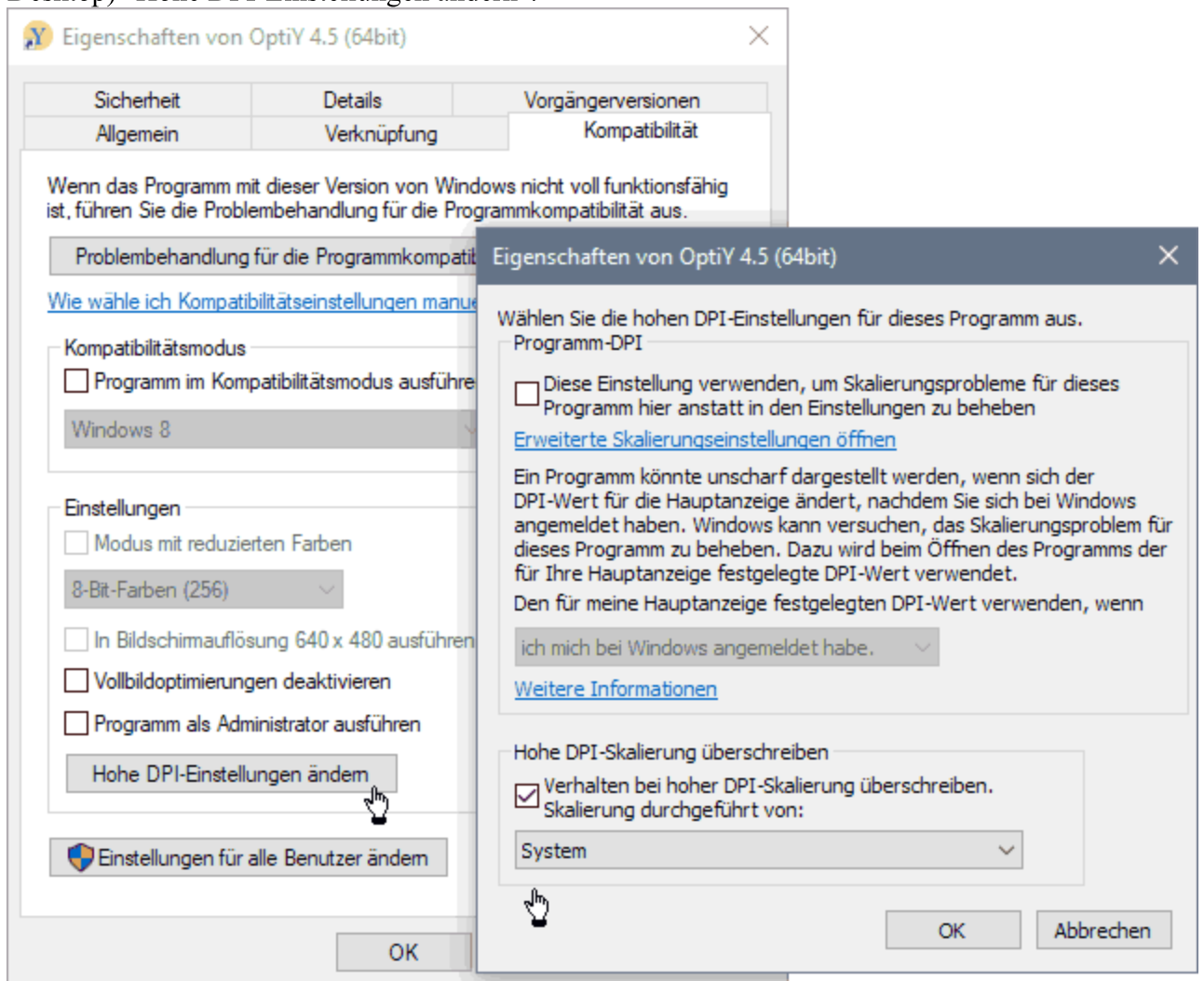


- Das **Akzeptieren der Lizenz-Vereinbarungen** ist im nächsten Schritt Voraussetzung, damit man den Installationsprozess fortsetzen kann.
- Den **Standard-Programmordner** "C:\Program Files\OptiY GmbH\OptiY 4.x" sollte man möglichst beibehalten und die **Installation für alle Computer-Nutzer** bestätigen.
- Wenn der Installer danach feststellt, dass der Computer bereit ist für die Installation, kann man die eigentliche Installation im nächsten Schritt veranlassen.
- Nach Bestätigung der Windows-Abfrage, ob die App diese Änderungen auf dem Computer wirklich vornehmen darf, dauert die darauffolgende Installation nur wenige Sekunden:

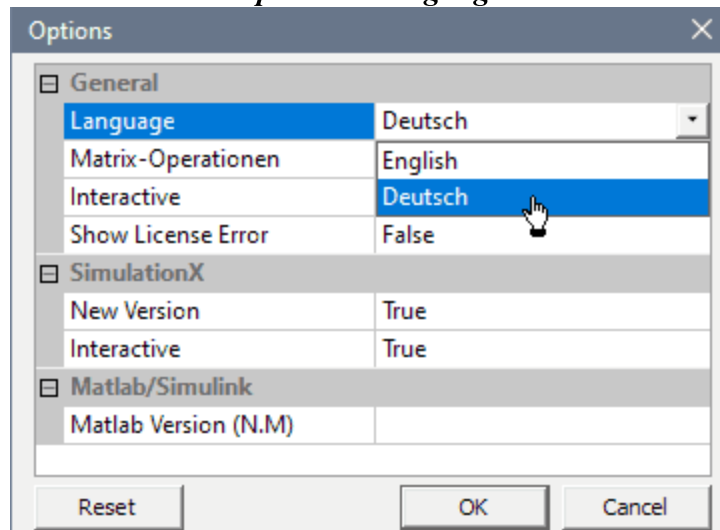


- **.NET Framework** 4.6 oder neuer wird von *OptiY* vorausgesetzt. Dieses ist normalerweise ein Bestand des Betriebssystems *MS Windows*. Falls dies nicht der Fall ist, erfolgt später durch *OptiY* eine Meldung und man kann dann diese Software kostenlos von der Microsoft Website herunterladen und auf dem Computer installieren.
- **Registrierung als COM-Objekt im Windows** erfordert einen einmaligen Start von *OptiY* mit Administrator-Rechten nach der Installation:
 - Unter **Eigenschaften - Kompatibilität** des Programms (Kontextmenü der rechten Maustaste auf *OptiY*-Symbol z.B. auf dem Desktop) wählt man die Berechtigungsstufe "Als Administrator ausführen".
 - Nach Beenden von *OptiY* muss diese Administrator-Berechtigung in den Kompatibilitätseigenschaften wieder deaktiviert werden!
- **Nutzung hochauflösender Displays:**

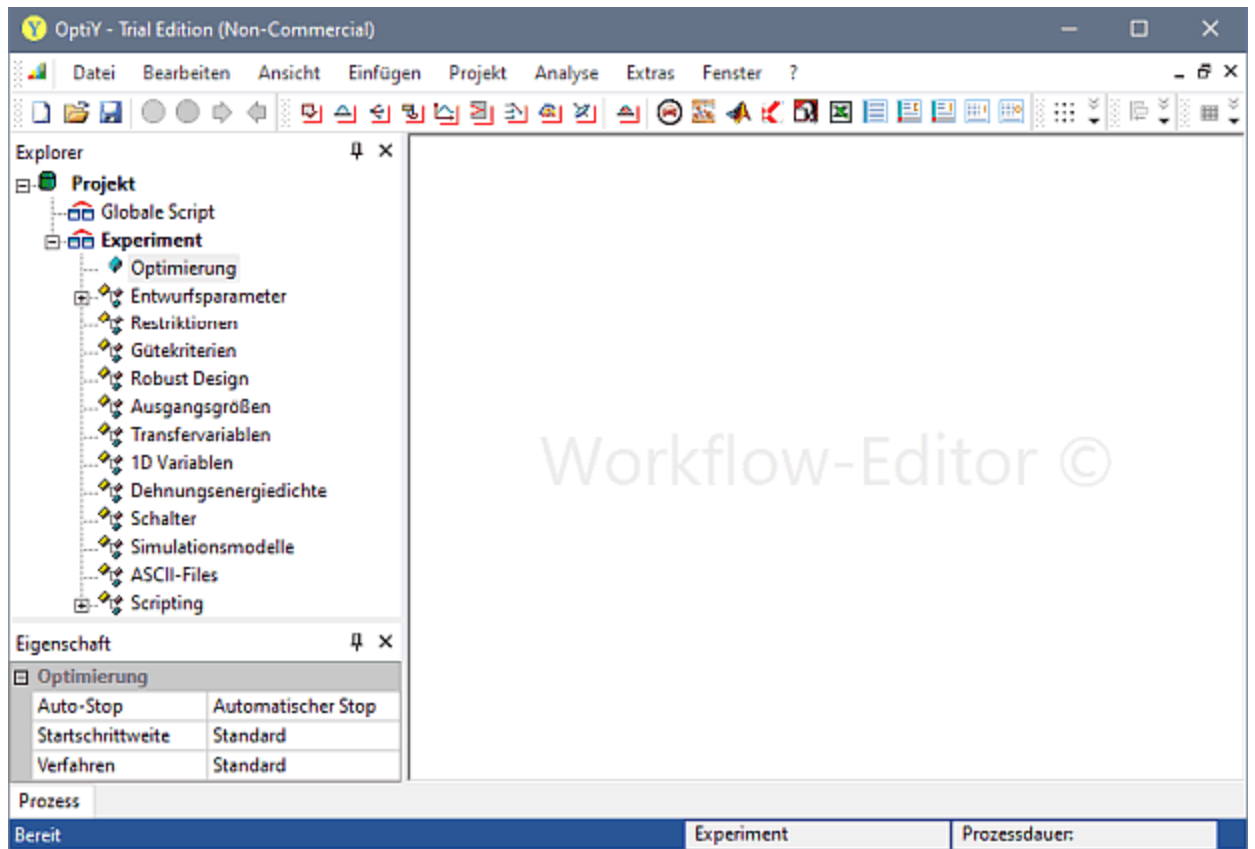
- Viele Programme haben Probleme mit extrem hochauflösenden Bildschirmen, wie sie verstärkt z.B. in modernen Tablett-Computern zum Einsatz kommen. In *OptiY* äußert sich dies in teilweise unlesbar kleinen Diagramm-Legenden oder sich überlappenden Listen-Elementen.
- Abhilfe schaffen hier die Kompatibilitätseinstellungen (Kontextmenü des *OptiY*-Icon auf dem Desktop) "Hohe DPI-Einstellungen ändern":



- Es ist der Modus "Verhalten bei hoher DPI-Skalierung überschreiben" zu aktivieren und "Skalierung durchgeführt von System" einzustellen. Danach sollten alle Programm-Komponente in aufeinander abgestimmter Skalierung dargestellt werden.
- **Umschalten der Bedienoberfläche auf die Sprache "Deutsch"** (um Konsistenz mit den deutschen Übungsanleitungen herzustellen):
 - Nach Start von *OptiY* im Menü *Tools > Options > Language > Deutsch*



- Danach *OptiY* beenden, da erst nach erneutem Start die Sprache der Oberfläche komplett umgeschaltet ist:



← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Vorbereitung_-_Software-Installation&oldid=27725“

Software: SimX - Nadelantrieb - Vorbereitung - Individuelle Modelldatei

Aus OptiYummy

↑

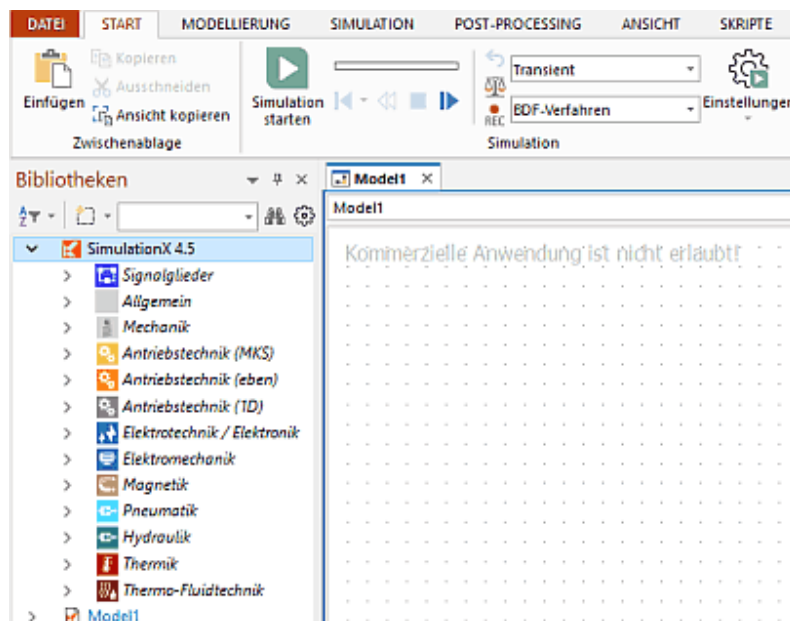
← →

Nutzung der teilnehmer-spezifischen Modelldatei

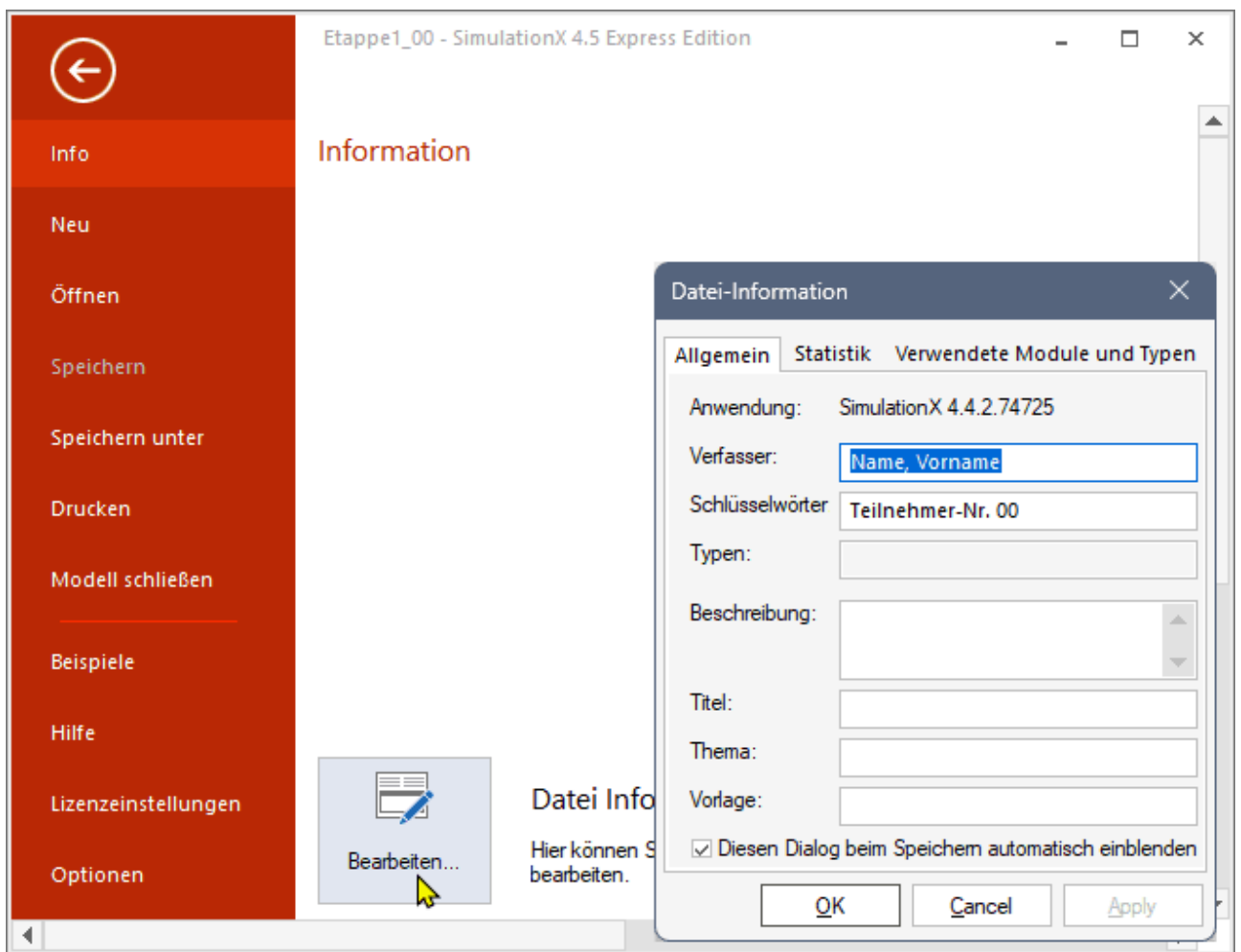
Wichtig: für Teilnehmer der **Lehrveranstaltung Optimierung** (TU Dresden, Institut für Feinwerktechnik und Elektronik-Design):

- Die individuellen Ergebnisse der einzelnen Bearbeitungsetappen sind direkter Bestandteil der Modulnote.
- Deshalb **muss** jeder Teilnehmer dieser Lehrveranstaltung entsprechend seiner Teilnehmer-Nummer **xx** (**xx**=01..99) die bereitgestellte "signierte" Modelldatei "**Etappe1_xx.isx**" als Grundlage für die Modellierung benutzen.
- Die bereitgestellten Modelldateien befinden sich im ZIP-Archiv "**Opt_Modelle.zip**" (Download).
- Nach dem Download ist das ZIP-Archiv zu öffnen und die individuelle Modelldatei "**Etappe1_xx.isx**" entsprechend der zugewiesenen Teilnehmer-Nr. **xx** für die Lehrveranstaltung in einen eigenen Arbeitsordner zu kopieren.

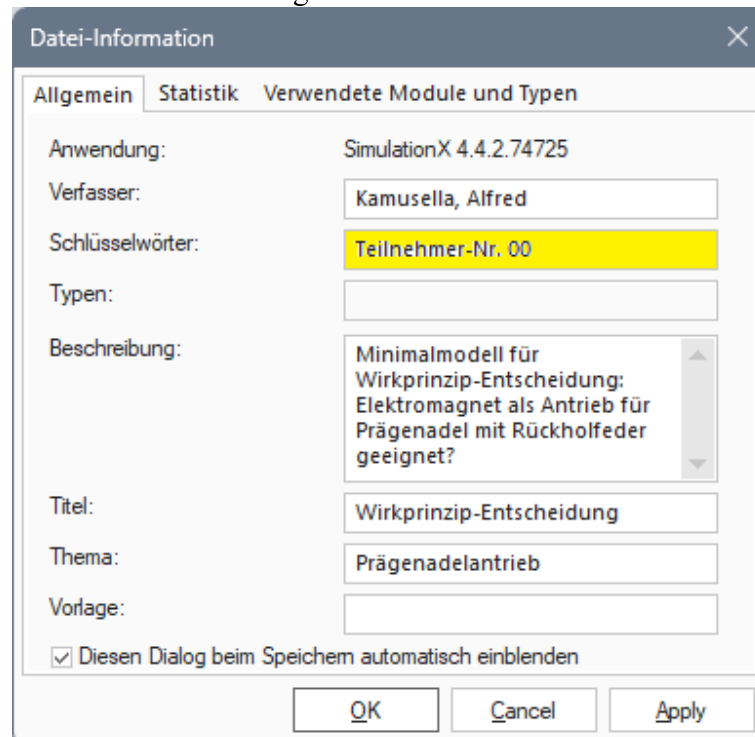
Nach dem Start von *SimulationX* ist standardmäßig ein leeres Modell "**Modell1**" innerhalb der grafischen Oberfläche geöffnet:



- Wer **nicht** an der **Lehrveranstaltung Optimierung** teilnimmt, kann dieses leere Modell nutzen und mittels **Datei > Speichern unter** unter dem Namen **Etappe1.isx** im eigenen Arbeitsordner als "**SimulationX Projekt**" mit Eingabe der danach geforderten **Datei-Informationen** speichern.
- Individuelle Modelldatei innerhalb der **Lehrveranstaltung Optimierung**:
 1. **Datei > Modell schließen**
 2. **Datei > Öffnen** aus dem eigenen Arbeitsordner die Modelldatei **Etappe1_xx.isx** (**xx**=Teilnehmernummer)
 3. **Datei > Info > Bearbeiten** (hier am Beispiel Teilnehmernummer=00):



- **Anwendung:** bezeichnet die SimulationX-Version, mit welcher die Modell-Datei ursprünglich erstellt wurde
- **Verfasser:** "Name, Vorname" durch den **eigenen Namen** ersetzen
- **Schlüsselwörter:** "Teilnehmer-Nr. xx" der "signierten" Modelldatei **nicht verändern!** (im Beispiel xx=00):



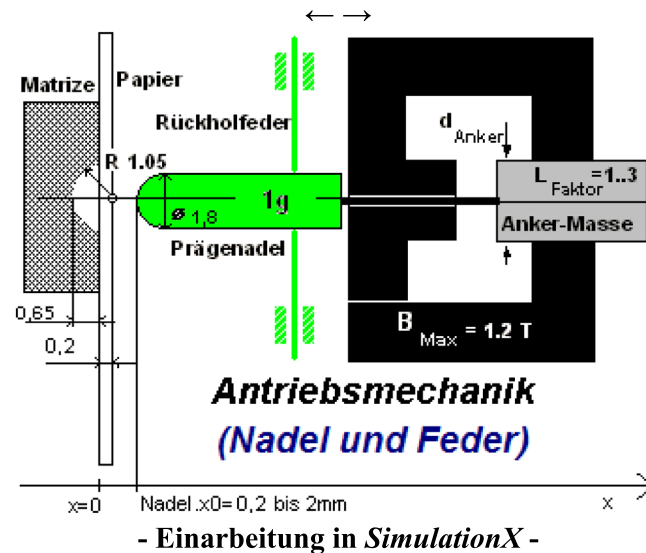
- **Beschreibung:** Inhalt und Zielstellung des Modells
- **Titel:** markante Bezeichnung für aktuelle Arbeitsetappe, z.B. → **Wirkprinzip-Entscheidung**
- **Thema:** markante Bezeichnung des bearbeiteten Projektes, z.B. → **Prägenadelantrieb**
- **OK=Anwenden:** → danach zurück zur grafischen Oberfläche von *SimulationX*.

Damit ist das "leere" Modell nun soweit vorbereitet, dass man im nächsten Schritt mit der eigentlichen Modellierung beginnen kann.

Software: SimX - Nadelantrieb - Vorbereitung - Modell Etappe 1

Aus OptiYummy

↑



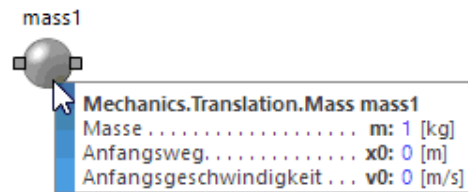
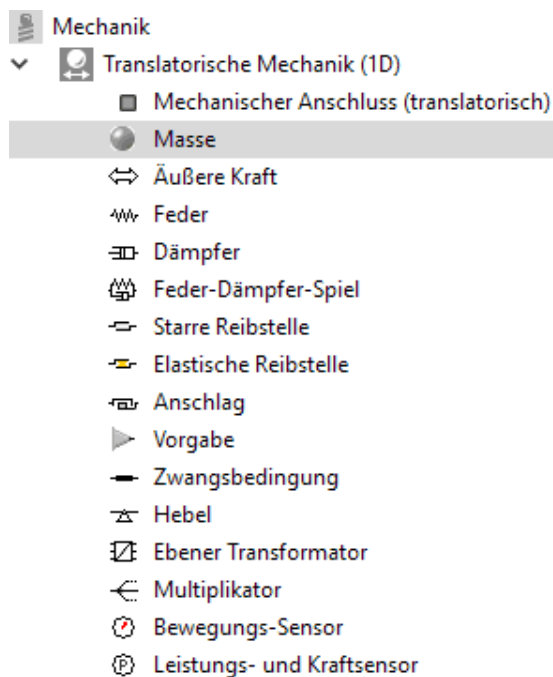
□

Modell-Koordinatensystem

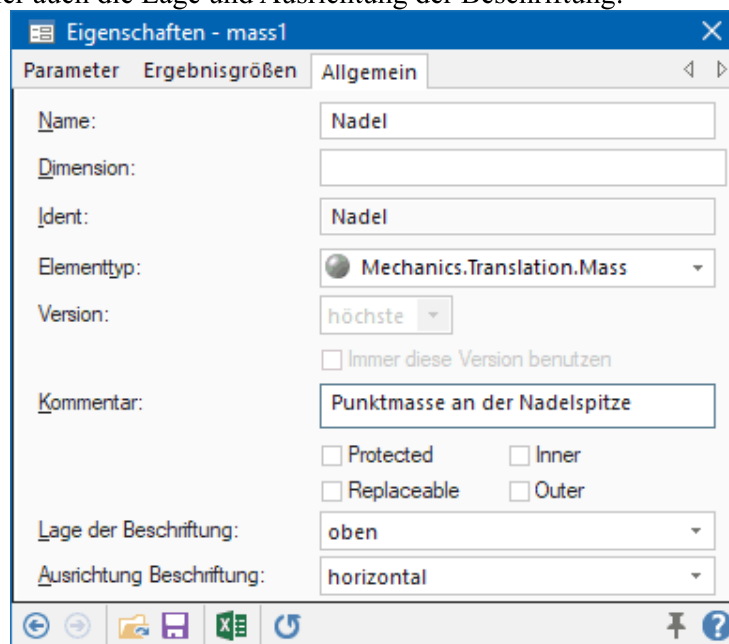
- Alle Mechanik-Elemente eines Modells müssen sich auf das gleiche globale Modell-Koordinatensystem beziehen. Deshalb sollte man die Modellbildung mit den Festlegungen zum Koordinatensystem beginnen.
- Im Prinzip ist es egal, wo man das Koordinatensystem hinlegt und wie man es ausrichtet - man muss sich danach nur daran halten!
- Die Festlegung sollte zu sinnvollen Koordinatenwerten führen.
- Wir benötigen für unseren Antrieb nur einen translatorischen Freiheitsgrad (x-Achse).
- Den Ursprung der x-Achse legen wir aus Sicht der Präge-Nadel auf die Unterseite des Papiers (Siehe obiges Bild).
- Die positive x-Richtung wird entgegen der Präge-Richtung der Nadel festgelegt.
- Im *SimulationX* ist zur Festlegung des Modell-Koordinatensystems keine Aktion erforderlich. Diese Festlegung erfolgt z.B. außerhalb des *SimulationX* in einer geometrischen Skizze des Antriebs.

Präge-Nadel

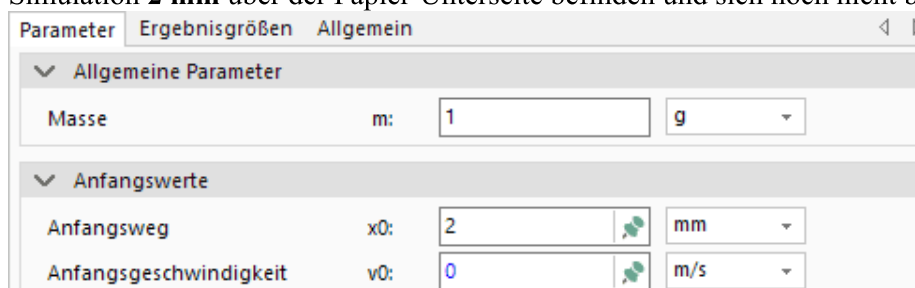
- Die Nadel wird als Punktmasse in der Position der Nadel-Spitze idealisiert.
- Es wird im Weiteren ein Schätzwert von 1 g für die Nadelmasse genutzt, da die konkrete Geometrie noch unbekannt ist.
- In der Mechanik-Bibliothek wird im Ordner *Translatorische Mechanik (1D)* der Element-Typ *Masse* bereitgestellt. Dabei handelt es sich um das Dynamik-Modell einer Punktmasse, die sich nur in x-Richtung bewegen kann.
- Per *Drag&Drop* platziert man diese *Masse* als Element auf die Arbeitsfläche der Modell-Datei (Strukturansicht des Modells). Wenn man den Mauszeiger über ein Modell-Element bewegt, so erscheint ein Quickinfo-Fenster mit den aktuellen Parameterwerten:



- Das Eigenschaftsfenster des Modell-Elements öffnet man mit Doppelklick:
 - Im Register *Allgemein* tragen wir den Namen "Nadel" ein und ergänzen einen sinnvollen Kommentar. Festlegen kann man hier auch die Lage und Ausrichtung der Beschriftung:



- Über das Register *Parameter* versteht man die Nadel mit einer Masse von **1 g**. Die Nadelspitze soll sich am Anfang der Simulation **2 mm** über der Papier-Unterseite befinden und sich noch nicht bewegen:



Hinweis: Die hinter den Anfangswerten der Zustandsgrößen befindlichen "Reißzwecken" dienen als Steuer-Elemente für den Solver. In der abgebildeten Default-Einstellung sind diese "Pin" fest, d.h. der Solver darf bei der Berechnung der Anfangswerte (bei $t=t_{\text{Start}}$) die eingetragenen Werte nicht verändern, auch wenn dies zu Widersprüchen mit anderen Anfangswerten führt.

- Unter *Ergebnisgrößen* sind die Element-Variablen aufgelistet, die für eine Protokollierung / Visualisierung zur Verfügung stehen. Davon wählen wir *Weg* und *Geschwindigkeit* für die Protokollierung aus. Die simulierten Zeitverläufe dieser Größen werden dann in eine Protokolldatei geschrieben und können bei Bedarf auch als Signalverlauf dargestellt werden. Dafür sollte man bereits eine anschauliche Maßeinheit für die protokollierten Größen wählen:

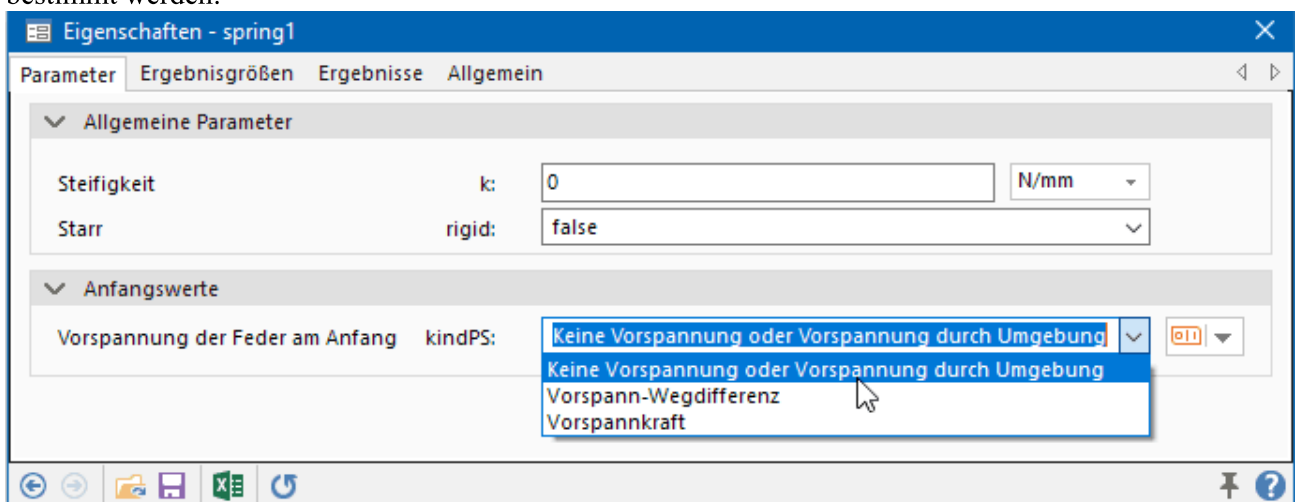


Hinweis: Man sollte nur die Größen für die Protokollierung auswählen, die man wirklich benötigt, da jeder "Datenmüll" Platz und Zeit kostet! Die Wahl einer geeigneten Maßeinheit für die Ergebnisdarstellung erleichtert die Interpretation.

- Alle Eigenschaftsänderungen werden bereits nach dem Eintrag in die entsprechenden Dialogfelder in das Modell übernommen. Das Schließen des Eigenschaftsfensters hat auf die Übernahme der Werte keinen Einfluss!

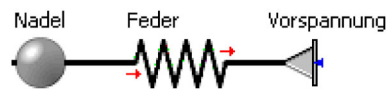
Rückhol-Feder

- Eine Feder soll die Nadel nach dem Prägevorgang hinreichend schnell in die Ruhelage zurückholen.
- Hier sind verschiedenste Feder-Formen denkbar. Günstig für eine stabile Ruhelage ist sicher eine Vorspannung.
- Wir idealisieren die vorgespannte Feder als lineare Kraft-Kennlinie und erweitern die Modellstruktur um ein *Feder*-Element aus der Mechanik-Bibliothek.
- Die Eigenschaften dieses Elements ändern wir analog zur Nadel-Masse:
 - Name=Feder / Kommentar=Rückholfeder.
 - Da die Bewegung im Millimeter-Bereich stattfindet, ist die Einheit [N/mm] für die Steifigkeit eine gute Wahl. Die Steifigkeit setzen wir vorläufig auf Null.
 - Die Vorspannung der Feder in der Ruhelage der Nadel soll durch ihre Einspannung in der Umgebung bestimmt werden:



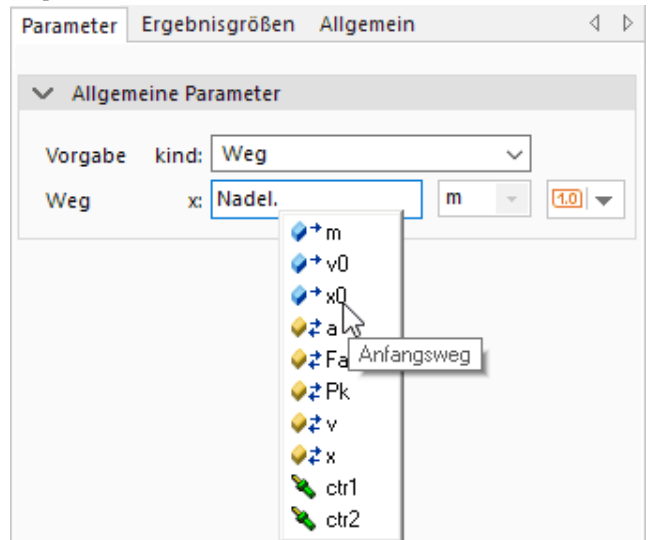
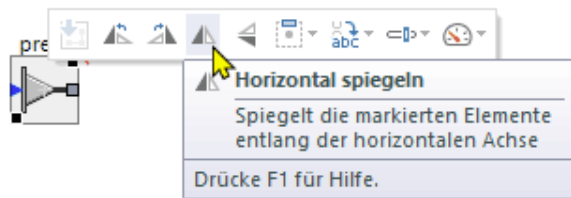
- Zu protokollieren ist nur die Gesamtkraft der Feder als Ergebnis-Größe (Innere Kraft F_i).
- Das eine Ende der Feder ist mit der Nadel zu verbinden:
 - Wenn man sich mit dem Mauszeiger über einen Elementanschluss bewegt, so färbt sich der Anschluss-Pin rot und am Mauszeiger erscheint ein Verbindungssymbol.
 - Die Verbindung wird hergestellt durch Bewegen des Mauszeigers zwischen den Anschluss-Pins mit gedrückter linker Maustaste ("Loslassen" am Ziel).

Feder-Vorspannung



Das nicht verbundene Ende der Feder ist standardmäßig bei $x=0$ fest eingespannt.

- Wir benötigen eine Einspann-Position relativ zur Ruhelage der Nadel, um eine definierte Vorspannung einzustellen.
- Dazu verwenden wir das *Vorgabe*-Element aus der Mechanik-Bibliothek und verbinden es mit dem offenem Feder-Ende:
 - Wir nutzen die Funktion "Weg"-Vorgabe. Vor dem endgültigen Platzieren und Einbinden in der Modellstruktur kann man sehr einfach über das eingblendete Tool die erforderlichen Drehungen bzw. Spiegelungen vornehmen:



- Der Vorgabe-Weg (=Position) wird vorläufig auf die Position der Nadel-Ruhelage gesetzt (noch ohne Vorspannung). Durch Nutzung der Variablen *Nadel.x0* passt sich die Feder-Position automatisch an eine Veränderung der Nadel-Ruhelage an.

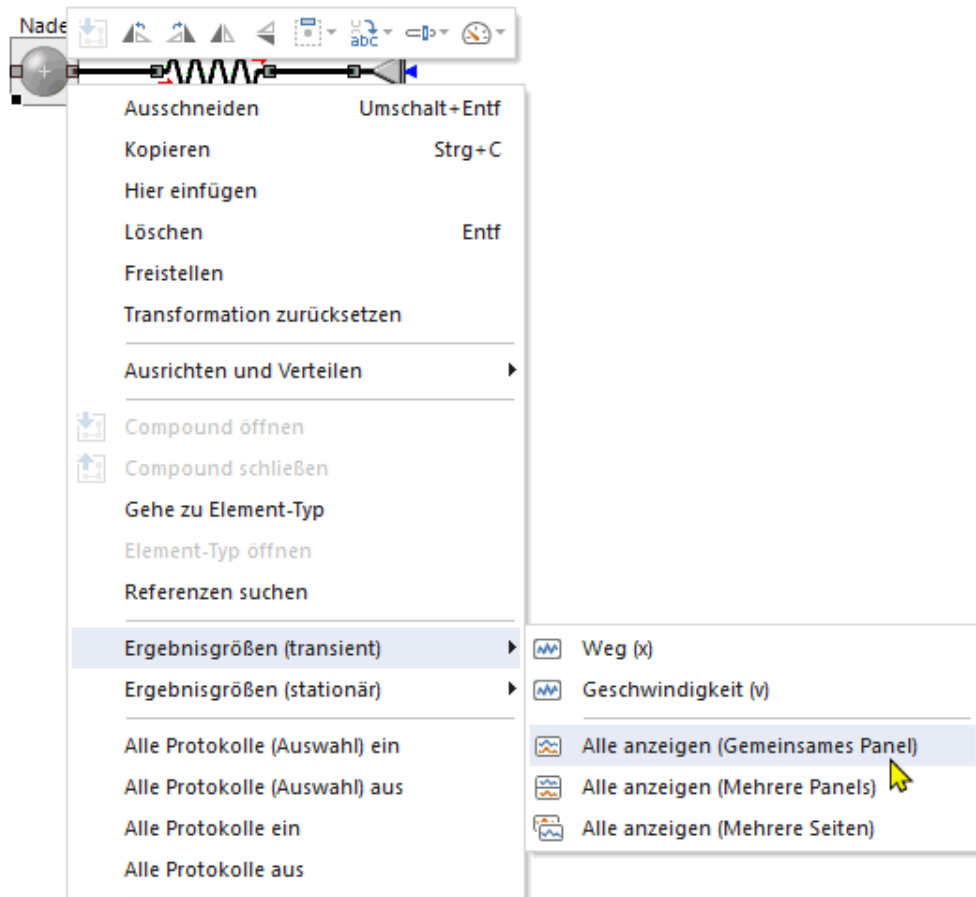
Validierung des Modellverhaltens

Es ist sinnvoll, ein Modell schrittweise in Betrieb zu nehmen, um eventuelle Fehler noch mit vertretbarem Aufwand zu finden! Die bereits modellierte Antriebsmechanik ist ausreichend für die Durchführung erster Simulationsexperimente. Folgende Signalverläufe sollen dafür in einem gemeinsamen Ergebnisdiagramm dargestellt werden:

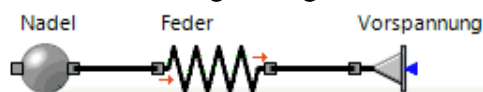
1. Nadel-Position
2. Nadel-Geschwindigkeit
3. Feder-Kraft

Für das Anlegen und Konfigurieren von Ergebnisfenstern werden in *SimulationX* verschiedene Möglichkeiten bereitgestellt.

- Voraussetzung ist die vorherige Auswahl der darzustellenden Ergebnisgrößen in den Modell-Elementen für die Protokollierung (bereits geschehen für **Nadel.x**, **Nadel.v** und **Feder.Fi**).
- Damit man der Verlauf des Nadel-Bewegung während der zeitlichen Simulation betrachten kann, muss man diese Ergebnisgröße für die Darstellung einem Ergebnisfenster zuweisen (z.B. mittels rechten Mausklick auf die "Nadel"):



- Auch innerhalb dieses Kontext-Menüs stehen verschiedene Möglichkeiten für die Ergebnisfenster-Zuordnung zur Verfügung → "**Alle anzeigen (Gemeinsames Panel)**" realisiert mit einem Klick unseren Wunsch nach gemeinsamer Darstellung der Signalverläufe in einem gemeinsamen Diagramm:



- Da noch keine Ergebnis-Signale berechnet wurden, enthält das gemeinsame Signal-Panel keine Signal-Verläufe und die Y-Achsen sind mit Standardgrenzwerten skaliert.

Auch über den Ergebnis-Bereich des Modellexplorers kann man mit wenigen Mausklicks Signalverläufe in neue oder vorhandene Ergebnisfenster ziehen:

- Zuvor muss man dafür das entsprechende Modell-Element auswählen (entweder in der zentralen grafischen Strukturansicht oder im Modellexplorer auf der rechten Seite der Benutzeroberfläche).
- Per *Drag & Drop* kann man das gewünschte Ergebnis an seiner Protokoll-Markierung "aufnehmen" und auf die Zielposition *Ziehen und Ablegen*, z.B. in ein vorhandenes Signal-Panel:

Model1 (Etappe1_00.isx)*

Model1

Kommerzielle Anwendung ist nicht erlaubt!

Nadel Feder Vorspannung

Ergebnisse

Kommentar	Name	Aktueller Wert	Maßeinheit	Protokoll
Innere Kraft	Fi	0	N	✓
Federkraft	Fs	0	N	✗
Wegdifferenz	dx	0	m	✗
Geschwindigkeitsdifferenz	dv	0	m/s	✗
Änderung der potentiellen Energie	Pp	0	kW	✗

Weg - Nadel.x , ...

DATEI BEARBEITEN ANSICHT HINZUFÜGEN ANALYSE DARSTELLUNG FENSTER ?

Weg - Nadel.x
Geschwindigkeit - Nadel.v

0.1 mm
0.0
-0.1

0.0001 m/s
0.0000
-0.0001

0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 s 1.0

Simulationzeit: 0.00000000 s

Simulation Starten (F5)
Startet die Simulationsrechnung.
Drücke F1 für Hilfe.

Wir starten einfach mal einen Simulationslauf, um zu sehen, was passiert:

Simulation starten

Simulation

Transient

BDF-Verfahren

Einstellungen

Schwingungsvorgänge dürften danach im Ergebnisfenster nicht sichtbar werden, nur eine "Null-Linie" der Signalverläufe. Da die Federsteife noch den Wert Null besitzt, kann von der Feder keine Kraft auf die Nadelmasse ausgeübt werden!

Wir werden im Folgenden die Federsteifigkeit mit einem Wert belegen, der in der richtigen Größenordnung liegt:

- Die Feder muss so steif sein, dass sie in der Lage ist, in ungefähr der halben Zeit eines Prägezyklusses von **3,6 ms** die Nadelmasse (einschließlich der Ankermasse) vom Matrizenboden in die Ruhelage zurück zu holen.
- Diese **1,8 ms** entsprechen einer Viertel Periode einer vollen Schwingung. Die Resonanzfrequenz des Feder-Masse-Systems sollte also gleich $1/(7,2 \text{ ms})$ sein, das sind ca. **139 Hz**.
- Die Masse des Magnet-Ankers ist noch unbekannt und soll vorläufig unberücksichtigt bleiben.

- Mit einer Nadelmasse $m=1\text{ g}$ berechnen wir die Federsteifigkeit k anhand der maximal zulässige Periodendauer $T=7,2\text{ ms}$ mit:

$$T = 2 \cdot \pi \cdot \sqrt{\frac{m}{k}}$$

- **Beachte:** Falls sich im weiteren Verlauf der Validierung nicht die gewünschte Periodendauer von $T=7,2\text{ ms}$ ergibt, so liegt dies meist einem Flüchtigkeitsfehler bei der Berechnung der Federsteifigkeit (Formel falsch umgestellt, Maßeinheiten nicht korrekt umgerechnet). Natürlich sollte man auch den korrekten Eintrag des Wertes von 1 g für die Nadelmasse überprüfen (so führt z.B. 1 kg zu einer extrem größeren Periodendauer!).

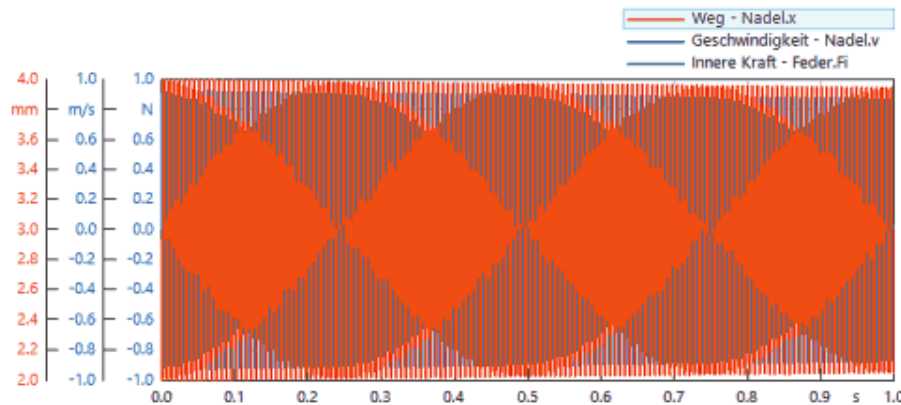
◀ Auf Anfangszeitpunkt setzen
Setzt die Simulation auf den Anfangszeitpunkt zurück.

Nach dem Zurücksetzen der Simulation starten wir die Simulation erneut. Auch diesmal dürften wieder keine Schwingungen sichtbar sein, denn die Feder ist infolge der Position ihrer Aufhängung nicht vorgespannt:

- Wir werden die Feder um 1 mm entgegen der Prägerichtung vorspannen. Dabei muss man beachten, dass automatisch alle Werte mit SI-Grundeinheiten versehen werden, wenn sie in zusammengesetzten Ausdrücken auftreten:

Vorgabe . . . kind: Weg
Weg x: Nadel.x0+0.001 [m]

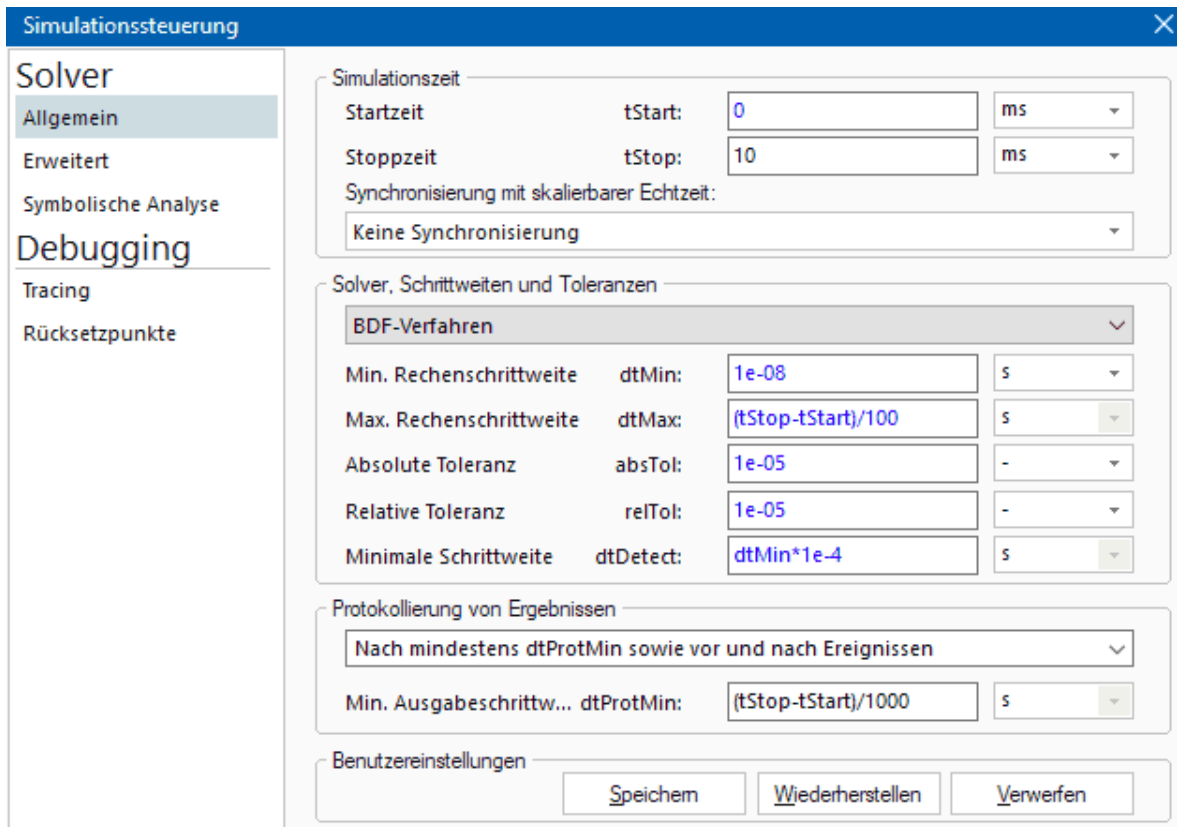
- Nach erneutem Rücksetzen und Start der Simulation wird man vor lauter Schwingungen fast nicht mehr erkennen und es werden in Abhängigkeit von der Fenstergröße "eigenartige" Muster sichtbar, welche man bei einer ungedämpften Schwingung nicht erwartet. Der berechnete Zeitbereich von 1 s ist für das Beispiel viel zu lang:




⚙ Simulations-Einstellungen
Öffnet den Einstellungsdialog für das aktuell gewählte Simulationsverfahren

Mittels der Möglichkeiten der Simulationssteuerung werden wir den berechneten Zeitbereich auf 10 ms begrenzen und eine optimale Darstellung der Signalverläufe konfigurieren:

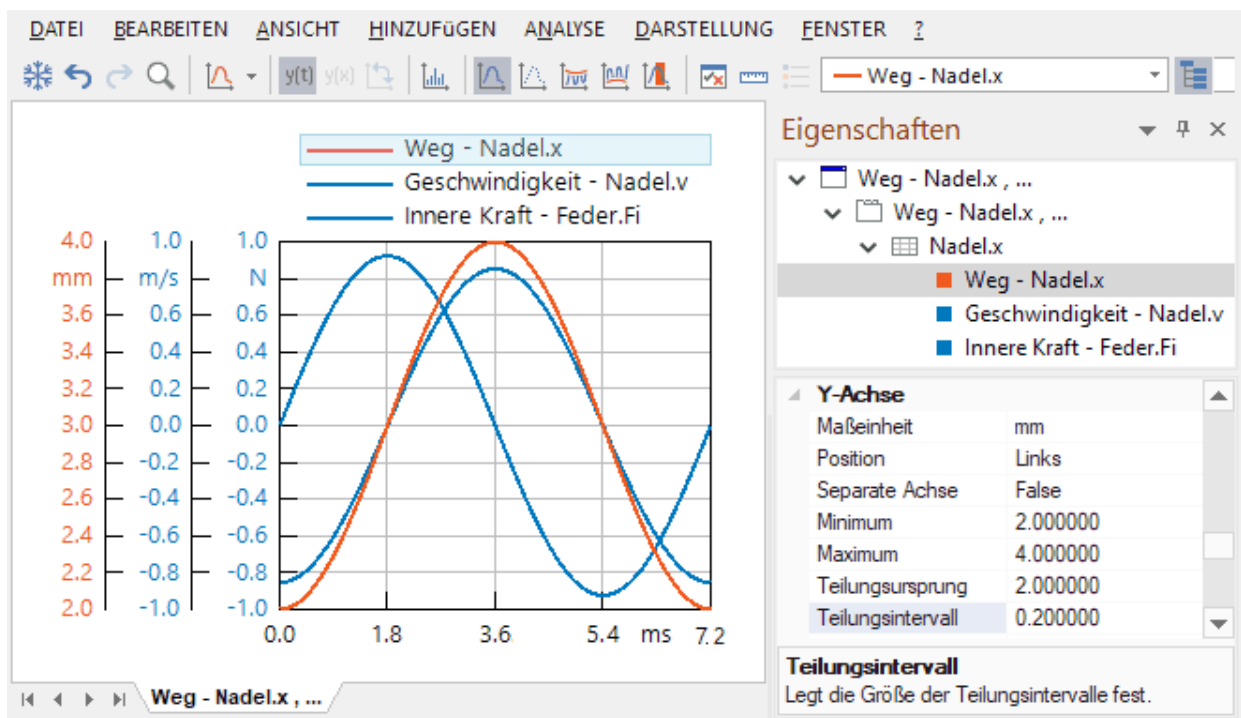
- Zuerst verringern wir unter *Solver - Allgemein* den Wert von **tStop** auf 10 ms (möglichst gleiche Maßeinheit für **tStart**!) und starten die Simulation erneut.
- Statt der erwarteten Sinusschwingung sieht man nun recht zackige Verläufe.
- Ursache ist die "min. Ausgabeschrittweite" **dtProtMin**=1 ms. Die Sinuskurven werden demzufolge im Beispiel aus Geradenstücken von 1 ms Länge approximiert!
- Damit Kurvendarstellungen unabhängig vom Zeitbereich einigermaßen "rund" sind, sollte man die Ausgabeschrittweite abhängig vom berechneten Zeitbereich machen. Für die Max. Rechenschrittweite ist z.B. standardmäßig " $dtMax=(tStop-tStart)/100$ " eingetragen.
- Wir verändern in diesem Sinne **dtProtMin** durch Eintrag eines solchen Ausdrucks. Dabei muss **dtMax>dtProtMin** sein (z.B. " $dtProtMin=(tStop-tStart)/1000$!") Nun müssten nach erneuter Simulationsrechnung schön gerundete Signalverläufe erscheinen.
- Zusätzlich sollte man alle Unstetigkeiten protokollieren ("Nach mindestens dtProtMin sowie vor und nach Ereignissen"):



Standardmäßig werden die X- und Y-Achse in den Ergebnis-Fenstern automatisch skaliert. Dies passiert während der Simulationsrechnung anhand der bis zum aktuellen Zeitpunkt registrierten Extremwerte. Wenn man diese Grenzwerte bereits kennt, kann das ständige Neuskalieren des Koordinatensystems vermeiden:

- Wir schalten dafür im Ergebnisfenster *vor dem Rücksetzen des Simulationslaufes* in der Eigenschaftsleiste des Ergebnisfensters  für das Panel die automatische Anpassung für die beiden Koordinatenachsen ab.
- Den Zeitbereich für die X-Achse kann man unabhängig vom berechneten Zeitbereich festlegen. Um die Länge einer Schwingung besser überprüfen zu können, setzen wir das Maximum auf 7,2 ms. Als "Teilungsintervall" wählen wir den Wert 1,8 ms.
- Für die Y-Achsen setzen wir die Grenzen auf den nächstliegenden ganzzahligen Wert und wählen eine sinnvolle Teilung.
- Die Maßeinheiten sollten nach DIN 461 an den Achsen angetragen werden.

Wenn das Simulationsmodell bisher richtig konfiguriert ist, müsste "exakt" eine Schwingung im betrachteten Zeitbereich abgebildet werden. Anderenfalls muss man sich auf Fehlersuche begeben:



Ergebnisse der Übungsvorbereitung

Mit der Entwicklung eines einfachen Modells für die Wechselwirkung zwischen der Prägenadel und der einseitig eingespannten Rückholfeder haben wir einen wichtigen Meilenstein in unserem Übungsbeispiel erreicht:

- Wir wissen nun, in welchen Abschnitt des konstruktiven Entwicklungsprozesses für einen Blindenschriftpräger sich unsere Übungsetappen einordnen.
- Die erforderliche Software wurde auf dem privaten Computer installiert und in Betrieb genommen (soweit dafür die technischen Voraussetzungen existierten).
- Es erfolgte eine grundlegende Einarbeitung in das Programm *SimulationX*, welches wir zur Systemsimulation nutzen.
- Wir haben anhand der Parametrisierung des Feder-Masse-Modells ein Gefühl für Zeitabläufe und die Größenordnungen von beschleunigter Masse und wirksamer Rückholfeder entwickelt.

Im Ergebnis dieser Übungsvorbereitung existiert ein parametrisiertes, lauffähiges Simulationsmodell **Etappe1_xx.isx** (Teilnehmer-Nummer **xx=01...99**) mit einem für die Periodendauer von **7,2 ms** konfigurierten Ergebnisfenster:

- Teilnehmer der **Lehrveranstaltung Optimierung** (TU Dresden, Institut für Feinwerktechnik und Elektronik-Design) können für diesen Stand der Vorbereitung einen Bonuspunkt erhalten!
- Dazu ist der aktuelle Zustand des Modells zu speichern und wie auf der Titelseite dieser Anleitung beschrieben einzusenden.

← →

Abgerufen von „http://index.php?title=Software:_SimX_-_Nadelantrieb_-_Vorbereitung_-_Modell_Etappe_1&oldid=27730“